

AI METHODS FOR STOCK-PRICE FORECASTING IN THE INDIAN MARKET: THE EFFECT OF MACROECONOMIC FACTORS

Ankit Tripathi¹, Oldřich Trenz¹

¹Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic

DOI: https://doi.org/10.11118/ecos2026_038

ABSTRACT

This article compares four artificial-intelligence forecasting methods (LSTM, SARIMAX, XGBoost and FB-Prophet) on a decade of daily prices of Reliance Industries, one of the largest companies on the Indian stock market. Most published forecasting studies use US, Chinese or European data; the Indian market is less frequently used as a test bed. In the source thesis, a Scopus query for “time series forecasting in India” refined by “financial market” returned 137 documents, while the broader bibliometric dataset contained 2,202 documents. The dataset covers 2 December 2013 to 30 November 2023 (over 2,600 trading days). Each model was first trained on price features alone and then re-evaluated after merging five macroeconomic indicators (gold price, USD/INR, the 10-year government-bond yield, the Wholesale Price Index and the Consumer Price Index). Accuracy was measured by MSE, MAE and the coefficient of determination R^2 . Adding macroeconomic factors did not help every model. XGBoost improved across all three metrics (MSE 39.35 to 24.58) and was the only method that gained without producing implausibly perfect numbers; LSTM improved only with the larger 90% training split, while FB-Prophet overfitted badly (MSE 11.17 to 1,912.35). The results are specific to one company and a retrospective forecasting experiment on historical data; they do not represent live trading performance.

Keywords: time series forecasting, LSTM, XGBoost, SARIMAX, Indian stock market, macroeconomic factors

JEL Code: C53



This article is licensed under CC BY-SA 4.0 Attribution-ShareAlike 4.0 International license. <https://creativecommons.org/licenses/by-sa/4.0/deed.en>

1 INTRODUCTION

Stock-price series are hard to predict. They are non-stationary, nonlinear and noisy, and they jump on political and economic news that no model sees coming. The classical statistical approach assumes a fixed data-generating process; financial markets rarely cooperate with that assumption. This is the reason machine-learning and deep-learning methods, which learn the structure from the data rather than imposing it, have moved into the field over the past decade.

There is a geographical imbalance in the published work. Forecasting models are usually tested on the US, Chinese or European markets, and far less often on India, even though India was the world's fifth-largest economy by GDP in 2023. The source thesis reports that a Scopus query for "time series forecasting in India" refined by "financial market" returned 137 documents; in contrast, its broader bibliometric dataset for time-series forecasting, artificial intelligence and financial markets contained 2,202 documents. The earlier bibliometric survey by Janková (2021) drew on 353 articles for the global topic. Part of the explanation lies in infrastructure rather than interest. The source thesis cites the Digital Quality Index 2023 as a contextual limitation: India ranked 52nd overall, 91st of 121 countries for electronic infrastructure, and its internet-affordability rank fell from 8 in 2020 to 28 in 2023. Add the full source to the references, or omit this contextual sentence from the conference version. The aim of this work is to benchmark four AI forecasting methods on Indian data and to test a concrete question: does adding macroeconomic context improve the forecast, and for which method?

1.1 Forecasting methods and prior results

The four methods cover the main families used in practice. Long Short-Term Memory (LSTM) networks were introduced by Hochreiter and Schmidhuber (1997) to keep gradient information over long sequences; Fischer and Krauss (2018) applied them to S&P 500 constituents and found they beat memory-free classifiers, which is also why that paper is among the most-cited in the area (1,007 citations). ARIMA and its exogenous extension SARIMAX follow the Box-Jenkins tradition (Box *et al.*, 2015) and remain a strong linear baseline. XGBoost, the gradient-boosted tree ensemble of Chen and Guestrin (2016), has become a default tabular learner and was used for stock prediction together with LSTM by Gumelar *et al.* (2020). FB-Prophet decomposes a series into trend and seasonal components and was applied to forecasting by Sharma *et al.* (2022).

Hybrid models recur in the literature. Zhang (2003) and Khashei and Bijari (2011) combined ARIMA with neural networks to capture linear and nonlinear structure at once; the latter work has accumulated 1,493 citations. For India specifically, Aithal *et al.* (2019) identified the macroeconomic indicators that move the Indian market, Srivastava *et al.* (2023) and Kumar *et al.* (2023) compared machine-learning models on Indian exchanges, and the second of these contrasted ARIMA with an RNN-LSTM directly. These studies motivated both the model selection here and the decision to test macroeconomic factors as inputs.

