Forecasting with Deep Learning: Beyond Average of Average of Average Performance*

Vitor Cerqueira[1,2], Luis Roque[1,2], and Carlos Soares[1,2,3]

[1] Faculdade de Engenharia da Universidade do Porto, Porto, Portugal
vcerqueira@fe.up.pt
[2] Laboratory for Artificial Intelligence and Computer Science (LIACC), Portugal
[3] Fraunhofer Portugal AICOS, Portugal

Abstract. Accurate evaluation of forecasting models is essential for ensuring reliable predictions. Current practices for evaluating and comparing forecasting models focus on summarising performance into a single score, using metrics such as SMAPE. We hypothesize that averaging performance over all samples dilutes relevant information about the relative performance of models. Particularly, conditions in which this relative performance is different than the overall accuracy. We address this limitation by proposing a novel framework for evaluating univariate time series forecasting models from multiple perspectives, such as one-step ahead forecasting versus multi-step ahead forecasting. We show the advantages of this framework by comparing a state-of-the-art deep learning approach with classical forecasting techniques. While classical methods (e.g. `ARIMA`) are long-standing approaches to forecasting, deep neural networks (e.g. `NHITS`) have recently shown state-of-the-art forecasting performance in benchmark datasets. We conducted extensive experiments that show `NHITS` generally performs best, but its superiority varies with forecasting conditions. For instance, concerning the forecasting horizon, `NHITS` only outperforms classical approaches for multi-step ahead forecasting. Another relevant insight is that, when dealing with anomalies, `NHITS` is outperformed by methods such as `Theta`. These findings highlight the importance of aspect-based model evaluation.

Keywords: Forecasting · Time Series · Deep Learning · Evaluation

1 Introduction

Time series forecasting is a relevant problem in various application domains, such as finance, meteorology, or industry. The generalized interest in this task

* This work was partially funded by projects AISym4Med (101095387) supported by Horizon Europe Cluster 1: Health, ConnectedHealth (n.º 46858), supported by Competitiveness and Internationalisation Operational Programme (POCI) and Lisbon Regional Operational Programme (LISBOA 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF) and NextGenAI - Center for Responsible AI (2022-C05i0102-02), supported by IAPMEI, and also by FCT plurianual funding for 2020-2023 of LIACC (UIDB/00027/2020 UIDP/00027/2020)

led to the development of several solutions over the past decades. The accurate evaluation of forecasting models is essential for comparing different approaches and ensuring reliable predictions. The typical approach for evaluating forecasts is conducted by averaging performance across all samples using metrics such as SMAPE (symmetric mean absolute percentage error) [23]. As such, the estimated accuracy of a model is an average computed over multiple time steps and forecasting horizons and also across a collection of time series.

Averaging forecasting performance into a single value is convenient because it provides a simple way of quantifying the performance of models and selecting the best one among a pool of alternatives. However, these averages dilute information that might be relevant to users. For instance, conditions in which the relative performance of several models is different than the overall accuracy, or scenarios in which models do not behave as expected. The real-world applicability of a model may depend on how it performs under certain conditions[4] that are not captured by averaging metrics over all samples.

We address this limitation by proposing a novel framework for evaluating univariate time series forecasting models. Our approach deviates from prior works by controlling forecasting performance by various factors. We aim to uncover insights that might be obscured when error metrics are condensed into a single value. By controlling experiments across several conditions such as sampling frequency or forecasting horizon, we provide a more nuanced understanding of how different models perform under diverse scenarios. A more granular analysis enables us to pinpoint the strengths and weaknesses of different methods. This knowledge is valuable for practitioners as well as future research on forecasting methods.

We showcase the advantages of the proposed framework by comparing a state-of-the-art deep learning approach with classical forecasting techniques. While traditional methods such as `ARIMA` [16] or exponential smoothing [14] are well-established, deep learning has recently emerged as a powerful alternative [26]. Several deep neural network architectures have exhibited competitive performance in benchmark competitions. These include `ES-RNN` [27], `N-BEATS` [26], or `NHITS` [9], among others. The comparison of forecasting methods based on artificial neural networks with classical approaches is a topic that has been studied for a long time [28,24].

We conduct an extensive empirical analysis comparing the deep learning approach `NHITS` [9] with several classical forecasting methods, including `ARIMA` or `Seasonal Naive` [16]. We select `NHITS` in particular as it has shown competitive forecasting performance with other neural networks, including `N-BEATS` [26], and state-of-the-art recurrent neural networks and transformers [9]. We evaluate several approaches in different conditions, such as varying sampling frequency, anomalous observations, or increasing forecasting horizons. While `NHITS` generally performs best, its superiority varies with forecasting conditions. For instance, in terms of forecasting horizon, `NHITS` only outperforms classical approaches for

[4] Other factors may be relevant, such as computational efficiency, ease of implementation, or interpretability, but these are out of the scope of this work.

multi-step ahead forecasting. When dealing with anomalies, `NHITS` is outperformed by methods such as `Theta`. In the interest of reproducible science, the experiments are available and fully reproducible[5].

