DEMS
Department of Economics, Management and Statistics

DEMS WORKING PAPER SERIES

Do global forecasting models require frequent retraining?

Marco Zanotti

No. 551 – April 2025

Department of Economics, Management and Statistics
University of Milano – Bicocca
Piazza Ateneo Nuovo 1 – 2016 Milan, Italy
http://dems.unimib.it/

Do global forecasting models require frequent retraining?

Marco Zanotti[a,*]

[a] *Department of Economics, Management and Statistics, University of Milano-Bicocca, Milan, Italy*

Abstract

In an era of increasing computational capabilities and growing environmental consciousness, organizations face a critical challenge in balancing the accuracy of their forecasting models with computational efficiency and sustainability. Global forecasting models, which leverage data across multiple time series to improve prediction accuracy, lowering the computational time, have gained significant attention over the years. However, the common practice of retraining these models with new observations raises important questions about the costs of producing forecasts. Using ten different machine learning and deep learning models, we analyzed various retraining scenarios, ranging from continuous updates to no retraining at all, across two large retail datasets. We showed that less frequent retraining strategies can maintain the forecast accuracy while reducing the computational costs, providing a more sustainable approach to large-scale forecasting. We also found that machine learning models are a marginally better choice to reduce the costs of forecasting when coupled with less frequent model retraining strategies as the frequency of the data increases. Our findings challenge the conventional belief that frequent retraining is essential for maintaining forecasting accuracy. Instead, periodic retraining offers a good balance between predictive performance and efficiency, both in the case of point and probabilistic forecasting. These insights provide actionable guidelines for organizations seeking to optimize forecasting pipelines while reducing costs and energy consumption.

Keywords: Time series, Demand forecasting, Forecasting competitions, Cross-learning, Global models, Machine learning, Deep learning, Green AI, Conformal predictions

JEL: C53, C52, C55

*Corresponding author

Email address: zanottimarco17@gmail.com (Marco Zanotti)

1. Introduction

Forecasting plays a critical role in decision-making processes across industries, from supply chain management to energy demand planning. Traditionally, time series forecasting has relied on a Local Modeling approach (LM), meaning that separate models [1] are trained for each series in isolation. However, recent advances have demonstrated the potential of Global forecasting Models (GM), a paradigm also known as cross-learning, which in turn consists of fitting a single forecasting model on the whole data set (Januschowski, Gasthaus, Wang, Salinas, Flunkert, Bohlke-Schneider & Callot, 2020). This novel approach allows to leverage information across multiple time series to improve accuracy and generalization by employing machine learning or deep learning architectures [2] . Global modeling has affirmed itself as one of the most relevant innovations in time series forecasting of recent years (Semenoglou, Spiliotis, Makridakis & Assimakopoulos, 2021), and its success is not only due to the demonstrated forecast accuracy in most time series competitions, but also to the particular computational benefit it provides. Indeed, while the local approach to time series forecasting is potentially beneficial for forecast accuracy, it comes with substantial computational overhead. Moreover, since the forecasts are usually produced through some cloud computing service that relies on pay-as-you-go pricing (Fotios Petropoulos & Spiliotis, 2024), higher computing time and resources directly translate into higher costs of forecasting for organizations.

The retail industry particularly exemplifies the complexity of modern forecasting challenges. Retailers must predict demand for thousands or even millions of products across hundreds of stores while accounting for seasonality, promotions, pricing strategies, and external factors such as weather and local events. Accurate demand forecasts directly impact inventory decisions: overestimation leads to excess inventory and waste, particularly critical for perishable goods, while underestimation results in stockouts and direct money loss for companies (Fildes, Ma & Kolassa, 2022). Traditional approaches involving separate models for each product-store combination have proven increasingly inadequate, especially in contexts where the number of series (SKUs) to forecast is high, leading to the adoption of global forecasting models that can learn patterns across the entire product hierarchy

[1]We use the words model and method interchangeably even if the two have distinct statistical meanings (Svetunkov & Boylan, 2023).

[2]We distinguish between traditional machine learning models, like linear regression or tree-based methods, and deep learning models, those that are based on some sort of neural network architecture.

and can drastically reduce the costs of producing forecasts [3] .

Nevertheless, despite the growing adoption of global models, it is still common practice to update the forecasting models when new observations are available, often motivated by the assumption that continuous updates lead to better adaptability to changing patterns and more accurate predictions. Unlike local models, global models benefit from learning shared dynamics across multiple time series, potentially leading to more robust representations that are less sensitive to frequent retraining. In the context of local models, Spiliotis & Petropoulos (2024) demonstrated that less frequent model updates do not harm forecasting accuracy. However, in the context of global models, the effects of retraining are not well understood. Is continuous retraining necessary to maintain forecast performance, or can global models remain effective without frequent updates? This aspect remains largely unexplored in the forecasting literature and investigating the impact of retraining is crucial for both methodological and practical reasons. From a methodological perspective, understanding whether global models degrade in performance without frequent retraining provides insights into their stability and adaptability. If global models can maintain strong predictive accuracy with less frequent updates, it would challenge conventional wisdom on the necessity of continuous retraining in forecasting. Moreover, from a practical perspective, frequent retraining has significant computational and environmental costs. Training large-scale forecasting models requires substantial computational power, contributing to energy consumption and carbon emissions (Schwartz, Dodge, Smith & Etzioni, 2020). Indeed, the energy consumption of model training extends beyond direct computational costs. In recent years, the environmental impact of machine learning models has become a growing concern. Data centers running these models contribute significantly to global carbon emissions, with estimates suggesting that training a single large deep learning model can emit as much carbon as five cars over their lifetimes Strubell, Ganesh & McCallum (2019). The frequency of model retraining multiplies this impact, particularly in large organizations handling millions of series, emphasizing the need for more computationally efficient training practices. The implication is straightforward: reducing the retraining frequency of forecasting models could contribute substantially to sustainability by lowering energy consumption, possibly without harming accuracy performance.

[3] Most of the cost of producing forecasts is due to the algorithm's training time. The global modeling approach cuts it by training the model only once on the whole data set.

1.1. Research Question

We aim to address the question *"Is frequent retraining necessary in the context of global forecasting models?"*. Specifically, we study whether skipping retraining when new observations are available harms the forecasting performance of global models. To answer this question, we use the two most recent and comprehensive retail forecasting datasets, namely the M5 and the VN1 competitions' data. Moreover, we also aggregate these two time series data sets to be able to further investigate the research question across different frequencies.

To generally understand how retraining the forecasting models affects their performances, we consider ten different global forecasting methods (five from the "classical" machine learning domain and five often used deep neural network architectures), and several possible retraining scenarios, from continuous retraining to no retraining at all. We also explore intermediate periodic retraining strategies to broadly cover the most reasonable and effective scenarios.

We also focus on the investigation of trade-offs between accuracy and sustainability, in terms of the computational cost of resources, to produce the forecasts. This cost is indeed significant for large-scale applications, like the retail industry, and being able to reduce (or control) it somehow may result in direct and significant business savings.

1.2. Contributions

Our contribution is threefold:

- We provide the first comprehensive study of the relationship between retraining frequency and forecast accuracy using 10 different global models, a diverse set of real-world datasets, and focusing on both point and probabilistic forecasting.
- We compare different retraining scenarios (e.g., continuous, periodic, and no retraining) on different datasets to quantify the impact of frequent retraining in terms of cost of forecasting.
- We present practical guidelines for organizations and practitioners on when and how often to retrain global forecasting models to balance accuracy and cost.

By addressing these points, this paper contributes to both the forecasting and machine learning communities, offering insights into the trade-offs between accuracy, efficiency, and sustainability in global forecasting models.

1.3. Overview

The rest of this paper is organized as follows. After a brief review of related works (Section 2), in Section 3 we describe the design of the experiment used in our study. The datasets and their characteristics are presented in 3.1, the methods adopted for global forecasting are discussed in 3.2, and the concepts related to model update and retrain scenario are explained in 3.3. The evaluation setup through rolling origin validation is presented in 3.4, and the metrics used to evaluate the model performances are shown in 3.5. In Section 4 we discuss the empirical findings of our study, including forecast accuracy, computing time, and cost analysis of the different scenarios. Finally, Section 5 contains our summary and conclusions.

2. Related works

The literature on cross-learning approach has evolved significantly in recent years. Nowadays, most of the works related to time series forecasting include at least some benchmark comparison with global models, demonstrating their relevance in the field. Semenoglou et al. (2021) extensively showed their accuracy on the M4 competition dataset, Hewamalage, Bergmeir & Bandara (2022) evaluated the conditions when global forecasting models are competitive, and Montero-Manso & Hyndman (2021) and (Montero-Manso, 2023) theoretically demonstrated that GM are at least as accurate as local models with less complexity and without any assumption on the similarity of the data. GM emerged as the most accurate approach in many forecasting areas, such as gas consumption (Gaweł & Paliński, 2024), electricity demand (Buonanno, Caliano, Pontecorvo, Sforza, Valenti & Graditi, 2022), water demand (Groß & Hans, 2024), crop production (Ibañez & Monterola, 2023), and retail demand (Spiliotis, Makridakis, Semenoglou & Assimakopoulos (2022), Bandara, Shi, Bergmeir, Hewamalage, Tran & Seaman (2019), Juan R Trapero & Fildes (2015)). Moreover, GM effectiveness shined during the M5 competition (Makridakis, Spiliotis & Assimakopoulos, 2022a), where tree-based models leveraging cross-learning were among almost all the top forecasting solutions (Januschowski, Wang, Torkkola, Erkkilä, Hasson & Gasthaus, 2022). Several techniques like clustering (Godahewa, Bandara, Webb, Smyl & Bergmeir (2021a), Bandara, Bergmeir & Smyl (2020)), and data augmentation (Bandara, Hewamalage, Liu, Kang & Bergmeir, 2021) have been tested to further increase the performances of GM. Furthermore, new machine learning (Godahewa, Webb, Schmidt & Bergmeir, 2023) and deep learning (Oreshkin, Carpov, Chapados & Bengio, 2020) architectures specifically designed for cross-learning have been developed. Finally, a novel area of research is emerging that tries to improve the ability of GM to capture local patterns (Sen, Yu & Dhillon, 2019), and their explainability (Rajapaksha, Bergmeir & Hyndman, 2023).

From a forecasting evaluation perspective, most of the literature on GM is focused on point prediction accuracy, possibly because most machine learning and deep learning methods do not directly output probabilistic forecasts (Makridakis, Spiliotis, Assimakopoulos, Chen, Gaba, Tsetlin & Winkler, 2022c). Nevertheless, in many forecasting contexts (like supply chain) it is very important to be able to produce and evaluate predictions in a probabilistic way (being intervals, quantiles or density based) (Fildes et al., 2022). Among others, Vovk, Gammerman & Shafer (2005) introduced a novel tool for uncertainty quantification under any machine learning model, namely Conformal Inference, that can also be applied in time series forecasting experiments (Stankeviciute, M. Alaa & van der Schaar, 2021).

In the context of model retraining and updating strategies, instead, Spiliotis & Petropoulos (2024) is the main only work related to time series forecasting. The authors explored extensively the effects of different retraining scenarios and different forms of model parameter updates on the model performance, although, they focused on the exponential smoothing family of models following the traditional local approach. (Huber & Stuckenschmidt, 2020) briefly discussed retraining in the context of retail demand, but with few models and retraining scenarios, and on a proprietary daily dataset only. Despite the findings being promising, there has been little direct exploration of whether global models specifically require frequent retraining or if they can retain competitive accuracy with less frequent updates. However, this topic is under consideration of the broader machine learning community (Getzner, Charpentier & Günnemann, 2023), advocating for Green AI (Schwartz et al., 2020).

