Leveraging Large Language Models for Time Series Prediction on Low-Frequency Data

Rodrigo Parracho[1], Fernando Alexandrino[1,2], Matheus de Souza Figueiredo [1], Lucas Pereira da Silva [1], Bruno Dutra de Macedo [1], Arthur Lamblet Vaz[1], Davi Louback [1], Victor Coculilo Desouzart [1], Rebecca Salles[3], Fabio Porto[4], Diego Carvalho[1], Eduardo Ogasawara[1]

[1]CEFET/RJ - Federal Center for Technological Education of Rio de Janeiro

[2]IFSP - Federal Institute of São Paulo

[3]INRIA

[4]LNCC - National Laboratory for Scientific Computing

{rodrigo.parracho, matheus.figueiredo.1}@cefet-rj.br,

{arthur.lamblet, lucas.pereira.1, bruno.macedo}@cefet-rj.br,

{davi.louback, victor.desouzart}@cefet-rj.br,

fernando.alexandrino@ifsp.edu.br, rebecca.pontes-salles@inria.fr

fporto@lncc.br, dcarvalho@ieee.org, eogasawara@ieee.org

Abstract. *Time series prediction is critical in domains such as economics, industry, and agriculture. Real-world scenarios often involve challenges like low data frequency, high variability, and non-repetitive patterns. Traditional statistical models and machine learning approaches, including Long Short-Term Memory (LSTM) networks, underperform in low-data contexts due to overfitting risks, intensive training requirements, and the lack of benchmarks tailored to such scenarios. Large language models (LLMs) have emerged as tools for time series forecasting, leveraging their ability to generalize and capture temporal dependencies and patterns across datasets without requiring extensive task-specific feature engineering. This study investigates the potential of time series foundation models (TSFMs), specifically Lag-Llama and Chronos, on low-frequency datasets by comparing zero-shot prediction across one-step-ahead and multi-step-ahead approaches. Our findings evaluate predictive error, robustness, efficiency, and applicability, showing how TSFMs address these limitations and enhance forecasting in data-scarce scenarios.*

1. Introduction

Time series (TS) prediction is a fundamental task with applications in economics, industry, and agriculture [Masini et al., 2021; Bao et al., 2025]. In real-world scenarios, challenges often emerge due to low data frequency, non-repetitive patterns, and high variability. These characteristics hinder the performance of classic methods, such as statistical models, which are effective on linear and stationary series but tend to fail in nonstationary or highly variable contexts [Petropoulos, 2022].

Recent advances in machine learning (ML) have enabled the modeling of nonlinear and complex patterns in TS [Semenoglou et al., 2023]. Long Short-Term Memory (LSTM) networks, in particular, are capable of capturing long-term temporal dependencies. However, these models require preprocessing, hyperparameter tuning, and are prone to overfitting when data is scarce. Forecasting under such low-data conditions poses significant challenges, as there are insufficient examples to capture recurring patterns or trends.

This limitation impacts both statistical and ML models, which typically depend on large datasets for training and validation. In data-scarce scenarios, the risk of overfitting increases, impairing the models' ability to generalize to unseen data [Goubeaud et al., 2021; Iglesias et al., 2023; Maior and Silva, 2024]. Even when more data is available, situations affected by concept drift [Ogasawara et al., 2025], such as those observed in the post-pandemic period, may render retraining impractical or ineffective.

Moreover, the absence of specific benchmarks for low-data TS impedes the systematic evaluation of novel models. Many existing approaches are designed for large datasets and do not address the specific challenges of short, intermittent, or low-granularity series. In these cases, reliance on historical data may not improve forecasting. As highlighted in Lucas's critique in economics, predictions based on past data can become unreliable when the underlying structures change, reducing the relevance of historical observations in dynamic contexts [McKay and Wolf, 2023].

Large language models (LLMs) offer an alternative due to their ability to generalize in data-scarce conditions [Gruver et al., 2023], drawing on knowledge embedded in their training corpora. Time series foundation models (TSFMs), inspired by LLMs in both architecture and capability, extend this potential to temporal data. However, evaluating their performance in low-data scenarios requires caution to ensure that results are both reliable and applicable. Although TSFMs have demonstrated superior performance compared to traditional forecasting methods, existing studies often focus on specific domains, such as energy and commodity pricing [Lin et al., 2024; He et al., 2024]. Broader evaluations across diverse application areas — particularly those involving low-frequency and short time series — remain scarce. Methodological adaptations are needed to address data variability and ensure robustness and scalability under such constraints.

This study investigates the application of TSFMs in forecasting scenarios characterized by data scarcity. We compare two prominent models, Lag-Llama and Chronos, with traditional linear and ML approaches, considering both one-step-ahead and multi-step-ahead forecasting tasks on low-frequency datasets. Our evaluation considers accuracy, robustness, efficiency, and applicability, employing a framework that supports zero-shot prediction across domains. TSFMs address key limitations of conventional models and improve forecasting performance in low-data environments [Lin et al., 2024]. While computationally demanding [Gupta et al., 2024a,b], their requirements are considerably lower in small-data settings, making them a viable and practical alternative.

