Evaluating the Chronos Foundation Model for Daily Stock Index Forecasting

Stanisław Łaniewski
University of Warsaw / Dep. of Quantitative Finance and Machine Learning / QFRG
Warsaw, Poland *s.laniewski@uw.edu.pl*

Robert Ślepaczuk
University of Warsaw / Dep. of Quantitative Finance and Machine Learning / QFRG
Warsaw, Poland *rslepaczuk@wne.uw.edu.pl*

Abstract

This study empirically evaluates the performance of Chronos, a recent foundation model pre-trained on a large corpus of time series data, for the task of daily stock index forecasting. Using a rolling window framework on historical Nasdaq-100 and S&P 500 data from 1995 to early 2025, we compare zero-shot and fine-tuned Chronos variants against a diverse set of established forecasting methods, including statistical benchmarks (AutoARIMA, ETS), standard deep learning models (DeepAR, DLinear, SimpleFeedForward), other Transformer-based architectures (PatchTST), and ensemble approaches. Our results, based on standard forecasting metrics and simulated trading performance, indicate that zero-shot Chronos provides competitive forecasting accuracy. It is statistically comparable to the best traditional methods, but its derived trading performance lags top benchmarks. The fine-tuned Chronos variant statistically underperformed the zero-shot version in forecast accuracy. These findings highlight the potential of foundation models and underlines the significant challenges in effective fine-tuning.

Keywords: Time Series Forecasting, Foundation Models, Chronos, Machine Learning, Algorithmic Trading

1. Introduction

Financial time series forecasting is a cornerstone of quantitative finance, crucial for portfolio allocation, risk management, and algorithmic trading strategy development. However, it is difficult to achieve correct predictions due to the noisy, non-stationary, and often regime-dependent nature of financial markets [16]. Traditional econometric models like the ARIMA and GARCH families often struggle to capture complex non-linear dynamics [4]. Standard machine learning and deep learning approaches, including Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM) [7] or Convolutional Neural Networks (CNNs), have shown promise but typically require significant task-specific training data and careful hyperparameter tuning.

Recently, the paradigm of large pre-trained foundation models, highly successful in Natural Language Processing (NLP) and Computer Vision, has been extended to time series analysis [2]. These models are trained on vast, diverse datasets and aim to provide strong zero-shot or few-shot forecasting capabilities across various domains. Chronos [1] represents a state-of-the-art example, utilizing language model architectures (specifically T5) to process tokenized time series data and perform probabilistic forecasting. By learning general temporal patterns from a massive corpus, Chronos potentially offers a robust alternative to models trained only on specific target series. Despite the theoretical appeal, the empirical performance of such foundation models, particularly in the challenging domain of daily financial forecasting, requires thorough investigation.

This paper addresses this gap by systematically comparing Chronos models against a range

of established forecasting techniques. Specifically, using historical daily data for the Nasdaq-100 and S&P 500 indices (from 1995 to early 2025), we evaluate the performance of Chronos in two modes: zero-shot and simply fine-tuned (using default settings without extensive hyperparameter optimization) on the target index data. We employ a rolling walk-forward evaluation framework to determine the relative effectiveness of Chronos, both as an off-the-shelf tool and with default automated fine-tuning, compared to established specialized and baseline models in finance.

Our experiments show that Chronos zero-shot variant achieves forecasting accuracy on par with the best traditional models, whereas simply fine-tuning Chronos with default parameters does not improve performance and even underperforms the zero-shot model. In terms of trading outcomes, Chronos forecasts yield only moderate returns, falling short of the top benchmark strategies. These results highlight the potential of foundation models for forecasting while emphasizing the challenges in effectively fine-tuning them for complex financial tasks.

The remainder of this paper is structured as follows: Section 2 briefly reviews related work on financial forecasting models and foundation models for time series. Section 3 details the methodology, including data, compared models, evaluation setup, and metrics. Section 4 presents the empirical results and analysis. Section 5 concludes the study, highlighting limitations and potential avenues for future research.

