ISSN : 2455-135X \ © 2025 International Scientific and Academic Research (ISAR) Publisher

Comparative Analysis of Deep Learning and Econometric Models for Per-Ticker Stock Price Forecasting in NSE Large and Small Cap Equities

Ceena Mathews[1]

[1]Department of Computer Science, Prajyoti Niketan College, Kerala, India

[1]ceenamathews@prajyotiniketan.edu.in

Received: Revised: Accepted: Published:

Abstract *- Stock price forecasting is a challenging task due to the non-stationarity, volatility, and nonlinearity of financial time series. Hence, it is necessary to evaluate the performance of stock price forecasting under varying market conditions using both conventional econometric and deep learning models. In this study, a comparative analysis of ARIMA, GARCH, LSTM, transformer encoder, TCN, and neural ensemble models is conducted for next-day stock price forecasting of NSE equities. Experiments are conducted on both large-cap and small-cap stocks using daily OHLCV data from 2015 to 2025. Per-ticker models are trained on log-return sequences augmented with rolling volatility features and evaluated using performance metrics including MAE, RMSE, and MAPE. The results indicate that ARIMA and GARCH models achieve higher forecasting accuracy for relatively stable large-cap stocks. In contrast, deep learning models significantly outperform classical methods in highly volatile small-cap equities. These findings suggest that forecasting performance is strongly influenced by market segment characteristics, highlighting the importance of volatility-aware model selection in equity price prediction.*

Keywords *- deep learning, econometric models, NSE, per-ticker, stock price forecasting*

1. Introduction

Equity returns are characterised by non-stationarity, volatility clustering, heavy tails, and structural breaks, making stock price forecasting a long-standing and practically important problem. Conventional statistical forecasting methods often struggle to capture the nonlinear patterns and abrupt regime shifts inherent in financial markets [1]. In contrast, deep learning techniques are capable of modelling complex nonlinear and temporal dependencies within financial time series, thereby improving the prediction of future stock movements from historical data [2].

According to the Securities and Exchange Board of India (SEBI), stocks are categorised into large-cap, mid-cap, and small-cap segments based on market capitalisation rankings. Large-cap stocks, comprising the top 100 companies, generally exhibit stable price dynamics and lower volatility, whereas small-cap stocks, ranked 251 and beyond, are more sensitive to market movements and display higher volatility. Existing studies typically focus on either large-cap or small-cap equities in isolation, with limited attention given to evaluating both categories within a unified framework. This study addresses this gap by providing a joint evaluation of econometric models and deep learning approaches across both market segments.

In many prior studies, forecasting models are trained on multiple tickers because stocks exhibit common temporal dynamics and can benefit from shared representation learning [3], [4]. However, Gu et al. and Tsantekidis et al. suggest that financial time series exhibit strong cross-sectional heterogeneity, and that per-ticker modelling can more effectively capture firm-specific behaviours and distinct volatility regimes [5], [6]. Accordingly, a per-ticker modelling strategy is adopted in this work, where independent models are trained for each stock using only its historical data.

All models are evaluated on the task of next-day closing price prediction for selected National Stock Exchange (NSE) large-cap and small-cap equities.

2. Related Work

Financial time-series forecasting has received sustained research attention due to the complex statistical characteristics of asset prices, including non-stationarity, volatility clustering, and regime shifts. Sezer et al suggest a gradual transition from econometric approaches toward deep learning–based forecasting frameworks, particularly for modelling nonlinear and high-variance financial data [7].

ISSN : 2455-135X \ © 2025 International Scientific and Academic Research (ISAR) Publisher

Deep recurrent architectures, most notably Long Short-Term Memory (LSTM) networks, have been extensively applied to stock price forecasting. Empirical results indicate that LSTM-based models can outperform linear statistical methods under volatile market conditions by capturing nonlinear temporal dependencies [8], [9]. However, recurrent architectures rely on sequential computation, which limits scalability and computational efficiency when applied to long time-series sequences.