The remainder of this paper is organized as follows. Section 2 presents the theoretical background, with emphasis on general TSFM pipelining. Section 3 discusses related work. Section 4 describes our proposed framework for TS prediction with low-frequency data. Section 5 details the datasets, evaluation protocols, and results. Finally, Section 6 summarizes our findings and outlines future directions.

2. Background

A time series X is a sequence of observations x_t, which can be decomposed into three components: T_t (trend), S_t (seasonal or cyclic component), and E_t (noise) [Ogasawara et al., 2025]. The Autoregressive Integrated Moving Average (ARIMA) model is widely employed for forecasting, combining an autoregressive (AR) component that captures dependencies on lagged values, an integration (I) term that ensures stationarity through differencing, and a moving average (MA) component that models residual errors from past observations [Box et al., 2015]. Typically expressed as (p, d, q), ARIMA is commonly used as a baseline for comparing ML models in TS prediction [Salles et al., 2017].

2.1. ML & Data Preprocessing

ML models are computational tools for TS forecasting that extract patterns to predict future observations [Benidis et al., 2022]. Algorithms such as Support Vector Machine, Random Forest, Gradient Boosting, and Multilayer Perceptron have shown competitive performance in time series tasks [Masini et al., 2021; Mello et al., 2024]. Deep neural networks, in particular, are recognized for their ability to model complex relationships [Pacheco et al., 2022; Salles et al., 2023]. LSTM networks address the limitations of traditional recurrent neural networks by selectively retaining or discarding historical information, allowing them to capture long-term dependencies and forecast in scenarios characterized by trends and seasonality [Benidis et al., 2022].

Despite their capabilities, LSTM models typically require large datasets to effectively capture temporal patterns, which constrains their applicability in data-scarce scenarios [Benidis et al., 2022]. Preprocessing techniques such as normalization, differencing (*diff*), and data augmentation can mitigate these limitations. Normalization scales the data to reduce the influence of outliers and enhance pattern visibility [Han et al., 2022; Salles et al., 2023], while differencing addresses nonstationarity [Salles et al., 2019]. Augmentation methods, such as *jittering*, introduce controlled noise to increase data diversity [Iglesias et al., 2023].

LLMs have emerged as alternatives for TS forecasting by learning patterns from extensive corpora and transferring this knowledge across time series tasks. Their generalization capabilities help overcome limitations associated with challenging datasets [Gruver et al., 2023; He et al., 2024].

2.2. TSFMs

Inspired by LLMs, TSFMs are trained on heterogeneous corpora from multiple domains, frequencies, and scales, which enables inference without the need for task-specific training [Rasul et al., 2024; Ansari et al., 2024]. Designed for forecasting, these models generate values or distributions corresponding to future observations. The general process involves three main stages: model building (preprocessing and pretraining) and zero-shot prediction, as illustrated in Figure 1.

In the preprocessing stage (Step A), raw TS data is transformed into an appropriate model input through scaling and discretization, producing discrete *tokens*. This transformation aims to preserve the fundamental properties of the original series. The resulting tokens define sequences that serve as context for model training.

