Benchmarking Zero-Shot Foundation Time Series Forecasting Models for Industrial Applications

Benjamin Pasquier[1,*,†], Frédéric Montet[1,†] and Beat Wolf[1]

[1] *iCoSys, HEIA-FR, HES-SO University of Applied Sciences and Arts Western Switzerland*

Abstract

Time-series forecasting foundation models recently emerged with zero-shot capabilities, leveraging generalized training on diverse datasets. This study compares zero-shot foundation models to traditional statistical, machine learning, and deep learning methods using industrial and academic multivariate datasets. Results show foundation models, particularly Moirai large, often outperform traditional methods while reducing dataset-specific tuning needs. These findings highlight their industrial potential by allowing for simpler, yet more accurate forecasting.

Keywords

zero-shot forecasting, foundation models, time series analysis, industrial applications, model benchmarking

1. Introduction

Forecasting is indispensable across various domains, including weather prediction, financial markets, energy management, anomaly detection, and many more. All of these areas benefit from robust predictors that can accurately fit the data for a subsequent task. Influenced by trends in the deep learning community, the technologies used to create state-of-the-art predictors have begun to shift in recent years. For some tasks, statistical methods still hold value, while in more complex use cases, deep learning methods such as Transformer-based or other frontier models are increasingly needed. In this context, foundation models are introducing a modeling approach that challenges traditional practices by training on all kinds of time-series data to make predictions on a target time series.

One of the strengths of foundation models is their ability to produce qualitative forecasts in a zero-shot context, i.e., where the model has never seen the data it is asked to predict. In this paper, we explore three research questions. **RQ1**: What is the performance of those models in a zero-shot setting compared to traditional machine learning approaches? **RQ2**: What are those models performance on yet unreleased and industrial datasets, given the complexity to assess either overfitting or data leakage? **RQ3**: Given the many parameters such as window length, co-variates, and gaps between inputs and targets, how can we evaluate those models in a fair and reproducible way?

This study addresses these challenges by proposing a structured approach to evaluate foundation models and benchmark them against traditional methods across diverse datasets. Evaluating and comparing forecasters on these questions empowers practitioners to make informed decisions when selecting models for specific projects.

In the following sections, we introduce our approach to benchmarking multiple foundation and classical models on five different datasets from industrial and academic sources. The paper follows the classical structure of scientific method sections, concluding with a discussion that assesses the quality of the tested models.

AI days HES-SO '25 January 27–29, 2025, Switzerland

*Corresponding author.

†These authors contributed equally.

benjamin.pasquier@hefr.ch (B. Pasquier); frederic.montet@hefr.ch (F. Montet); beat.wolf@hefr.ch (B. Wolf)

0009-0009-7414-7279 (B. Pasquier); 0000-0003-0439-5559 (F. Montet); 0000-0002-9307-7212 (B. Wolf)

Table 1
Overview of the forecasting methods used in this research.

Model	Type	Prediction type	# Parameters
NaïveSeasonal	Statistical	Univariate	Not applicable
AutoARIMA [2]	Statistical	Univariate	< 100
GRU [7]	Deep learning	Multivariate	5-20k
TSMixer [3]	Deep learning	Multivariate	26-48k
Chronos Tiny [6]	Foundation	Univariate	8m
Chronos Large [6]	Foundation	Univariate	710m
Moirai small [5]	Foundation	Multivariate	14m
Moirai large [5]	Foundation	Multivariate	311m

2. Methods

The aim of our experiments is to have a comparison between models that is reliable and performed across multiple datasets. For that purpose, we extended the onTime time series analysis library with a benchmarking framework [1]. This extension allows us to minimize the implementation work required for new models, as well as benefiting from available academic datasets, requiring custom data loading code only for industrial datasets.

2.1. Models

To evaluate how well foundation models perform compared to existing dataset-specific approaches, we select several well-established methods in the field of time series forecasting. First, Naïve Seasonal forecasting and AutoARIMA [2] models serve as baseline benchmarks. Additionally, we incorporate more advanced deep learning methods, including a GRU model and the TSMixer [3] model, which have demonstrated their effectiveness in prior studies. These four models are implemented using the wrappers provided by the Darts library, applied in their simplest forms with default parameters [4]. For the deep learning methods, only the input and output chunk lengths are adjusted to align with the specific dataset's input and target lengths. Finally, we evaluate two foundation models, Moirai [5] and Chronos [6], which are currently the only ones integrated into our benchmarking framework. We use checkpoints of the smallest, respectively the biggest models, available on the Hugging Face platform[1,2]. Table 1 presents the model details.