To mitigate these limitations, attention-based transformer architectures have been introduced for time-series forecasting. Models such as the temporal fusion transformer enable interpretable multi-horizon forecasts and have demonstrated strong predictive performance across diverse application domains [10]. More recent transformer variants, including Autoformer and FEDformer, further improve forecasting accuracy through explicit time-series decomposition and frequency-domain representations [11], [12].

Temporal convolutional networks (TCN) have also emerged as a competitive alternative to recurrent and attention-based architectures. By employing dilated causal convolutions and large receptive fields, TCNs achieve stable training and strong predictive performance for sequential data [13], [14]. These characteristics make TCNs particularly suitable for financial time series that exhibit long-range temporal dependencies.

Another important research direction concerns modelling strategies across multiple assets. Global forecasting models trained jointly across many stocks assume the presence of shared temporal dynamics and common latent structures. However, Sirignano et al suggests that strong cross-sectional heterogeneity and asset-specific volatility regimes can reduce the effectiveness of such universal models [15]. As a result, per-ticker modelling approaches have gained increasing attention for their ability to capture firm-specific price dynamics more accurately.

Despite these advances, limited work has focused on a unified comparison of econometric methods and deep learning models across both large-cap and small-cap equities. Furthermore, the relationship between market-cap–driven volatility regimes and forecasting model performance remains insufficiently explored. This study addresses these gaps by conducting a systematic per-ticker evaluation of econometric and deep learning approaches on NSE equities.

3. Materials and Methods

3.1. Dataset

Equities listed on the NSE of India are categorised into large-cap and small-cap segments based on market capitalisation norms defined by SEBI. Large-cap stocks correspond to the top 100 listed companies and are generally characterised by high liquidity, lower volatility, and relatively stable price dynamics. In contrast, small-cap stocks comprise companies ranked 251 and below, which typically exhibit higher volatility, irregular price movements, and lower levels of institutional participation.

In this study, both large-cap and small-cap equities are analysed using daily historical market data. The dataset consists of OHLCV (Open, High, Low, Close, and Volume) features covering the period from January 1, 2015, to November 28, 2025. The data are sourced from Yahoo Finance using the yfinance Python package.

Two non-overlapping sets of tickers are selected to represent the respective market-cap segments:

- Large-cap equities: RELIANCE, HDFCBANK, ICICIBANK, INFY, KOTAKBANK, TCS, HINDUNILVR, SBIN, AXISBANK, ASIANPAINT.
- Small-cap equities: BBTC, APOLLOTYRE, CROMPTON, DALBHARAT, RBLBANK, PNBHOUSING, MCX, INDIACEM, BEML.

To account for cross-sectional heterogeneity in price behaviour, each stock is modelled independently using only its own historical data. This per-ticker approach avoids the homogeneity assumptions inherent in global multi-asset forecasting models and enables a more accurate representation of asset-specific dynamics.

3.2 Feature Engineering

Financial time series are inherently non-stationary due to trends, volatility clustering, and structural breaks. Since most forecasting models benefit from approximately stationary inputs, return-based transformations of price data are employed.

For each ticker, daily simple returns r_t and logarithmic returns ℓ_t are computed as in Equation 1.

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}}, \qquad \ell_t = \log\left(\frac{P_t}{P_{t-1}}\right) \tag{1}$$

where P_t denotes the closing price on day t.

To capture local market uncertainty, a rolling volatility estimate σ_t is computed from log-returns using a sliding window of $W = 60$ trading days, as in Equation 2.

$$\sigma_t = \sqrt{\frac{1}{W} \sum_{i=t-W+1}^{t} (\ell_i - \bar{\ell})^2} \tag{2}$$

where $\bar{l}$ denotes the mean log-return within the window.

For each trading day, the input features consist of the standard OHLCV variables augmented with the rolling volatility measure. Fixed-length temporal sequences of $L = 60$ trading days are constructed using a rolling-window approach, with the model trained to predict the next-day log-return.

