International Journal of Basic and Applied Sciences, 14 (SI-2) (2025) 131-140

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/6wpxv706
Research paper

Performance Analysis of LSTM, Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), Convolutional LSTM (ConvLSTM) and LSTM with Attention in Stock Market Prediction

Santosh Kumar Sahu [1,3]*, A. S. Mokahde [2], Jagdish Chakole [3], Samir N. Ajani [4], Manish M. Goswami [5]

[1] *Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology (VNIT), Nagpur, India*
[2] *Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology (VNIT), Nagpur, India*
[3] *Department of Computer Science and Engineering, Indian Institute of Information Technology (IIIT), Nagpur, India*
[4] *School of Computer Science and Engineering, Ramdeobaba University (RBU), Nagpur, India*
[5] *Department of Computer Science and Engineering, SB Jain Institute of Technology, Management and Research, Nagpur, India*
**Corresponding Author E-Mail: sksahu.earth@gmail.com*

Received: June 6, 2025, Accepted: June 18, 2025, Published: July 27, 2025

Abstract

Predicting the stock market is a complex and challenging endeavor because financial time series are inherently volatile and nonlinear. Recurrent neural networks (RNNs) and their advanced variants have shown significant promise in modeling sequential data. This study evaluates and contrasts the effectiveness of five deep learning models Classic LSTM, Bidirectional LSTM (BiLSTM), GRU, Convolutional LSTM (ConvLSTM), and LSTM with Attention Mechanism in forecasting stock market closing prices. Our evaluation employs historical stock index data, with model performance quantified using standard regression metrics—MSE, RMSE, MAE, MAPE, and R^2. Our results show that GRU consistently outperforms other models in capturing intricate temporal patterns, Results show that LSTM with Attention Mecha-nism achieves the highest accuracy (MSE: 0.000370 MAE: 0.014482, RMSE: 0.019242, MAPE: 6.676856 R2: 0.873357) while GRU offers the best balance of accuracy and efficiency. These findings provide actionable insights for selecting appropriate RNN models for financial forecasting applications.

Keywords: *Deep Learning, Attention Mechanisms, Recurrent neural networks, LSTM, GRU, BiLSTM, Stock Market*

1. Introduction

Stock market forecasting is vital for investors, traders, and financial institutions. Traditional statistical models like ARIMA and GARCH, while useful, often fail to capture complex non-linear dependencies in financial time series. Deep learning models, particularly RNN-based architectures, have emerged as powerful alternatives. Stock market prediction remains challenging due to non-stationary price dynamics. While traditional technical indicators (e.g., MACD, RSI) have limitations in high-frequency regimes, deep learning models capture non-linear time-based relationships.

Predicting stock prices is inherently difficult because financial time-series data exhibit high volatility, nonlinear trends, and noise. Among deep learning approaches, Recurrent Neural Networks (RNNs) and their advanced variants—such as LSTM, BiLSTM, GRU, ConvLSTM, and LSTM with Attention—have demonstrated strong capabilities in capturing complex temporal dependencies. This paper evaluates the performance of these five architectures in predicting stock prices, focusing on their accuracy, computational efficiency, and ability to handle market fluctuations.

Day trading involves two primary positions: long and short. A long position requires purchasing stocks with the expectation of selling them at a higher price, while a short position involves selling borrowed stocks first, aiming to repurchase them at a lower price. Conversely, in a short position, the trader sells stocks first and buys them back later, betting on a price decline. Market sentiment significantly influences stock price trends. Good news generally increases stock prices, while bad news usually makes them drop quickly. Unlike delivery-based (positional) trading, day trading involves no overnight risk and allows short selling. Since day trading requires closing positions the same day, selecting highly liquid stocks with substantial trading volume is crucial [1]. Predicting stock movements remains a challenging re-search problem. Traders use three main analysis types: quantitative (QA) for statistical models, technical (TA) for price patterns, and fundamental (FA) for company/value assessment - all helping interpret market data and news [2]. Li et al. systematically reviewed 229 studies from management information systems (MIS) and examining the relationship between web-based information and stock markets. Their work categorized media types and explored techniques for converting textual data into actionable insights [3]. Stock market predictions attract significant interest, yet achieving 100% accuracy is impossible due to external factors like geopolitical events, economic conditions, and investor psychology. Saini et al. compared fundamental and technical analysis, along with various forecasting approaches such as time series models and artificial neural networks [4].

