

Deep Learning for Emerging Market Forecasting: LSTM Performance on Indian IT Sector Stocks with ARIMA Benchmark Comparison

Kulkarni Parag M.^{1*}, and Sanjay Azade²

¹Department of Computer Science and IT, SBES College of Science, Chh. Sambhajinagar, Maharashtra, India

²Department of Computer Science and IT, Dr. G.Y. Pathrikar College of CS & IT, MGM University, Chh. Sambhajinagar, Maharashtra, India

Corresponding author: Kulkarni Parag M, parag_sbes@outlook.com

Abstract: Forecasting stock prices in emerging markets presents unique challenges rooted in structural volatility, episodic macro shocks, and non-stationary return distributions. While classical econometric approaches such as ARIMA have served as the standard baseline for decades, the question of whether deep learning architectures offer a measurable and reliable advantage in such environments has not been conclusively answered—particularly for individual stock-level prediction in India’s information technology sector. This paper investigates whether a stacked Long Short-Term Memory (LSTM) deep learning model can outperform an ARIMA benchmark when applied to ten years of daily closing price data for five NSE-listed Indian IT stocks: Infosys (INFY), HCL Technologies (HCLTECH), Tech Mahindra (TECHM), Wipro (WIPRO), and Tata Consultancy Services (TCS), covering January 2016 to June 2026. A two-layer LSTM with dropout regularization was trained on 80/10/10 chronological train-validation-test splits, evaluated across three feature configurations—univariate, price-based, and full technical indicator sets—against pre-computed ARIMA forecasts as a benchmark. Seven evaluation criteria were applied: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Coefficient of Determination (R^2), directional accuracy, and Theil’s U statistic. On all five stocks, the LSTM architecture outperformed ARIMA by a wide margin—RMSE fell by 74.1% for INFY, 84.66% for HCLTECH, 87.22% for TECHM, 86.15% for WIPRO, and 88.35% for TCS. The R^2 values of the best LSTM models spanned 0.77 to 0.94, whereas ARIMA returned negative R^2 on four of the five series—a clear sign it cannot keep pace with shifting price dynamics. Structured autocorrelation and non-Gaussian residuals nevertheless persisted in the LSTM outputs, pointing to a need for explicit volatility modeling in future work. The results affirm that deep learning provides a strong empirical advantage over classical time series models in emerging market stock forecasting, while also motivating hybrid approaches capable of capturing both trend dynamics and volatility clustering.

Keywords: LSTM; deep learning; stock price forecasting; NSE; Indian IT sector; ARIMA benchmark; emerging markets; time series

1. Introduction

Financial time series forecasting has been a persistent area of inquiry in both economics and computer science, owing to its theoretical complexity and practical stakes. In emerging economies like India, these challenges are amplified: equity markets respond not only to company fundamentals but also to currency volatility, sectoral regulatory shifts, and exogenous shocks such as global pandemics. The Indian technology sector, represented on the National Stock Exchange (NSE), is of particular interest because it combines global demand exposure with domestic macroeconomic sensitivity—a combination that challenges standard forecasting paradigms.



Autoregressive Integrated Moving Average (ARIMA) models have occupied the default role of baseline forecaster in univariate financial time series work for decades. Their transparent Box-Jenkins derivation and well-understood theoretical properties have made them the yardstick against which competing methods are judged [1][2]. The trouble is that ARIMA rests on assumptions—linearity, stationarity, and Gaussian disturbances—that stock returns regularly breach. The COVID-19 pandemic laid this bare: within weeks of March 2020, equity prices underwent sharp structural breaks, sudden regime shifts, and bursts of volatility clustering that no linear model is equipped to track [3].

Long Short-Term Memory (LSTM) networks, a variant of recurrent neural networks introduced by Hochreiter and Schmidhuber [4], address these limitations directly. Their gating mechanism allows them to learn long-range dependencies in sequential data without the vanishing gradient problem that afflicts vanilla RNNs. Since their introduction to finance by Bao et al. [5], LSTMs have been applied widely to equity forecasting—but comparisons against classical baselines on individual emerging-market stocks remain underrepresented in the literature. Most existing work either focuses on index-level prediction, confines itself to mature markets, or uses rolling-window ARIMA variants as impromptu competitors rather than properly fitted benchmarks.

As part of a broader research program on Indian IT sector forecasting, the authors conducted a parallel ARIMA structural break analysis on the same five stocks and identical ten-year dataset. That analysis demonstrated that while ARIMA adequately characterized the autocovariance structure of individual stock series, it failed to generate competitive short-horizon forecasts in the post-COVID recovery period, yielding negative R^2 values on all five stocks during the evaluation window. The present work determines whether deep learning resolves that forecasting gap and quantifies the magnitude of improvement.

This study makes the following contributions:

- A rigorous ten-year LSTM analysis of five NSE-listed Indian IT stocks (INFY, HCLTECH, TECHM, WIPRO, TCS), with data spanning January 2016 to June 2026, using a standardized 80/10/10 chronological split to prevent look-ahead bias.
- Three feature configurations—univariate close-price, price-feature enriched, and full technical indicator sets—evaluated per stock to establish which input regime performs best in practice.
- A direct RMSE-ranked ARIMA vs. LSTM benchmark comparison using ARIMA forecasts computed by the authors on the same dataset and test period under a standardized 13-step analysis pipeline, providing paired performance data on identical stock series.
- Residual diagnostic analysis (Ljung-Box, Jarque-Bera, ARCH-LM) to characterize the nature of LSTM forecast errors and identify directions for future hybrid modeling.