To ensure consistent evaluation across different model families, predicted log-returns are transformed back into price space using Equation 3.

$$\hat{P}_{t+1} = P_t \exp\left(\hat{\ell}_{t+1}\right) \quad (3)$$

where $\hat{l}_{t+1}$ denotes the model-predicted log-return.

All numeric input features are standardised independently for each ticker using z-score normalisation, ensuring zero mean and unit variance while preserving stock-specific distributions. The dataset is partitioned chronologically, with 80% of observations used for training and validation, and the remaining 20% reserved for out-of-sample testing.

3.3. Model Architectures

This section summarises the neural and statistical forecasting models employed for each ticker. All neural architectures are optimised using the Adam optimiser with mean squared error (MSE) loss.

3.3.1 LSTM

These networks are recurrent architectures designed to capture long-range temporal dependencies through gated memory mechanisms. By regulating information flow via input, forget, and output gates, they mitigate the vanishing gradient problem commonly observed in standard recurrent neural networks (RNN).

In this study, the model consists of two stacked LSTM layers followed by a fully connected regression head that predicts the next-day log-return. This architecture enables modelling of non-linear temporal interactions and evolving dependencies in financial time series.

3.3.2 Transformer Encoder Model

To leverage self-attention mechanisms, a transformer encoder architecture with sinusoidal positional encoding is employed. Since transformers do not incorporate recurrence or convolution, temporal order is explicitly injected through positional encoding.

The encoder block consists of:

- Linear projection of input features to a latent space of dimension d_{model},
- Additive sinusoidal positional encoding,
- Multi-Head Self-Attention (MHSA),
- Residual connections with Layer Normalisation,
- Position-wise feed-forward network (FFN),
- Global average pooling and a dense regression head.

Given a multivariate input sequence, the features are first projected into a latent space:

$$H = XW_p + b_p \quad (4)$$

where H is the embedded representation used by the transformer, X is the input feature matrix, W_p is the learnable projection matrix, and b_p is the bias vector.

$$Z_0 = H + E \quad (5)$$

where E is the sinusoidal positional encoding matrix, Z_0 is the position-aware input to the transformer, obtained by augmenting the projected features with sinusoidal positional encodings.

The encoder block applies MHSA and FFN, each wrapped with residual connections and layer normalisation:

$$Z_1 = \text{LayerNorm}\left(Z_0 + \text{MHSA}(Z_0)\right) \quad (6)$$

$$Z_2 = LayerNorm(Z_1 + FFN(Z_1)) \quad (7)$$

Global average pooling over the temporal dimension is applied before a dense regression head to predict the next day log-return.

3.3.3 Temporal Convolutional Network (TCN)

It is a fully convolutional sequence model based on dilated causal convolutions, ensuring that predictions depend only on past inputs while preserving sequence length. Increasing dilation rates exponentially expands the receptive field without increasing model depth.

In this work, the TCN architecture comprises stacked causal convolutional layers with increasing dilation factors, followed by pooling and dense layers for next-day return prediction.

3.3.4 Neural Ensemble

An ensemble predictor is constructed by averaging the outputs of the LSTM, Transformer, and TCN models. This ensemble exploits architectural diversity to reduce variance and improve forecasting robustness.

3.3.5 ARIMA

As a statistical baseline, an ARIMA(p,q) model is applied to the log-return series, where p denotes the order of the autoregressive component, q denotes the order of the moving-average component.

ISSN : 2455-135X \ © 2025 International Scientific and Academic Research (ISAR) Publisher

3.3.6 GARCH

To capture time-varying volatility, a GARCH (1,1) model is fitted to the log-return series. The conditional mean forecast is subsequently transformed into price predictions using the last observed closing price.

3.4. Evaluation Metrics

For all models, predicted log-returns are transformed back to the original price scale before evaluation. Forecasting performance is assessed on reconstructed prices to ensure consistency across different model families.