2 METHODOLOGY

The case study uses Reliance Industries Ltd. (ticker RELIANCE.NS), with daily historical prices from Yahoo Finance for 2 December 2013 to 30 November 2023; the source thesis describes the dataset as containing over 2,600 rows after preprocessing. The columns are Date, Open, High, Low, Close, Adjusted Close and Volume; the target is Adjusted Close. All models were implemented in Python on Google Colaboratory and scored with the same three metrics: mean squared error (MSE), mean absolute error (MAE) and the coefficient of determination R^2 .

LSTM was tried in two forms. The first fed the network three past values of the target (input shape 3×1) through one LSTM layer of 30 memory units and two dense layers; it overfitted, with test MSE 1,825.23 against training MSE 574.16. The second form learned from the correlated price features (Open, High, Low, Close), used a StandardScaler, LSTM and dense layers of 64 and 32 units and 20% dropout, and removed the overfitting. SARIMAX followed a Box-Jenkins identification from the ACF and PACF plots, giving order $(p = 0, d = 1, q = 2)$, that is no autoregressive term, first-order differencing and two moving-average terms, with price columns added as exogenous regressors. XGBoost used the XGBRegressor; two rows with missing values were dropped and Volume was removed for low correlation. FB-Prophet was set to model weekday data only, with trend, weekly, yearly and daily components, the last 500 rows held out for testing.

The second part of the experiment added five macroeconomic indicators from Investing.com India: the gold price, the USD/INR exchange rate, the 10-year government-bond yield, the Wholesale Price Index (WPI) and the Consumer Price Index (CPI). Their correlations with Adjusted Close range from strong to weak: gold 0.93, USD/INR 0.90, WPI 0.48, CPI 0.16 and the bond yield -0.29 . The two price indices are published monthly, which limits their resolution against daily prices. Each model was scored twice, once on prices alone (“before”) and once on the merged dataset (“after”), so that the contribution of the macroeconomic context could be read off the change in the metrics.

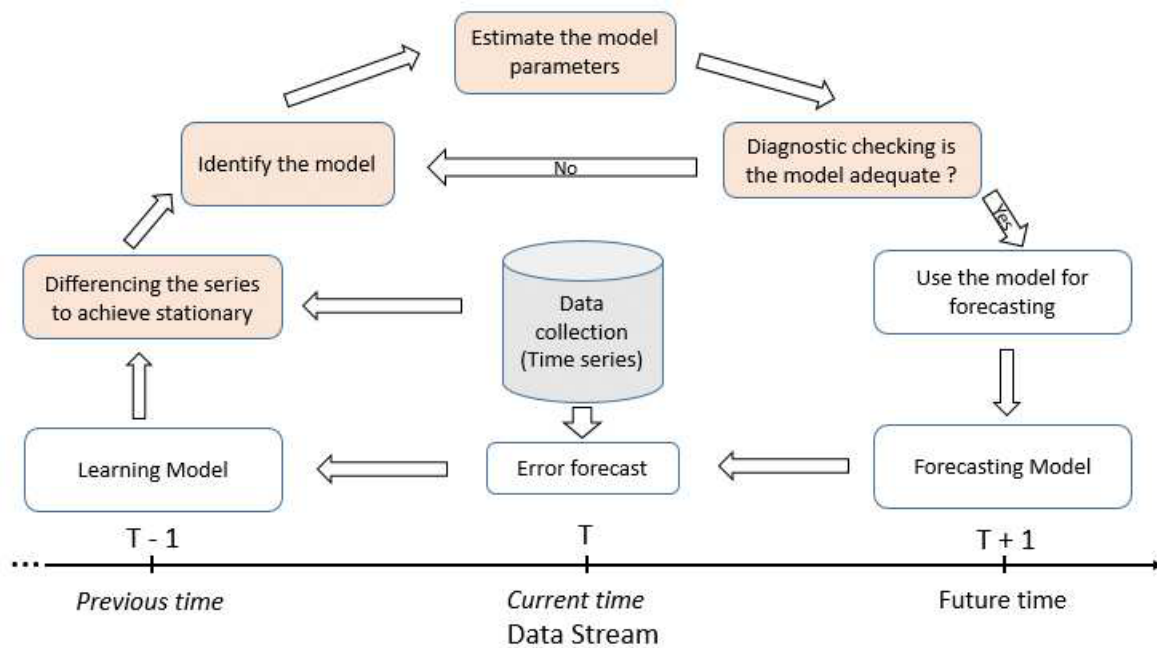


Fig. 1: Box-Jenkins iterative workflow for ARIMA-based time series forecasting; model identification, parameter estimation and diagnostic checking are repeated until the model is adequate for forecasting (adapted from Tripathi, 2023).