Section 2 surveys the relevant literature. Section 3 covers the dataset, feature engineering, and model architecture. Section 4 details the experimental configuration. Section 5 presents and interprets the empirical results. Section 6 concludes and outlines future research directions.

2. Literature Review

2.1 Classical Time Series Methods in Stock Forecasting

The dominance of ARIMA in financial time series analysis has been well-established since Box and Jenkins formalized the model selection procedure [1]. Numerous studies have applied ARIMA to equity markets with reasonable short-term accuracy on stationary transformed series, particularly in low-volatility periods. Working on Indian equity indices, Kumar and Thenmozhi [6] showed that ARIMA bests a random walk benchmark at the one-day horizon, though accuracy erodes quickly as the forecast window widens. On NYSE-listed securities, Adebisi et al. [7] reached a similar verdict—ARIMA held its own for next-day prediction but faltered around corporate earnings announcements. Across these and related studies, a shared weakness emerges: ARIMA cannot accommodate nonlinear dynamics or abrupt regime changes, a shortcoming that bites especially hard in emerging markets where policy surprises and macro shocks are common [8].

2.2 LSTM and Recurrent Neural Networks for Financial Forecasting

Among deep learning methods designed for sequential data, LSTM networks have attracted by far the most attention from financial forecasters. Hochreiter and Schmidhuber [4] introduced the cell-state gating mechanism that prevents gradients from vanishing over long sequences; Bao et al. [5] then showed, on Chinese stock indices, that

stacking LSTMs with autoencoders produces gains over shallow classifiers. Fischer and Krauss [9] conducted one of the most methodologically careful assessments, testing LSTM on all S&P 500 constituents and documenting outperformance over ARIMA, random forests, and plain deep nets. In a direct head-to-head comparison on general financial time series, Siami-Namini et al. [10] found that LSTM cut RMSE by 85–87% relative to ARIMA—a range that aligns closely with the margins observed in the present study.

The Indian market has also received attention. Selvin et al. [11] pitted LSTM against ARIMA and a sliding-window neural network on NSE data and reported that LSTM returned lower RMSE on the majority of tested stocks. Pawar et al. [12] applied LSTM to BSE Sensex, finding good directional accuracy but also observing that price-level RMSE shifted considerably depending on the chosen lookback window—a hyperparameter left unoptimized in that work. Mehtab et al. [13] took a convolutional-LSTM approach on NSE equities and reported that richer technical-indicator inputs boosted direction accuracy while sometimes raising point-estimate RMSE—a tension our own experiments echo for INFY and HCLTECH, where the univariate configuration outperformed the feature-enriched variants.

2.3 Comparative and Hybrid Approaches

Recognition that no single model dominates every context has driven sustained interest in hybrid designs. Zhang’s foundational framework [14] split a financial series into a linear component (modeled by ARIMA) and a nonlinear remainder (captured by an ANN)—a decomposition idea that Chen and He [15] later extended by substituting LSTM for the neural network, reporting gains on Chinese blue-chip equities. In the Indian setting, Jain and Tripathi [16] showed that adding an LSTM correction layer on top of ARIMA residuals beat either method solo on Nifty 50 data. On the volatility side, Kim and Won [17] showed that combining LSTM with multiple GARCH-type models for stock index volatility forecasting consistently outperformed any single GARCH specification—a finding that directly motivates the hybrid volatility-aware architecture we are developing in follow-on work.

The COVID-19 shock added further evidence. Ashraf [18] tracked model performance through the sharp drawdown of March–April 2020 and found machine learning methods held up considerably better than ARIMA during those turbulent weeks. Across European indices, Lyócsa et al. [19] reported similar resilience in LSTM-based volatility estimates relative to GARCH-class benchmarks. Both findings reinforced our interest in examining whether the LSTM advantage extends to the Indian IT sector, which experienced severe disruption in that same period.

2.4 Gaps Addressed by This Study

Despite the extensive literature, several gaps persist. First, most Indian IT stock studies use sector index data rather than individual stocks, masking heterogeneity that individual-level analysis reveals. Second, direct comparison of properly fitted ARIMA forecasts with LSTM on the same dataset over the same period is rare. Third, studies covering the extended post-pandemic period through 2025–2026 are nascent. The present study addresses all three gaps through individual stock analysis, paired benchmark comparison, and a dataset extending to mid-2026.

3. Dataset and Methodology

3.1 Dataset

Daily close price histories were assembled for five technology companies listed on India’s National Stock Exchange: Infosys (INFY), HCL Technologies (HCLTECH), Tech Mahindra (TECHM), Wipro (WIPRO), and Tata Consultancy Services (TCS). For INFY, HCLTECH, TECHM, and WIPRO, the records run from January 2016 through January 2026, providing between 2,465 and 2,468 trading sessions each. TCS data spans June 2016 to June 2026, totalling 2,460 sessions. INFY, HCLTECH, TECHM, and WIPRO carry six fields—Date, Open, High, Low, Close, and Volume—while TCS contains only Date and Close, which constrained its feature engineering to quantities derived from the closing price alone.

Fig. 1 illustrates the full ten-year close price series for INFY, representative of the broader dataset. The series exhibits the characteristic COVID-19 trough in March 2020 followed by a strong recovery, a bull run through 2021–2022, and subsequent consolidation—structural dynamics that ARIMA cannot model adaptively.



Fig. 1. Close price time series for INFY (January 2016 – January 2026) with COVID-19 structural break markers (March 11 and March 24, 2020).