The rest of this paper is organized as follows. Section 2 provides a background to this work, including the definition of the forecasting problem and several modeling approaches used to tackle it. In Section 3, we describe the materials and methods used in the empirical analysis carried out. The experiments and respective results are presented in Section 4 and discussed in Section 5. We conclude the paper in Section 6.

2 Background

This section overviews several topics related to our work. We start by defining the problem and outlining a few time series models (Section 2.1). In Section 2.2, we elaborate on auto-regressive approaches focusing on how deep learning methods leverage multiple time series to build global forecasting models. Section 2.3 overviews past works that compare artificial neural networks with classical approaches for univariate time series forecasting. Finally, we briefly overview evaluation practices used in forecasting problems (Section 2.4).

2.1 Time Series Forecasting

A univariate time series is defined as a temporal sequence of values $Y = \{y_1, y_2, \ldots, y_t\}$, where $y_i \in \mathcal{Y} \subset \mathbb{R}$ is the value of Y at the i-th timestep and t is the size of Y. We address univariate time series forecasting tasks, where the goal is to predict the value of upcoming observations of the time series, $y_{t+1}, \ldots, y_{t+H}$, where H denotes the forecasting horizon.

There are several approaches to tackle this problem. One of the simplest methods is seasonal naive, which predicts the future values of a time series according to the last known observation of the same season. `ARIMA` and exponential smoothing are two long-standing classical approaches to forecasting [15]. `ARIMA` models time series according to a linear combination of past values along with a linear combination of past errors, plus a differencing operation for integrated time series. Similarly to auto-regression, exponential smoothing models time series based on a linear combination of past observations. The simple exponential smoothing model involves a weighted average of the past values, where the weight decays exponentially as the observations are older [10].

2.2 Forecasting with Deep Learning

With machine learning, forecasting problems are framed as a supervised learning problem according to an auto-regressive type of modeling. A dataset is built using time delay embedding [5]. Time delay embedding denotes the process

[5] `https://github.com/vcerqueira/modelradar`

of reconstructing a time series into the Euclidean space by applying sliding windows. This results in a dataset $\mathcal{D} = \{< X_i, y_i >\}_{i=p+1}^{t}$ where y_i represents the i-th observation and $X_i \in \mathbb{R}^p$ is the i-th corresponding set of p lags: $X_i = \{y_{i-1}, y_{i-2}, \ldots, y_{i-p}\}$.

Forecasting problems often involve time series databases that contain multiple univariate time series. We define a time series databases as $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_n\}$, where n is the number of time series in the collection. In these scenarios, forecasting approaches fall into one of two categories: local or global [17]. Local methods build a model for each time series in a database. Classical forecasting techniques usually follow this approach. On the other hand, global methods train a single model using all time series in the database. Using several time series to train a model has been shown to lead to better forecasting performance [11]. The intuition for this effect is that the time series in a database are often related, for example, the demand time series of different related retail products. Global models can learn useful patterns in some time series that are not revealed in others, while local approaches only learn dependencies across time.

The training process of global forecasting models involves combining the data from various time series during the data preparation stage. Specifically, the training dataset $\mathcal{D}$ for a global model is composed of a concatenation of the individual datasets: $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$, where $\mathcal{D}_j$ is the dataset corresponding to the time series Y_j. The auto-regressive formulation described above is applied to the combined dataset.

Several neural architectures have recently shown competitive forecasting performance in benchmark competitions. One of these is `NHITS` [9], short for Neural Hierarchical Interpolation for Time Series Forecasting. Similarly to its predecessor `N-BEATS` [26], `NHITS` is based on stacks that contain blocks of multi-layer perceptrons (`MLP`) along with residual connections. The architecture behind `NHITS` also features other relevant aspects, such as multi-rate input sampling that models data with different scales or hierarchical interpolation for better long-horizon forecasting. `NHITS` has shown state-of-the-art forecasting performance relative to other deep learning approaches, including various transformers and state-of-the-art recurrent-based neural networks [9]. Moreover, `NHITS` is significantly superior in terms of computational scalability relative to other neural-based approaches.

2.3 Comparing Deep Learning with Classical Methods

Several previous works have addressed the comparison of methods based on artificial neural networks with classical approaches for forecasting. Hill et al. [13] pioneering work shows that `MLPs` exhibit a competitive performance with classical approaches such as `ARIMA`. Tang et al. [28] also compare `MLPs` with `ARIMA`-based methods and report that `MLPs` have a competitive forecasting performance. One key finding is that the neural network performed better for long-term forecasting, while `ARIMA` was better for the short-term.