2. Related Work

Time series forecasting is a mature field with established statistical methods like ARIMA and Exponential Smoothing (ETS) serving as strong baselines [10]. However, capturing volatility changes and the complex non-linearities inherent in financial data often requires more advanced approaches [16]. Deep learning models, including RNNs, LSTMs [7], and specialized architectures like DeepAR [14] and selected Transformer-based model, PatchTST [13]), have demonstrated significant improvements in accuracy for specific forecasting tasks, although under task-specific training and hyperparameter tuning.

The recent emergence of large pre-trained foundation models offers an opportunity for time series analysis [2]. These models aim to learn universal temporal patterns from vast datasets, enabling effective zero-shot or few-shot forecasting. Examples include TimeGPT [8] and TimesFM [5], which have shown competitive zero-shot performance against traditional and deep learning models on diverse benchmarks.

This study focuses specifically on Chronos [1], a family of T5-based foundation models pre-trained by tokenizing time series values. Their success in zero-shot forecasting motivates evaluation in finance. Preliminary applications, such as [17] testing Chronos on stock returns, suggest potential for identifying weak predictive signals. However, practical profitability remains limited, and simpler specialized models often still performed better. Notably, that study highlighted a performance gap between using Chronos in zero-shot mode versus fine-tuning it on financial data.

Our work extends this line of study by systematically comparing publicly available Chronos variants (both zero-shot and using default fine-tuning settings via AutoGluon) against a curated set of established statistical, deep learning, and Transformer baselines on major stock indices (Nasdaq-100, S&P 500). With a rolling-window framework we add empirical evidence to the ongoing evaluation of foundation models in quantitative finance.

3. Methodology

3.1. Data and Preprocessing

This study utilizes daily historical data for two major US stock indices: the S&P 500 (`^GSPC`) and the Nasdaq-100 (`^NDX`). The data spans approximately 30 years, from January 1995 to

early 2025, sourced from Yahoo Finance. These indices provide challenging test cases, the Nasdaq-100 known for its technology focus and higher volatility, while the S&P 500 represents the broader US market. Daily closing prices P_t are transformed into logarithmic return rates $r_t = \log(P_t) - \log(P_{t-1})$ to improve stationarity. All analysis uses a business day frequency. Standard preprocessing, including feature scaling within each training window, is handled in a way to prevent lookahead bias.

3.2. Forecasting Models Compared

We compare the performance of Chronos foundation models against several established baseline and state-of-the-art methods.

- **Chronos [0-Shot]:** The Amazon Chronos model evaluated using its pre-trained weights without any fine-tuning on the target index data [1].
- **Chronos [Fine-tuned]:** We follow Chronos authors [1] to fine-tune the pre-trained model using suggested fine-tuning settings: learning rate = $1e^{-5}$, steps (number of gradient update steps) = 1000, batch size = 32. (These default training parameters were chosen to mimic an out-of-the-box fine-tuning scenario without extensive hyperparameter search.)
- **Statistical Baselines:** AutoARIMA and ETS, representing classical econometric approaches, with parameters automatically selected [10].
- **Standard Deep Learning Models:** DeepAR [14], DLinear [18], and a SimpleFeedForward (MLP) network.
- **Transformer-Based Model:** PatchTST [13], a recent high-performing Transformer architecture for time series.
- **Tabular/Ensemble Models:** RecursiveTabular and WeightedEnsemble, both generated by AutoML frameworks like AutoGluon library [15], which was used for model fitting and prediction.

3.3. Evaluation Framework

A rolling walk-forward validation procedure ensures robust out-of-sample performance assessment [3]. The data for each index is iterated through using sequential, non-overlapping windows:

- **Training Window Length (T_{train}):** 150 business days (approx. 7 months).
- **Testing (Prediction) Window Length (T_{test}):** 50 business days (approx. 2.5 months).
- **Step Size:** 50 business days (non-overlapping test sets).

Models are trained (or applied zero-shot) on T_{train} to forecast the subsequent T_{test} period. The window then slides forward by the step size for retraining and re-evaluation.

3.4. Performance Metrics

Model performance is evaluated using both standard forecast accuracy metrics and metrics derived from a simple trading simulation. For the trading evaluation, we consider a naïve long-short strategy: go long (buy) if the model predicts an upward move above a small threshold, go short (sell) if it predicts a sufficiently negative move, or stay flat if the predicted return is within the threshold. We assumed a signal threshold $\tau = 0.1\%$ (in log-returns) and zero transaction costs for this hypothetical strategy. Appropriate statistics were selected based on [11]. Table 1 provides the definitions of these metrics and their interpretation.