Conventional statistical approaches, such as ARIMA and GARCH, often struggle to model the intricate non-linear patterns inherent in financial time series data. In contrast, deep learning techniques, particularly those based on Recurrent Neural Networks (RNNs), have shown significant potential in capturing complex temporal relationships. However, the volatile and non-stationary nature of stock prices poses ongoing challenges. Traditional indicators like Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI) are less effective in high-frequency trading scenarios, whereas deep learning models excel in identifying long-term dependencies. This paper evaluates these five architectures for stock price prediction, focusing on their accuracy, computational efficiency, and robustness. The study aims to guide practitioners in selecting appropriate models for financial forecasting. This paper presents a comparative study of five RNN-based models for stock price prediction:

- The Standard LSTM Network.
- The Bidirectional LSTM Network
- The Gated Recurrent Unit
- The Convolutional LSTM and
- The LSTM with Attention Mechanism

After reviewing existing literature in Section 2, we present our methodology in Section 3. Experimental results follow in Section 4, with analysis in Section 5. We conclude with key takeaways in Section 6 and future research opportunities in Section 7.

2. Related Work

The Efficient Market Hypothesis (EMH) states that stock prices reflect all available information. Since stock prices adjust gradually to new data, consistently outperforming the market on a risk-adjusted basis is considered impossible. Under this hypothesis, investors cannot reliably buy undervalued stocks or sell overvalued ones, as stock movements are influenced by numerous unpredictable factors. Early models suggest stock prices cannot be forecasted because they respond only to new information not historical or current prices [5]. Art Paspanthong et al. (Stanford University) applied machine learning models including LSTM, support vector machines (SVM), logistic regression and convolutional neural networks (CNN) to predict minute-by-minute trading activity. Among these, SVM with a polynomial kernel delivered the best performance when used for portfolio optimization [6]. Jagdish Chakole et al. developed a Q-learning reinforcement agent that learns trend-following strategies by rewarding profitable trades. This approach, tested on Indian and U.S. stock market data, demonstrated improved forecasting accuracy and profitability [7][8][9]. Jigar Patel et al. focused on predicting stock movements in Indian markets using Naive Bayes, ANN, random forests (RF), and SVM. Two approaches were compared: Using 10 technical indicators (open, high, low, close prices). Converting these indicators into trend-deterministic data. The latter method significantly improved prediction accuracy [10].

Zheng Tan et al. employed a random forest (RF) model for stock selection in China's market, combining technical, fundamental, and momentum features to forecast short- and long-term trends. The model classified training data to predict future market behavior [11]. M. Ananthi et al. proposed a hybrid method using regression analysis and candlestick patterns to predict a company's stock price over several days [12]. This study investigates the predictive capacity of prior-day trading patterns to generate pre-market LONG/SHORT signals for intraday equity trading, with positions initiated at market open and liquidated at session close.[13]. The paper examines CNN, LSTM, and hybrid LSTM-CNN models for NSE stock price forecasting, showing their robustness and accuracy, especially in volatile markets.[14]. Attention-LSTM outperforms all other models, achieving the lowest Root Mean Squared Error (RMSE) in stock price forecasting. Hyper-parameter-tuned ARIMA is more accurate than standard ARIMA but less precise than deep learning models. LSTM effectively captures temporal relationships and reduces prediction errors, with Attention-LSTM further improving forecast accuracy by dynamically weighting historical data through its attention mechanism[15]. The findings show that the LSTM model consistently surpasses the ANN model in accuracy and trend detection. For example, over a 30-day period, the LSTM accurately predicted 8 out of 10 stocks, while the ANN predicted only four[16]. The proposed solution improves daily forecasting accuracy by 83.37-84.05% versus LSTM variants and by 55.8% versus LSTM-ARIMA-GARCH, while also exceeding LSTMNP, AHMPSO-LSTM, and DLWR-LSTM performance[17].