3 RESULTS

Table 1 lists the baseline accuracy of each model on prices alone. The two tree- and decomposition-based models (XGBoost, FB-Prophet) started far ahead of the sequence models on MSE, which already hints that for this single, fairly trending series the harder-to-train recurrent network was not the obvious winner.

Table 2 reports selected before/after metrics after the five macroeconomic indicators were merged in. Because the compact conference table does not include every metric for every model, the text should state explicitly that it presents the key metrics needed for comparison. XGBoost (in bold) is the only method that improved on the reported MSE, MAE and R^2 values while keeping plausible, non-perfect scores.

The training-split size mattered for LSTM. At the 80% split the macroeconomic inputs made it worse (MSE rose from 628.12 to 1,070.10); at the 90% split they helped (MSE fell from 577.59 to 323.08, R^2 rose to 0.9796). SARIMAX reported MSE and MAE of zero with R^2 of 1 after the merge. For daily stock prices that perfect fit almost certainly signals information leakage from the strongly collinear Open/High/Low/Close regressors rather than genuine predictive skill, so it cannot be read as a real improvement. FB-Prophet moved the opposite way and overfitted, its MSE jumping from 11.17 to 1,912.35. XGBoost gained quietly and consistently: MSE 39.35 to 24.58, MAE 4.75 to 3.27, R^2 0.9999 to 0.99996.

Tab. 1 Baseline accuracy on price features only (80% training split; LSTM second approach)

Model	MSE	MAE	R^2
LSTM	628.12	21.45	0.9602
SARIMAX	59.22	6.08	1.00
FB-Prophet	11.17	2.39	1.00
XGBoost	39.35	4.75	0.9999

Tab. 2 Effect of macroeconomic factors on each model, before vs. after merging (Tripathi, 2023)

Model	Training	Metric	Before	After
LSTM	80%	MSE	628.12	1070.10
LSTM	90%	MSE	577.59	323.08
LSTM	90%	MAE	21.37	14.66
LSTM	90%	R^2	0.9607	0.9796
SARIMAX	80%	MSE	59.22	0
FB-Prophet	80%	MSE	11.17	1912.35
FB-Prophet	80%	R^2	1.00	0.88
XGBoost	80%	MSE	39.35	24.58
XGBoost	80%	MAE	4.75	3.27
XGBoost	80%	R^2	0.9999	0.99996

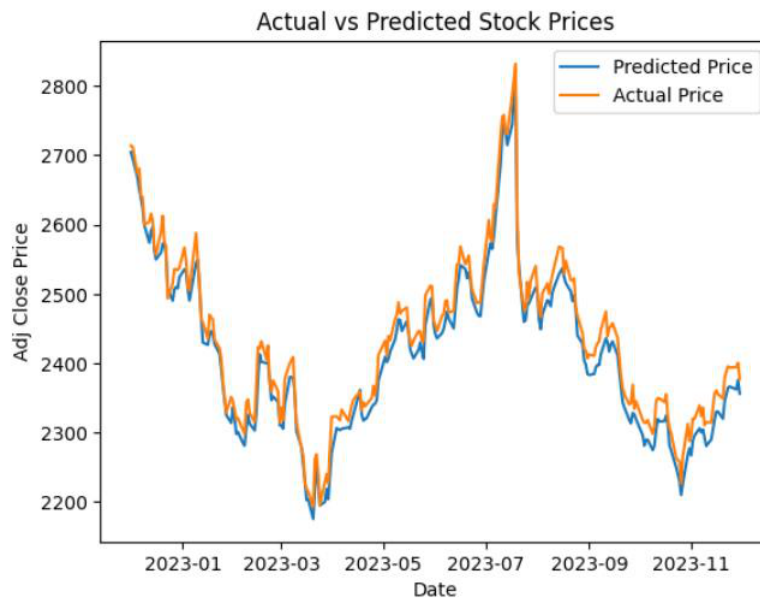


Fig. 2: Actual vs. predicted adjusted close prices on the held-out test set (LSTM, second approach). The forecast tracks the level and most turning points of the 2023 price path but lags the sharpest moves (adopted from Tripathi, 2023)