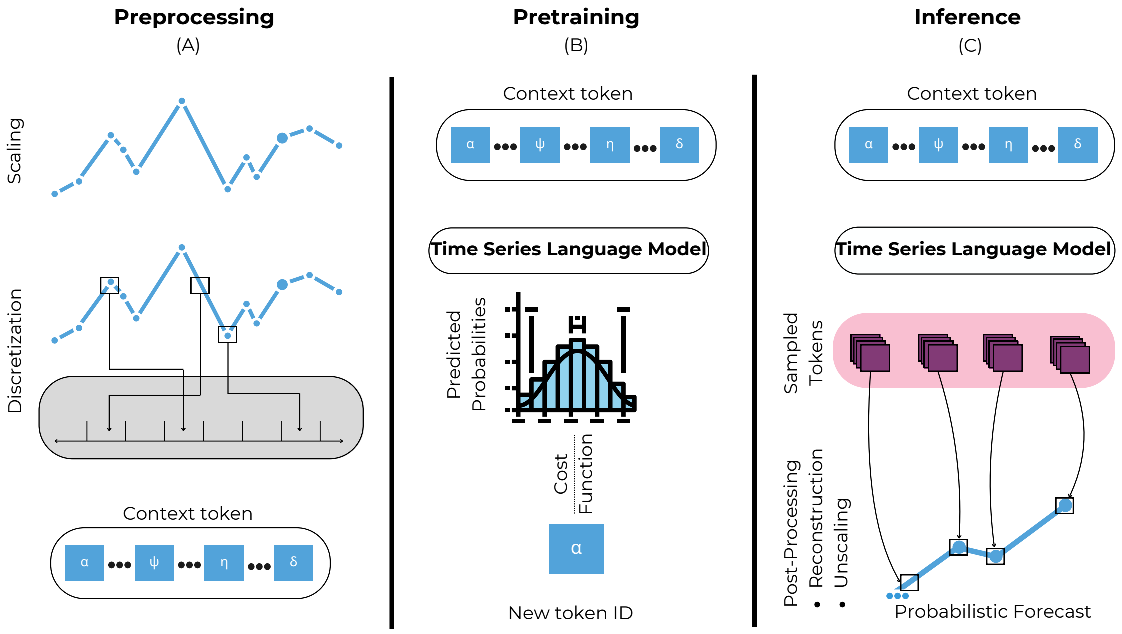

Figure 1. General TSFM Pipelining [Ansari et al., 2024]

During the pretraining phase (Step B), the model is fed with token sequences (e.g., α, ..., ψ, ..., η, ..., δ) and learns to predict the next token by estimating a probability distribution over possible continuations. A new token (e.g., α) is sampled by minimizing a cost function, with data augmentation applied to reduce overfitting. Through this process, the model internalizes TS patterns such as trends, seasonality, and correlations [Iglesias et al., 2023; Rasul et al., 2024].

Zero-shot prediction (Steps A-B-C) uses the same pipeline to infer token distributions without additional fine-tuning. The output tokens are then postprocessed to revert to the original data scale. Multiple sampling rounds generate a set of forecasts, which can be summarized using statistical measures such as the mean or median.

Lag-Llama and Chronos employ distinct preprocessing and training strategies. Lag-Llama applies lag features and robust standardization, using the Negative Log-Likelihood loss function [Rasul et al., 2024]. Chronos adopts standard normalization and quantization, combined with categorical cross-entropy loss [Ansari et al., 2024]. These variations illustrate how TSFMs can be adapted to different forecasting challenges. Nevertheless, their architectures and pretraining processes entail significant computational demands, particularly in large-scale or real-time applications [Gupta et al., 2024a].

3. Related Work

We conducted a literature review to investigate the potential of TSFMs in forecasting. Given the rapid development of the field, our search encompassed peer-reviewed publications and preprints available on SpringerLink, IEEE Xplore, and arXiv. We employed keywords such as *“TSFM”*, *“foundation models”*, and *“time series forecasting”*, prioritizing studies that benchmark TSFMs against statistical or ML-based methods.

Models such as Lag-Llama and Chronos have shown superior performance over traditional approaches in zero-shot scenarios, reducing reliance on dataset-specific training. Gruver et al. [2023] emphasize the ability of TSFMs to generalize across domains by leveraging innovations from language models. He et al. [2024] applied LLM-based methods to oil price forecasting, obtaining lower prediction errors than ARIMA and Random Walk models. Similarly, Bahelka and de Weerd [2024] demonstrated the advantage of Lag-Llama in inflation nowcasting using multivariate data, achieving higher accuracy and stronger correlation with real-world observations.

Other studies explored optimization strategies to enhance TSFM performance. Gupta et al. [2024b] employed Chronos with Low-Rank Adaptation (LoRA) to forecast vital signs in sepsis patients, achieving reduced computational costs. In another work, Gupta et al. [2024a] incorporated Fourier-based techniques to improve performance with minimal parameter tuning, highlighting the scalability of TSFMs for large datasets.

Two studies specifically addressed data-limited forecasting scenarios. Liao et al. [2024] used Chronos for electric load forecasting, comparing it to ARIMA and neural networks with 15 days of hourly data. Chronos outperformed the other models by approximately 30% in 48-hour forecasts. Lin et al. [2024] evaluated an even more constrained case, using only three days of data at multiple frequencies to predict energy consumption across countries. TSFMs, including Chronos, outperformed Support Vector Machines and Gaussian Processes, demonstrating efficacy with few observations and no external variables.

These studies suggest that TSFMs can match or surpass traditional forecasting methods without the need for complex architectures or large training datasets. However, most works concentrate on domains such as energy [Saravanan et al., 2024], finance, and healthcare, without evaluating generalization across broader application contexts. While He et al. [2024], Gupta et al. [2024a], and Lin et al. [2024] address aspects related to model size, and Liao et al. [2024] examines prediction horizons, there is limited analysis regarding distributional metrics, model size, and the use of TS as prompts for both short- and long-term forecasts. This study addresses these gaps by evaluating TSFMs across diverse domains with smaller datasets, highlighting their simplicity and efficiency without requiring task-specific preprocessing or retraining.

4. Methods

We apply TSFMs to the challenge of TS prediction with low-frequency data. Figure 2 illustrates the main steps of our methodology. We benchmark TSFMs against two reference methods: a traditional ML-based approach and ARIMA, representing a statistical baseline.