Table 1 gives a statistical snapshot of each series. INFY and TCS lean slightly to the left (negative skewness), whereas HCLTECH, TECHM, and WIPRO skew positively. Every stock produces a kurtosis below zero, meaning extreme daily moves occur less often than a normal distribution would imply—a pattern typical of large-cap equities traded over a long, relatively stable horizon.

Table 1: Descriptive Statistics of NSE Indian IT Sector Stocks (January 2016 – June 2026)

Stock	N	Mean	Std Dev	Min	Max	Skewness	Kurtosis
INFY	2,466	640.29	200.01	235.66	1,012.90	-0.185	-0.896
HCLTECH	2,468	906.60	458.78	354.38	1,995.10	0.505	-1.033
TECHM	2,465	963.47	404.34	376.30	1,806.10	0.316	-1.161
WIPRO	2,465	181.13	74.08	81.18	360.75	0.376	-1.133
TCS	2,460	2,684.43	955.94	1,050.58	4,553.75	-0.132	-1.211

3.2 Feature Engineering

Three feature sets were constructed for model training, increasing in complexity:

(i) **Univariate set:** Close price only. This configuration tests the hypothesis that temporal dependencies in the closing price sequence alone are sufficient for prediction.

(ii) **Price features set:** Close price augmented with SMA-10, SMA-50, EMA-10, EMA-50, one-day and five-day lagged close prices, and 10-day and 30-day rolling standard deviation. All indicators derived exclusively from Close and available for all five stocks including TCS.

(iii) **Full features set:** All price features plus RSI-14, MACD, MACD Signal, Bollinger Bands (upper and lower), and ATR-14. This set requires Open, High, Low, and Volume data and was not applicable to TCS, for which the Price Features set served as the expanded configuration.

3.3 Data Preprocessing and Sequence Construction

Each stock’s data was split chronologically: 80% for training, 10% for validation, and 10% for testing, corresponding to approximately 1,928–1,935 training, 241–242 validation, and 242 test observations per stock. A MinMaxScaler fitted exclusively on the training partition normalized all features to $[0, 1]$; the fitted scaler was applied to validation and test data to prevent information leakage. Sequences of length 60 trading days were constructed as model inputs, with the subsequent day’s close price as the target. All error metrics were computed on inverse-transformed original-scale predictions.

3.4 LSTM Architecture

The base LSTM model consists of six layers: a first LSTM layer with 128 units returning full sequences, followed by 20% dropout; a second LSTM layer with 64 units, followed by 20% dropout; a Dense layer with 32 ReLU-activated units; and a final linear Dense layer producing a single scalar output (118,081 trainable parameters). The Adam optimizer with learning rate 0.001 and mean squared error loss was used throughout. Training was bounded by 100 epochs with early stopping (patience = 15) monitoring validation loss, restoring best weights; a ReduceLROnPlateau scheduler halved the learning rate when validation loss plateaued for seven consecutive epochs.

Fig. 2 shows the epoch-by-epoch loss trajectory for the INFY Univariate model. Validation loss moves in step with training loss without diverging, a reassuring sign that the dropout layers and early-stopping criterion kept overfitting in check.

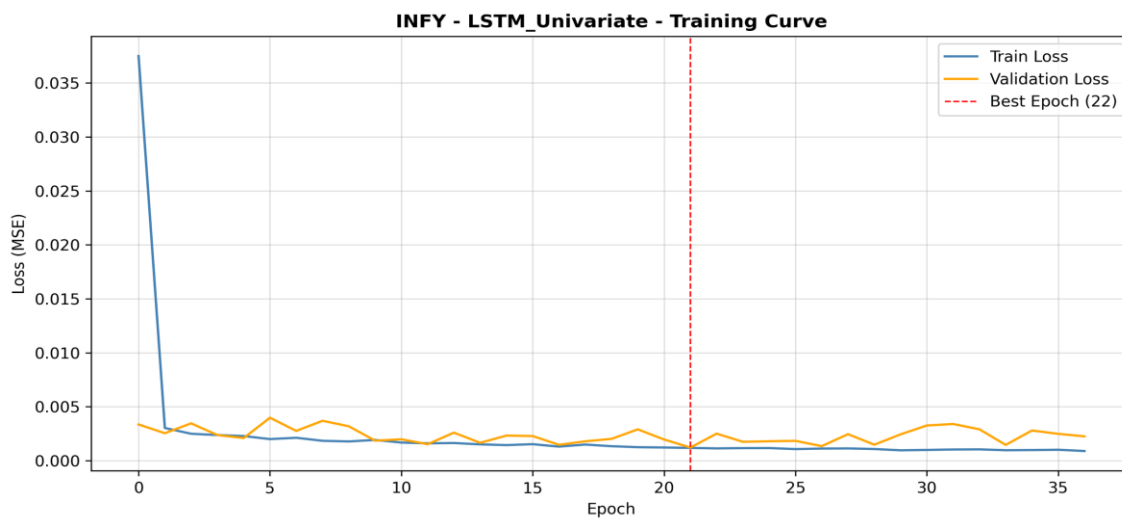


Fig. 2. Training and validation loss curves for INFY – LSTM Univariate model. Early stopping triggered at epoch 22; best weights restored from that checkpoint.

3.5 Evaluation Metrics

Model performance was assessed using seven metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Coefficient of Determination (R^2), Directional Accuracy (DA), Maximum Absolute Error, and Theil’s U statistic. Residual diagnostics were performed via the Ljung-Box test (lags 10 and 20), Jarque-Bera normality test, and ARCH-LM test (lag 5).