Our study builds upon these existing works, directly investigating the necessity of retraining in global forecasting models. By evaluating many different retraining strategies and their impact on the forecast accuracy of several global models, we aim to provide both theoretical insights and practical recommendations for sustainable forecasting practices.

3. Experimental design

This section describes the empirical analysis we conducted to study whether less frequent retraining scenarios may produce similar accuracy results concerning the baseline scenario (that with the highest retraining frequency). First, we describe the datasets used in the experiments, and then explain the different machine learning and deep learning models adopted. The performance measures, the several possible scenarios, and the strategy used to evaluate the forecasts are also discussed.

3.1. Datasets

For the experiment, we used two retail forecasting datasets: the M5 and the VN1 competition datasets. The M5 competition was organized by Spyros Makridakis and his colleagues as part of the M-competitions series, which aimed to compare different forecasting methods in the context of retail demand (Makridakis, Spiliotis & Assimakopoulos, 2022b). The M5 (Howard & Makridakis, 2020) is a well-known and well-studied dataset containing 3.049 daily time series of Walmart's unit sales of products. It covers the sales of three categories of products (Food, Hobbies, and Household) sold into ten different stores located in three US states (California, Texas, and Wisconsin). The time period spans from 2011 until 2016. The time series are highly intermittent and are hierarchically organized, allowing forecasting at multiple levels such as individual products, product categories, stores, and States. Exogenous information that can influence sales, such as product prices, promotions, and special events (e.g., holidays) are also available. The VN1 Forecasting - Accuracy Challenge competition was jointly organized by Flieber, Syrup Tech, and SupChains in October 2024, and it is the first of its series (Vandeput, 2024). The dataset contains weekly sales of 15.053 products sold from 2020 until 2024 from e-vendors. In particular, they were mostly online from the US, and the products were directly sold to the final consumers. Contrary to the M5 dataset, where all the products were sold by a single retailer (Walmart) and from just a few stores, the VN1 dataset includes product sales of 328 warehouses from 46 different retailers. As far as we know, we are among the first to test forecasting models on this data. These sets of data are the most recent and comprehensive time series datasets related to retail demand, allowing for a good generalization of the results in the context of demand forecasting.

In both cases, in our experiment, we focused on the most disaggregated level (SKUs), since the potential benefits of retraining the models less frequently are much larger at lower levels of

Table 1: Characteristics of the different datasets used in the experiments.

Dataset	Frequency	N. Series	Min Obs per Series
M5	Daily (7)	28.298	730
VN1	Weekly (52)	15.053	157

aggregation. Moreover, we did not consider the whole set of time series to be able to consistently apply the evaluation setup described in Sections 3.4. In particular, for daily data, we kept only time series with at least two years of data (730 observations), while weekly and monthly SKUs we considered only those that had at least three years of data (157 and 36 observations respectively).

3.2. Forecasting models

In this section, we provide an overview of the global models employed for our experiments.

Let define $\mathcal{Y}$ as the set of all available time series in a dataset, such that Y_i represents a single component, and let $\mathcal{F}$ be the set of possible predictive functions, so that F corresponds to a single model [4] . Without loss of generality, we can assume that all the necessary information for prediction are contained in $\mathcal{Y}$. Under the local approach, predictions for the forecast horizon h are obtained training a model for each time series in the data set, implying that each time series has its own model, defined by its own parameters' values.

$$Y_i^h = F(Y_i, \theta_i). \tag{1}$$

On the contrary, following the global modeling framework, forecasts for are each time series are produced by a model trained on the whole data set.

$$Y_i^h = F_k(\mathcal{Y}, \Theta). \tag{2}$$

Notice how in the cross-learning methodology, the parameters Θ are not series-specific, but are common to all time series.

[4]In our setting, F can be any predictive model in the machine learning or deep learning framework. We do not consider classical statistical forecasting methods, like ARIMA or Exponential Smoothing, since in their common formulations they are purely local models.

In our experiment, we focused on analyzing the performances of global methods only since we are mainly interested in testing whether this approach, as opposed to the local one, can benefit from retraining the models less frequently. Indeed, nowadays cross-learning is the go-to approach for most industries that are involved with huge time series datasets, as in the context of retail demand forecasting, where the forecasts of many different SKUs have to be provided regularly (Januschowski et al., 2020). For a comprehensive evaluation of global modeling approaches in demand forecasting we used both traditional machine learning models and cutting-edge deep learning methodologies. The models were selected for their established performance in time series forecasting tasks and their diverse methodological approaches, enabling a wide comparison.

All the global models were trained using Python under Nixtla's framework (Nixtla, 2022). The *mlforecast* and the *neuralforecast* libraries were used to train machine learning and deep learning models efficiently.

3.2.1. Machine learning models

Machine learning models have demonstrated their effectiveness in forecasting tasks due to their ability to capture non-linear relationships in the data. Moreover, they are often easy to train and usually produce very accurate results leveraging the cross-learning approach. In this study, we experimented with Linear (Pooled) Regression and four different tree-based methods.

Linear Regression (LR) is a classical statistical model that assumes a linear relationship between input features and the target variable. Despite its simplicity, LR can be effective for time series forecasting when combined with appropriate feature engineering. The Pooled Regression is considered a solid benchmark for global model performance evaluation ((Montero-Manso & Hyndman, 2021), (Godahewa, Bergmeir, Webb, Hyndman & Montero-Manso, 2021b)), and it has also proven to be quite effective (Bandara, Hewamalage, Godahewa & Gamakumara, 2022).

Random Forest (RF) is an ensemble learning method based on regression trees. It constructs multiple trees during training and aggregates their predictions through averaging. RF excels in capturing non-linear patterns and interactions between variables and it is robust to overfitting making it a strong tree-based model for demand forecasting (Breiman, 2001). It was the method used by Amazon until 2015 to forecast e-commerce products demand (Januschowski et al., 2022) [5].

[5] For the cost of computation, the Random Forest model is tested only on the VN1 weekly dataset.

Extreme Gradient Boosting (XGBoost) is a gradient-boosted decision tree algorithm designed for speed and performance. Iteratively optimizing an objective function combines weak learners' predictions to form a solid predictive model (Chen & Guestrin, 2016). XGBoost incorporates regularization techniques to prevent overfitting, making it particularly effective for datasets with complex relationships (Shwartz-Ziv & Armon, 2022).

Light Gradient Boosting Machine (LGBM) is another gradient-boosted tree algorithm that emphasizes efficiency and scalability. It uses a histogram-based approach to split features and a leaf-wise growth strategy to reduce computation time. LGBM is particularly well-suited for large datasets and high-dimensional data, often outperforming other boosting algorithms in terms of speed and accuracy (Ke, Meng, Finley, Wang, Chen, Ma, Ye & Liu, 2017). In demand forecasting, it is among the top solutions of all the major competitions (Favorita, Rossmann, M5, and VN1) (Makridakis et al. (2022a), Makridakis et al. (2022c), In & Jung (2022), Lainder & Wolfinger (2022)).

Categorical Boosting (CatBoost) is a gradient-boosted decision tree algorithm specifically designed to handle categorical data. By leveraging ordered boosting and other innovations it fosters accuracy and enhances generalization (Prokhorenkova, Gusev, Vorobev, Dorogush & Gulin, 2018). CatBoost's ability to handle categorical features without extensive preprocessing makes it advantageous for forecasting tasks involving categorical covariates. During the last years, it consistently showed top performance on many tabular data studies, becoming the go-to solution for many practitioners in the field (Shmuel, Glickman & Lazebnik (2024), Ye, Liu, Cai, Zhou & Zhan (2025)).

Machine learning models have the advantage of being easier to train with respect to their deep learning alternatives. However, they require extensive and careful feature engineering to produce accurate results (Januschowski et al., 2022). For this reason, we followed simplified versions of the M5 and VN1 top solutions to build time series features. In particular, we used lags, rolling averages, expanding averages, calendar features (year, quarter, month, week, day of week, day) and static features (store, category, location, product identifiers) based on the frequency of the dataset. Moreover, for the M5 data, we used also external features related to special events since available. The most relevant hyperparameters of the models were selected based on top performant solutions, otherwise the default values suggested by the software provider were adopted.

3.2.2. Deep learning models

Deep learning models have gained prominence in time series forecasting due to their capacity to model longer-term dependencies in the data and to easily learn hierarchical representations from raw data (Goodfellow, Bengio & Courville, 2016). In our experiment, we employed five different neural network architectures, two well-known methods and three state-of-the-art models.

Multi-Layer Perceptron (MLP) is a feedforward neural network consisting of an input layer, one or more hidden layers, and an output layer. Each layer applies a non-linear activation function to capture complex relationships in the data (Rosenblatt, 1958). MLPs are a versatile and very efficient solution. For this reason, many deep learning models specifically created for time series forecasting are MLP-based.

Recurrent Neural Networks (RNN) are designed to model sequential data by maintaining a hidden state that captures temporal dependencies. Standard RNNs, however, suffer from vanishing gradient problems, limiting their ability to learn long-term dependencies (Cho, van Merriënboer, Gulcehre, Bahdanau, Bougares, Schwenk & Bengio, 2014). Variants such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) address this limitation and are widely used for time series forecasting (Hochreiter & Schmidhuber, 1997) [6].

For a long time in deep learning, sequence modeling was synonymous with recurrent networks, yet several papers have shown that simple convolutional architectures can outperform canonical recurrent networks like LSTMs by demonstrating longer effective memory. *Temporal Convolutional Networks (TCN)* are convolutional architectures tailored for sequential data that, by employing causal convolutions and dilations, capture long-range dependencies (Van den Oord, Dieleman, Zen, Simonyan, Vinyals, Graves, Kalchbrenner, Senior & Kavukcuoglu, 2016).

Neural Basis Expansion Analysis for Time Series (NBEATS) is a deep learning model specifically designed for time series forecasting. It employs a sequence of fully connected layers organized into blocks. Each block outputs both trend and seasonal components, enabling the model to decompose and predict time series effectively. The network has an interpretable configuration that sequentially projects the signal into polynomials and harmonic basis to learn trend and seasonality components(Oreshkin et al., 2020). The *Neural Basis Expansion Analysis with Exogenous*

[6] In our work, we used only the LSTM version but, for the cost of computation, it was trained only on the VN1 weekly dataset.

(NBEATSx), allows for the inclusion of exogenous temporal variables available at the time of the prediction (Olivares, Challu, Marcjasz, Weron & Dubrawski, 2023). This method has shown state-of-the-art performance on several benchmark datasets and competitions (Anderer & Li, 2022).

Neural Hierarchical Interpolation for Time Series (NHITS) builds upon the success of NBEATS by incorporating hierarchical interpolation mechanisms to better capture time-hierarchies in time series data. Multi-rate input pooling, hierarchical interpolation, and backcast residual connections together induce the specialization of the additive predictions in different signal bands, reducing memory and computational time, thus improving the architecture's parsimony and accuracy (Challu, Olivares, Oreshkin, Garza, Mergenthaler-Canseco & Dubrawski, 2022).