Each time series is partitioned into training and test sets. The training set contains the observations used to fit the model, while the test set comprises the most recent, unseen data used for evaluation. Given a series X with n observations and a split point t, the training set is defined as $x_1, \ldots, x_t$ and the test set as $x_{t+1}, \ldots, x_n$.

Each approach follows a distinct methodological pipeline. TSFMs operate as described in Figure 1, relying on pretraining over diverse corpora to generalize to previously unseen series. Initially, the training data is scaled to normalize values and mitigate the effects of magnitude variation. The scaled series is then tokenized, converting continuous

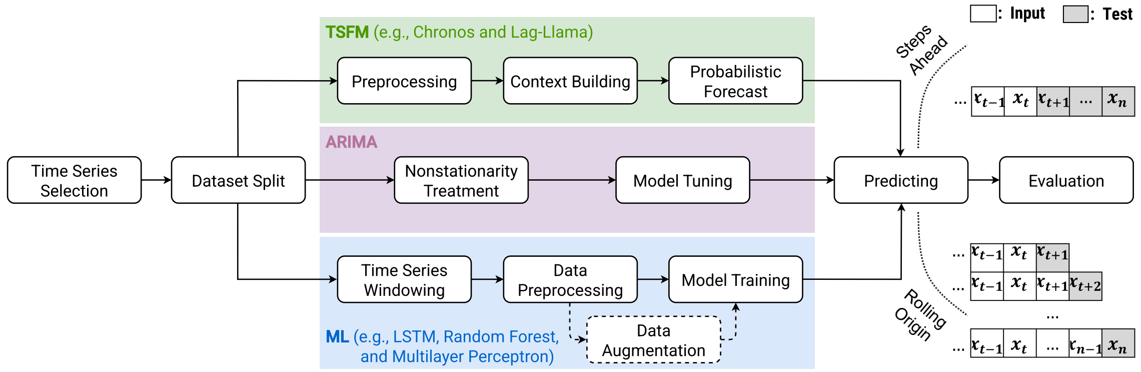
: Input
: Test
TSFM (e.g., Chronos and Lag-Llama)
Preprocessing
Context Building
Probabilistic Forecast
ARIMA
Time Series Selection
Dataset Split
Nonstationarity Treatment
Model Tuning
Predicting
Evaluation
Time Series Windowing
Data Preprocessing
Model Training
Data Augmentation
ML (e.g., LSTM, Random Forest, and Multilayer Perceptron)
Steps Ahead
Rolling Origin
... x_{t-1} x_t x_{t+1} ... x_n
... x_{t-1} x_t x_{t+1}
... x_{t-1} x_t x_{t+1} x_{t+2}
...
... x_{t-1} x_t ... x_{n-1} x_n

Figure 2. Framework for evaluating TSFMs in time series prediction

values into discrete tokens suitable for model input. TSFMs identify temporal patterns and dependencies within these sequences. During inference, the model estimates token probabilities and produces zero-shot predictions based on the patterns learned during pre-training.

The ARIMA method comprises two main steps. First, the series is differenced d times to address nonstationarity, thereby stabilizing its mean and variance. Then, an optimization procedure identifies the optimal ARIMA parameters. The resulting (p, d, q) model is configured and used for forecasting.

In the ML-based approach, the time series is transformed into sliding windows. Let $seq_{k,m}(X)$ be a subsequence of size k starting at position m, defined as $\{x_m, \ldots, x_{m+k-1}\}$, where $1 \leq m \leq n - k + 1$. This transformation produces input subsequences that capture temporal dependencies. The data is normalized, and data augmentation may be applied to enhance robustness. Hyperparameter tuning is performed via grid search. Once trained, the model is applied to the test set.

During testing, all fitted models generate predictions for $x_{t+1}, \ldots, x_n$. The preprocessing steps used during training are consistently applied to the test data. For TSFMs, final predictions are obtained by summarizing multiple outputs, typically using the mean or the median.

Two evaluation strategies are adopted: rolling origin and steps ahead. In rolling origin, the model generates sequential predictions for $t + 1$ using observations up to t, iteratively updating the input window. In the steps ahead strategy, the model predicts the entire test set in a single run using all prior data.

Finally, prediction quality is assessed using a performance metric computed over the test set.

5. Experimental Evaluation

5.1. Datasets

This study uses ninety time series collections encompassing economic, environmental, and agricultural indicators obtained from the FAO database [FAO, 2025]. The indicators include gross domestic product (GDP), climate change metrics, fertilizer production, greenhouse gas emissions, and pesticide use. Data were gathered for the ten largest global economies in 2024, as identified by the International Monetary Fund. Each series contains between 30 and 60 annual observations, characterizing a data-scarce scenario. Table 1 provides a summary of the datasets.