4. Experimental Setup

4.1 Hardware and Software Environment

The computational stack comprised Python 3.11 with TensorFlow 2.15 and Keras for model building, pandas and NumPy for data handling, scikit-learn for scaling and metric computation, the ta library for technical indicators, and statsmodels together with arch for residual diagnostics. All training runs used a local workstation (Intel Core i7, 16 GB RAM) with no GPU—a deliberate choice to demonstrate that the proposed workflow is reproducible on commodity hardware common in academic research settings.

4.2 Model Configurations

Three LSTM configurations—LSTM_Univariate, LSTM_Price_Features, and LSTM_Full_Features—were trained for each stock, yielding fifteen models in total (TCS received LSTM_Price_Features as its highest feature tier). Training times per model ranged from approximately 260 to 440 seconds.

4.3 ARIMA Benchmark

ARIMA benchmark metrics were obtained from a parallel analysis conducted by the authors on the same dataset and identical test period. That analysis applied a 13-step pipeline covering ADF stationarity testing, AIC-guided grid search over ARIMA(p, d, q) orders, residual diagnostics, and structural break testing using the Zivot-Andrews and Bai-Perron procedures. The ARIMA models consistently selected differencing orders of $d = 1$ or $d = 2$ across all five stocks, with autoregressive and moving average orders ranging from (1,1,1) to (3,1,2). ARIMA RMSE, MAE, MAPE, and R^2 values from that analysis are used here for direct benchmark comparison.

4.4 Reproducibility

Random seeds were fixed at 42 for all stochastic operations in NumPy, Python random, and TensorFlow. LSTM models were saved in Keras .keras format following training. The MinMaxScaler instances were fitted only on training data and stored for consistent inverse transformation during evaluation.

5. Results and Discussion

5.1 LSTM Model Performance by Feature Configuration

Table 3 reports the full performance matrix across all fifteen LSTM model-stock combinations. WIPRO exhibited the lowest absolute RMSE values (5.81–6.80 INR), consistent with its lower price range (mean INR 181.13). TCS produced the largest absolute errors (RMSE 85.36–86.66 INR) owing to its high price level (mean INR 2,684.43), though proportional accuracy (MAPE 2.31–2.39%) was comparable to other stocks. The influence of feature configuration on performance was stock-dependent rather than monotonic. For INFY, adding features hurt rather than helped: the univariate RMSE of 26.32 INR was noticeably lower than the 32.88 and 40.57 INR produced by the richer configurations, suggesting the extra predictors introduced noise that the model struggled to filter out. In contrast, WIPRO and TECHM demonstrated the expected monotonic improvement with richer feature sets.

Fig. 3 plots the actual and predicted close price for INFY over the test window under the best-performing Univariate LSTM. The two series move in close alignment throughout, with only modest divergences in stretches where prices shifted direction quickly—a common challenge for any point-forecast model.

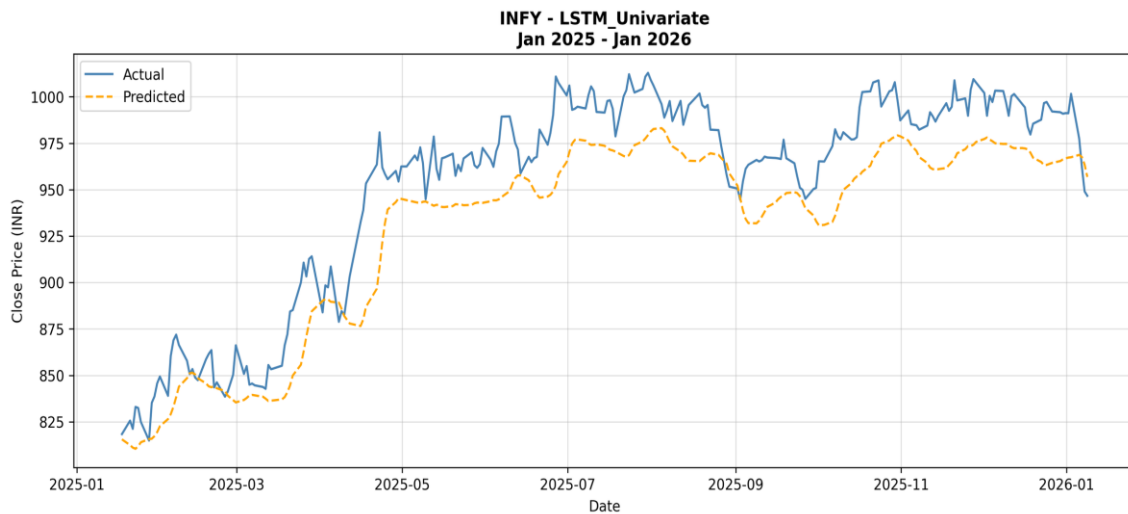


Fig. 3. Actual vs. predicted close price for INFY – LSTM Univariate model (test period). Blue: actual; orange dashed: predicted.

The TCS forecast under the Price Features LSTM is shown in Fig. 4. TCS trades at the highest absolute price levels among the five stocks, yet the model follows the underlying trend faithfully and delivers an R^2 of 0.941—the strongest individual result in the study.