It is worth noting that Naïve Seasonal forecasting, AutoARIMA, and Chronos are univariate models, meaning they do not account for cross-correlations between features. Consequently, their univariate predictions are combined to produce a single multivariate prediction.

2.2. Datasets and Pre-processing

Five different multivariate datasets are tested, as presented in Table 2. For each of them, a realistic predictive task is defined by setting the input and target length for one prediction.

ETTh1 is data from an electricity transformer temperature. This dataset is multivariate, containing oil temperature data and six power load features, with a one hour resolution. The most common task with this data is to predict the future power oil temperature as a function of the power load.

The Energy dataset showcases the hourly energy demand generation and weather in Spain from 2015 to the end of 2018. 28 features are available including different type of energy generation such as coal, oil, geothermal, etc. However, we remove eight constant features from the dataset as they offer no predictive value and could potentially cause division by zero errors when calculating specific metrics.

[1]Chronos checkpoints used : https://huggingface.co/amazon/chronos-t5-tiny and https://huggingface.co/amazon/chronos-t5-large
[2]Moirai checkpoints used : https://huggingface.co/Salesforce/moirai-1.0-R-small and https://huggingface.co/Salesforce/moirai-1.0-R-large

Table 2
Datasets used for the benchmark

Dataset	Type	Nb. Features	Resolution	Input length	Target length
ETTh1 [8]	Academic	7	1 hour	4 days	1 day
Energy [9]	Academic	20	1 hour	4 days	1 day
HEIA10min	Industrial	24	10 minutes	1 day	2 hours
HEIA1h	Industrial	24	1 hour	4 days	1 day
MeteoSwiss [10]	Industrial	8	10 minutes	1 day	2 hours

HEIA10min and HEIA1h are datasets that represent the electrical consumption of the University of Applied Sciences in Fribourg, Switzerland. Two variants of this dataset are available with a 10 minutes resolution as well as an hourly one. In this data, six buildings do provide electrical measurement of sub-circuits for: the main circuit, the lightning, the "UPS power" (Uninterruptible Power Supply) and the emergency power circuit.

The last dataset, MeteoSwiss, represents meteorological data from a national weather station located in the Fribourg/Grangeneuve region. The data has a 10 minutes resolution and includes eight features such as pressure, wind speed, wind direction, relative humidity, etc.

In terms of pre-processing, the time series from each dataset are first divided into training and test sets using a standard 80%-20% split. Subsequently, the testing data is segmented into samples with varied input (the portion used for prediction) and target (the portion to be predicted) splits, simulating possible real-world use cases. Missing values in the datasets are handled using linear interpolation to ensure continuity.

2.3. Evaluation

Although some datasets are designed such that certain features are used as inputs while others are exclusively forecasted, we adopt a systematic approach where all available features are used for both input and forecast.

To assess the models' forecasts, we employ a rolling evaluation on the test set with a stride equal to the prediction length. This approach ensures that all time steps in the dataset are utilized, while avoiding repeated use of the same point as a target in multiple samples.

For evaluation metrics, we select two that are scale-invariant, allowing us to aggregate performance across features even when they have different units or scales. The first metric is the Mean Absolute Scaled Error (MASE), which measures the forecast MAE relative to a naïve seasonal baseline. MASE is computed as shown in Equation 1:

$$\text{MASE} = \frac{\frac{1}{T}\sum_{t=t_p+1}^{t_p+T} |y_t - \hat{y}_t|}{\frac{1}{t_p-m}\sum_{t=m}^{t_p} |y_t - y_{t-m}|} \tag{1}$$

Here, y_t represents the true values, $\hat{y}_t$ the predicted values, T is the prediction length, t_p is the end of the training period, and m is the seasonal lag. In our case, we use a naïve forecast with $m = 1$, where the forecast at each step simply repeats the value from the previous time step.

The second metric is the Symmetric Mean Absolute Percentage Error (sMAPE), which evaluates the relative error as a percentage, symmetrically penalizing over- and under-predictions. sMAPE is calculated as shown in Equation 2:

$$\text{sMAPE} = 200 \times \frac{1}{T}\sum_{t=1}^{T} \frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)} \tag{2}$$

Unlike traditional MAPE, sMAPE avoids division by zero and ensures that the metric remains bounded, making it well-suited for datasets with values near zero.