Deep learning models typically do not require the extensive feature engineering step needed by their machine learning counterpart, since they create such features (like lags and rolling averages) internally. Nevertheless, they usually are more difficult to train since they have much more hyperparameters to select affecting the forecasting performance (Smyl, 2020). We trained the global deep learning models by adding static, calendar, and external features only, and relying on top competitions' solutions to set the hyperparameters' values.

3.3. Retrain scenarios

To answer our research question we explored several possible retraining scenarios. A retrain scenario or retrain window, r, is identified as a positive integer representing the frequency at which the model is re-trained or updated. Essentially, it represents how many data points need to be passed before a new training step is performed. The retrain scenarios strongly depend on the frequency of the dataset considered because the frequency drives both the forecast horizon and the business review periods. Therefore, we defined different set of retrain scenarios based on the frequencies of our datasets:

- for weekly data, $r = \{1, 2, 3, 4, 6, 8, 10, 13, 26, 52\}$
- for daily data, $r = \{7, 14, 21, 30, 60, 90, 120, 150, 180, 364\}$

For instance, in the case of daily data, $r = 7$ implies that the model is retrained every 7 new observations come in, that is, every week. Each set contains ten different values, being as exhaustive and computationally feasible as possible.

Note that we are training global models, hence the training data set is composed by the training set of each time series in the dataset. Since the datasets we considered are all aligned, this implies that, at every retrain period, we fit the model on a new training data set containing r new observations for each time series (where r is the chosen retrain scenario). Moreover, note also that we are considering only two different forms of model update: a model can either be completely updated when retraining is performed, or used as is to produce forecasts for r subsequent periods. Therefore, we are not examining the effect of hyperparameter tuning within each retrain scenario, given the expensive computational cost of this process and minor changes are expected.

Table 2: The retraining scenarios associated with each dataset. The test windows and the forecasting horizons are chosen based on the frequency of the dataset.

Dataset	Frequency	Retraining Scenarios (r)	Test Window (T)	Horizon (h)
M5	Daily (7)	7, 14, 21, 30, 60, 90, 120, 150, 180, 364	364	28
VN1	Weekly (52)	1, 2, 3, 4, 6, 8, 10, 13, 26, 52	52	13

The scenario $r = 1$ is the so-called *continuous* retraining, and it is the most expensive since it implies that the model is retrained every new observation. Theoretically, but usually also practically, it should be the most accurate forecasting scenario, because the model used to predict has been trained on all the available data points up to that moment. For this reason, we considered this scenario as the benchmark both in terms of forecasting accuracy and computational cost. For daily data only, however, the benchmark scenario is $r = 7$, because it is not common to retrain a global model every day, and usually the update is performed once a week. The scenario $r = T$ is the *no retraining* scenario, meaning that the forecasting model is fitted just once on the initial training set, then it is used to produce the forecasts for the entire test set T. It is the lowest computationally expensive but it should also be the lowest accurate. All the other scenarios such that $1 < r < T$ are considered as *periodic* updating strategies. Both the accuracy and the computing time should be non-increasing functions of r.

3.4. Evaluation setup

In time series forecasting, out-of-sample testing is crucial for assessing how well a model can generalize to unseen data. This is particularly important because patterns in historical data may not fully represent future trends, and unexpected changes in the data may occur (Tashman, 2000).

As suggested by Bergmeir & Benítez (2012), rolling origin evaluation is the most widely used and correct method for conducting such tests, offering a systematic way to evaluate forecasting models over multiple iterations.

The rolling origin evaluation process begins by dividing the time series into two parts: a training set, which is used to build the model, and a test set, which is used for evaluation. The test set always follows the training set chronologically to maintain the temporal order of the data. At each step, the model is trained on the training set and generates forecasts for a specified number of future points, the forecast horizon h. These forecasts are then compared to the actual values in the test set using a chosen evaluation metric (see Section 3.5). What makes rolling origin evaluation unique is its iterative nature. After each forecasting step, the forecast origin (the last point in the training set) is shifted forward by a fixed number of periods, the step size. The model is then retrained using the updated training set, which may either expand to include all available data up to the new forecast origin or remain fixed to a specific window size. This process is repeated until the test set is fully utilized, and the overall performance of the model is calculated by averaging the results across all the iterations.

Compared to the fixed origin evaluation setup, that allows for a single evaluation step, the rolling origin evaluation is preferable for its ability to provide a more comprehensive assessment of a model's performance under different conditions, such as varying seasonal patterns, level shifts, or data trends. By simulating multiple forecast cycles, indeed, this method reduces the risk of bias that can arise from evaluating a model a single time and at a single forecast origin Bergmeir & Benítez (2012). This approach is beneficial in industries like retail and supply chain management, where forecasts need to be continuously updated to reflect real-time changes. By simulating multiple scenarios and evaluating a model's performance across different cycles, rolling origin evaluation provides valuable insights into the robustness and reliability of the forecasting methods adopted.

In practice, the rolling origin evaluation setup can be adjusted based on the data and forecasting goals. For instance, the training set can be fixed to a rolling window of the most n recent observations (fixed window), or it can expand to include the full historical data (expanding window) Bergmeir & Benítez (2012). In real-life applications, most forecast practitioners use this second option, especially when the length of the time series is small (Petropoulos & et al., 2022). In our study, we adopted the expanding window approach, because we are interested in simulating a forecasting experiment as close as possible to the business reality. Moreover, it is also the only reasonable option when dealing with short time series, like the case for the weekly VN1 datasets.

The size of the test set, the length of the forecast horizon, and the step size for shifting the forecast origin are also customizable parameters that depend on the specific use case. Table 2 shows the parameters selected in our experiments. We set the size of the test set to cover at least one complete year so that the evaluation of the different scenarios does not depend on intra-year variations, such as any specific season or period of the year. We selected the forecast horizons based on the type of business decisions that usually daily and weekly forecasts support, i.e. operational vs strategic planning. Finally, the step size was set always equal to one to maximize the number of evaluations of each scenario.

3.5. Performance metrics

The accuracy evaluation of point predictive models is a controversial topic in time series forecasting: many metrics are available to capture models' performances but no consensus has been reached in the literature on whether one metric is better than others (Hewamalage, Ackermann & Bergmeir, 2023). However, since we are dealing with demand forecasting at SKU-level, implying that data intermittency is very much likely, all the metrics based on absolute or percentage errors are not optimal since they optimize for the median (Kolassa, 2020). Moreover, being the scale possibly different among the series, a scaled accuracy measure is preferable. For this reason, we considered the Root Mean Squared Scaled Error (RMSSE), proposed by Hyndman & Koehler (2006), to evaluate the point forecast accuracy of the models. The RMSSE is defined as:

$$\text{RMSSE} = \sqrt{\frac{\frac{1}{h}\sum_{t=n+1}^{n+h}(y_t - \hat{y}_t)^2}{\frac{1}{n-s}\sum_{t=s+1}^{n}(y_t - y_{t-s})^2}}. \tag{3}$$

The RMSSE measures the relative prediction accuracy of a forecasting method by comparing the mean squared errors of the prediction and the observed value against the mean squared errors of the seasonal naive model. This metric was the official metric used to assess the performance of models in the M5 competition (with $s = 1$) Makridakis et al. (2022b), which is one of the datasets we used in our experiments. Lower RMSSE values suggest higher model accuracy.

Our work is not limited to the evaluation of point forecasting accuracy, but we aim also to assess the probabilistic performance of the models throughout each different retrain scenario.

However, since the machine learning and deep learning models adopted for the experiments do not output density (or probabilistic) predictions by default, we relied on Conformal Inference as a general framework to build prediction intervals. Conformal Inference is a powerful tool that allows quantification of the uncertainty associated with out-of-sample predictions simply using a validation set on a point forecaster model to produce intervals at the desired levels (Vovk et al., 2005). It was originally proposed for non-temporal data, because the data exchangeability assumption, nevertheless in recent years it has been extended to be used also in the context of time series data (Stankeviciute et al., 2021). In particular, given its guaranteed coverage, distribution-free, model-agnostic, computational efficiency, and low dataset size requirements, Conformal Inference is likely the only tool available to evaluate the uncertainty in a global forecasting setting where different models are compared among several datasets and scenarios.

Once the prediction intervals are obtained, we used the Multi-Quantile Loss to comprehensively measure the goodness of the probabilistic predictions. The Quantile Loss (also known as Pinball Loss) and the Multi-Quantile Loss are defined as:

$$\mathrm{QL} = \frac{1}{h} \sum_{t=n+1}^{n+h} \left(q \cdot (y_t - \hat{y}_t) \cdot \mathbb{I}_{y_t \geq \hat{y}_t} + (1-q) \cdot (\hat{y}_t - y_t) \cdot \mathbb{I}_{y_t < \hat{y}_t} \right), \tag{4}$$

$$\mathrm{MQL} = \frac{1}{\mathcal{Q}} \sum_{q \in \mathcal{Q}} \mathrm{QL}(q). \tag{5}$$

The QL, being a proper scoring rule, allows for the correct evaluation of the probabilistic forecasts (Kolassa, 2016). Moreover, the (Weighted) MQL was the official metric used to evaluate the overall performance of the forecasting methods during the M5 Uncertainty competition (Makridakis et al., 2022c).

Note that we considered the median and 6 central prediction intervals, namely 60%, 70%, 80%, 90%, 95% and 99%, for a total of 13 different quantiles. The median and the 60% and 70% central prediction intervals provide a good description of the center of the forecast distribution, while the 90%, 95%, and 99% give information about its tails that, in retail demand forecasting problems are fundamental to determine the appropriate safety stock levels (Barrow & Kourentzes, 2016). These quantiles provide sufficient information about the uncertainty of the forecasts and allow for the effective description of the whole distribution.

Note also that we computed conformal prediction intervals on a validation set at least twice as long as the forecast horizon. This was the main reason that forced us to reduce the number of time series for weekly and monthly datasets, that is to be able to obtain a reliable estimate of the model uncertainty through prediction intervals.

Finally, we addressed the problem of evaluating the cost associated to producing the forecasts by calculating the Computing Time (CT). CT is defined as the time in seconds required for training and predicting the next h time steps ahead with the model. Among the possible measures of complexity, such as the number of parameters, number of iterations, model's depth, etc, CT is the most directly related to the monetary cost of forecasting since the forecasts are usually produced through some sort of cloud computing service that relies on pay-as-you-go fares (Spiliotis & Petropoulos, 2024). Hence, CT is analyzed for each forecasting model and retrain scenario combination. Lower CT values imply lower forecasting costs.

Note that we related each evaluation metric result to the baseline retraining scenario, depending on the frequency of the dataset, to be able to easily compare the performance among different models and retrain windows. Furthermore, to statistically verify our research question we tested the scenario's results using the Friedman-Nemenyi test for multiple comparisons (Demšar, 2006).

For the experiments, we used a cloud computing machine NC6s_v3 hosted on Microsoft Azure, with Linux Ubuntu 24 operating system, 1 Graphical Processing Unit, 6 Computing Processing Units, 112GB of memory. Parallelization and GPUs were used whenever possible within the model libraries. The *utilsforecast* library was used to evaluate the model performance.

4. Results and discussion

In general, the models tested performed better, in absolute terms, on the M5 dataset compared to the VN1 (see the RMSSE and MQL tables in the supplementary materials of the article) [7]. This may be due to different factors like the dataset size, the frequency of the time series, the availability of external regressors to account for promotions, events, etc, and the absence of reference hyperparameter values to use during the training process.