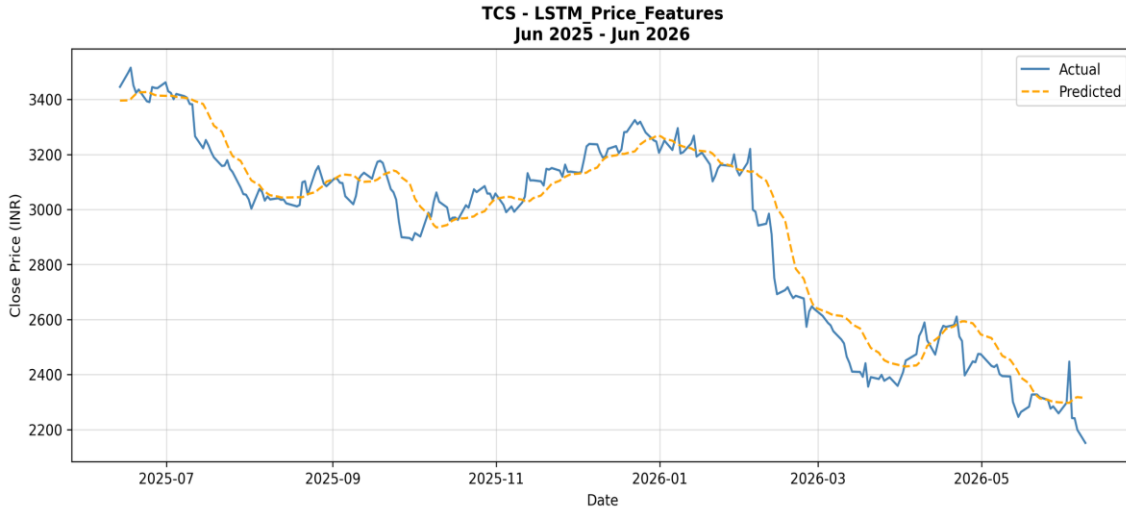


Fig. 4. Actual vs. predicted close price for TCS – LSTM Price Features model (test period). Blue: actual; orange dashed: predicted.

Table 2: Best LSTM Hyperparameters by Stock

Stock	Units L1	Units L2	Dropout	Learn Rate	Batch Size	Best Epoch	Val Loss
INFY	128	64	0.10	0.001	32	22	0.00117
HCLTECH	64	64	0.10	0.001	32	27	0.00090
TECHM	128	64	0.20	0.001	32	14	0.00051
WIPRO	64	64	0.10	0.001	32	21	0.00052
TCS	64	64	0.10	0.001	64	45	0.00134

Table 3: LSTM Model Performance — All Configurations (Test Set, INR)

Stock	Model	RMSE	MAE	MAPE %	R^2	DA %	Max Err	Theil U
INFY	Univariate	26.32	23.05	2.39	0.773	47.30	72.61	0.02758
	Price Features	32.88	29.83	3.10	0.645	47.30	78.63	0.03444
	Full Features	40.57	38.35	3.97	0.460	51.04	74.83	0.04250

HCLTECH	Univariate	37.11	27.28	1.74	0.870	49.79	150.46	0.02336
	Price Features	45.89	33.68	2.16	0.801	46.47	169.71	0.02889
	Full Features	43.03	32.54	2.07	0.825	50.21	162.05	0.02709
TECHM	Univariate	45.11	35.25	2.35	0.793	51.87	161.19	0.02938
	Price Features	42.55	32.92	2.19	0.816	50.21	151.26	0.02771
	Full Features	38.47	28.80	1.92	0.849	50.21	138.53	0.02505
WIPRO	Univariate	6.20	4.57	1.75	0.912	53.94	23.24	0.02377
	Price Features	6.80	4.95	1.90	0.894	51.45	23.98	0.02607
	Full Features	5.81	4.27	1.62	0.923	47.72	20.47	0.02227
TCS	Univariate	86.66	66.58	2.39	0.939	52.70	315.08	0.02935
	Price Features	85.36	64.46	2.31	0.941	52.28	312.25	0.02891

Bold values indicate best-performing model per stock. DA = Directional Accuracy. TCS Full Features equivalent to Price Features due to data availability.

5.2 Best LSTM Model Performance

Table 4 summarizes the best LSTM model per stock by minimum RMSE. R² values for winning models ranged from 0.773 (INFY) to 0.941 (TCS), indicating that LSTM captured a substantial proportion of close price variance across all five stocks. WIPRO and TCS achieved particularly high R² (0.923 and 0.941), reflecting more persistent trend behavior in their price series during the test window.

Table 4: Best LSTM Model Performance Summary

Stock	Best Model	RMSE	MAE	MAPE %	R ²	DA %	Max Error	Theil U
INFY	Univariate	26.32	23.05	2.39	0.773	47.30	72.61	0.02758
HCLTECH	Univariate	37.11	27.28	1.74	0.870	49.79	150.46	0.02336
TECHM	Full Features	38.47	28.80	1.92	0.849	50.21	138.53	0.02505

WIPRO	Full Features	5.81	4.27	1.62	0.923	47.72	20.47	0.02227
TCS	Price Features	85.36	64.46	2.31	0.941	52.28	312.25	0.02891

5.3 ARIMA vs. LSTM Benchmark Comparison

Table 5 presents the paired ARIMA vs. LSTM comparison across all five stocks. LSTM outperforms ARIMA on every stock and every metric. ARIMA RMSE values ranged from 41.96 INR (WIPRO) to 732.68 INR (TCS), while the corresponding LSTM best-model RMSE values were 5.81 and 85.36 INR—improvements of 86.15% and 88.35% respectively. All five ARIMA models yielded negative R^2 values on the test period, implying worse performance than a simple mean prediction. LSTM MAPE values remained below 2.40% on all stocks, compared to ARIMA MAPE values of 9.56%–20.58%.