Three standard regression metrics are employed: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). MAE measures the average magnitude of prediction errors and is defined as in Equation 8.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \left|P_t - \widehat{P}_t\right| \quad (8)$$

where P_t and $\widehat{P}_t$ denote the actual and predicted closing prices, respectively, and N is the number of test observations.

RMSE penalises larger forecast errors more heavily due to the squaring operation, making it particularly sensitive to large deviations. MAPE expresses forecast errors as a percentage of the actual price, facilitating scale-independent comparison across different equities, although it may become unstable when prices approach zero.

4. Results and Discussion

All neural models are trained using a batch size of 32 and optimised with the Adam optimiser using mean squared error (MSE) on log-returns as the loss function. A lookback window of 60 trading days is employed for large-cap equities, while a longer window of 120 trading days is used for small-cap stocks to better capture extended temporal dependencies under higher volatility. Training is performed for up to 15 epochs, with early stopping based on validation loss and a patience of three epochs.

Per-ticker forecasting performance for four NSE large-cap stocks is shown in Table 1 using MAE, RMSE, and MAPE. Representative next-day closing price forecasts for Asian Paints and Axis Bank are shown in Figure 1.

Across the large-cap universe, classical time-series models consistently achieve the strongest performance. As shown in Table 1, ARIMA and GARCH yield the lowest MAE and MAPE values for most tickers, with percentage errors close to or below 1%. These results indicate that the relatively smooth price dynamics and high liquidity of large-cap equities are well captured by linear autoregressive structures, with GARCH further accounting for volatility clustering. Neural models exhibit mixed performance in this regime. While LSTM and TCN remain competitive on certain tickers, they do not consistently outperform the classical baselines. Transformer-based models, in particular, show higher error on several large-cap stocks, suggesting that the additional model complexity and self-attention mechanisms do not provide a clear advantage when price dynamics are predominantly linear.

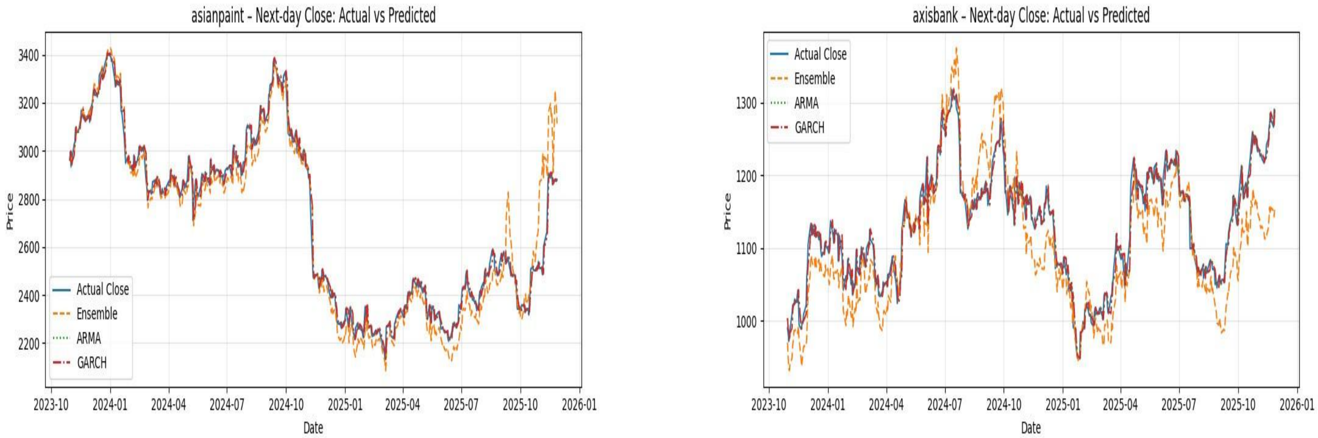
Table 1. Per-ticker performance metrics for all models on NSE large-cap stocks