Figure 1 shows the forecast accuracy of each model along the different retraining scenarios for the M5 and the VN1 datasets. To simplify comparisons, for each dataset we report the results in

[7] Results are obtained by averaging over the whole dataset after trimming almost 0.5% of the time series at the extremes of the distribution of each metric. This is a standard practice, since those time series that exhibit completely out-of-scale values of the evaluation metric should be modeled separately.

relative terms with respect to the corresponding baseline scenario, that is $r = 7$ for M5 and $r = 1$ for VN1. With the only exception of the CatBoost model, the RMSSE profiles are very stable over the retraining frequencies. Indeed, the accuracy remains practically the same in the M5 dataset and even improves in the VN1 dataset, regardless of the retraining scenario considered. Especially for low periodic retraining scenarios, the performances of most global models are indistinguishable from the baseline, and even if some deterioration is present for higher retraining frequencies, this is less than 5% also for the no retraining setup. These results imply that less frequent retraining does not harm the point forecast accuracy of global models. This can be explained by the fact that, if the data remains stable without significant trends or concept drifts, as is the case for both the M5 and VN1 datasets, the forecasts will accurately track demand over extended periods.

Figure 2 statistically confirms the above results for the M5 dataset [8]. It is clear that the retraining scenarios are not statistically different in terms of point forecast accuracy at the 5% level. Indeed, even though the mean rank of the continuous retraining scenario is lower, the intervals intersect with the other scenarios, hence we can not differentiate between them. For the VN1 dataset (see supplementary materials), less frequent retrainings are even statistically significant, meaning that periodic retraining does improve accuracy over continuous retraining.

In a similar fashion to Figure 1, Figure 3 summarizes the relative accuracy in a probabilistic forecasting setting, as defined in 3.5. In this context, we observe that for the M5 dataset the accuracy (as measured by Multi Quantile Loss) is clearly an increasing function of the retrainig scenario, meaning that less frequent updates harm the probabilistic forecasting performance of the models. This is true regardless of the method used. Nonetheless, the differences in accuracy are irrelevant for low retraining scenarios and slightly more pronounced for higher retraining levels, but in any case less than 5/6% points. We also note a small difference in the performances of machine learning and deep learning models, where the formers perform consistently better as the retraining scenario increases. For the VN1 dataset, instead, we observe an almost convex relationship between the accuracy and the retraining period. On average, models' performance improves for low retraining scenarios and then starts deteriorating around $r = 4$. The only exception are NBEATSx and NHITS models, designed also for long-term forecasting, which are consistently better compared to the

[8]The Friedman-Nemenyi test is usually adopted to compare and rank the performance of different models. Here we use it to compare the accuracy produced by the same model over different retraining scenarios.

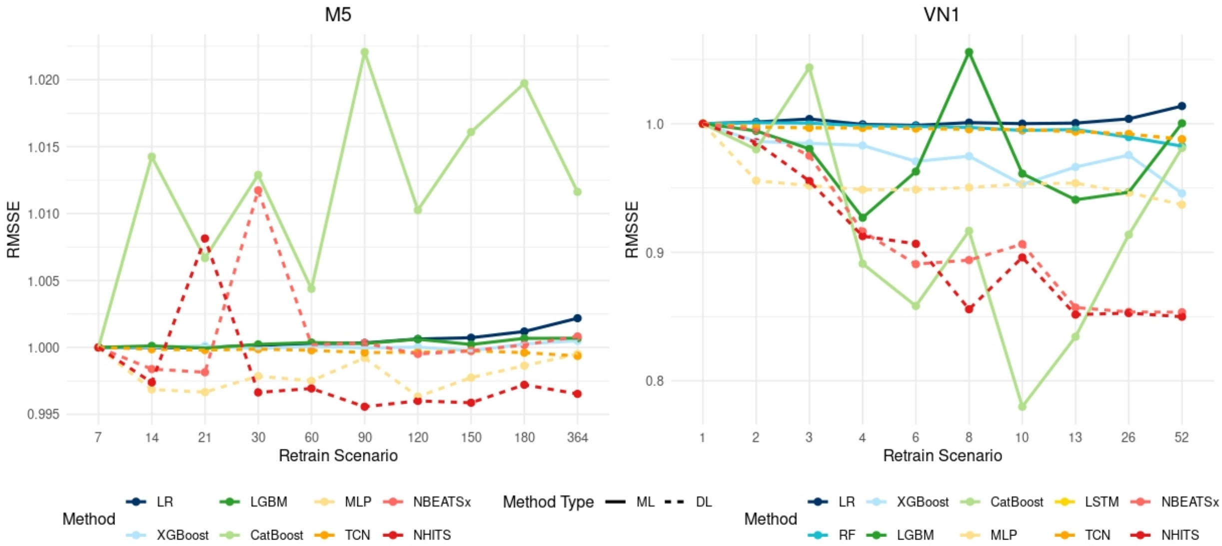

Figure 1: RMSSE results for each method and retrain scenario combination in relative terms with respect to the baseline scenario, $r = 7$ for the M5 dataset and $r = 1$ for the VN1 dataset.

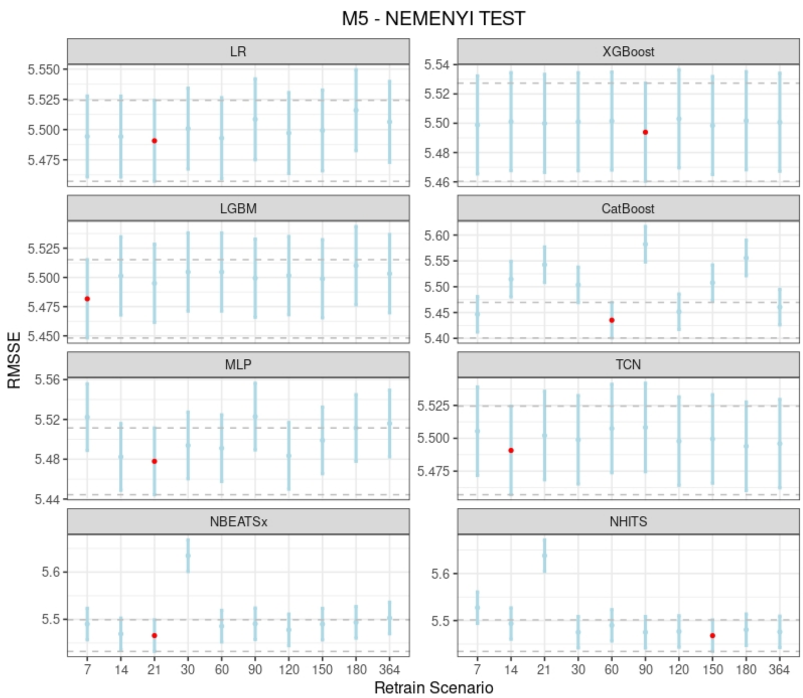

Figure 2: M5 Friedman-Nemenyi test results based on RMSSE.

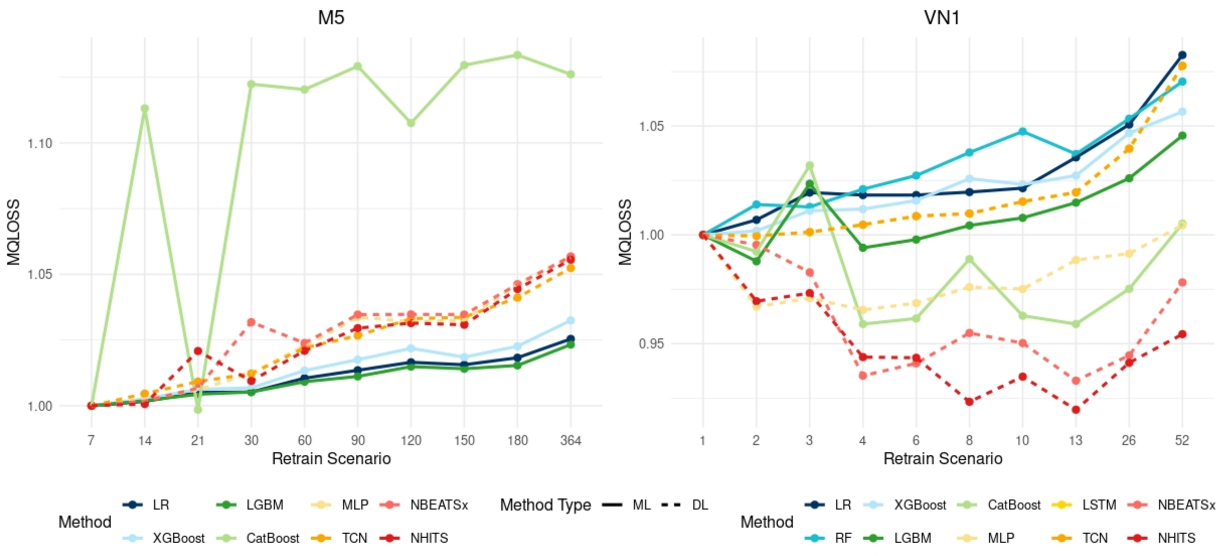

Figure 3: MQL results for each method and retrain scenario combination in relative terms with respect to the baseline scenario, $r = 7$ for the M5 dataset and $r = 1$ for the VN1 dataset.

others for longer retraining scenarios, both for point and probabilistic forecasting.

The statistical test to compare probabilistic forecasting performances among the different retraining scenarios on the M5 dataset as depicted in Figure 4 confirms that there are significant differences in accuracy. In particular, lower restraining scenarios tend to produce more accurate probabilistic forecasts, that is, less frequent retraining effectively harms the accuracy. However, this is not true for low retraining scenarios where the differences are not statistically significant, implying that even in the context of probabilistic forecasting the frequency of retaining may be reduced to at least once a month (or every two months). For the VN1 (results in supplementary materials), instead, some level of periodic retaining is better than continuous retraining most of the time.

Figure 5 shows the relative computational time of each retraining scenario for the two datasets. We observe that CT decreases exponentially as the retraining scenario increases. On average, reductions are similar for both, the M5 and the VN1: going from the baseline to the lowest periodic retraining scenario ($r = 14$ or $r = 2$) almost halves CT, retraining the models every month often reduces computing time by 75%, and this reduction reaches 90% in the no retraining scenario, where the models are trained just once. However, we observe that there is a strong difference between machine learning and deep learning models in the M5 data. While the former continuously

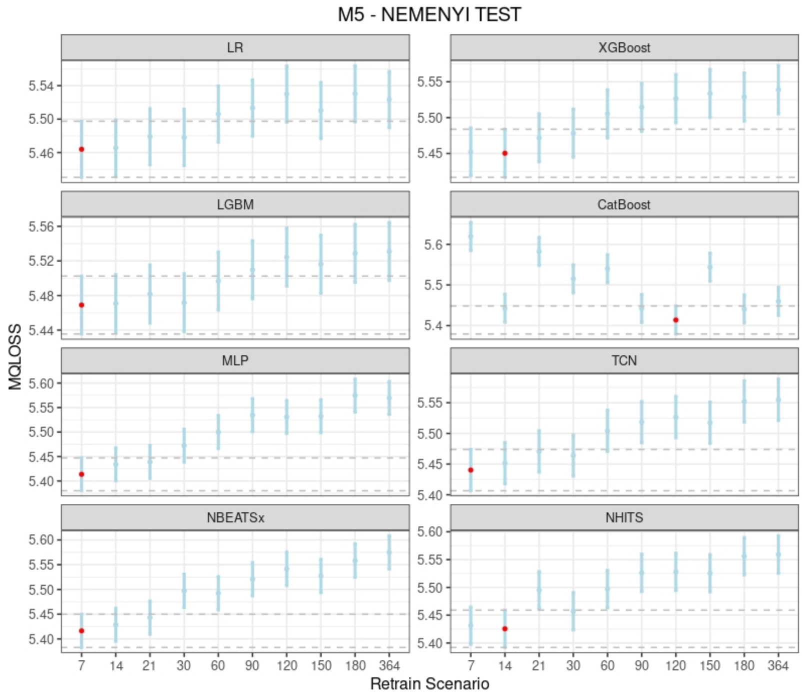

Figure 4: M5 Friedman-Nemenyi test results based on MQL.

undergoes CT reductions, the latter seems to plateau at 50%, experiencing much lower gains over increasing retraining scenarios. As expected, this difference is less evident in the VN1 dataset, since it is smaller and of lower frequency compared to the M5. These results will have direct implications in terms of costs of forecasting, which in turn may guide in the choice of the model to adopt.