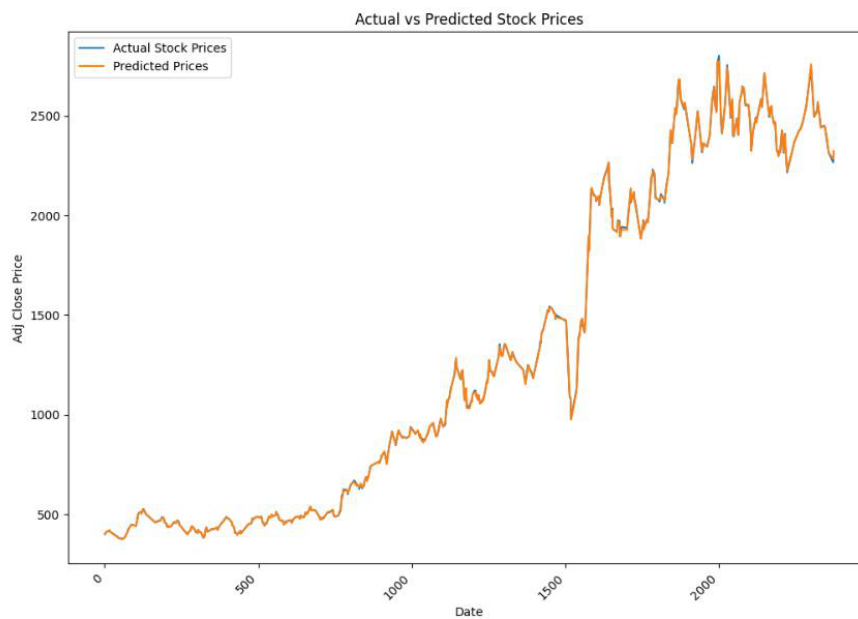


Fig. 3: Actual vs. predicted adjusted close prices over the full 2013–2023 sample after merging macroeconomic factors; the predicted curve overlaps the actual one across the whole range, including the steep rise after index 1,500 (adopted from Tripathi, 2023)

4 DISCUSSION

Why did the same five macroeconomic factors help XGBoost but hurt FB-Prophet? The answer is in how each model treats extra inputs. XGBoost can treat additional regressors as candidate split features and may re-duce their influence when they do not improve the objective function; therefore, a weak monthly series such as CPI (correlation 0.16) is less likely to distort the fit than in a model that explicitly adds external regressors to its trend structure. FB-Prophet, by contrast, folds external regressors into an additive model on top of its trend and seasonality; a noisy low-resolution input there shifts the whole fit, and the held-out 500-row test then pays for it. The increase in FB-Prophet's MSE from 11.17 to 1,912.35 is much larger than five-fold; it is more accurate to describe it as an approximately 170-fold increase, consistent with severe degradation after adding the macroeconomic regressors.

The LSTM result is about data, not architecture. The recurrent network has many more parameters to estimate than a tree ensemble, and its gates can only learn the relationship between, say, the USD/INR rate and the price if they see enough examples. At 80% training the extra inputs were closer to noise and the network degraded; the additional ten percentage points of data at the 90% split were enough for the gates to start using them, and the error dropped by almost half. This is the familiar behaviour of deep models on small tabular problems: they reward more data more steeply than shallow models do.

The SARIMAX numbers deserve scepticism rather than celebration. A reported MSE of zero on out-of-sample daily adjusted closing prices is not plausible in a real forecasting setting and therefore requires a leakage or evaluation-protocol check. The most likely cause is that the exogenous Open, High and Low values for the same day are near-perfect predictors of the Adjusted Close of that day, so the model is partly reading the answer from its inputs. This should be treated as a potential leakage or evaluation artefact rather than as genuine predictive superiority. The practical ranking in this study should therefore rest mainly on XGBoost, whose improvement is large enough to be useful and small enough to be plausible.