Ticker	Model	MAE	RMSE	MAPE(%)
Asianpaint	ARIMA	24.5206	34.6218	0.9124
Asianpaint	GARCH	24.5552	34.6718	0.9139
Asianpaint	LSTM	70.1820	90.4438	2.7140
Asianpaint	TCN	84.2734	229.8145	3.1937
Asianpaint	Transformer	84.1426	102.1467	3.1978
Asianpaint	Ensemble	59.1964	86.4720	2.2585
Axisbank	ARIMA	11.7270	16.3864	1.0379
Axisbank	GARCH	11.7457	16.3905	1.0399
Axisbank	LSTM	24.1185	33.0313	2.0911
Axisbank	TCN	138.0417	194.6483	11.9905
Axisbank	Transformer	208.6480	226.5364	18.1841
Axisbank	Ensemble	40.1933	48.6515	3.5163
Reliance	ARIMA	13.1734	18.3588	0.9551
Reliance	GARCH	13.1691	18.3551	0.9547
Reliance	LSTM	48.8609	57.4507	3.4505
Reliance	TCN	13.8741	19.0809	1.0059
Reliance	Transformer	15.5560	20.4562	1.1327
Reliance	Ensemble	20.6737	25.8810	1.4830
TCS	ARIMA	35.2968	49.5354	0.9391
TCS	GARCH	35.3012	49.5374	0.9393
TCS	LSTM	41.7381	54.9598	1.1181
TCS	TCN	35.0629	49.5036	0.9321
TCS	Transformer	105.1018	153.4525	2.6623
TCS	Ensemble	53.5489	71.7759	1.3932

Average performance metrics across large-cap stocks, summarised in Table 2, reinforce these observations. ARIMA and GARCH achieve the lowest average MAE and MAPE, while neural models display higher average error. The ensemble model improves robustness in some cases but does not consistently outperform the best individual model, reflecting the sensitivity of simple averaging when constituent models differ in strength.

Per-ticker results for NSE small-cap stocks are reported in Table 3, with representative forecasts for Apollo Tyres and Crompton shown in Figure 2. In contrast to the large-cap regime, classical models exhibit substantially degraded performance on small-cap equities. ARIMA and GARCH frequently produce MAPE values in the range of 20–40%, highlighting their limited robustness under conditions of higher volatility, abrupt price movements, and non-linear behaviour.

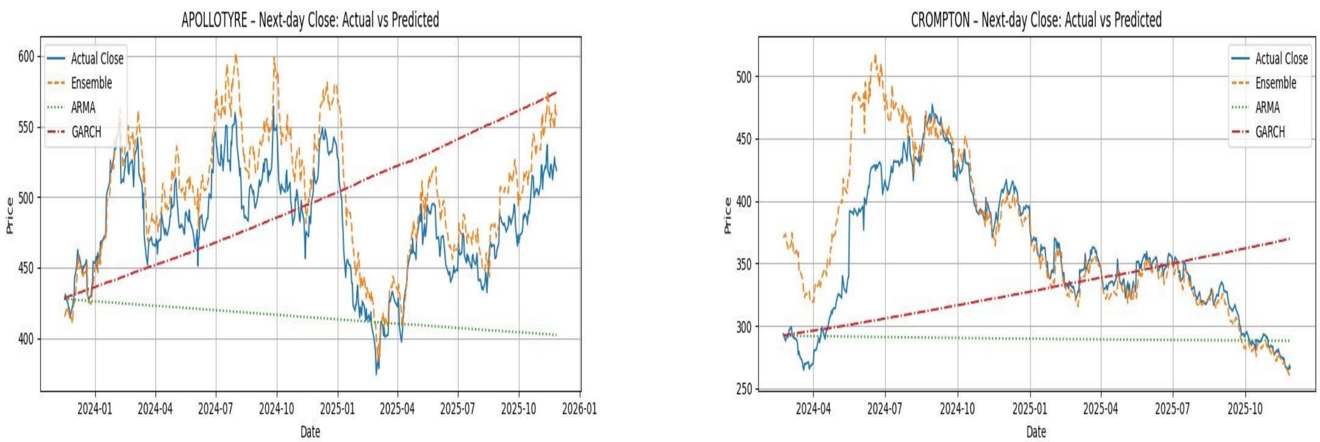
ISSN : 2455-135X \ © 2025 International Scientific and Academic Research (ISAR) Publisher



(a) Asian Paints (Large-cap)

(b) Axis Bank (Large-cap)

Fig. 1 Comparison of actual closing price vs model predictions for two large-cap NSE tickers.