Overall, the results of Figures 1, 3, and 5, combined with the Friedman-Nemenyi tests, suggest that retraining the global models less frequently does not harm (or even improves) forecast accuracy, while at the same time significantly reduces the computing time of producing the forecasts. Thus, extending the retraining time from the standard practice of continuous retaining to some level of periodic retraining allows to effectively manage the computing time, and in turn the costs of forecasting. Indeed, as already observed, the computational time can be directly translated into actual costs for the company. Following Nikolopoulos & Petropoulos (2018) and Fotios Petropoulos & Spiliotis (2024), we assumed some standard costs for computing services to estimate the costs of forecasting associated with each retrain scenario. Note that costs and savings are normalized by the number of SKUs in each datasets so that they can be directly compared and conclusions can be drawn in terms of the frequency of the time series. Figures 6, and 7 show the costs and associated

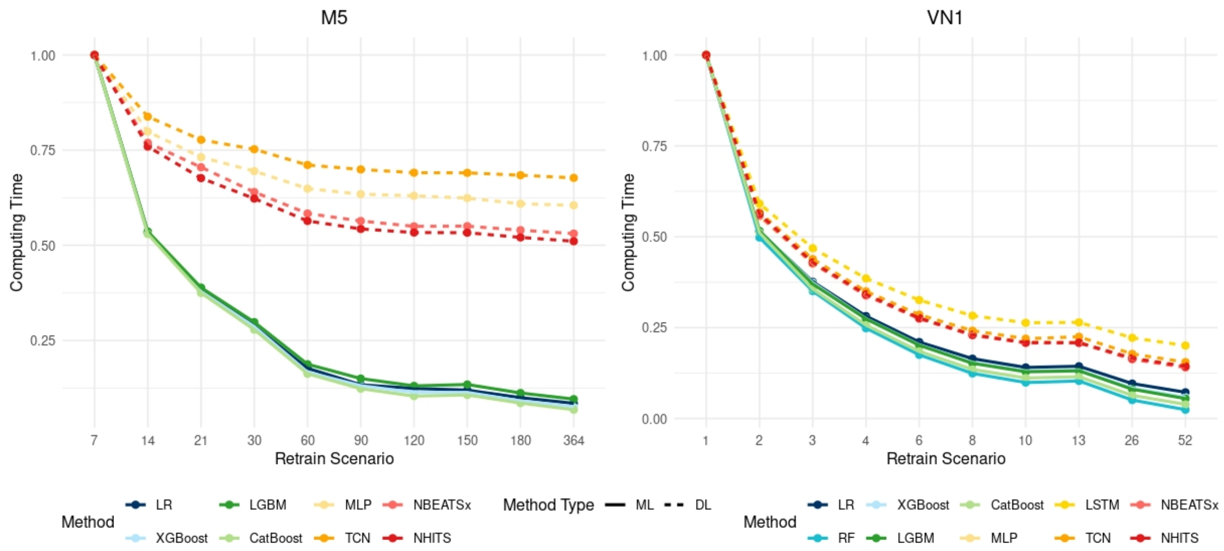

Figure 5: CT results for each method and retrain scenario combination in relative terms with respect to the baseline scenario, $r = 7$ for the M5 dataset and $r = 1$ for the VN1 dataset.

savings for a large retailer, given a fixed computing service cost of \$3.5/hour, 200,000 unique SKUs and 5,000 stores, which is approximately the size of the forecasting problem of Walmart (Spiliotis & Petropoulos, 2024). As expected, the costs decrease exponentially with less frequent retraining. For daily data, forecasting with machine learning models usually costs less than with deep learning models. On average, the continuous retraining scenario costs approximately \$750,000 and this cost drops down to almost \$250,000 in the no retraining scenario, implying direct savings of more than 60%. Moreover, machine learning models permit to reach higher savings. Indeed, even if the models have to be updated, going from retraining every week to retraining every month allows to get direct savings of almost 75%, while it can be less than 30% for deep learning models. This implies that, machine learning methods are a marginally better choice to reduce the costs of forecasting when coupled with less frequent model retraining as the frequency of the data increases. For weekly data, these differences are less evident, meaning that machine learning and deep learning models have usually very similar costs and savings profiles (without considering the Random Forest model, which is usually more than 10 times slower than other methods). This is to be expected since, lower frequencies imply also smaller datasets and lower computing time. In this case, the average cost of forecasting under a continuous retraining scenario is \$250,000 (1/3 of that of daily data) and it drops to \$15,000 in the least frequent retraining scenario, producing direct savings of 90%.

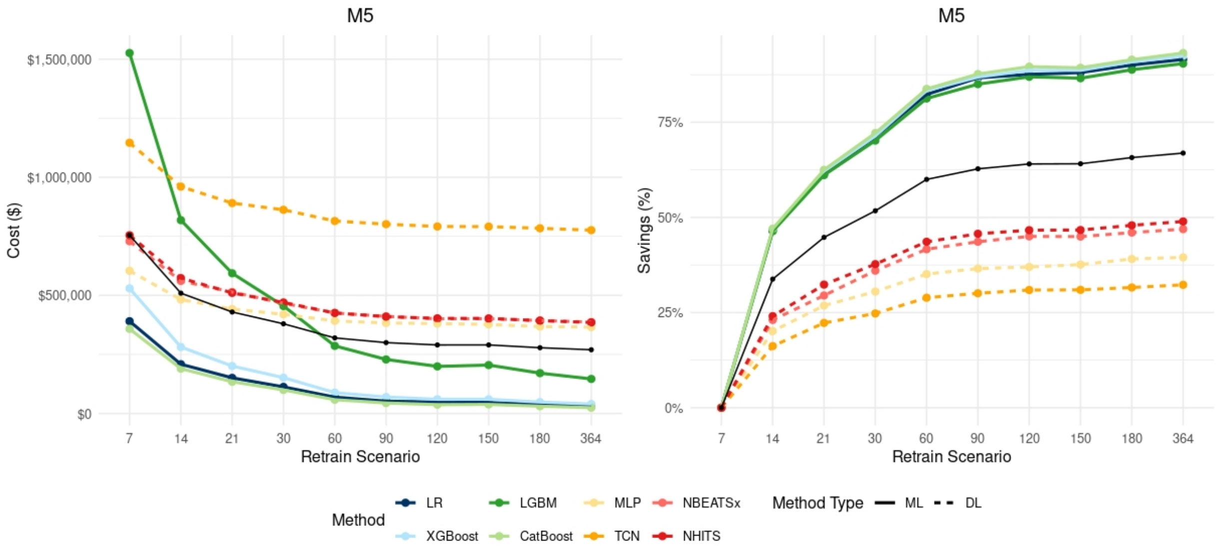

Figure 6: Daily data estimated costs and percentage savings for each method and retrain scenario combination. The black line represents the average. Costs are in real values, while savings are expressed in relative terms with respect to the baseline scenario, $r = 7$.

Nevertheless, for weekly data too, moving from retraining a model every week ($r = 1$) to just once a month ($r = 4$) allows to reduce the costs of almost 75%.

It may be argued that for a large retailer (like Walmart) these costs (and savings) may be negligible. However, it has to be noted that the cost reduction obtained from less frequent retraining comes with no shortcomings in terms of forecasting accuracy (especially for point forecasting). Indeed, as already mentioned, models under a periodic retraining scenario result in at least the same (if not even better) forecasting performance compared to the usual practice of continuous retraining. Moreover, periodic retraining may be a good practice even when probabilistic forecasts are needed, balancing costs and accuracy effectively.

5. Conclusions

In this study, we went beyond the traditional evaluation of forecasting models by also exploring the computational cost associated with generating forecasts. We examined the effects of retraining frequency on the accuracy and computational efficiency of global time series forecasting models. By systematically evaluating various retraining scenarios, ranging from continuous retraining to no retraining at all, across multiple machine learning and deep learning models, we aimed to address

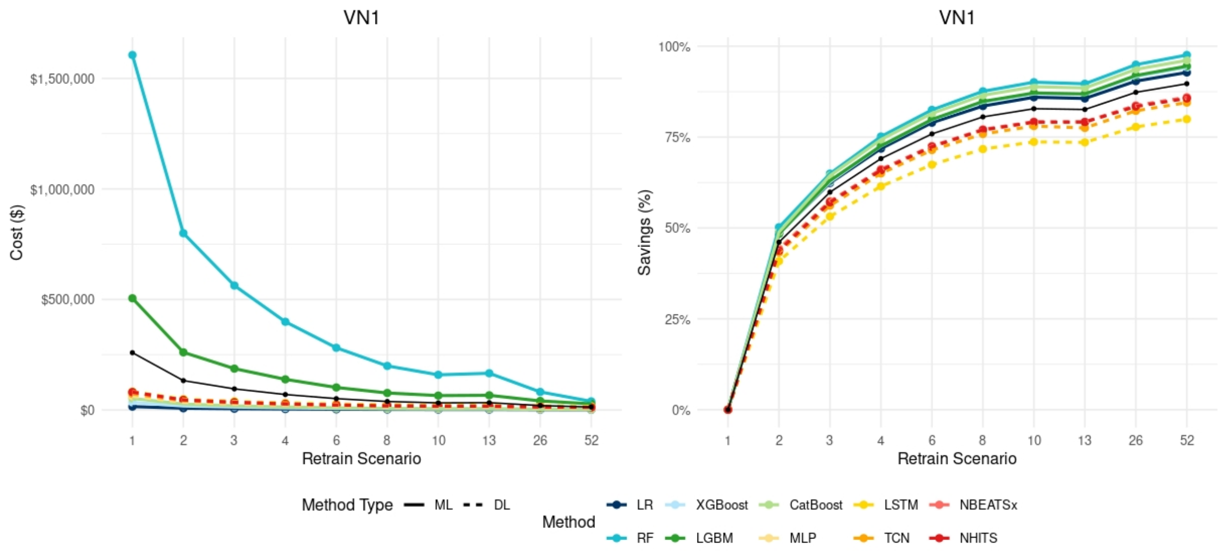

Figure 7: Weekly data estimated costs and percentage savings for each method and retrain scenario combination. The black line represents the average. Costs are in real values, while savings are expressed in relative terms with respect to the baseline scenario, $r = 1$.

a key question in forecasting: is frequent retraining necessary for maintaining high predictive performance? Our findings, derived from an extensive evaluation of ten different global models across two large-scale, real-world retail datasets, challenge the conventional wisdom of continuous model retraining. Indeed, our results indicate that the conventional practice of continuous retraining may not always be justified, and that periodic retraining strategies can offer substantial benefits in terms of computational efficiency without a significant loss in accuracy.