Figs. 5 and 6 present the ARIMA vs. LSTM RMSE and MAPE comparisons visually. The magnitude of improvement is unambiguous across all five stocks, with LSTM bars uniformly a fraction of their ARIMA counterparts.

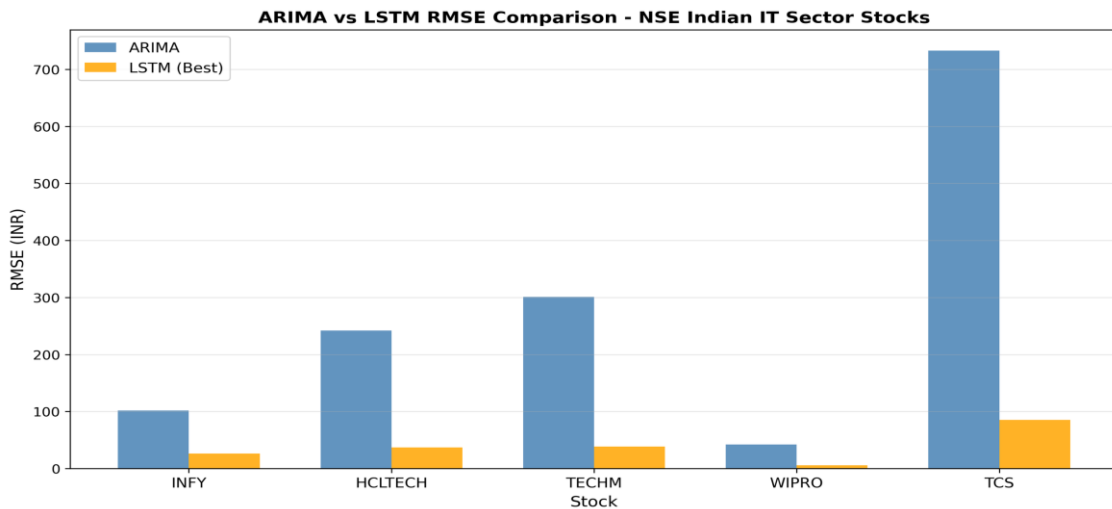


Fig. 5. ARIMA vs. LSTM RMSE comparison across five NSE Indian IT sector stocks (test period). Lower is better.

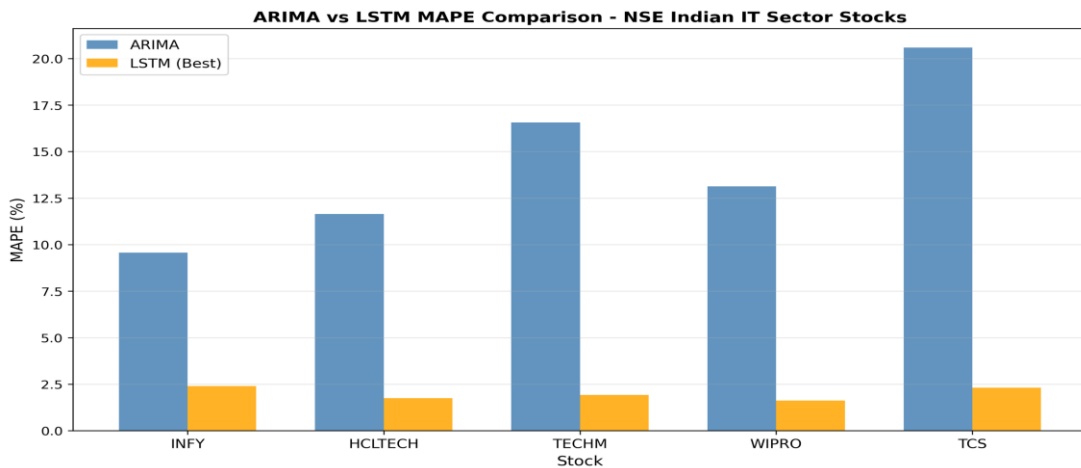


Fig. 6. ARIMA vs. LSTM MAPE (%) comparison across five NSE Indian IT sector stocks (test period). Lower is better.

Table 5: ARIMA vs. LSTM Benchmark Comparison

Stock	ARIMA RMSE	LSTM RMSE	ARIMA MAE	LSTM MAE	ARIMA MAPE%	LSTM MAPE%	RMSE Impr.%	Winner
INFY	101.62	26.32	85.39	23.05	9.56	2.39	74.10%	LSTM
HCLTECH	241.91	37.11	198.92	27.28	11.64	1.74	84.66%	LSTM
TECHM	300.93	38.47	262.03	28.80	16.56	1.92	87.22%	LSTM
WIPRO	41.96	5.81	35.70	4.27	13.13	1.62	86.15%	LSTM
TCS	732.68	85.36	622.01	64.46	20.58	2.31	88.35%	LSTM

5.4 Residual Diagnostics

Table 6 presents residual diagnostic statistics for the best LSTM model per stock. The Ljung-Box test (lags 10 and 20) produced p-values below 0.001 across all five stocks, rejecting the null hypothesis of no autocorrelation. The ARCH-LM test returned $p < 0.001$ on all five series, establishing that conditional variance in the residuals still clusters over time. The Jarque-Bera test likewise rejected normality on every stock, pointing to fatter tails than a Gaussian error assumption would produce. In combination, these results confirm that while LSTM absorbs most of the trend signal and dramatically narrows the gap relative to ARIMA, it leaves conditional heteroscedasticity largely unaddressed. A composite framework pairing LSTM with a GARCH or EGARCH volatility module represents the logical next step.