Metric	Abbrev.	Formula	Additional Information
Forecast Accuracy Metrics			
Mean Absolute Error	MAE	$\frac{1}{N}\sum \lvert r_t - \hat{r}_t \rvert$	Lower is better.
Root Mean Squared Error	RMSE	$\sqrt{\frac{1}{N}\sum (r_t - \hat{r}_t)^2}$	Lower is better.
Directional Accuracy	DA	$\%(\text{sign}(\hat{r}_t) == \text{sign}(r_t))$	Higher is better (accuracy of sign prediction).
Mean Abs. Scaled Error	MASE	$MAE/MAE_{naive,train}$	Error relative to in-sample naïve forecast; <1 is good. [9]
Avg. Pinball Loss	AvgPinball	$\text{mean}(\text{PinballLoss}_q)$	Average loss across quantiles [0.1,..,0.9]. Lower is better. [12]
Diebold-Mariano Stat.	DM	(See [6])	Tests significance of loss difference between models.
Trading Simulation Metrics (Threshold $\tau = 0.001$, Cost $c = 0.0$)			
Annualized Return	aRC	$(\prod(1 + r_{strat,t}))^{252/N} - 1$	Compounded annualized strategy return.
Annualized Std. Dev.	aSD	$\sqrt{252} \times \text{std}(r_{strat,t})$	Annualized volatility of strategy returns.
Information Ratio	IR	aRC/aSD	Sharpe Ratio (risk-free rate = 0).
Sortino Ratio	Sortino	$aRC/aSD_{downside}$	Uses std. dev. of negative strategy returns only.
Maximum Drawdown	MD	$\min(Equity_t / \max_{i \le t} Equity_i - 1)$	Max peak-to-trough equity decline (negative).
Calmar Ratio	Calmar	$aRC/\lvert MD \rvert$	Return relative to max drawdown.
Max Loss Duration	MLD	$\max(\text{duration below peak})/252$	Longest time (years) to recover previous peak.
Average Max Drawdown	AMD	$\text{mean}(MD_{yearly})$	Mean of yearly maximum drawdowns.
Number of Trades	N Trades	$\sum \lvert \Delta \text{signal}_t \rvert$	Count of signal changes.

Table 1. Performance statistics definitions.

r_t=actual log return, $\hat{r}_t$=forecast log return, $r_{strat,t}$=strategy daily log return. B&H strategy used as benchmark.

4. Results

This section presents the empirical results comparing the zero-shot and fine-tuned forecasting performance of Chronos against baseline and state-of-the-art models on daily Nasdaq-100 and S&P 500 log-returns (1995-2025) using the rolling window framework.

4.1. Overall Trading Performance

Table 2 and Table 3 summarize the key trading performance metrics over the entire 30-year period for Nasdaq-100 and S&P 500, respectively, based on the directional strategy with a 0.1% signal threshold and zero transaction costs.

Model	N Trades	aRC (%)	aSD (%)	IR (Sharpe)	Sortino	MD (%)	Calmar
WeightedEnsemble	2419	16.7	27.7	0.60	0.81	-50.2	0.33
ETS	116	15.2	27.7	0.55	0.71	-59.7	0.25
SimpleFeedForward	1129	14.3	25.3	0.57	0.67	-59.6	0.24
Chronos[0-shot]	962	13.7	24.2	0.56	0.58	-57.7	0.24
BuyHold	1	12.7	27.7	0.46	0.60	-87.9	0.14
DeepAR	2460	5.5	25.9	0.21	0.26	-71.2	0.08
Chronos[Fine-tuned]	2335	5.0	26.7	0.19	0.24	-79.8	0.06
DLinear	3947	4.5	26.6	0.17	0.22	-65.4	0.07
RecursiveTabular	3266	1.5	27.2	0.05	0.07	-86.7	0.02
AutoARIMA	424	0.8	18.1	0.05	0.03	-76.4	0.01
PatchTST	4045	-0.1	26.5	-0.00	-0.00	-74.4	-0.00