To characterize the statistical properties of the series, we applied the Augmented Dickey-Fuller (ADF) and Phillips-Perron (PP) tests [Box et al., 2015; Ogasawara et al., 2025]. The results indicate that most series are nonstationary, with the exception of Climate Change. This diagnosis supports the interpretation of model performance under different structural conditions.

Table 1. Overview of real-world datasets with low-frequency observations

Dataset	Description	Coverage	Observations
Climate Change	Mean absolute variation in surface temperature by country, expressed in ^{o}C	1961–2023	63
Emissions (CH_4)	Amount of methane (CH_4) released into the atmosphere by country, in kilotons	1961–2021	61
Emissions (CO_2)	Amount of carbon dioxide (CO_2) released into the atmosphere by country, in kilotons	1990–2021	32
Emissions (N_2O)	Amount of nitrous oxide (N_2O) released into the atmosphere by country, in kilotons	1961–2021	61
Fertilizers (K_2O)	Tonnes of potassium (K_2O) manufactured into fertilizer products by country	1961–2022	62
Fertilizers (N)	Tonnes of nitrogen (N) manufactured into fertilizer products by country	1961–2022	62
Fertilizers (P_2O_5)	Tonnes of phosphorus (P_2O_5) manufactured into fertilizer products by country	1961–2022	62
GDP	Total value of goods and services produced within a country, in dollars	1970–2023	54
Pesticides	Agricultural use of pesticides and related chemicals by country, measured in tonnes	1990–2022	33

5.2. Experimental Setup

Experiments were conducted using the DAL Toolbox [Ogasawara et al., 2023], with integrated support for Chronos and Lag-Llama. The last five observations of each time series were reserved for testing. TSFM predictions were generated in zero-shot mode using 30, 60, 90, 200, 500, and 1,000 samples. Final predictions were obtained by computing either the mean or the median of the generated outputs. Chronos was evaluated using pretrained models of varying sizes: tiny (8M parameters), small (46M), base (200M), and large (710M). ARIMA was optimized using the Auto-ARIMA procedure [Hyndman and Athanasopoulos, 2018], and LSTM was adopted as a baseline ML model for comparison.

The LSTM model consists of a single-layer recurrent architecture with input and hidden size equal to 6, followed by a linear output layer. The network uses the hyperbolic tangent activation function. Training is performed for up to 10,000 epochs using the Adam optimizer (learning rate 10^{-3}, batch size 8). Input sequences are constructed using a sliding window of size 6, with normalization applied via differencing. Data augmentation is performed using jittering, doubling the size of the training set.

Prediction accuracy is assessed using the Symmetric Mean Absolute Percentage Error (SMAPE), defined in Equation 1. SMAPE enables proportional comparisons and consistent evaluation across datasets, where lower values indicate more accurate forecasts. Execution times are also reported. All experiments were conducted on a system equipped with an Intel Xeon W3-2423 processor (16 threads, 4.20 GHz), 512 GB RAM, and running Ubuntu 22.04. Scripts and datasets used in this study are publicly available [Alexandrino et al., 2025].

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

5.3. TSFM Setup

To facilitate visualization, we applied a logarithmic transformation to SMAPE values. Figure 3 presents the distributions of $\log(\text{SMAPE})$ for mean and median estimators, as well as the effect of sample size on prediction accuracy. Lower values on the x-axis correspond to better performance. Peaks along the y-axis reflect estimator consistency, with narrower curves indicating greater stability. A leftward shift in the distribution denotes improved predictive accuracy.

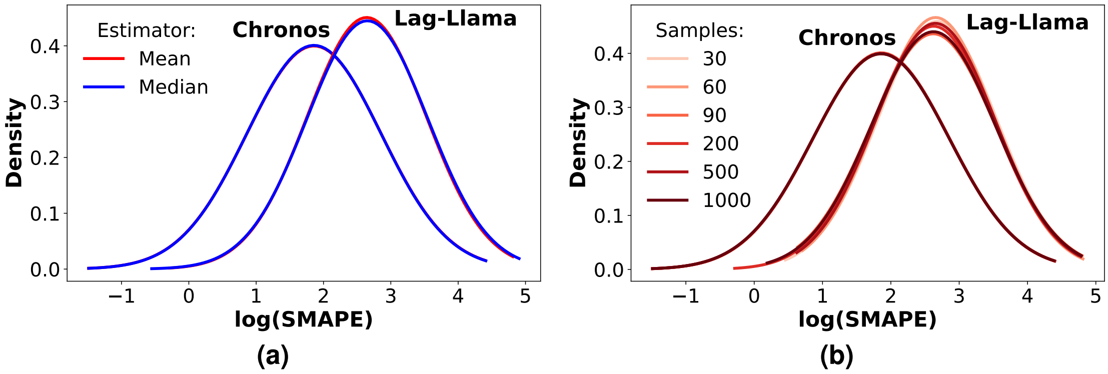

Figure 3. Evaluation of TSFM setups based on $\log(\text{SMAPE})$. (a) Analysis of mean- and median-based estimations, and (b) the impact of sample sizes on prediction performance.