Fig. 7 displays the ACF and PACF of INFY LSTM residuals. The spikes at multiple lags extend well beyond the confidence bands, visually confirming the structured dependence that the Ljung-Box test detected numerically.

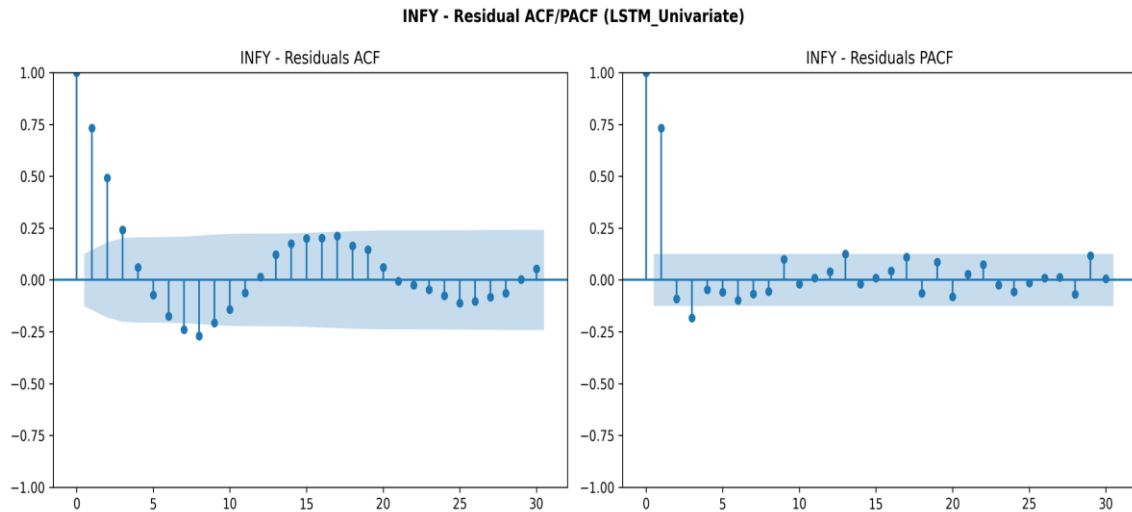


Fig. 7. ACF and PACF of LSTM forecast residuals for INFY – Univariate model (test period). Significant spikes confirm residual autocorrelation.

Table 6: Residual Diagnostics for Best LSTM Model per Stock

Stock	Model	Mean Residual	Std Residual	L-B p (lag 10)	L-B p (lag 20)	JB p-val	ARCH-LM p
INFY	Univariate	22.46	13.73	< 0.001	< 0.001	0.0013	< 0.001
HCLTECH	Univariate	2.27	37.04	< 0.001	< 0.001	< 0.001	< 0.001
TECHM	Full Features	0.16	38.47	< 0.001	< 0.001	0.0011	< 0.001
WIPRO	Full Features	-0.01	5.81	< 0.001	< 0.001	0.0005	< 0.001
TCS	Price Features	-27.84	80.69	< 0.001	< 0.001	0.0014	< 0.001

L-B = Ljung-Box autocorrelation test. JB = Jarque-Bera normality test. ARCH-LM = Engle ARCH Lagrange Multiplier test. All residuals in INR on original price scale.

5.5 Discussion

The results provide strong empirical support for the hypothesis that deep learning outperforms classical ARIMA models in emerging market stock forecasting. LSTM’s advantages are most pronounced for high-price-range, trend-dominant stocks (TCS, TECHM) where ARIMA’s stationarity assumptions fail most severely in the extended post-COVID recovery period. For WIPRO, which exhibited the most price-stable behavior relative to its absolute level, LSTM still outperformed ARIMA by 86.15% in RMSE but did so with the smallest absolute error values.

The cases where simpler inputs won—most clearly for INFY and HCLTECH—carry a practical warning. When many correlated technical indicators are fed into a 60-day rolling window, collinearity and noise can offset whatever informational content the features hold, leaving the model worse off than if it had used close price alone. Attention-based or adaptive feature-selection layers embedded within the LSTM may resolve this; systematic evidence on which technical indicators add value and which introduce noise is provided by Htun et al. [20] in their broad survey of feature selection techniques for stock market prediction.

The persistent autocorrelation and ARCH effects in LSTM residuals indicate that LSTM, while vastly superior to ARIMA for point-estimate accuracy, does not fully model the conditional data generating process. Volatility clustering remains in the residuals, motivating hybrid architectures that combine LSTM’s trend-capturing capability with explicit volatility regime modeling—a direction the authors are pursuing in subsequent work through a volatility-regime-aware adaptive fusion framework incorporating SHAP explainability at the individual stock level.

Finally, the near-chance directional accuracy across models confirms that the market for large-cap NSE IT stocks is informationally semi-efficient at the daily frequency [21]. LSTM provides strong RMSE improvements by tracking price levels accurately, but level accuracy and directional accuracy are not equivalent; the latter requires reliable inference about return sign, which remains an open problem in financial machine learning [9].

6. Conclusion

This study examined whether deep learning, specifically a stacked LSTM architecture, outperforms classical ARIMA forecasting for NSE-listed Indian IT sector stocks over a ten-year period spanning January 2016 to June 2026. Using a standardized 80/10/10 chronological train-validation-test split and evaluating three feature configurations per stock, we compared LSTM against pre-computed ARIMA benchmarks on identical datasets.