Ahmed et al. [1] compare different machine learning algorithms for time series forecasting and conclude that `MLPs` and Gaussian Processes exhibit the best performance. In a seminal work, Makridakis et al. [24] extend the study by

Ahmed et al. [1] by including classical approaches such as `ARIMA` or exponential smoothing. They conclude that most classical approaches, including naive, outperformed machine learning methods, including neural network algorithms. However, this study is biased towards time series dataset with a low sample size [7], where neural networks become heavily over-parametrized [29].

The M4 forecasting competition [23], which featured 100,000 from various application domains, represents an important mark for understanding the relative performance of forecasting methods. This competition was won by an approach called `ES-RNN` [27] that combines exponential smoothing with an `LSTM` neural network trained globally. The subsequent M5 forecasting competition [25] included 42,840 time series from a retail company. One of the main findings from this competition is that machine learning approaches outperformed classical methods. The winning solution was based on gradient boosting using `lightgbm` [18].

2.4 Evaluation Metrics

There are several measures to evaluate the performance of point forecasts. These fall into different categories, such as scale-dependent, scale-independent, percentage, or relative metrics. Hewamalage et al. [12] survey a comprehensive list of metrics and provide recommendations for which ones should be used in different scenarios. Overall, there is no consensus concerning what the best metric is. Nonetheless, for a sufficiently large sample size, most metrics agree on what the best forecasting model is [19,8].

In the benchmark M4 forecasting competition [23], two metrics were used for evaluation: SMAPE and MASE (mean absolute scaled error). These are defined as follows:

$$\text{SMAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{(|\hat{y}_i| + |y_i|)/2} \tag{1}$$

$$\text{MASE} = \frac{\frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\frac{1}{n-m}\sum_{i=m+1}^{n} |y_i - y_{i-m}|} \tag{2}$$

where $\hat{y}_i$, and y_i are the forecast and actual value for the i-th instance, respectively, n is the number of observations and m is the seasonal period. These and other metrics are usually computed across all available predictions points, which include multiple time steps, forecasting horizons, and time series.

3 Materials and Methods

This section describes the materials and methods used in the experimental study. First, we present the datasets and briefly summarise their characteristics (Section 3.1). Then, we list the forecasting methods tested in the experiments (Section 3.2). Then, we describe the training methodology (Section 3.3) and evaluation framework (Section 3.4).

3.1 Data

We use the following benchmark datasets that were part in past forecasting competitions:

- **M3** [22] contains a set of 3,003 time series from various application domains. The time series are split over three sampling frequencies: monthly, quarterly, and yearly;
- **Tourism** [3] contains 1,311 time series related to tourism. These also exhibit a monthly, quarterly, and yearly sampling frequency.
- **M4** [23] is a benchmark dataset with 100,000 time series from different application domains and sampling frequencies. In the interest of consistency, we use the subset of 95.000 time series that exhibit a monthly, quarterly, or yearly sampling frequency.

Table 1 provides a brief summary of the datasets. Overall, the datasets contain a total of 99,140 time series with 14,898,364 observations.

Table 1: Summary of the datasets: number of time series, number of observations, forecast horizon, number of lags, and frequency.

		# time series	# observations	H	p	Frequency
M3	Monthly	1428	167562	18	23	12
	Quarterly	756	37004	8	10	4
	Yearly	645	18319	6	8	1
M4	Monthly	48000	11246411	18	23	12
	Quarterly	24000	2406108	8	10	4
	Yearly	23000	858458	6	8	1
Tourism	Monthly	366	109280	18	23	12
	Quarterly	427	42544	8	10	4
	Yearly	518	12678	6	8	1
Total		99140	14898364	-	-	-

In terms of input size[6], we follow the heuristic described by Bandara et al. [4], which leads to competitive forecasting performance [21]. They suggest using an input size based on the forecasting horizon and the frequency of the time series. The idea is to take the maximum value between the forecasting horizon and the frequency and then multiply the result by a factor of 1.25. We also take the ceiling to get an integer value. The resulting input size varies by sampling frequency and is reported in Table 1 (column p). We remark that this approach for selecting the input size is only adopted for deep learning. The configuration of classical approaches, such as the order of auto-regression of `ARIMA`, is selected automatically according to the process detailed in the next section.