Our empirical evaluation provides evidence that the forecasting accuracy of global models remains stable even when retraining frequency is significantly reduced. For point forecasting, the root mean squared scaled error (RMSSE) results indicate that models trained less frequently do not exhibit substantial performance degradation. In fact, in some cases, particularly for periodic retraining frequencies, we observe marginal improvements in accuracy. For probabilistic forecasting, the results are slightly more nuanced. While less frequent retraining does lead to minor reductions in accuracy as measured by the Multi-Quantile Loss (MQL), the degradation is relatively small (typically within 5-6%). This implies that for most practical applications, especially those where forecasting costs are key considerations, periodic retraining provides a favorable balance between accuracy and efficiency. This suggests that global models, that learn shared dynamics across multiple time series, are robust to the absence of frequent updates, especially in contexts with relatively stable demand patterns.

Considering periodic retraining scenarios allows to effectively manage the computing time, and in turn the costs of forecasting. Indeed, we have shown how the computational time can be directly translated into actual costs for the company. One of the most significant findings of this study is the exponential reduction in computational costs as retraining frequency decreases. The computational time (CT) analysis demonstrates that moving from continuous retraining to a monthly retraining schedule can reduce computational costs by approximately 75%. In the extreme case of no retraining, the cost reductions approach 90%, representing a resource-saving opportunity with no-shortcomings. Furthermore, the cost analysis performed on realistic retail settings (e.g., Walmart-sized operations) reveals that periodic retraining can yield relevant financial savings. The estimated costs of forecasting, which scale with computing service expenses, decrease sharply when retraining is performed less frequently. This suggests that organizations can achieve economic benefits by optimizing their retraining schedules, without compromising the forecast quality.

The implications of these findings are highly relevant for both academic research and industry applications. First, they challenge the prevailing assumption that forecasting models require frequent updates to maintain high predictive performance. Instead, our results suggest that global models remain effective over extended periods, meaning that retraining can be strategically planned rather than performed continuously. For practitioners, our results provide concrete guidelines on how often retraining should occur. In general, retraining every monthly appears to be a viable option to balance probabilistic accuracy and costs. Instead, if the forecasting objective is on point forecast, then even longer retraining scenarios may be adopted. Moreover, our results shed some light on the computational comparison between machine learning and deep learning models. We found that the former benefit more from less frequent retraining as the frequency of the data increases. This implies that, for large-scale applications, like the retail industry, where the forecasts of many different SKUs have to be provided regularly, machine learning models are a marginally better choice to reduce the costs of forecasting when coupled with less frequent model retraining strategies as the frequency of the data increases. These insights also have broader implications for sustainability in machine learning and AI-driven forecasting. The computational cost savings associated with reduced retraining directly translate into lower energy consumption, making forecasting operations more environmentally sustainable. This aligns with recent discussions on "Green AI", which emphasizes the importance of optimizing computational resources to reduce the environmental impact of machine learning applications.

While our findings provide strong evidence for the feasibility of less frequent retraining in global forecasting models, some limitations remain. First, our study focused on two large retail datasets (M5 and VN1), which may not fully capture all possible time series patterns. Moreover, in this paper, we assumed that the data generating process remains stable without exhibiting significant trends or concept drifts, but in some real-world applications this assumption may simply not be valid. Future research could extend this analysis to other domains, such as financial time series or industrial production data, to verify the general validity of these findings. Furthermore, our study did not explore adaptive retraining strategies, where models are updated only when a significant drift in data distribution is detected. Such adaptive approaches could provide an even more refined balance between accuracy and computational efficiency. Future work could investigate the integration of drift detection mechanisms to optimize retraining schedules dynamically. Another direction for future studies may be the analysis of the effect of different retraining scenarios on the stability of global models, that is, answering the question *"Does global models produce stable forecasts even if not continuously retrained?"*. Extending the analysis to some very recent global forecasting methods that demonstrate state-of-the-art performance (e.g. transformers) would also be a natural path for future research. Finally, we encourage further studies to explore the interaction between retraining frequency and hyperparameter optimization. Indeed, different models may respond differently to retraining strategies depending on their hyperparameter configurations, and understanding these interactions could lead to even more efficient forecasting strategies.

In summary, our study shows that frequent retraining is not necessarily required to maintain high forecasting accuracy in global models. Less frequent retraining can significantly reduce computational costs while maintaining competitive forecasting performance. These findings have important implications for organizations seeking to optimize their forecasting pipelines, offering a pathway toward more efficient and sustainable forecasting practices. By shifting from a continuous retraining paradigm to a periodic retraining approach, businesses can achieve relevant cost savings while still ensuring high predictive performance.

Acknowledgements

The DataLab of the University of Milano-Bicocca for the IT resources provided.

References

Anderer, M., & Li, F. (2022). Hierarchical forecasting with a top-down alignment of independent-level forecasts. *International Journal of Forecasting*, *38*, 1405–1414. URL: https://www.sciencedirect.com/science/article/pii/S0169207021002211. doi:https://doi.org/10.1016/j.ijforecast.2021.12.015. Special Issue: M5 competition.

Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, *140*, 112896. URL: https://www.sciencedirect.com/science/article/pii/S0957417419306128. doi:https://doi.org/10.1016/j.eswa.2019.112896.

Bandara, K., Hewamalage, H., Godahewa, R., & Gamakumara, P. (2022). A fast and scalable ensemble of global models with long memory and data partitioning for the m5 forecasting competition. *International Journal of Forecasting*, *38*, 1400–1404. URL: https://www.sciencedirect.com/science/article/pii/S0169207021001783. doi:https://doi.org/10.1016/j.ijforecast.2021.11.004. Special Issue: M5 competition.

Bandara, K., Hewamalage, H., Liu, Y.-H., Kang, Y., & Bergmeir, C. (2021). Improving the accuracy of global forecasting models using time series data augmentation. *Pattern Recognition*, *120*, 108148. URL: https://www.sciencedirect.com/science/article/pii/S0031320321003356. doi:https://doi.org/10.1016/j.patcog.2021.108148.

Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., & Seaman, B. (2019). Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In T. Gedeon, K. W. Wong, & M. Lee (Eds.), *Neural Information Processing* (pp. 462–474). Cham: Springer International Publishing.

Barrow, D. K., & Kourentzes, N. (2016). Distributions of forecasting errors of forecast combinations: Implications for inventory management. *International Journal of Production Economics*, *177*, 24–33. URL: https://www.sciencedirect.com/science/article/pii/S0925527316300226. doi:https://doi.org/10.1016/j.ijpe.2016.03.017.

Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192–213. URL: https://www.sciencedirect.com/science/article/pii/S0020025511006773. doi:https://doi.org/10.1016/j.ins.2011.12.028. Data Mining for Software Trustworthiness.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. doi:10.1023/A:1010933404324.

Buonanno, A., Caliano, M., Pontecorvo, A., Sforza, G., Valenti, M., & Graditi, G. (2022). Global vs. local models for short-term electricity demand prediction in a residential/lodging scenario. *Energies*, *15*. URL: https://www.mdpi.com/1996-1073/15/6/2037. doi:10.3390/en15062037.

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., & Dubrawski, A. W. (2022). N-hits: Neural hierarchical interpolation for time series forecasting. *ArXiv*, *abs/2201.12886*. URL: https://api.semanticscholar.org/CorpusID:246430557.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* KDD '16 (p. 785–794). New York, NY, USA: Association for Computing Machinery. URL: https://doi.org/10.1145/2939672.2939785. doi:10.1145/2939672.2939785.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics. URL: https://aclanthology.org/D14-1179/. doi:10.3115/v1/D14-1179.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30. URL: http://jmlr.org/papers/v7/demsar06a.html.

Fildes, R., Ma, S., & Kolassa, S. (2022). Retail forecasting: Research and practice. *International Journal of Forecasting*, *38*, 1283–1318. URL: https://www.sciencedirect.com/science/article/pii/S016920701930192X. doi:https://doi.org/10.1016/j.ijforecast.2019.06.004. Special Issue: M5 competition.

Fotios Petropoulos, E. S., Yael Grushka-Cockayne, & Spiliotis, E. (2024). Wielding occam's razor: Fast and frugal retail forecasting. *Journal of the Operational Research Society*, *0*, 1–20. URL: https://doi.org/10.1080/01605682.2024.2421339. doi:10.1080/01605682.2024.2421339. arXiv:https://doi.org/10.1080/01605682.2024.2421339.

Gaweł, B., & Paliński, A. (2024). Global and local approaches for forecasting of long-term natural gas consumption in poland based on hierarchical short time series. *Energies*, *17*. URL: https://www.mdpi.com/1996-1073/17/2/347. doi:10.3390/en17020347.

Getzner, J., Charpentier, B., & Günnemann, S. (2023). Accuracy is not the only metric that matters: Estimating the energy consumption of deep learning models. URL: https://arxiv.org/abs/2304.00897. arXiv:2304.00897.

Godahewa, R., Bandara, K., Webb, G. I., Smyl, S., & Bergmeir, C. (2021a). Ensembles of localised models for time series forecasting. *Knowledge-Based Systems*, *233*, 107518. URL: https://www.sciencedirect.com/science/article/pii/S0950705121007802. doi:https://doi.org/10.1016/j.knosys.2021.107518.

Godahewa, R., Webb, G. I., Schmidt, D., & Bergmeir, C. (2023). Setar-tree: a novel and accurate tree algorithm for global time series forecasting. *Mach. Learn.*, *112*, 2555–2591. URL: https://doi.org/10.1007/s10994-023-06316-x. doi:10.1007/s10994-023-06316-x.

Godahewa, R. W., Bergmeir, C., Webb, G. I., Hyndman, R., & Montero-Manso, P. (2021b). Monash time series forecasting archive. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. URL: https://openreview.net/forum?id=wEc1mgAjU-.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Groß, M., & Hans, L. (2024). Leveraging potentials of local and global models for water demand forecasting. *Engineering Proceedings*, *69*. URL: https://www.mdpi.com/2673-4591/69/1/129. doi:10.3390/engproc2024069129.

Hewamalage, H., Ackermann, K., & Bergmeir, C. (2023). Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, *37*, 788–832. doi:10.1007/s10618-022-00894-5.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2022). Global models for time series forecasting: A simulation study. *Pattern Recognition*, *124*, 108441. URL: https://www.sciencedirect.com/science/article/pii/

S0031320321006178. doi:https://doi.org/10.1016/j.patcog.2021.108441.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, *9*, 1735–1780. URL: https://doi.org/10.1162/neco.1997.9.8.1735. doi:10.1162/neco.1997.9.8.1735.

Howard, A., & Makridakis, S. (2020). M5 forecasting - accuracy. https://kaggle.com/competitions/m5-forecasting-accuracy. Kaggle.

Huber, J., & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, *36*, 1420–1438. URL: https://www.sciencedirect.com/science/article/pii/S0169207020300224. doi:https://doi.org/10.1016/j.ijforecast.2020.02.005.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*, 679–688. URL: https://www.sciencedirect.com/science/article/pii/S0169207006000239. doi:https://doi.org/10.1016/j.ijforecast.2006.03.001.

Ibañez, S. C., & Monterola, C. P. (2023). A global forecasting approach to large-scale crop production prediction with time series transformers. *Agriculture*, *13*. URL: https://www.mdpi.com/2077-0472/13/9/1855. doi:10.3390/agriculture13091855.