By combining these two complementary metrics, we obtain a comprehensive evaluation of the models' performance across diverse datasets and features.

3. Results

Table 3 presents a comparison of different forecasting models across multiple datasets, evaluated using the MASE and sMAPE metrics. Among the models, Moirai large consistently achieves the best performance, as evidenced by its lowest MASE and sMAPE values across most datasets, including the ETTh1, Energy, HEIA10min, HEIA1h, and MeteoSwiss datasets. The Chronos models also perform competitively, with results often close to Moirai, particularly on datasets like HEIA10min and Energy, where the differences in metrics are relatively small.

Surprisingly, the Naïve Seasonal model demonstrates strong performance in certain cases, such as on the HEIA10min and MeteoSwiss datasets, where it achieves the best sMAPE, while showing comparable MASE to more advanced models. However, traditional machine learning and deep learning approaches, such as GRU and TSMixer, tend to underperform. For instance, GRU shows particularly poor results on the Energy and MeteoSwiss datasets.

Overall, the table underscores the strength of foundation models, particularly Moirai, in multivariate forecasting tasks while highlighting the limitations of simpler baselines and older deep learning models in this context.

Table 3
Comparison of models across datasets (MASE and sMAPE metrics).

Dataset	ETTh1		Energy		HEIA10min		HEIA1h		MeteoSwiss	
Metric	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE
Model										
NaïveSeasonal	3.21	73.02	5.70	35.22	3.27	**14.29**	2.95	33.48	3.73	**29.75**
AutoARIMA	2.99	81.95	5.86	43.56	3.43	15.78	3.48	52.58	3.52	44.37
GRU	2.89	60.92	77.83	133.43	15.36	44.01	19.50	56.67	485.55	81.23
TSMixer	2.53	56.77	19.74	52.78	14.31	36.68	12.67	43.90	15.98	48.07
Chronos Tiny	2.08	46.80	5.93	33.74	4.69	16.79	3.08	29.14	9.90	50.25
Chronos Large	2.13	47.41	6.02	33.74	4.21	17.08	6.15	27.45	19.40	50.33
Moirai small	1.83	43.61	4.52	31.38	2.44	15.53	2.47	33.11	2.54	46.25
Moirai large	**1.78**	**42.95**	**4.45**	**30.18**	**2.28**	14.81	**2.09**	**27.35**	**2.48**	43.87

Table 4 compares training and inference times across models and datasets. Naïve Seasonal and AutoARIMA skip explicit training, integrating their fitting process into inference, with Naïve Seasonal being the fastest by simply reusing historical values. Deep learning models like TSMixer require offline training (e.g., 1,079 seconds on MeteoSwiss) but achieve rapid inference (0.01–0.02 seconds). Foundation models, operating in zero-shot mode, avoid training entirely; Moirai delivers significantly faster inference (0.02–0.06 seconds) compared to Chronos, whose larger variants take up to 0.57 seconds. Overall, Moirai combines scalability with fast inference, outperforming Chronos in efficiency.

4. Discussion

In this paper, we compared eight models performance across five different datasets with two metrics: MASE and sMAPE. Such a benchmarking task allowed us to gather knowledge about the complexity inherent to model comparison. Indeed, the number of parameters that one has to choose to compare different models between each other influences the results. Furthermore, the metric used is also a particularly important choice.

To come back to the research questions stated in the introduction, we could take a position related to each of them. In relation to RQ1 *How do those models perform compared to traditional machine learning*

Table 4
Comparison of training and inference time across datasets, in seconds.