[6] also referred to as the number of lags, or lookback window

3.2 Methods

The experiments include a total of 7 forecasting approaches, 1 of which is a deep learning method. The following list describes the classical approaches:

- `ARIMA`: The auto-regressive integrated moving average method that is a standard benchmark for univariate time series forecasting. The model configuration is optimized using the Akaike Information Criterion (AIC) [16];
- `ETS`: The error, trend, and seasonality exponential smoothing method that is also optimized using AIC [14];
- `SNaive`: The seasonal naive method where forecasts are the last known observation of the same period;
- `RWD` (Random walk with drift) [15]: a variant of the naive method where the forecasts are adjusted according to the historical average of the time series;
- `SES`: The simple exponential smoothing method, with the smoothing parameter optimize by squared error minimization [14];
- `Theta` [2]: The `Theta` method, with the configuration being optimized by squared error minimization.

Regarding deep learning, we include a single architecture on the experiments for conciseness. As mentioned before, we focus on `NHITS` [9] (c.f. Section 2.3), for two main reasons: i) it is significantly more computationally efficient than other architectures (50 times faster than transformers according to Challu et al. [9]); and ii) it has shown state-of-the-art forecasting performance when compared with several other deep neural networks (e.g. [9,6]). We resorted to the nixtla framework[7] to implement all the above methods. Classical approaches are available on the *statsforecast* Python package, while `NHITS` is implemented on *neuralforecast* package.

3.3 Training Methodology

Each classical approach follows a local methodology. On the other hand, we train `NHITS` in a global fashion according to the approach described in Section 2.2. We train one `NHITS` model for each dataset listed in Table 1. For instance, one model is created with all monthly time series in the M3 dataset.

We use the default configuration of `NHITS` available on *neuralforecast*. Precisely, `NHITS` models are built with 3 stacks with a block of `MLPs`. Each `MLP` features 2 hidden layers, each with 512 hidden units. The activation function is set to the rectified linear unit, and `NHITS` is trained for a maximum of 1500 training steps using ADAM optimizer and a learning rate of 0.001. We use early stopping (with 50 patience steps) and model checkpointing to drive the training process.

[7] `https://nixtlaverse.nixtla.io/`

3.4 Evaluation Framework

The test set is composed of the last H (one complete forecasting horizon) observations of each time series in the corresponding dataset. For example, for each monthly time series, the last 18 observations are held out for testing.

We use SMAPE as the evaluation metric, defined in Section 2.4, and apply it in three different ways:

- Overall performance: The standard approach of computing forecasting performance using SMAPE on a given dataset.
- Expected shortfall: We use the SMAPE expected shortfall to compare different forecasting models. Expected shortfall is a financial risk measure that quantifies the expected return of a portfolio on a percentage of worst cases. We adopt this idea to our study and measure forecasting accuracy on the 5% of time series where a given model shows the worst scores. We compute the SMAPE for each model in each time series. Then, take the average score in the 5% of cases. This metric helps quantify and compare the models regarding their worst-case scenarios.
- Win/Loss ratios: Counting how many times a model outperforms another across all time series based on SMAPE. Ratios provide a non-parametric way of comparing different models, which mitigates the effect of outliers.

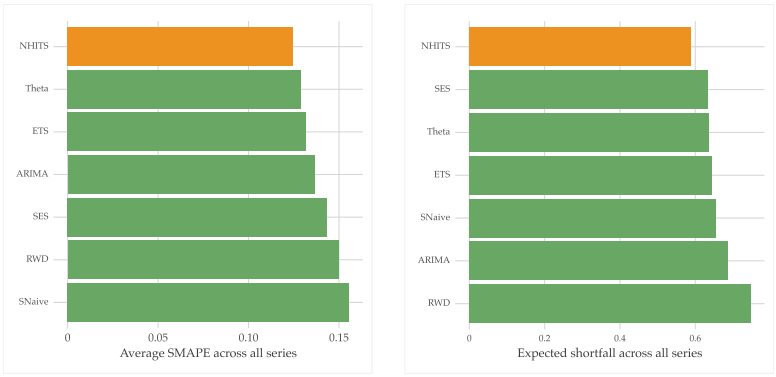
These metrics are computed in different dataset conditions, specifically:

- All data: Following a standard approach, we compute the metrics over all samples;
- Different horizons: We evaluate models in different forecasting horizons to assess if the relative performance varies across the horizon;
- Varying sampling frequency: We include datasets with three different sampling frequencies: monthly, quarterly, and yearly;
- Difficult problems: Some time series may exhibit easy-to-model patterns. In that case, an approach with a more flexible functional form, such as deep neural networks, may be unnecessary. We control for the *difficulty* of a time series, which is defined in the next section.
- Anomalies: Finally, we analyse how models perform when forecasting anomalous observations. In this work, we consider an observation to be an anomaly if its value falls outside of the 99% prediction interval of the `SNaive` model.