The results unambiguously favor LSTM: it achieved RMSE improvements of 74–88% over ARIMA across all five stocks, with R^2 values ranging from 0.77 to 0.94 compared to uniformly negative R^2 values from ARIMA. The performance advantage was robust across low-price (WIPRO), mid-price (HCLTECH, TECHM, INFY), and high-price (TCS) stocks. MAPE values for the best LSTM configurations remained below 2.40% on all stocks, compared to ARIMA MAPE values ranging from 9.56% to 20.58%.

At the same time, residual diagnostics revealed persistent autocorrelation, volatility clustering, and non-Gaussian error distributions across all LSTM models, indicating that the deep learning architecture does not fully capture the conditional second-order structure of financial returns. These findings directly motivate a subsequent research phase in which an LSTM-based adaptive hybrid is integrated with an explicit volatility model, with SHAP-based explainability providing interpretable attribution of prediction components.

Analyzing individual securities rather than sector indices was a conscious methodological decision, not an incidental one. Each of the five IT stocks carries its own volatility profile, trend dynamics, and information environment; pooling them into an index suppresses exactly the variation that makes stock selection consequential. By establishing where LSTM holds advantages and identifying the conditions under which parsimonious feature sets outperform richer ones, this study gives portfolio managers and quantitative analysts in the Indian equity space concrete, stock-specific guidance they can act on.

Several constraints bound the present results. Macroeconomic covariates, news sentiment scores, and order-book microstructure data are absent from the input set. The test window also falls entirely in the post-COVID era, limiting inferences about pre/post-break performance differences. Planned extensions include incorporating earnings event windows, broadening the analysis across sectors, and applying the framework to the Bank Nifty index to test whether conclusions transfer beyond the IT segment.

References

1. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ, USA: Wiley, 2015.
2. R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts, 2021.
3. B. N. Ashraf, "Stock markets' reaction to COVID-19: Cases or fatalities?" *Research in International Business and Finance*, vol. 54, p. 101249, Dec. 2020. doi: 10.1016/j.ribaf.2020.101249.
4. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. doi: 10.1162/neco.1997.9.8.1735.
5. W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLOS ONE*, vol. 12, no. 7, p. e0180944, Jul. 2017. doi: 10.1371/journal.pone.0180944.
6. M. Kumar and M. Thenmozhi, "Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models," *International Journal of Banking, Accounting and Finance*, vol. 5, no. 3, pp. 284–316, 2014. doi: 10.1504/IJBAAF.2014.064605.
7. A. A. Adebisi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and artificial neural networks models for stock price prediction," *Journal of Applied Mathematics*, vol. 2014, p. 614342, 2014. doi: 10.1155/2014/614342.
8. H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. C. Khatri, "Predicting stock market index using LSTM," *Machine Learning with Applications*, vol. 9, p. 100320, Sep. 2022. doi: 10.1016/j.mlwa.2022.100320.
9. T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, Oct. 2018. doi: 10.1016/j.ejor.2017.11.054.
10. S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Orlando, FL, USA, 2018, pp. 1394–1401. doi: 10.1109/ICMLA.2018.00227.
11. S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proc. Int. Conf. Advances in Computing, Comm. Inform. (ICACCI)*, Udupi, India, 2017, pp. 1643–1647. doi: 10.1109/ICACCI.2017.8126078.
12. K. Pawar, R. S. Jalem, and V. Tiwari, "Stock market price prediction using LSTM RNN," in *Emerging Trends in Expert Applications and Security*, V. E. Balas, N. Sharma, and A. Chakrabarti, Eds. Singapore: Springer, 2019, pp. 493–503. doi: 10.1007/978-981-13-2285-3_58.
13. S. Mehtab, J. Sen, and A. Dutta, "Stock price prediction using machine learning and LSTM-based deep learning models," in *Machine Learning and Metaheuristics Algorithms, and Applications*, S. Thampi et al., Eds. Singapore: Springer, 2021, pp. 88–106. doi: 10.1007/978-981-16-0419-5_8.
14. G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003. doi: 10.1016/S0925-2312(01)00702-0.
15. S. Chen and G. He, "Stock prediction using convolutional neural network," *IOP Conf. Series: Materials Science and Engineering*, vol. 435, p. 012026, 2018. doi: 10.1088/1757-899X/435/1/012026.

16. A. Jain and S. Tripathi, "Forecasting and trading in financial markets using machine learning," in *Advances in Intelligent Systems and Computing*. Singapore: Springer, 2021, pp. 289–299. doi: 10.1007/978-981-15-7041-3_26.
17. H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, Aug. 2018. doi: 10.1016/j.eswa.2018.03.002.
18. B. N. Ashraf, "Economic impact of government interventions during the COVID-19 pandemic: International evidence from financial markets," *Journal of Behavioral and Experimental Finance*, vol. 27, p. 100371, Sep. 2020. doi: 10.1016/j.jbef.2020.100371.
19. S. Lyócsa, P. Molnár, T. Plíhal, and M. Širaňová, "Impact of macroeconomic news, regulation and hacking exchange markets on the volatility of bitcoin," *Journal of Economic Dynamics and Control*, vol. 119, p. 103980, Oct. 2020. doi: 10.1016/j.jedc.2020.103980.
20. H. H. Htun, M. Biehl, and N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction," *Financial Innovation*, vol. 9, no. 1, p. 26, Jan. 2023. doi: 10.1186/s40854-022-00441-7.
21. E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *Journal of Finance*, vol. 25, no. 2, pp. 383–417, May 1970. doi: 10.2307/2325486.