Table 2. Overall Trading Performance Metrics (Nasdaq-100, Thresh=0.001, Cost=0.0)

The trading simulation results highlight interesting differences between the indices. For the Nasdaq-100 (Table 2), several active strategies, WeightedEnsemble, ETS, SimpleFeedForward, and Chronos[0-shot], outperformed the B&H benchmark on both raw (aRC) and risk-adjusted (IR, Sortino, Calmar) metrics. Conversely, for the S&P 500 (Table 3), the B&H strategy remained superior across most metrics. The best active models based on IR for SP500 was

Model	N Trades	aRC (%)	aSD (%)	IR	Sortino	MD (%)	Calmar
BuyHold	1	8.9	19.0	0.47	0.59	-59.8	0.15
SimpleFeedForward	1434	5.5	17.1	0.32	0.34	-45.8	0.12
Chronos[0-shot]	991	5.2	13.3	0.39	0.33	-34.2	0.15
DeepAR	2759	4.7	16.4	0.29	0.32	-64.9	0.07
Chronos[Fine-tuned]	2414	2.4	17.1	0.14	0.17	-61.4	0.04
WeightedEnsemble	2423	2.3	14.9	0.16	0.15	-55.3	0.04
ETS	273	2.0	11.4	0.17	0.12	-33.7	0.06
DLinear	4114	1.2	17.6	0.07	0.08	-67.2	0.02
RecursiveTabular	3284	0.8	18.5	0.04	0.05	-60.8	0.01
AutoARIMA	403	0.5	8.4	0.06	0.03	-49.0	0.01
PatchTST	4305	-0.2	17.3	-0.01	-0.01	-78.4	-0.00

Table 3. Overall Trading Performance Metrics (SP500, Thresh=0.001, Cost=0.0)

Chronos[0-shot], which achieved competitive risk-adjusted returns (IR=0.39 vs 0.47 for B&H). The Chronos[Fine-tuned] version performed worse than the zero-shot one, however, it was more active in the market (2335 vs 962 and 2414 vs 991 trades respectively).

4.2. Forecast Accuracy Comparison

Item ID	Model	MAE	RMSE	DA (%)	MASE	AvgPinballLoss
Nasdaq-100						
	ETS	0.01195	0.01585	53.95	1.13	0.00438
	AutoARIMA	0.01197	0.01585	73.78	1.13	0.00437
	Chronos[0-shot]	0.01201	0.01594	53.35	1.30	0.00524
	WeightedEnsemble	0.01214	0.01608	53.52	1.15	0.00447
	SimpleFeedForward	0.01238	0.01637	53.82	1.17	0.00468
	DeepAR	0.01293	0.01698	52.44	1.22	0.00511
	PatchTST	0.01364	0.01779	50.32	1.29	0.00536
	Chronos[Fine-tuned]	0.01377	0.01803	51.28	1.13	0.00561
	DLinear	0.01393	0.01809	50.89	1.31	0.00544
	RecursiveTabular	0.01923	0.02469	50.23	1.82	0.01355
S&P 500						
	Chronos[0-shot]	0.00795	0.01089	52.81	1.74	0.00365
	AutoARIMA	0.00796	0.01089	73.93	1.50	0.00301
	SimpleFeedForward	0.00816	0.01115	53.47	1.54	0.00319
	WeightedEnsemble	0.00833	0.01148	51.98	1.57	0.00316
	DeepAR	0.00852	0.01149	51.64	1.60	0.00346
	ETS	0.00877	0.01280	53.14	1.65	0.00337
	PatchTST	0.00913	0.01226	50.76	1.72	0.00369
	Chronos[Fine-tuned]	0.00924	0.01246	50.73	1.50	0.00376
	DLinear	0.00930	0.01236	51.04	1.75	0.00368
	RecursiveTabular	0.01311	0.01728	50.25	2.47	0.00943

Table 4. Overall Forecast Accuracy Metrics (1995-2025, Daily Log-Returns)

We next examine the raw forecast accuracy using MAE and RMSE, alongside directional accuracy (DA) and MASE (Table 4). It reveals that the models with the best trading performance (e.g., WeightedEnsemble, SimpleFeedForward on Nasdaq) do not necessarily translate into the lowest forecast errors. Statistical models like AutoARIMA and ETS often achieve lower MAE/RMSE, particularly for the SP500. Chronos (0-shot variant) demonstrates competitive MAE/RMSE, ranking among the best models, but its DA is only slightly above 50%.