We remark that we conduct the analysis of results using all datasets jointly and not by each dataset listed in Table 1.

4 Experiments

The evaluation framework described in the previous section is applied in a comparison of deep learning with classical forecasting techniques. In particular, the central research question posed is the following: "How does `NHITS`, a state-of-the-art deep learning forecasting method, perform relative to classical approaches for univariate time series forecasting?"



(a) Average SMAPE (b) SMAPE expected shortfall

Fig. 1: Average SMAPE (a) and expected shortfall (b) for each model across all time series

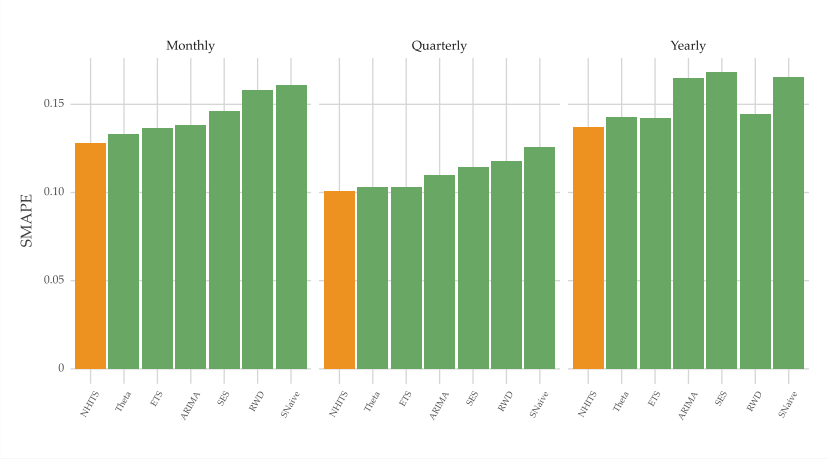


Fig. 2: SMAPE scores by model and sampling frequency.

4.1 Performance on all data

We start by summarising forecasting performance across all time series using SMAPE. The results are shown in Figure 1a, where `NHITS` presents the best score, outperforming all classical approaches. Among these, the Theta method exhibits the best performance. Figure 1b shows the SMAPE expected shortfall (c.f. Section 3.4). From a worst-case scenario perspective, `NHITS` also stands out and shows the best performance.

Then, we evaluate and compare each approach by controlling for several factors. Figure 2 reports the SMAPE scores controlling for sampling frequency.

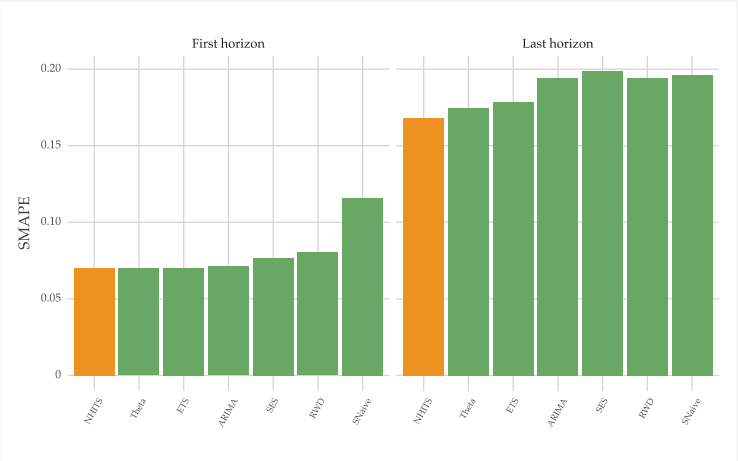


Fig. 3: SMAPE scores by model and forecasting horizon.

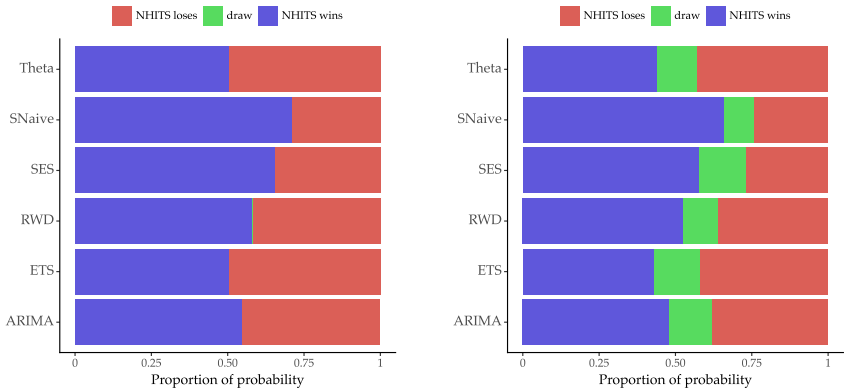
`NHITS` shows the best performance in all cases, though the relative advantage varies in each of these.