3. Methodology

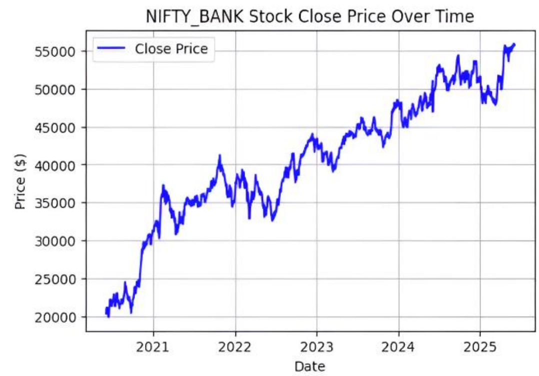
3.1 Data Collection

The dataset having daily closing prices of the four different sectors of Nifty index Nifty Bank, Nifty IT, Nifty Energy, Nifty Pharma and Nifty Auto from June 2020 to June 2025, sourced from Yahoo Finance. Data is normalized to 0 to 1 and split in 70%, 20% ,10% as training, validation and testing sets. We used daily historical data from the following indices from the last five years:

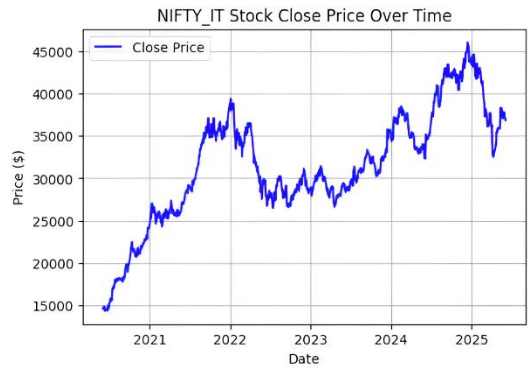
- NIFTY BANK : Tracks the performance of India's top banking stocks, including major public and private sector banks. It is highly sensitive to interest rate changes and economic policies.
- NIFTY IT : Represents leading Indian IT companies, reflecting the growth of the tech sector driven by global demand for software services and digital transformation.
- NIFTY ENERGY : Covers key players in the oil, gas, and power sectors, influenced by global crude prices and domestic energy demand.
- NIFTY PHARMA : Includes major pharmaceutical and biotechnology firms, impacted by drug exports, regulatory approvals, and R&D innovations.

Table 1: Stock indices and number of transactions

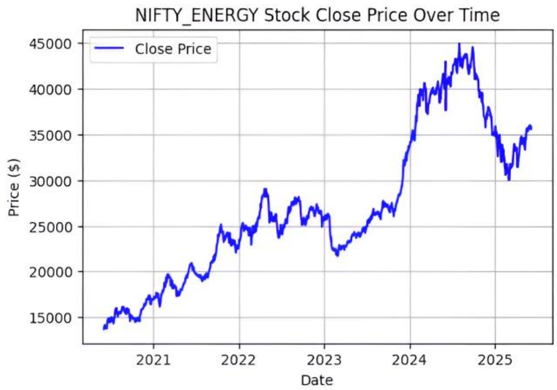
SN	Stock Indices	No. of Transaction
1	Nifty Bank	1234
2	Nifty IT	1234
3	Nifty Energy	1234
4	Nifty Pharma	1234



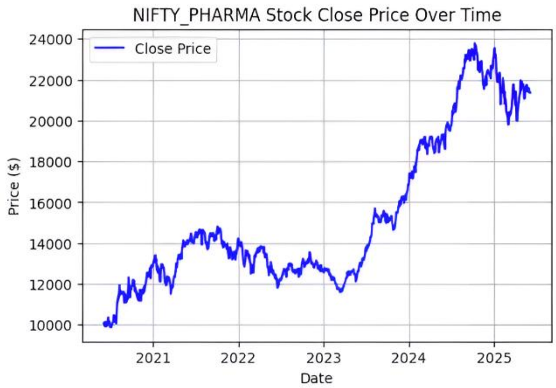
Nifty Bank: Price over Date



Nifty IT: Price over Date



Nifty Energy: Price over Date



Nifty Pharma: Price over Date

Fig. 1: Price movement chart of different Stock indices

3.2 Model Architecture: Following models were applied in this study:

- LSTM: Uses memory cells and gates to capture long-term dependencies.
- BiLSTM: Processes input from both past and future directions.
- GRU: A simplified version of LSTM with fewer gates.
- ConvLSTM: Incorporates convolution operations into LSTM, suitable for spatio-temporal data.
- Attention-based LSTM: Combines LSTM with attention mechanism to focus on important time steps.

3.2.1 Classic LSTM

The Classic LSTM processes sequential data through gated memory cells (input, forget, output), with our specific architecture containing two 128-neuron LSTM layers and a dense output layer.