The MASE values, generally above 1, indicate that most models struggle to consistently outperform a simple naïve persistence forecast in terms of MAE. The low DA for AutoARIMA seems anomalous, it had often zero forecast.

Probabilistic forecast accuracy, represented by Average Pinball Loss, largely mirrors the RMSE rankings. Statistical models and simpler DL ensembles tend to show better calibration across quantiles compared to the more complex Transformer models or Chronos variants in this evaluation.

4.3. Statistical Significance of Forecast Errors

Model 1	Model 2	DM (MAE)	p-MAE	DM (MSE)	p-MSE
Chronos[Fine-tuned]	SimpleFeedForward	19.02	0.000***	6.74	0.000***
Chronos[Fine-tuned]	ETS	12.27	0.000***	-3.62	0.000***
Chronos[Fine-tuned]	DeepAR	11.13	0.000***	4.94	0.000***
Chronos[Fine-tuned]	Chronos[0-shot]	24.43	0.000***	6.83	0.000***
Chronos[0-shot]	SimpleFeedForward	-9.65	0.000***	-1.50	0.133
Chronos[0-shot]	ETS	-5.47	0.000***	-4.10	0.000***
Chronos[0-shot]	DeepAR	-17.01	0.000***	-1.35	0.176

Table 5. Diebold-Mariano Test Results (Selected Pairs, Nasdaq-100). Positive DM stat indicates Model 1 loss is higher (worse). *** p<0.001, ** p<0.01.

To check whether the observed differences in forecast accuracy are statistically significant, we perform Diebold-Mariano tests that compare loss differentials (MAE and MSE) between selected model pairs (Table 5). The results show that the fine-tuned Chronos variant produces forecasts that are statistically significantly less accurate (higher MAE and MSE loss) than the zero-shot variant (Chronos[0-shot]) and SimpleFeedForward ($p < 0.001$ for both MAE and MSE). It also performs significantly worse than DeepAR ($p<0.001$). It appears statistically worse than ETS based on MAE, but surprisingly better based on MSE loss (DM stat is negative), suggesting complex error distribution.

Comparing the zero-shot Chronos to others, we find it is significantly more accurate (lower loss) than SimpleFeedForward and DeepAR based on MAE, but not significantly different based on MSE. There is no statistically significant difference in accuracy between zero-shot Chronos or ETS.

In summary, these tests suggest that the default fine-tuning procedure significantly degraded the forecasting performance of the Chronos model, rather than improving it, highlighting the importance of more robust hyperparameter optimization.

4.4. Performance Across Markets

Finally, we examine the performance of the models in different markets. Figures 1 and 2 display the overall equity curves for all models on the Nasdaq-100 and S&P 500, respectively.

Overall, the results illustrate both the promise and the limitations of using a large pre-trained model like Chronos for daily stock index forecasting. In the following section, we discuss the key findings, practical limitations, and avenues for future work based on these results.

5. Discussion and Conclusion

5.1. Findings

Our empirical evaluation provides several key findings. First, foundation models like Chronos can achieve forecasting accuracy comparable to top traditional methods. Chronos[0-shot] error