We also controlled the experiments for forecasting horizon. We measured performance in the first and last horizon of each series, where the former equates to one-step-ahead forecasting. The forecasting horizon varies by sampling frequency (c.f. Table 1). In effect, the last horizon is different in different sampling frequencies. The results (Figure 3) suggest that, for the first horizon, `NHITS` shows comparable performance with several classical approaches, such as `Theta` and `ETS`. However, in the last horizon, `NHITS` outperforms other approaches. This result is similar to the findings by Tang et al. [28], mentioned in Section 2.3.

We also controlled the experiments by individual time series and computed how often `NHITS` outperformed other approaches. Figure 4a shows that, while `NHITS` exhibits the overall best performance, there is a reasonable chance that it is outperformed by any other method. For example, `NHITS` outperforms `Theta` in about 50% of the 99140 time series. We also carried out this analysis using the principles behind practical equivalence [20]. We set the region of practical equivalence (ROPE) to 5%, so we consider two models to perform similarly if their absolute percentage difference in SMAPE is below 5%. The results (Figure 4b) show that `NHITS` remains competitive with all approaches in this scenario. However, there is also a reasonable chance that a given classical approach outperforms it by at least 5%.

4.2 Performance on difficult problems

In the previous analysis, we considered all 99140 time series. However, some time series may exhibit patterns easily captured by a simple model. Thus, we repeat the analysis only considering difficult problems. We took a data-driven



(a) No ROPE (b) ROPE=5%

Fig. 4: Probability of NHITS outperforming other approaches across all time series

and model-based approach to define a difficult problem based on the performance of a baseline, namely `SNaive`.

Figure 5 shows the distribution of SMAPE performance by `SNaive` across all time series. The vertical line depicts the 95% score percentile. We consider a difficult problem to be any time series corresponding to the right side of the vertical line.

We present the results of the repeated analysis in Figure 6. `NHITS` also shows the best performance in difficult problems. However, the advantage is considerably smaller relative to the results using all time series.

4.3 Performance on anomalies

Time series often exhibit unexpected or anomalous observations. Sometimes, these instances significantly impact the corresponding application domain, making it important to accurately forecast this type of case.

Figures 7a and 7b shows the performance of each model in anomalous observations across all time series. In these instances, `NHITS` is outperformed by `ETS` in terms of overall SMAPE and by `SES` and `Theta` in terms of expected shortfall.

5 Discussion

As reported in previous studies, we found that `NHITS` shows an overall better univariate forecasting performance relative to classical approaches, according to SMAPE [9]. However, we also discovered several factors that give a more nuanced perspective about their relative performance:

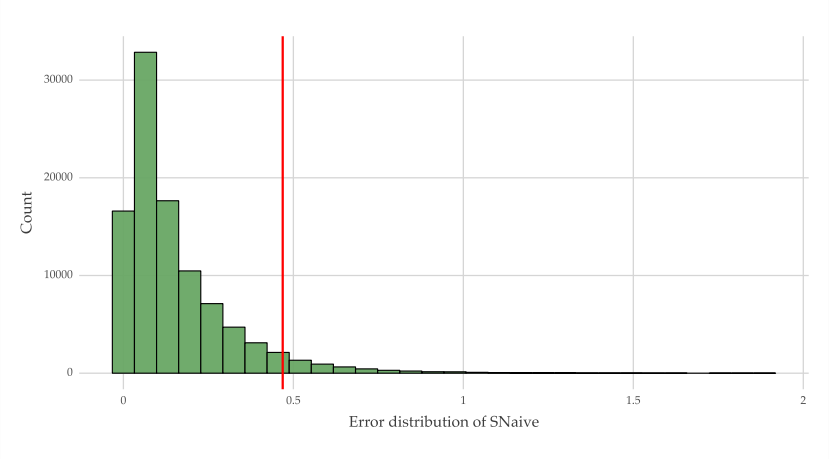
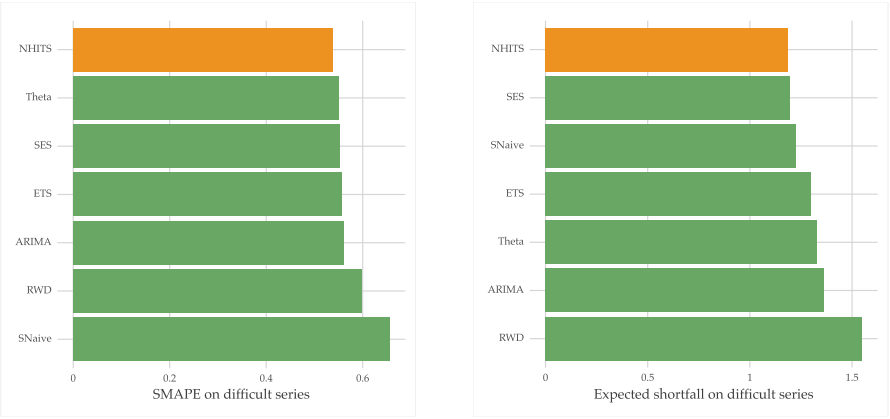