(a) Apollo Tyres (Small-cap)

(b) Crompton (Small-cap)

Fig. 2 NSE small-cap tickers showing actual close price and model predictions.

Table 2. Average performance across tickers on NSE large-cap stocks

Model	MAE	RMSE	MAPE(%)
ARIMA	16.7632	23.8145	0.9483
GARCH	16.7743	23.8227	0.9490
LSTM	32.2789	41.2428	1.9457
TCN	41.4288	67.4440	2.8210
Ensemble	35.5525	46.5401	2.2434

Neural models demonstrate a clear advantage in this setting. LSTM-based models consistently achieve low MAE and MAPE across most small-cap tickers, while the ensemble model further stabilises predictions by reducing model-specific variance. Transformer models perform well on selected small-cap stocks but exhibit higher variability across the set. TCN models effectively capture local temporal structure in some cases, but are less consistent when performance is averaged across tickers.

Average results in Table 4 highlight the magnitude of this contrast. While classical models maintain average MAPE values exceeding 20%, LSTM and ensemble models reduce MAPE to approximately 4–5%, representing a substantial improvement in forecasting accuracy under high-volatility conditions.

To assess whether the observed performance differences are statistically meaningful, paired comparisons of absolute forecast errors were conducted between classical and neural models across the test period for each ticker. In the small-cap regime, the reduction in MAE achieved by LSTM and ensemble models relative to ARIMA and GARCH is statistically significant at the 5% level under a paired t-test, confirming that the observed improvements are unlikely to be due to random variation. In contrast, for large-cap stocks, differences between ARIMA, GARCH, and neural models are

ISSN : 2455-135X \ © 2025 International Scientific and Academic Research (ISAR) Publisher

generally not statistically significant, consistent with the comparable performance observed in Table 2.

Table 3. Per-ticker metrics for NSE small-cap stocks using ARIMA, GARCH, LSTM, TCN, transformer, ensemble models.

Ticker	Model	MAE	RMSE	MAPE(%)
Apollotyre	ARIMA	67.4165	76.3999	13.5160
Apollotyre	GARCH	53.9641	63.3071	11.4726
Apollotyre	LSTM	40.1294	45.3031	8.1531
Apollotyre	TCN	195.7442	196.8852	40.7255
Apollotyre	Transformer	219.9889	221.5429	45.7462
Apollotyre	Ensemble	22.8359	26.4095	4.6908
BBTC	ARIMA	746.2314	849.5069	35.7022
BBTC	GARCH	363.3505	503.0229	16.3377
BBTC	LSTM	49.5902	73.9274	2.4184
BBTC	TCN	307.4474	352.9987	15.2092
BBTC	Transformer	110.3648	167.0038	5.4045
BBTC	Ensemble	138.8458	165.3116	6.8699
BEML	ARIMA	813.5648	883.8819	41.5792
BEML	GARCH	554.3736	622.8759	28.0195
BEML	LSTM	203.4605	254.8405	10.5831
BEML	TCN	630.3984	650.8712	33.5077
BEML	Transformer	615.0485	676.6495	32.0415
BEML	Ensemble	121.4547	149.9394	6.6067
Crompton	ARIMA	70.8149	86.6413	18.1864
Crompton	GARCH	55.6730	72.5806	14.7553
Crompton	LSTM	6.1236	8.3168	1.6975
Crompton	TCN	56.2530	101.2123	16.1537
Crompton	Transformer	15.1053	17.2909	4.3489
Crompton	Ensemble	21.0222	33.9006	6.0216