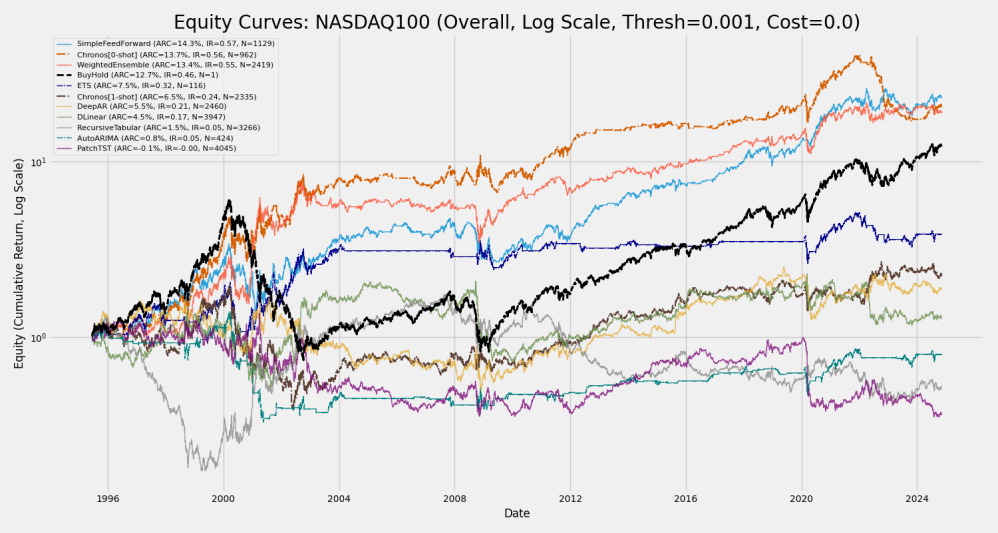

Fig. 1. Equity Curves: NASDAQ100 (Overall, Log Scale, Thresh=0.001, Cost=0.0)

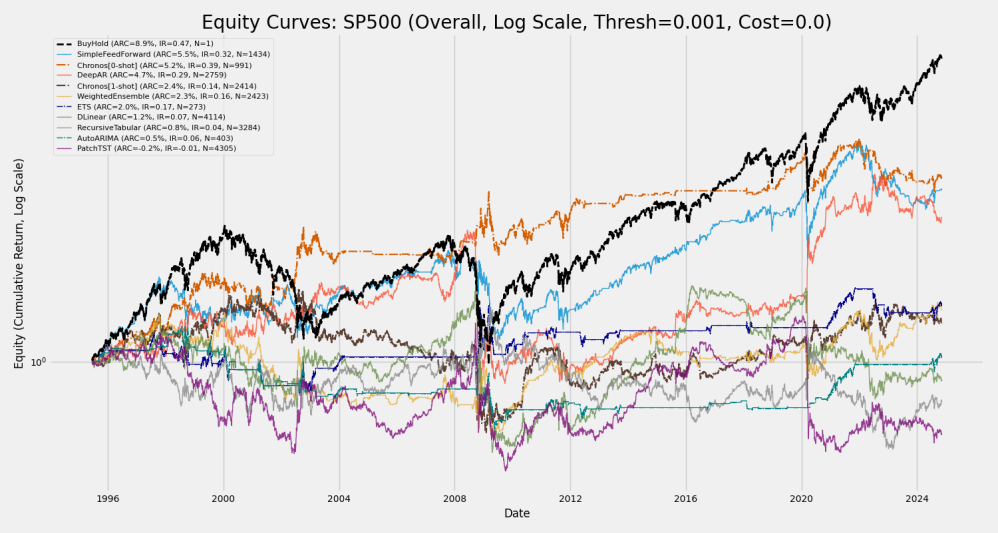

Fig. 2. Equity Curves: SP500 (Overall, Log Scale, Thresh=0.001, Cost=0.0)

rates (MAE, RMSE, Pinball Loss) were competitive with the best statistical models and deep learning benchmarks on both indices. However, Chronos fine-tuning (with default parameters) did not yield improvement - in fact, this version significantly underperformed the zero-shot variant in forecast accuracy, as confirmed by statistical tests. This counterintuitive result highlights the difficulty of effectively adapting a large pre-trained model to a specific financial time series without extensive tuning.

Second, in terms of trading performance, we found that the zero-shot Chronos strategy was among the better-performing active strategies (it achieved the highest Information Ratio for the S&P 500 and was among the top few for Nasdaq-100), yet it still outperformed by simpler statistical models (ETS, AutoARIMA). The fine-tuned Chronos, meanwhile, lagged substantially in trading outcomes.

In conclusion, foundation models like Chronos offer promising capabilities for time series forecasting, but their zero-shot application in complex domains like daily financial markets requires careful benchmarking.