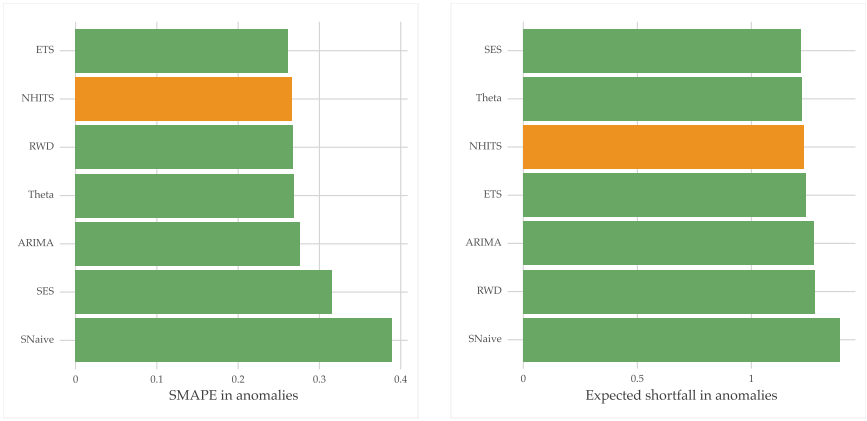
Fig. 5: SMAPE distribution of SNaive across all time series. The vertical line depicts the 95% score percentile.

1. Effect of sampling frequency: `NHITS` shows the best performance in all three sampling frequencies tested. However, `NHITS` is less competitive for time series with low sampling frequencies, such as yearly. This suggests that the effectiveness of `NHITS` may depend on the frequency at which data is collected. We note that our analysis was based on time series datasets with a monthly, quarterly, and yearly sampling frequency. This type of dataset tends to comprise many time series, but each of which is small. Notwithstanding, there is also evidence that `NHITS` shows state-of-the-art forecasting accuracy in time series with high sampling frequency [9].
2. Relative performance: While `NHITS` shows better SMAPE scores overall, there is a reasonable chance that classical approaches may outperform it, even with an equivalence margin of 5%. This implies that the superiority of `NHITS` is not guaranteed in all cases.
3. Worst-case scenarios: In worst-case scenarios, as measured by SMAPE-based expected shortfall, `NHITS` demonstrates better performance than classical methods. This suggests that `NHITS` may be more robust or reliable relative to classical approaches.
4. Forecasting horizon: `NHITS` is particularly suited in forecasting multiple steps ahead. This indicates that its strengths lie in long-term prediction rather than short-term forecasting. Indeed, `NHITS` was specially designed to handle long-horizon forecasting [9]. However, previous work also reported this effect when comparing `MLPs` with `ARIMA` [28].
5. Difficulty of problems: The advantage of `NHITS` diminishes on difficult forecasting problems, as measured by the `SNaive` worst-case performance. This implies that the advantage of `NHITS` may vary depending on the complexity or nature of the data being analyzed.



(a) Average SMAPE (b) Expected shortfall

Fig. 6: Average SMAPE (a) and expected shortfall (b) for each model across difficult time series



(a) Overall SMAPE (b) Expected shortfall

Fig. 7: Performance of each model in anomalous observations across all time series

6. Anomalous observations: NHITS is outperformed by classical methods when dealing with anomalous observations. This suggests that NHITS may struggle with handling outliers or unexpected data points compared to classical forecasting techniques.

Overall, these findings highlight the nuanced nature of the performance of NHITS compared to classical forecasting methods, with its strengths and weaknesses becoming apparent under different conditions. In future work, we plan to include additional perspectives to improve the characterization of the relative performance of forecasting models.

6 Conclusions

This paper presents an extensive empirical comparison of a state-of-art deep learning forecasting method and several classical approaches for univariate time series forecasting problems. Contrary to previous attempts at this task, we evaluate forecasting performance from different perspectives. This approach enabled a more granular analysis of the relative performance of different methods.

NHITS shows the overall best performance according to SMAPE, a commonly used forecasting evaluation metric. However, we found that NHITS is outperformed by classical approaches in a reasonable percentage of time series. We discovered other interesting aspects, such as the varying relative performance in forecasting horizon conditions. While NHITS is more robust than classical approaches in terms of worst-case performance, it presents a poor performance when predicting unexpected values. We believe that the nuanced analysis presented in this work will foster further research to develop better forecasting approaches.