In, Y., & Jung, J.-Y. (2022). Simple averaging of direct and recursive forecasts via partial pooling using machine learning. *International Journal of Forecasting*, *38*, 1386–1399. URL: https://www.sciencedirect.com/science/article/pii/S0169207021001813. doi:https://doi.org/10.1016/j.ijforecast.2021.11.007. Special Issue: M5 competition.

Januschowski, T., Gasthaus, J., Wang, Y., Salinas, D., Flunkert, V., Bohlke-Schneider, M., & Callot, L. (2020). Criteria for classifying forecasting methods. *International Journal of Forecasting*, *36*, 167–177. URL: https://www.sciencedirect.com/science/article/pii/S0169207019301529. doi:https://doi.org/10.1016/j.ijforecast.2019.05.008. M4 Competition.

Januschowski, T., Wang, Y., Torkkola, K., Erkkilä, T., Hasson, H., & Gasthaus, J. (2022). Forecasting with trees. *International Journal of Forecasting*, *38*, 1473–1481. URL: https://www.sciencedirect.com/science/article/pii/S0169207021001679. doi:https://doi.org/10.1016/j.ijforecast.2021.10.004. Special Issue: M5 competition.

Juan R Trapero, N. K., & Fildes, R. (2015). On the identification of sales forecasting models in the presence of promotions. *Journal of the Operational Research Society*, *66*, 299–307. URL: https://doi.org/10.1057/jors.2013.174. doi:10.1057/jors.2013.174. arXiv:https://doi.org/10.1057/jors.2013.174.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc. volume 30. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Kolassa, S. (2016). Evaluating predictive count data distributions in retail sales forecasting. *International Journal of Forecasting*, *32*, 788–803. URL: https://www.sciencedirect.com/science/article/pii/S0169207016000315. doi:https://doi.org/10.1016/j.ijforecast.2015.12.004.

Kolassa, S. (2020). Why the "best" point forecast depends on the error or accuracy measure. *International Journal* *of Forecasting*, *36*, 208–211. URL: https://www.sciencedirect.com/science/article/pii/S0169207019301359. doi:https://doi.org/10.1016/j.ijforecast.2019.02.017. M4 Competition.

Lainder, A. D., & Wolfinger, R. D. (2022). Forecasting with gradient boosted trees: augmentation, tuning, and cross-validation strategies: Winning solution to the m5 uncertainty competition. *International Journal of Forecasting*, *38*, 1426–1433. URL: https://www.sciencedirect.com/science/article/pii/S0169207021002090. doi:https://doi.org/10.1016/j.ijforecast.2021.12.003. Special Issue: M5 competition.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022a). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, *38*, 1346–1364. URL: https://www.sciencedirect.com/science/article/pii/S0169207021001874. doi:https://doi.org/10.1016/j.ijforecast.2021.11.013. Special Issue: M5 competition.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022b). The m5 competition: Background, organization, and implementation. *International Journal of Forecasting*, *38*, 1325–1336. URL: https://www.sciencedirect.com/science/article/pii/S0169207021001187. doi:https://doi.org/10.1016/j.ijforecast.2021.07.007. Special Issue: M5 competition.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., Chen, Z., Gaba, A., Tsetlin, I., & Winkler, R. L. (2022c). The m5 uncertainty competition: Results, findings and conclusions. *International Journal of Forecasting*, *38*, 1365–1385. URL: https://www.sciencedirect.com/science/article/pii/S0169207021001722. doi:https://doi.org/10.1016/j.ijforecast.2021.10.009. Special Issue: M5 competition.

Montero-Manso, P. (2023). How to leverage data for time series forecasting with artificial intelligence models: Illustrations and guidelines for cross-learning. In M. Hamoudia, S. Makridakis, & E. Spiliotis (Eds.), *Forecasting with Artificial Intelligence: Theory and Applications* (pp. 123–162). Cham: Springer Nature Switzerland. URL: https://doi.org/10.1007/978-3-031-35879-1_6. doi:10.1007/978-3-031-35879-1_6.

Montero-Manso, P., & Hyndman, R. J. (2021). Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, *37*, 1632–1653. URL: https://www.sciencedirect.com/science/article/pii/S0169207021000558. doi:https://doi.org/10.1016/j.ijforecast.2021.03.004.

Nikolopoulos, K., & Petropoulos, F. (2018). Forecasting for big data: Does suboptimality matter? *Computers & Operations Research*, *98*, 322–329. URL: https://www.sciencedirect.com/science/article/pii/S0305054817301193. doi:https://doi.org/10.1016/j.cor.2017.05.007.

Nixtla (2022). Nixtla - state-of-the-art time series and forecasting software. https://nixtlaverse.nixtla.io/. Nixtla.

Olivares, K. G., Challu, C., Marcjasz, G., Weron, R., & Dubrawski, A. (2023). Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with nbeatsx. *International Journal of Forecasting*, *39*, 884–900. URL: https://www.sciencedirect.com/science/article/pii/S0169207022000413. doi:https://doi.org/10.1016/j.ijforecast.2022.03.001.

Van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *CoRR*, *abs/1609.03499*. URL: http://dblp.uni-trier.de/db/journals/corr/corr1609.html#OordDZSVGKSK16.

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=r1ecqn4YwB.

Petropoulos, F., & et al. (2022). Forecasting: theory and practice. *International Journal of Forecasting*, *38*, 705–871. URL: https://www.sciencedirect.com/science/article/pii/S0169207021001758. doi:https://doi.org/10.1016/j.ijforecast.2021.11.001.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* NIPS'18 (p. 6639–6649). Red Hook, NY, USA: Curran Associates Inc.

Rajapaksha, D., Bergmeir, C., & Hyndman, R. J. (2023). Lomef: A framework to produce local explanations for global model time series forecasts. *International Journal of Forecasting*, *39*, 1424–1447. URL: https://www.sciencedirect.com/science/article/pii/S0169207022000978. doi:https://doi.org/10.1016/j.ijforecast.2022.06.006.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408. doi:10.1037/h0042519.

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green ai. *Commun. ACM*, *63*, 54–63. URL: https://doi.org/10.1145/3381831. doi:10.1145/3381831.

Semenoglou, A.-A., Spiliotis, E., Makridakis, S., & Assimakopoulos, V. (2021). Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, *37*, 1072–1084. URL: https://www.sciencedirect.com/science/article/pii/S0169207020301850. doi:https://doi.org/10.1016/j.ijforecast.2020.11.009.

Sen, R., Yu, H.-F., & Dhillon, I. S. (2019). Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc. volume 32. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/3a0844cee4fcf57de0c71e9ad3035478-Paper.pdf.

Shmuel, A., Glickman, O., & Lazebnik, T. (2024). A comprehensive benchmark of machine and deep learning across diverse tabular datasets. URL: https://arxiv.org/abs/2408.14817. arXiv:2408.14817.

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84–90. URL: https://www.sciencedirect.com/science/article/pii/S1566253521002360. doi:https://doi.org/10.1016/j.inffus.2021.11.011.

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *36*, 75–85. URL: https://www.sciencedirect.com/science/article/pii/S0169207019301153. doi:https://doi.org/10.1016/j.ijforecast.2019.03.017. M4 Competition.

Spiliotis, E., Makridakis, S., Semenoglou, A., & Assimakopoulos, V. (2022). Comparison of statistical and machine learning methods for daily sku demand forecasting. *Operational Research*, *22*, 3037–3061. doi:10.1007/s12351-020-00605-2. Publisher Copyright: © 2020, Springer-Verlag GmbH Germany, part of Springer Nature.

Spiliotis, E., & Petropoulos, F. (2024). On the update frequency of univariate forecasting models. *European Journal of Operational Research*, *314*, 111–121. URL: https://www.sciencedirect.com/science/article/pii/S0377221723006859. doi:https://doi.org/10.1016/j.ejor.2023.08.056.

Stankeviciute, K., M. Alaa, A., & van der Schaar, M. (2021). Conformal time-series forecasting. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in Neural Information Processing Systems* (pp. 6216–6228). Curran Associates, Inc. volume 34. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/312f1ba2a72318edaaa995a67835fad5-Paper.pdf.

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645–3650). Florence, Italy: Association for Computational Linguistics. URL: https://aclanthology.org/P19-1355/. doi:10.18653/v1/P19-1355.

Svetunkov, I., & Boylan, J. E. (2023). iets: State space model for intermittent demand forecasting. *International Journal of Production Economics*, *265*, 109013. URL: https://www.sciencedirect.com/science/article/pii/S0925527323002451. doi:https://doi.org/10.1016/j.ijpe.2023.109013.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, *16*, 437–450. URL: https://www.sciencedirect.com/science/article/pii/S0169207000000650. doi:https://doi.org/10.1016/S0169-2070(00)00065-0. The M3- Competition.

Vandeput, N. (2024). Vn1 forecasting - accuracy challenge. https://www.datasource.ai/en/home/data-science-competitions-for-startups/phase-2-vn1-forecasting-accuracy-challenge/description. DataSource.ai.

Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic Learning in a Random World*. Berlin, Heidelberg: Springer-Verlag.

Ye, H.-J., Liu, S.-Y., Cai, H.-R., Zhou, Q.-L., & Zhan, D.-C. (2025). A closer look at deep learning methods on tabular datasets. URL: https://arxiv.org/abs/2407.00956. arXiv:2407.00956.

Supplementary material

In this section, we provide tables and figures related to the empirical results of the M5 and VN1 datasets.

The Tables 3 and 4 show the forecast accuracy of the different models along the examined retrain scenarios for the M5 daily dataset, while Table 5 depicts the computing time in seconds.

Method	7	14	21	30	60	90	120	150	180	364
LR	0.777	0.777	0.777	0.777	0.777	0.777	0.777	0.777	0.778	0.779
XGBoost	0.755	0.755	0.755	0.755	0.755	0.755	0.755	0.754	0.755	0.755
LGBM	0.771	0.772	0.771	0.772	0.772	0.772	0.772	0.772	0.772	0.772
CatBoost	0.947	0.961	0.954	0.960	0.952	0.968	0.957	0.963	0.966	0.958
MLP	0.821	0.818	0.818	0.819	0.819	0.820	0.818	0.819	0.819	0.820
TCN	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.864
NBEATSx	0.815	0.814	0.813	0.825	0.815	0.815	0.815	0.815	0.815	0.816
NHITS	0.828	0.826	0.835	0.825	0.826	0.825	0.825	0.825	0.826	0.825

Table 3: M5 RMSSE values for each method and retrain scenario combination.

The Tables in 6 and 7 show the forecast accuracy of the different models along the examined retrain scenarios for the VN1 weekly dataset, while 8 depicts the computing time in seconds.

Figures 8 and 9 show the results of the Friedman-Nemenyi test on in the context of both point and probabilistic forecasting for the VN1 dataset.

The costs tables 9 and 10 show the estimated cost in real values of each scenario for daily and weekly data respectively.

Method	7	14	21	30	60	90	120	150	180	364
LR	0.267	0.268	0.269	0.269	0.270	0.271	0.272	0.271	0.272	0.274
XGBoost	0.258	0.259	0.260	0.260	0.262	0.263	0.264	0.263	0.264	0.267
LGBM	0.256	0.256	0.257	0.257	0.258	0.259	0.260	0.259	0.260	0.262
CatBoost	0.263	0.292	0.262	0.295	0.294	0.296	0.291	0.297	0.298	0.296
MLP	0.281	0.282	0.283	0.285	0.288	0.291	0.290	0.291	0.294	0.297
TCN	0.290	0.291	0.292	0.293	0.296	0.297	0.299	0.299	0.301	0.305
NBEATSx	0.279	0.280	0.281	0.288	0.286	0.289	0.289	0.289	0.292	0.295
NHITS	0.284	0.284	0.290	0.287	0.290	0.292	0.293	0.293	0.297	0.300

Table 4: M5 MQL values for each method and retrain scenario combination.