5.2. Limitations

This study has several important limitations. We focused on only two large stock indices (Nasdaq-100 and S&P 500) at a daily frequency, which may not generalize to other assets or higher-frequency trading. We evaluated only two configurations of Chronos (zero-shot and one particular fine-tuning setting), without exploring more extensive hyperparameter tuning. In addition, we used a single simple trading strategy to judge economic performance, assuming zero transaction costs.

5.3. Future Work

Future research could build on these findings in several ways. Broader evaluations could be conducted on different markets and other asset classes and at different frequencies (intraday data). Future studies can create sophisticated strategies based on foundation model's probabilistic outputs and compare them against more complex trading benchmarks. Additionally, they could improve the fine-tuning process for foundation models. Finally, as foundation models for time series evolve, investigating model interpretability and trustworthiness in forecasting financial markets will be important for real-world adoption.

6. Declarations

Large Language Models, namely ChatGPT and Gemini, were used in this research for evaluation, as well as for text, code, and table polishing.

No funding was received to assist with the preparation of this manuscript. The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] 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., Wang, Y.: Chronos: Learning the language of time series (2024), https://arxiv.org/abs/2403.07815

[2] Benidis, K., Rangapuram, S.S., Stella, L., Turbé, H., Paschali, K., Turkmen, C., Mercado, P., Callot, L., Januschowski, T.: Deep learning for time series forecasting: Tutorial and literature survey. ACM Computing Surveys 55(6), pp. 1–36 (2023), https://doi.org/10.1145/3533382

[3] Bergmeir, C., Benítez, J.M.: Note on the need for adequate evaluation strategies for classification. Neural Networks 25

[4] Cont, R.: Empirical properties of asset returns: stylized facts and statistical models. Quantitative Finance 1(2), pp. 223–236 (2001)

[5] Das, A., Kong, W., Sen, R., Zhou, Y.: A decoder-only foundation model for time-series forecasting. Proceedings of the 41st International Conference on Machine Learning (2024), https://arxiv.org/abs/2310.10688

[6] Diebold, F.X., Mariano, R.S.: Comparing predictive accuracy. Journal of Business & Economic Statistics 13(3), pp. 253–263 (1995)

[7] Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research 270(2), pp. 654–669 (2018)

[8] Garza, A., Mergenthaler-Canseco, M., Alcaraz, C.: Timegpt: Foundation models for zero-shot time series forecasting (2023), https://arxiv.org/abs/2310.03589

[9] Hyndman, R.J.: Another Look at Forecast Accuracy Metrics for Intermittent Demand. Foresight: The International Journal of Applied Forecasting (4), pp. 43–46 (June 2006), https://ideas.repec.org/a/for/ijafaa/y2006i4p43-46.html

[10] Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles and Practice. OTexts, 3rd edn. (2021), https://otexts.com/fpp3/

[11] Kijewski, M., Ślepaczuk, R., Wysocki, M.: Predicting prices of s&p 500 index using classical methods and recurrent neural networks. International Conference on Information Systems Development (2024), https://api.semanticscholar.org/CorpusID:272563359

[12] Koenker, R., Bassett Jr, G.: Regression quantiles. Econometrica: journal of the Econometric Society pp. 33–50 (1978)

[13] Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers (2023), https://arxiv.org/abs/2211.14730

[14] Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T.: Deepar: Probabilistic forecasting with autoregressive recurrent networks (2019), https://arxiv.org/abs/1704.04110

[15] Shchur, O., Turkmen, C., Erickson, N., Shen, H., Shirkov, A., Hu, T., Wang, B.: Autogluon-timeseries: Automl for probabilistic time series forecasting. ArXiv abs/2308.05566 (2023), https://arxiv.org/abs/2308.05566

[16] Tsay, R.S.: Analysis of Financial Time Series. Wiley Series in Probability and Statistics, John Wiley & Sons, 2nd edn. (2005)

[17] Valeyre, S., Aboura, S.: Llms for time series: an application for single stocks and statistical arbitrage (2024), https://arxiv.org/abs/2412.09394

[18] Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? Proceedings of the AAAI Conference on Artificial Intelligence 37(9), pp. 11121–11128 (Jun 2023), https://ojs.aaai.org/index.php/AAAI/article/view/26317