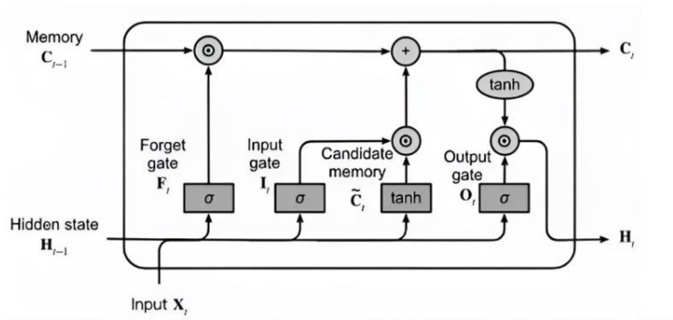


Fig. 2: LSTM Architecture

Xt = Input vector at present timestamp
ht = Hidden State at present timestamp
ht-1 = Hidden State at previous timestamp
Ct = Cell State
Ct-1 = Cell State at previous timestamp

3.2.2 Bidirectional LSTM (BiLSTM)

The BiLSTM model reads input data both forwards and backwards to understand complete sequence relationships. Our specific implementation uses two 128-neuron bidirectional layers and a final dense layer.

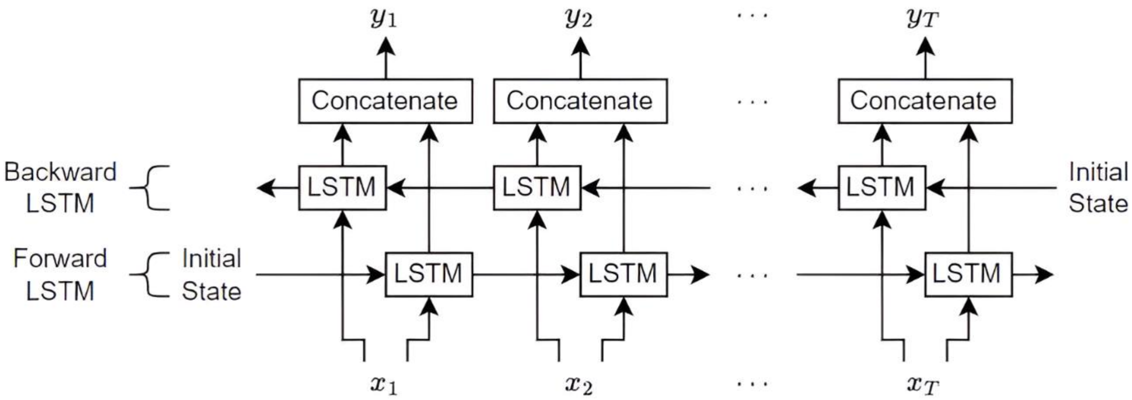


Fig. 3: Bidirectional LSTM (BiLSTM) Architecture

3.2.3 Gated Recurrent Unit

GRU reduces LSTM complexity through combined update gates, implemented here with two 128-unit GRU layers and a dense output layer.

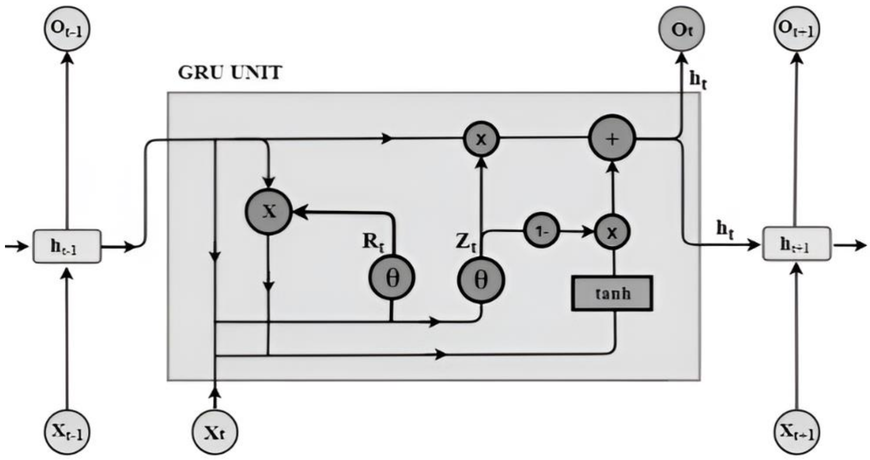


Fig. 4: Gated Recurrent Unit (GRU) Architecture

3.2.4 Convolutional LSTM

ConvLSTM combines convolutional and recurrent operations to model spatiotemporal relationships, implemented here with a 64-filter (3x3 kernel) layer and dense output.