Dataset	ETTh1		Energy		HEIA10min		HEIA1h		MeteoSwiss	
Time	Train.	Inf.	Train.	Inf.	Train.	Inf.	Train.	Inf.	Train.	Inf.
Model										
NaïveSeasonal	-	**0.01**	-	0.03	-	0.03	-	0.03	-	**0.01**
AutoARIMA	-	0.68	-	1.90	-	2.18	-	2.19	-	0.82
GRU	313.78	**0.01**	86.25	0.02	138.87	**0.01**	7.28	**0.01**	487.78	**0.01**
TSMixer	71.44	**0.01**	104.59	**0.01**	117.25	0.02	20.82	0.02	1079.56	0.02
Chronos Tiny	-	0.17	-	0.17	-	0.09	-	0.17	-	0.09
Chronos Large	-	0.57	-	0.57	-	0.32	-	0.57	-	0.30
Moirai small	-	0.02	-	0.02	-	0.03	-	0.02	-	0.02
Moirai large	-	0.05	-	0.05	-	0.06	-	0.05	-	0.05

approaches? and RQ2 *What are the performances of those models across various industrial datasets?*, the results of our benchmark allowed us to rank the performance of each model.

The table 3 presents those results and showcases the competitiveness of the selected foundation models against other approaches. Using those models in zero-shot context has shown a great developer experience with no complex training process needed to reach good performance across many datasets. Nevertheless, the benchmarking process should include hyperparameter optimization of all ML/DL models, in order to get the maximum performance for all of them; thus allowing for a fairer comparison.

An aspect to consider is the way predictions are calculated. In a case like Chronos, a multivariate forecast is calculated component after component. Therefore, it misses out potentially important cross-component correlations that could improve the forecast. Also, the inference time takes more time as it evolves linearly, the more components are to be predicted.

One notable finding from this benchmark is that the simplest models are still relevant. This highlights the importance of choosing a model that remains adapted to the data at hand and promotes a sort of technological sobriety.

Finally, the answer to RQ3 *Given the significant responsibility placed on developers when modeling time series data, how can we ensure reliable comparisons between different forecasting models?*, the answer is more subtle. Transparency of the pre-processing is of paramount importance, so are the metrics used. About the latter, none of the metrics we identified did provide an optimal way to compare multivariate time series forecasts without falling into issues such as division by zero, or else. Therefore, new metrics could be an area of improvement of the study in addition to more zero-shot models.

5. Conclusion

Our study reveals the challenges in comparing eight models in various datasets. foundation models perform impressively in zero-shot contexts, offering ease of use without extensive training, yet simpler models remain effective when well-matched to the data. The necessity of fine-tuning and choosing appropriate metrics is critical, as these factors greatly influence performance and fairness in comparisons. Limitations in current forecasting methods and metrics point to a need for new approaches that better capture cross-component correlations and improve the reliability of the evaluation. Future research should focus on improving transparency in pre-processing and developing innovative metrics to advance model benchmarking.

References

[1] ontime.re, onTime: Your library to work with time series, GitHub repository, 2024. https://github.com/ontime-re/ontime (accessed on 18.11.2024).

[2] R. J. Hyndman, Y. Khandakar, Automatic time series forecasting: The forecast package for r, Journal of Statistical Software 27 (2008) 1–22. URL: https://www.jstatsoft.org/index.php/jss/article/view/v027i03. doi:10.18637/jss.v027.i03.

[3] S.-A. Chen, C.-L. Li, N. Yoder, S. O. Arik, T. Pfister, Tsmixer: An all-mlp architecture for time series forecasting, 2023. URL: https://arxiv.org/abs/2303.06053. arXiv:2303.06053.

[4] J. Herzen, F. LÃ¤ssig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. V. Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin, M. Dumonal, J. KoÅ›cisz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik, G. Grosch, Darts: User-friendly modern machine learning for time series, Journal of Machine Learning Research 23 (2022) 1–6. URL: http://jmlr.org/papers/v23/21-1177.html.

[5] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, D. Sahoo, Unified training of universal time series forecasting transformers, 2024. URL: https://arxiv.org/abs/2402.02592. arXiv:2402.02592.

[6] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. Pineda Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. Gordon Wilson, M. Bohlke-Schneider, Y. Wang, Chronos: Learning the language of time series, arXiv preprint arXiv:2403.07815 (2024).

[7] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. URL: https://arxiv.org/abs/1406.1078. arXiv:1406.1078.

[8] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference, volume 35, AAAI Press, 2021, pp. 11106–11115.

[9] Energy consumption, generation, prices and weather, 2019. URL: https://www.kaggle.com/datasets/nicholasjhana/energy-consumption-generation-prices-and-weather, accessed: 2024-11-17.

[10] MeteoSwiss, Federal office of meteorology and climatology, 2024. URL: https://www.meteoswiss.admin.ch/, accessed: 2024-11-18.