Table 4: Average metrics across NSE small-cap stocks

Model	MAE	RMSE	MAPE (%)
ARIMA	615.4007	723.1573	29.1342
GARCH	435.9988	532.9442	20.4704
LSTM	94.4402	113.1673	4.5818
TCN	584.8908	647.5277	20.3281
Transformer	411.5326	568.2528	15.2354
Ensemble	100.1237	119.8205	4.7140

Overall, the results demonstrate that no single modelling approach dominates across all market regimes. Classical econometric models remain highly effective for stable, liquid large-cap equities, where linear temporal dependencies dominate. Neural models, particularly LSTM-based architectures and their ensembles, provide superior robustness and accuracy for small-cap stocks characterised by higher volatility and non-linear price dynamics. Forecasting log-returns as the prediction target contributes to numerical stability across both regimes and enables fair comparison between fundamentally different model families.

5. Conclusion

In this work, a comparative analysis of statistical and deep learning models for next-day stock price forecasting on NSE equities was conducted. The results indicate that ARIMA and GARCH models achieve strong performance on large-cap stocks characterized by stable trends and moderate volatility. However, their effectiveness is reduced when applied to small-cap stocks with higher volatility and nonlinear price movements.

Deep learning models, particularly LSTM and TCN architectures, demonstrate improved robustness on small-cap equities by effectively capturing nonlinear temporal dependencies. The neural ensemble model provides competitive results but does not consistently outperform the best individual models across all market segments. Overall, the findings confirm that forecasting performance is strongly dependent on market characteristics and volatility levels. Consequently, the selection of forecasting models should be guided by the structural properties of the target market segment.

Future research may focus on adaptive hybrid frameworks and volatility-conditioned model selection strategies to further enhance forecasting reliability.

References

[1] M. S. Chowdhury, N. Nabi, M. N. U. Rana, M. Shaima, H. Esa, A. Mitra, M. A. S. Mozumder, I. A. Liza, M. M. R. Sweet, and R. Naznin, "Deep learning models for stock market forecasting: A comprehensive comparative analysis," *Journal of Business and Management Studies*, vol. 6, no. 2, p. 95, 2024.

[2] W. Long, Z. Lu, and L. Cui, "Deep learning-based feature engineering for stock price movement prediction," *Knowledge-Based Systems*, vol. 164, pp. 163–173, 2019.

[3] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Dilated convolutional neural networks for time series forecasting," *Journal of Computational Finance*, vol. 22, no. 4, pp. 73–101, 2018.

[4] J. Sirignano and R. Cont, "Universal features of price formation in financial markets: perspectives from deep learning," in *Machine learning and AI in finance*. Routledge, 2021, pp. 5–15.

[5] S. Gu, B. Kelly, and D. Xiu, "Empirical asset pricing via machine learning," *The Review of Financial Studies*, vol. 33, no. 5, pp. 2223– 2273, 2020.

[6] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Using deep learning to detect price change indications in financial markets," in *2017 25th European signal processing conference (EUSIPCO)*. IEEE, 2017, pp. 2511–2515.

[7] O. B. Sezer, U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review," *Applied Soft Computing*, vol. 90, p. 106181, 2020.

[8] 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, 2020.

[9] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Stock price forecasting using deep learning: A comparative study," *Journal of King Saud University – Computer and Information Sciences*, vol. 33, no. 8, pp. 1018–1031, 2021.

[10] B. Lim, S. O. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[11] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22419–22430, 2021.

[12] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[13] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *Journal of Machine Learning Research*, vol. 21, no. 113, pp. 1–33, 2020.

[14] A. Borovykh, S. Bohte, and C. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *Journal of Computational Finance*, vol. 24, no. 3, pp. 1–39, 2021.

[15] J. Sirignano and R. Cont, "Universal features of price formation in financial markets: Perspectives from deep learning," *Quantitative Finance*, vol. 20, no. 9, pp. 1449–1459, 2020.