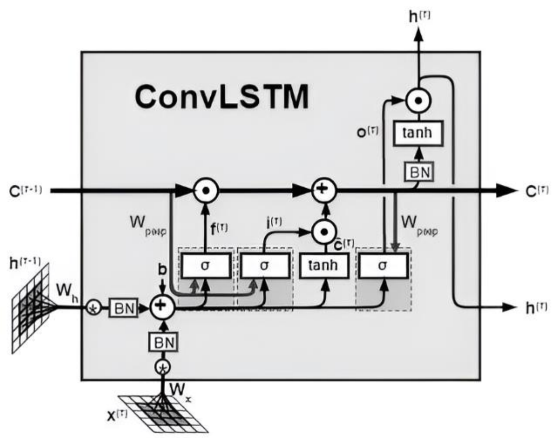


Fig. 5: ConvLSTM Architecture

3.2.5 LSTM with Attention Mechanism

This model augments Classic LSTM with an attention layer to focus on relevant time steps. It includes two LSTM layers (128 units each), an attention layer, and a dense layer

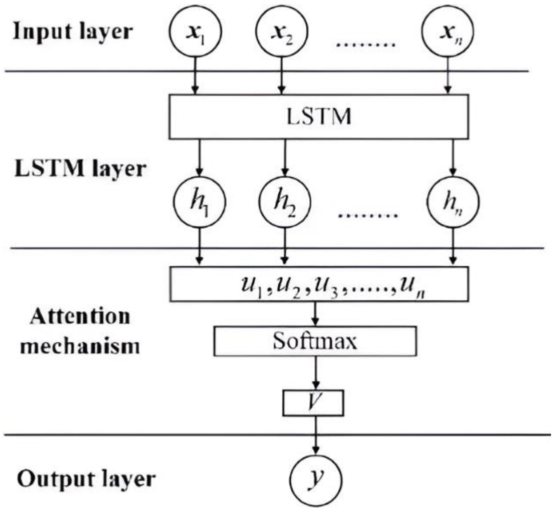


Fig. 6: LSTM with Attention Mechanism Architecture

3.3 Key Terms and Abbreviations for Standard LSTM

- LSTM: An advanced type of RNN that uses memory cells with input, forget, and output gates to remember information over extended time periods.
- RNN: Recurrent Neural Network, the base architecture for LSTM, processing sequential data.
- Cell State (C_t): Component of LSTM memory that communicates data across time steps.
- Hidden State (h_t): The LSTM's 'memory output' at each step - used both for current predictions and as input for the next time step
- Gates: Mechanisms to control information flow:
- Input Gate (i_t): Regulates the incorporation of new information into the cell state.
- Forget Gate (f_t): Modulates the retention/discard ratio of historical cell state information.
- Output Gate (o_t): Governs the exposure of cell state content to the hidden state.
- Candidate Cell State (C̃_t): A proposed cell state update based on current inputs and gated historical context.
- Sigmoid (σ): Activation function (0 to 1) used in gates to control information flow.
- Tanh: Hyperbolic tangent activation (-1 to 1) used for scaling cell state and candidate updates.
- Weights (W, U): Learnable parameters for input and recurrent connections.
- Bias (b): Additive term in gate computations.

3.4 Experimental Setup

- Framework: Tensor Flow and Keras
- Epochs: 20
- Batch size: 64
- Optimizer: Adam with default momentum parameters
- Loss Criterion: MSE computed across all output dimensions
- Evaluation Metrics: MSE, RMSE, MAE, MAPE, R^2

4. Result and Discussion

Following results, we have been found in this research.