Method	7	14	21	30	60	90	120	150	180	364
LR	11,372	6,068	4,388	3,275	2,012	1,517	1,410	1,358	1,133	964
XGBoost	15,417	8,189	5,843	4,413	2,567	2,022	1,748	1,750	1,402	1,160
LGBM	44,428	23,835	17,279	13,245	8,332	6,656	5,802	5,966	4,971	4,256
CatBoost	10,423	5,521	3,907	2,901	1,693	1,285	1,082	1,113	891	709
MLP	17,583	14,044	12,864	12,219	11,411	11,156	11,080	10,972	10,711	10,639
TCN	33,363	27,960	25,931	25,102	23,721	23,332	23,046	23,033	22,825	22,591
NBEATSx	21,226	16,331	14,965	13,583	12,387	11,969	11,670	11,681	11,458	11,265
NHITS	21,969	16,684	14,863	13,682	12,390	11,929	11,722	11,712	11,440	11,219

Table 5: M5 CT values for each method and retrain scenario combination.

Method	1	2	3	4	6	8	10	13	26	52
LR	6.549	6.557	6.573	6.545	6.540	6.554	6.549	6.552	6.574	6.639
RF	1.868	1.869	1.869	1.865	1.864	1.862	1.858	1.859	1.848	1.835
XGBoost	1.890	1.863	1.861	1.858	1.834	1.842	1.800	1.826	1.843	1.787
LGBM	3.542	3.523	3.473	3.283	3.411	3.74	3.405	3.333	3.354	3.544
CatBoost	5.762	5.647	6.014	5.134	4.945	5.282	4.494	4.807	5.264	5.653
MLP	1.543	1.475	1.469	1.464	1.464	1.467	1.471	1.472	1.46	1.446
LSTM	1.913	1.908	1.907	1.907	1.906	1.905	1.904	1.901	1.898	1.89
TCN	1.913	1.908	1.907	1.907	1.906	1.905	1.904	1.901	1.898	1.89
NBEATSx	1.698	1.69	1.656	1.555	1.512	1.518	1.539	1.455	1.449	1.449
NHITS	1.699	1.674	1.623	1.55	1.54	1.454	1.522	1.447	1.449	1.444

Table 6: VN1 RMSSE values for each method and retrain scenario combination.

Method	1	2	3	4	6	8	10	13	26	52
LR	2.896	2.916	2.953	2.949	2.949	2.953	2.959	2.999	3.043	3.136
RF	2.590	2.626	2.623	2.645	2.661	2.688	2.713	2.686	2.728	2.773
XGBoost	2.469	2.474	2.497	2.498	2.508	2.533	2.526	2.537	2.585	2.609
LGBM	2.625	2.593	2.687	2.609	2.619	2.636	2.646	2.664	2.693	2.745
CatBoost	2.845	2.823	2.936	2.728	2.736	2.813	2.739	2.728	2.774	2.860
MLP	2.492	2.410	2.420	2.406	2.414	2.432	2.430	2.463	2.471	2.503
LSTM	2.843	2.842	2.847	2.856	2.868	2.871	2.887	2.899	2.956	3.064
TCN	2.843	2.842	2.847	2.856	2.868	2.871	2.887	2.899	2.956	3.064
NBEATSx	2.626	2.613	2.580	2.456	2.470	2.507	2.495	2.450	2.480	2.568
NHITS	2.632	2.552	2.562	2.485	2.484	2.430	2.461	2.421	2.478	2.512

Table 7: VN1 MQL values for each method and retrain scenario combination.

Method	1	2	3	4	6	8	10	13	26	52
LR	236	121	89	66	50	39	33	34	23	17
RF	24,862	12,380	8,715	6,177	4,355	3,084	2,461	2,570	1,262	602
XGBoost	530	274	198	146	108	81	69	70	44	31
LGBM	7,824	4,037	2,894	2,144	1,576	1,188	1,007	1,027	630	429
CatBoost	805	412	287	207	149	109	90	93	51	31
MLP	962	534	409	327	265	223	202	202	159	138
LSTM	1,284	759	602	495	418	363	338	340	285	258
TCN	1,127	639	494	394	322	272	247	253	200	175
NBEATSx	1,244	694	531	421	342	285	258	259	203	175
NHITS	1,251	704	537	427	346	288	261	261	208	180

Table 8: VN1 CT values for each method and retrain scenario combination.

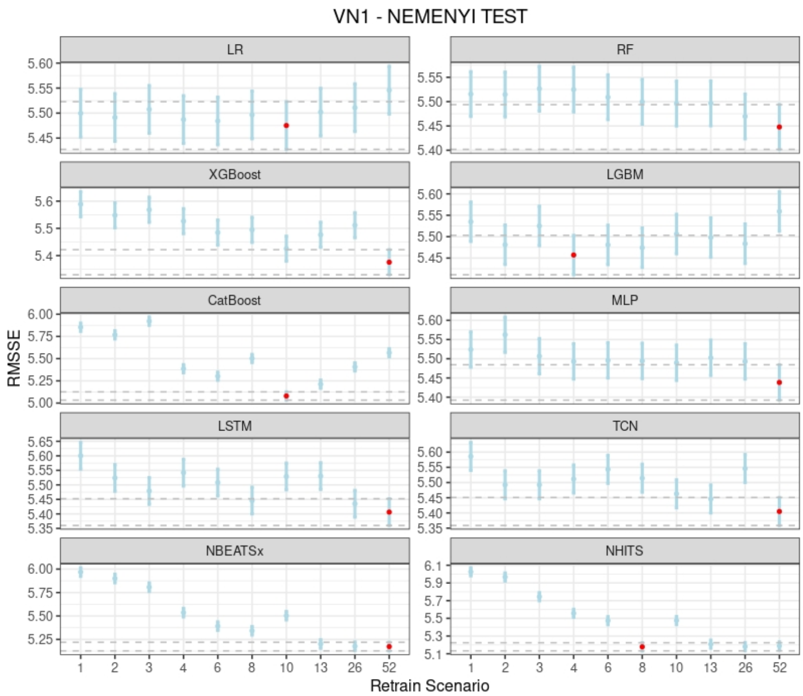

Figure 8: VN1 Friedman-Nemenyi test results based on RMSSE.

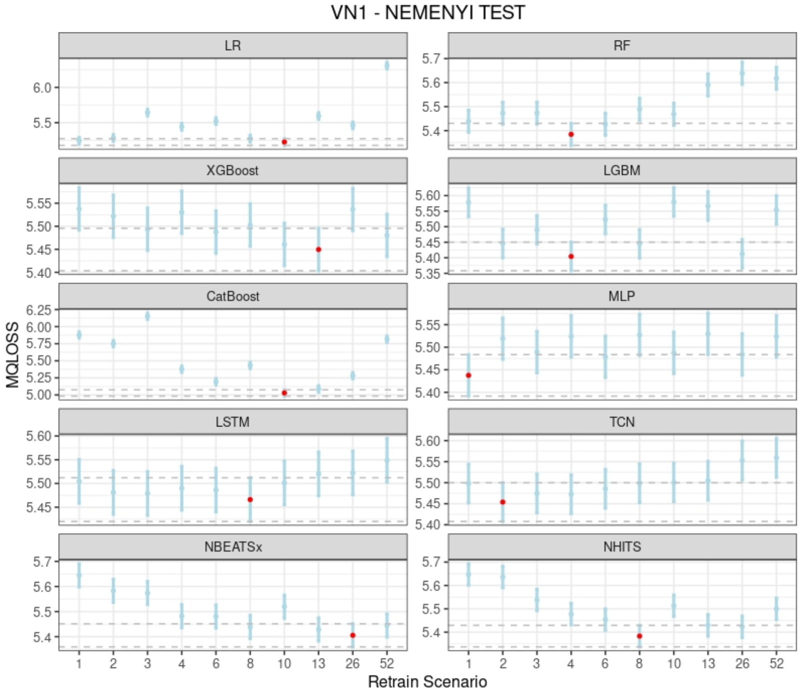

Figure 9: VN1 Friedman-Nemenyi test results based on MQL.

Method	7	14	21	30	60	90	120	150	180	364
LR	390,732$	208,499$	150,791$	112,547$	69,132$	52,131$	48,474$	46,678$	38,937$	33,151$
XGBoost	529,679$	281,358$	200,778$	151,635$	88,200$	69,495$	60,082$	60,155$	48,189$	39,861$
LGBM	1,526,424$	818,569$	593,676$	455,075$	286,262$	228,698$	199,337$	204,972$	170,802$	146,252$
CatBoost	358,123$	189,714$	134,258$	99,699$	58,167$	44,164$	37,206$	38,266$	30,622$	24,362$
MLP	604,120$	482,505$	441,981$	419,823$	392,046$	383,287$	380,697$	376,968$	368,023$	365,525$
TCN	1,146,256$	960,626$	890,924$	862,432$	815,006$	801,612$	791,801$	791,348$	784,201$	776,181$
NBEATSx	729,263$	561,105$	514,166$	466,683$	425,593$	411,232$	400,953$	401,352$	393,673$	387,058$
NHITS	754,783$	573,233$	510,655$	470,077$	425,684$	409,844$	402,731$	402,394$	393,071$	385,470$
Average	754,922$	509,451$	429,654$	379,746$	320,011$	300,058$	290,160$	290,267$	278,440$	269,733$

Table 9: M5 estimated costs for each method and retrain scenario combination.

Method	1	2	3	4	6	8	10	13	26	52
LR	15,234$	7,839$	5,731$	4,292$	3,205$	2,508$	2,140$	2,190$	1,463$	1,099$
RF	1,605,768$	799,597$	562,896$	398,954$	281,246$	199,214$	158,959$	165,975$	81,529$	38,865$
XGBoost	34,254$	17,687$	12,796$	9,413$	6,944$	5,262$	4,473$	4,534$	2,863$	2,005$
LGBM	505,304$	260,721$	186,904$	138,460$	101,775$	76,738$	65,013$	66,335$	40,721$	27,701$
CatBoost	52,021$	26,594$	18,515$	13,366$	9,611$	7,022$	5,798$	5,977$	3,311$	2,023$
MLP	62,102$	34,489$	26,431$	21,149$	17,121$	14,391$	13,027$	13,015$	10,255$	8,891$
LSTM	82,927$	49,019$	38,851$	31,962$	27,022$	23,465$	21,829$	21,932$	18,415$	16,652$
TCN	72,763$	41,291$	31,910$	25,464$	20,791$	17,579$	15,983$	16,365$	12,936$	11,284$
NBEATSx	80,337$	44,837$	34,283$	27,179$	22,057$	18,387$	16,686$	16,702$	13,085$	11,277$
NHITS	80,776$	45,458$	34,688$	27,585$	22,336$	18,607$	16,888$	16,855$	13,405$	11,601$
Average	259,149$	132,753$	95,301$	69,782$	51,211$	38,317$	32,080$	32,988$	19,798$	13,140$

Table 10: VN1 estimated costs for each method and retrain scenario combination.