Chronos outperforms Lag-Llama, with its distribution shifted toward lower values. The mean and median curves for Chronos largely overlap, indicating equivalent accuracy and consistency. Given its computational simplicity, the mean is selected as the preferred estimator for Chronos. Although performance differences are small, mean estimation requires fewer operations.

Lag-Llama presents a narrower distribution, suggesting lower variability. Its mean curve displays a higher peak density and superior performance in worst-case predictions. For this reason, the mean is also adopted for Lag-Llama.

Chronos exhibits consistent performance across different sample sizes. The similarity of density curves across attempts indicates limited benefit from increasing the number of samples. Therefore, smaller sample sizes — such as 30 or 60 — are computationally efficient without sacrificing accuracy.

Lag-Llama presents greater variability. A pronounced peak at 60 samples suggests this configuration provides improved reliability with performance comparable to that of larger sample sizes. Thus, 60 samples offer a suitable trade-off between computational cost and predictive performance.

Figure 4 compares different Chronos model sizes. The similarity among the density curves indicates that increasing model size yields limited improvement. However, average computation time grows with model size. As larger models do not provide significant performance gains, the base configuration offers the best compromise between robustness and efficiency.

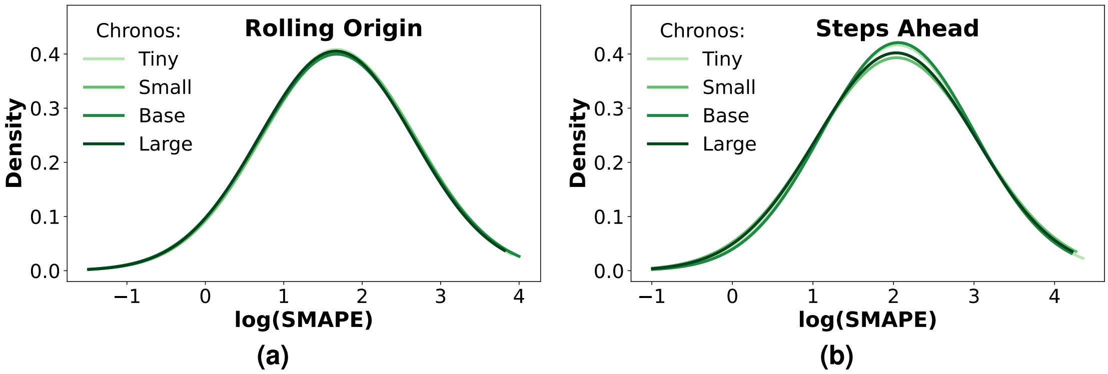

Figure 4. Evaluation of Chronos setups based on $\log(\mathrm{SMAPE})$. The figure illustrates the prediction performance of Chronos pretrained models for (a) one-step-ahead and (b) multi-step-ahead approaches

Based on these results, we select 60 samples, the mean estimator, and the Chronos base model for subsequent analyses.

5.4. Overall Performance Analysis

Table 2 presents the predictive performance and execution time of each model under the rolling origin evaluation strategy. Chronos outperforms the other models in five out of nine datasets. ARIMA exhibits the lowest computation time but achieves the best accuracy in only one dataset (Emissions N_2O). LSTM variants perform best in the Climate Change, Emissions CO_2, and GDP datasets. However, they are also associated with the highest execution times. Lag-Llama presents the weakest overall results, particularly in the Climate Change and Emissions datasets, where it yields higher prediction errors and computational costs compared to Chronos.

Chronos generalizes well across datasets. Its performance does not depend on the length of the time series, achieving good results in both shorter datasets (e.g., Pesticides, 28 observations) and longer ones (e.g., Fertilizers P_2O_5, 57 observations). The performance gap between the TSFM models is likely attributable to differences in their pretrained corpora. Lag-Llama was trained on a single dataset and has 3.5M parameters, whereas the Chronos base model comprises 200M parameters, enabling it to better capture complex temporal patterns in small and diverse datasets.

Another contributing factor may be the relative ineffectiveness of Lag-Llama's data augmentation methods (Freq-Mix and Freq-Mask) compared to Chronos' techniques

Table 2. Performance measured by SMAPE (%) under rolling origin evaluation

Dataset	ARIMA	Chronos	Lag-Llama	LSTM+diff	LSTM+diff+jitter
Climate Change	22.322	26.444	44.022	**18.991**	20.600
Emissions (CH_4)	2.204	**2.141**	6.089	3.041	3.352
Emissions (CO_2)	4.363	5.067	14.820	**4.233**	4.511
Emissions (N_2O)	**2.428**	2.552	5.069	3.823	2.807
Fertilizers (K_2O)	11.785	**11.452**	14.421	11.766	12.032
Fertilizers (N)	5.380	**4.912**	9.294	5.733	7.020
Fertilizers (P_2O_5)	11.015	**10.993**	17.373	11.266	11.400
GDP	7.578	7.352	11.924	6.672	**6.482**
Pesticides	6.525	**6.212**	10.048	7.227	8.471
Computation Time (s)	0.090	6.574	9.765	29.907	34.692

(TSMixup and KernelSynth) [Rasul et al., 2024; Ansari et al., 2024]. TSMixup combines both values and timestamps, while KernelSynth applies kernel-based transformations. These methods enhance Chronos' capacity to generalize from limited data.