References

1. Ahmed, N.K., Atiya, A.F., Gayar, N.E., El-Shishiny, H.: An empirical comparison of machine learning models for time series forecasting. Econometric Reviews **29**(5-6), 594–621 (2010)
2. Assimakopoulos, V., Nikolopoulos, K.: The theta model: a decomposition approach to forecasting. International journal of forecasting **16**(4), 521–530 (2000)
3. Athanasopoulos, G., Hyndman, R.J., Song, H., Wu, D.C.: The tourism forecasting competition. International Journal of Forecasting **27**(3), 822–844 (2011)
4. Bandara, K., Bergmeir, C., Smyl, S.: Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. Expert systems with applications **140**, 112896 (2020)
5. Bontempi, G., Ben Taieb, S., Le Borgne, Y.A.: Machine learning strategies for time series forecasting. Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures 2 pp. 62–77 (2013)
6. Cerqueira, V., Santos, M., Baghoussi, Y., Soares, C.: On-the-fly data augmentation for forecasting with deep learning. arXiv preprint arXiv:2404.16918 (2024)

7. Cerqueira, V., Torgo, L., Soares, C.: A case study comparing machine learning with statistical methods for time series forecasting: size matters. Journal of Intelligent Information Systems **59**(2), 415–433 (2022)
8. Cerqueira, V., Torgo, L., Soares, C.: Model selection for time series forecasting an empirical analysis of multiple estimators. Neural processing letters **55**(7), 10073–10091 (2023)
9. Challu, C., Olivares, K.G., Oreshkin, B.N., Ramirez, F.G., Canseco, M.M., Dubrawski, A.: Nhits: Neural hierarchical interpolation for time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 6989–6997 (2023)
10. Gardner Jr, E.S.: Exponential smoothing: The state of the art. Journal of forecasting **4**(1), 1–28 (1985)
11. Godahewa, R., Bandara, K., Webb, G.I., Smyl, S., Bergmeir, C.: Ensembles of localised models for time series forecasting. Knowledge-Based Systems **233**, 107518 (2021)
12. Hewamalage, H., Ackermann, K., Bergmeir, C.: Forecast evaluation for data scientists: common pitfalls and best practices. Data Mining and Knowledge Discovery **37**(2), 788–832 (2023)
13. Hill, T., O'Connor, M., Remus, W.: Neural network models for time series forecasts. Management science **42**(7), 1082–1092 (1996)
14. Hyndman, R., Koehler, A.B., Ord, J.K., Snyder, R.D.: Forecasting with exponential smoothing: the state space approach. Springer Science & Business Media (2008)
15. Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2018)
16. Hyndman, R.J., Khandakar, Y.: Automatic time series forecasting: the forecast package for r. Journal of statistical software **27**, 1–22 (2008)
17. Januschowski, T., Gasthaus, J., Wang, Y., Salinas, D., Flunkert, V., Bohlke-Schneider, M., Callot, L.: Criteria for classifying forecasting methods. International Journal of Forecasting **36**(1), 167–177 (2020)
18. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems **30** (2017)
19. Koutsandreas, D., Spiliotis, E., Petropoulos, F., Assimakopoulos, V.: On the selection of forecasting accuracy measures. Journal of the Operational Research Society **73**(5), 937–954 (2022)
20. Kruschke, J.K.: Rejecting or accepting parameter values in bayesian estimation. Advances in methods and practices in psychological science **1**(2), 270–280 (2018)
21. Leites, J., Cerqueira, V., Soares, C.: Lag selection for univariate time series forecasting using deep learning: An empirical study. arXiv preprint arXiv:2405.11237 (2024)
22. Makridakis, S., Hibon, M.: The m3-competition: results, conclusions and implications. International journal of forecasting **16**(4), 451–476 (2000)
23. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The m4 competition: Results, findings, conclusion and way forward. International Journal of forecasting **34**(4), 802–808 (2018)
24. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: Concerns and ways forward. PloS one **13**(3), e0194889 (2018)
25. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: M5 accuracy competition: Results, findings, and conclusions. International Journal of Forecasting (2022)

26. Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y.: N-beats: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437 (2019)
27. Smyl, S.: A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. International Journal of Forecasting **36**(1), 75–85 (2020)
28. Tang, Z., De Almeida, C., Fishwick, P.A.: Time series forecasting using neural networks vs. box-jenkins methodology. Simulation **57**(5), 303–310 (1991)
29. Triebe, O., Laptev, N., Rajagopal, R.: Ar-net: A simple auto-regressive neural network for time-series. arXiv preprint arXiv:1911.12436 (2019)