Two limits frame all of this. The evaluation is a retrospective forecasting experiment on one company, Reliance Industries, so the numbers say nothing directly about live trading, portfolio performance or other firms, where firm-specific variables and political shocks would matter. And the macroeconomic set is small and partly monthly; richer or higher-frequency indicators might change which model benefits. The geographical gap that motivated the study, few Indian forecasting papers and weak digital infrastructure, also means there is little local work to compare against, so these results are a starting point rather than a settled benchmark.

5 CONCLUSION

Within the single-asset experiment on a decade of Reliance Industries prices, XGBoost produced the most defensible improvement after adding macroeconomic factors. It gave a strong baseline and was the only one of the four methods that improved on every metric once macroeconomic factors were added, without the suspiciously perfect output that made the SARIMAX result unusable. For a practitioner testing a single Indian equity with a limited set of macroeconomic series, gradient-boosted trees appear to be a cautious first modelling choice in this proof-of-concept; the recurrent network benefited only when the training split was increased, while FB-Prophet degraded after adding low-resolution regressors. The SARIMAX results should be reported as requiring further

verification, because a perfect score on financial data usually indicates an evaluation artefact or information leakage rather than genuine forecasting ability. The next steps are to repeat the test across several companies and markets, to add firm-specific and event-based variables, and to replace monthly indices with higher-frequency macroeconomic data.

Acknowledgements

This paper was supported by the project CZ.02.1.01/0.0/0.0/16_017/0002334 Research Infrastructure for Young Scientists, which is co-financed from the Operational Programme Research, Development and Education.

Anthropic Claude was used to summarize the text of the final thesis (Tripathi, 2023) and transform it into the form of this compact paper. The paper was then reviewed and fine-tuned by the author. The author is responsible for the correctness of the final text.

REFERENCES

- AITHAL, P. K., DINESH, A. U., GEETHA, M. 2019. Identifying Significant Macroeconomic Indicators for Indian Stock Markets. *IEEE Access*. 7, 143829–143840. <https://doi.org/10.1109/ACCESS.2019.2945603>
- BOX, G. E. P., JENKINS, G. M., REINSEL, G. C., LJUNG, G. M. 2015. *Time Series Analysis: Forecasting and Control*. 5th ed. Hoboken, NJ: John Wiley & Sons. ISBN 978-1-118-67502-1.
- CHEN, T., GUESTRIN, C. 2016. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. <https://doi.org/10.1145/2939672.2939785>
- FISCHER, T., KRAUSS, C. 2018. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*. 270(2), 654–669. <https://doi.org/10.1016/j.ejor.2017.11.054>
- GUMELAR, A. B., SETYORINI, H., ADI, D. P. *et al.* 2020. Boosting the Accuracy of Stock Market Prediction using XGBoost and Long Short-Term Memory. In: *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp. 609–613. <https://doi.org/10.1109/iSemantic50169.2020.9234256>
- HOCHREITER, S., SCHMIDHUBER, J. 1997. Long short-term memory. *Neural Computation*. 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- JANKOVÁ, Z. 2021. A Bibliometric Analysis of Artificial Intelligence Technique in Financial Market. *Scientific Papers of the University of Pardubice*. 29(3). <https://doi.org/10.46585/sp29031268>
- KHASHEI, M., BIJARI, M. 2011. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Applied Soft Computing*. 11(2), 2664–2675. <https://doi.org/10.1016/j.asoc.2010.10.015>
- KUMAR, J. P. S., SUNDAR, R., RAVI, A. 2023. Comparison of stock market prediction performance of ARIMA and RNN-LSTM model — A case study on Indian stock exchange. *AIP Conference Proceedings*. 2875, 020010 <https://doi.org/10.1063/5.0154124>
- SHARMA, K., BHALLA, R., GANESAN, G. 2022. Time Series Forecasting Using FB-Prophet. In: *CEUR Workshop Proceedings*, pp. 59–65.
- SRIVASTAVA, S., PANT, M., GUPTA, V. 2023. Analysis and prediction of Indian stock market: a machine-learning approach. *International Journal of System Assurance Engineering and Management*. 14(5), 1567–1585. <https://doi.org/10.1007/s13198-023-01934-z>

- TRIPATHI, A. 2023. *The use of artificial intelligence methods for time series prediction*. Master's thesis, head: Trenz, O. Brno: Mendel University in Brno.
- ZHANG, G. P. 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*. 50, 159–175. [https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0)

Contact information

Ankit Tripathi: e-mail: qqtripat@node.mendelu.cz

Oldrich Trenz: oldrich.trenz@mendelu.cz