Table 3 presents results under the steps-ahead evaluation strategy, which assesses model performance over extended forecasting horizons. Chronos consistently outperforms Lag-Llama across all datasets but shows greater sensitivity to the training-testing gap in terms of both accuracy and computational cost. This may be explained by its dependence on patterns learned from the pretraining corpus, which may not fully generalize to long-range forecasts.

Table 3. Performance measured by SMAPE (%) under steps ahead evaluation

Dataset	ARIMA	Chronos	Lag-Llama	LSTM+diff	LSTM+diff+jitter
Climate Change	26.736	32.087	77.830	**21.283**	23.627
Emissions (CH_4)	3.318	**3.089**	14.164	7.949	7.330
Emissions (CO_2)	**5.668**	7.899	20.135	6.244	6.697
Emissions (N_2O)	4.807	**4.587**	14.898	6.676	5.073
Fertilizers (K_2O)	17.725	15.511	25.829	**15.301**	15.524
Fertilizers (N)	**10.444**	10.916	13.657	12.121	12.929
Fertilizers (P_2O_5)	16.157	15.370	24.275	**13.048**	14.166
GDP	6.796	**6.614**	26.647	6.914	7.481
Pesticides	**9.052**	10.036	18.351	11.305	13.398
Computation Time (s)	0.094	9.071	12.855	29.942	35.331

ARIMA and LSTM models exhibit better adaptability to long-term forecasting. An exception is observed in the GDP dataset, where Chronos outperforms its own results from the rolling origin strategy — possibly due to the presence of GDP-related data in its pretraining corpus. TSFMs present their weakest performance on the Climate Change dataset, which is primarily composed of stationary series. In such cases, LSTM models demonstrate a better fit to the underlying structure.

6. Conclusion

This study evaluated the zero-shot performance of TSFMs, specifically Chronos and Lag-Llama, for forecasting small, low-frequency time series. Comparisons with traditional approaches (ARIMA) and machine learning methods (LSTM) revealed performance differences across one-step-ahead and multi-step-ahead forecasting horizons. Chronos maintained competitive accuracy in one-step-ahead scenarios but exhibited greater sensitivity in multi-step-ahead predictions, including in terms of computational efficiency. The proposed evaluation framework supports robust comparisons and confirms that TSFMs are effective alternatives for forecasting in data-scarce conditions.

Although the datasets encompass multiple sectors, further evaluations in broader domains — or in tasks such as anomaly detection and event forecasting — could provide a more comprehensive assessment of the generalization capabilities of TSFMs. In addition, deeper error analyses on series with spikes or unstable trends, as well as insights into model architecture, would help clarify current limitations. Future work may also investigate strategies to mitigate the effects of data scarcity and nonstationarity, or explore fine-tuning mechanisms to improve TSFM alignment with specific time series characteristics. These efforts could enhance the long-term forecast stability of TSFMs in specialized and dynamic scenarios.

References

Alexandrino, F., Parracho, R., Carvalho, D., and Ogasawara, E. (2025). Code and Data Repository for LLM on LFD. https://github.com/cefet-rj-dal/tsfm.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Wilson, A. G., Bohlke-Schneider, M., and Wang, Y. (2024). Chronos: Learning the Language of Time Series. http://arxiv.org/abs/2403.07815.

Bahelka, A. and de Weerd, H. (2024). Comparative analysis of Mixed-Data Sampling (MIDAS) model compared to Lag-Llama model for inflation nowcasting. http://arxiv.org/abs/2407.08510.

Bao, W., Cao, Y., Yang, Y., Che, H., Huang, J., and Wen, S. (2025). Data-driven stock forecasting models based on neural networks: A review. *Information Fusion*, 113:102616.

Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, Y., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., Salinas, D., Stella, L., Aubet, F.-X., Callot, L., and Januschowski, T. (2022). Deep Learning for Time Series Forecasting: Tutorial and Literature Survey. *ACM Comput. Surv.*, 55(6).

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.

FAO (2025). Food and agriculture data. https://www.fao.org/faostat.

Goubeaud, M., Jousen, P., Gmyrek, N., Ghorban, F., and Kummert, A. (2021). White Noise Windows: Data Augmentation for Time Series. In *2021 International Conference on Optimization and Applications, ICOA 2021*.

Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. (2023). Large Language Models Are Zero-Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*, volume 36.

Gupta, D., Bhatti, A., and Parmar, S. (2024a). Beyond LoRA: Exploring Efficient Fine-Tuning Techniques for Time Series Foundational Models. http://arxiv.org/abs/2409.11302.

Gupta, D., Bhatti, A., Parmar, S., Dan, C., Liu, Y., Shen, B., and Lee, S. (2024b). Low-Rank Adaptation of Time Series Foundational Models for Out-of-Domain Modality Forecasting. http://arxiv.org/abs/2405.10216.

Han, J., Pei, J., and Tong, H. (2022). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Cambridge, MA, 4th edition edition.

He, K., Yu, L., and Zou, Y. (2024). Crude oil future price forecasting using pretrained transformer model. *Procedia Computer Science*, 242:288–293.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Iglesias, G., Talavera, E., González-Prieto, Á., Mozo, A., and Gómez-Canaval, S. (2023). Data Augmentation techniques in time series domain: a survey and taxonomy. *Neural Computing and Applications*, 35(14):10123 – 10145.

Liao, W., Yang, Z., Jia, M., Rehtanz, C., Fang, J., and Porté-Agel, F. (2024). Zero-Shot Load Forecasting with Large Language Models. http://arxiv.org/abs/2411.11350.

Lin, N., Yun, D., Xia, W., Palensky, P., and Vergara, P. P. (2024). Comparative Analysis of Zero-Shot Capability of Time-Series Foundation Models in Short-Term Load Prediction. http://arxiv.org/abs/2412.12834.

Maior, C. S. and Silva, T. (2024). Time-series failure prediction on small datasets using machine learning. *IEEE Latin America Transactions*, 22(5):362 – 371.

Masini, R. P., Medeiros, M. C., and Mendes, E. F. (2021). Machine learning advances for time series forecasting. *Journal of Economic Surveys*, 37(1):76 — 111.

McKay, A. and Wolf, C. K. (2023). What can time-series regressions tell us about policy counterfactuals? *Econometrica*, 91(5):1695–1725.

Mello, A., Giusti, L., Tavares, T., Alexandrino, F., Guedes, G., Soares, J., Barbastefano, R., Porto, F., Carvalho, D., and Ogasawara, E. (2024). D-AI2-M: Ethanol Production Forecasting in Brazil Using Data-Centric Artificial Intelligence Methodology. *IEEE Latin America Transactions*, 22(11):899–910.

Ogasawara, E., Castro, A., Borges, H., Carvalho, D., Santos, J., Bezerra, E., and Coutinho, R. (2023). daltoolbox: Leveraging Experiment Lines to Data Analytics. https://cran.r-project.org/web/packages/daltoolbox/index.html.

Ogasawara, E., Salles, R., Porto, F., and Pacitti, E. (2025). *Event Detection in Time Series*. Springer, Switzerland.

Pacheco, C., Guimaraes, M., Bezerra, E., Lobosco, D., Soares, J., González, P. H., Andrade, A., De Souza, C. G., and Ogasawara, E. (2022). Exploring Data Preprocessing and Machine Learning Methods for Forecasting Worldwide Fertilizers Consumption. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2022-July.

Petropoulos, F. (2022). Forecasting: theory and practice. *International Journal of Forecasting*, 38(3):705–871.

Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Bayazi, M. J. D., Adamopoulos, G., Riachi, R., Hassen, N., Biloš, M., Garg, S., Schneider, A., Chapados, N., Drouin, A., Zantedeschi, V., Nevmyvaka, Y., and Rish, I. (2024). Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting. http://arxiv.org/abs/2310.08278.

Salles, R., Assis, L., Guedes, G., Bezerra, E., Porto, F., and Ogasawara, E. (2017). A framework for benchmarking machine learning methods using linear models for univariate time series prediction. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2017-May, pages 2338 – 2345.

Salles, R., Belloze, K., Porto, F., Gonzalez, P. H., and Ogasawara, E. (2019). Nonstationary time series transformation methods: An experimental review. *Knowledge-Based Systems*, 164:274 – 291.

Salles, R., Pacitti, E., Bezerra, E., Marques, C., Pacheco, C., Oliveira, C., Porto, F., and Ogasawara, E. (2023). TSPredIT: Integrated Tuning of Data Preprocessing and Time Series Prediction Models. *Lecture Notes in Computer Science*, 14160 LNCS:41 – 55.

Saravanan, H. K., Dwivedi, S., Praveen, P., and Arjunan, P. (2024). Analyzing the Performance of Time Series Foundation Models for Short-term Load Forecasting. In *Proceedings of the 2024 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '24, pages 346 – 349, New York, NY, USA. Association for Computing Machinery.

Semenoglou, A.-A., Spiliotis, E., and Assimakopoulos, V. (2023). Data augmentation for univariate time series forecasting with neural networks. *Pattern Recognition*, 134.