A. Nifty Bank

Table 2: MSE, MAE, RMSE, MAPE and R^2 of different Model for Nifty Bank Indices

Models	MSE	MAE	RMSE	MAPE	R^2
LSTM	0.001077	0.026256	0.032823	6.404321	0.631500
BiLSTM	0.000784	0.021858	0.027996	6.602210	0.731921
GRU	0.000370	0.014482	0.019242	6.676856	0.873357
ConvLSTM	0.001737	0.033487	0.041679	5.994405	0.405827
LSTM+Attention	0.003709	0.048057	0.060905	5.491411	-0.268754

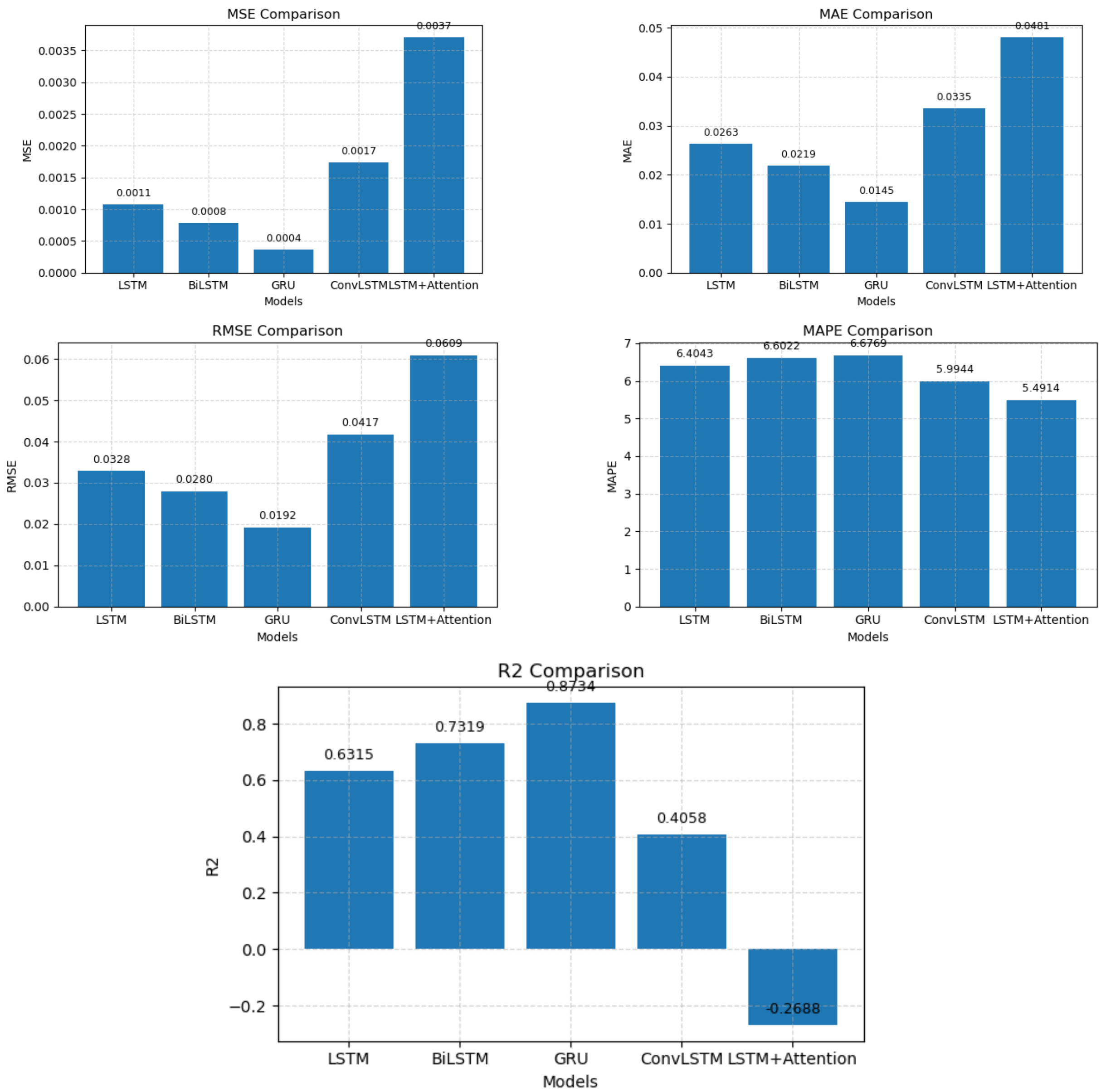


Fig. 7: MSE, MAE, RMSE, MAPE, R^2 of Nifty Bank

B. Nifty IT

Table 3: MSE, MAE, RMSE, MAPE and R^2 of different Model for Nifty IT Indices

Model	MSE	MAE	RMSE	MAPE	R^2
LSTM	0.001478	0.031279	0.038441	14.127451	0.853325
BiLSTM	0.000924	0.024111	0.030399	14.542692	0.908272
GRU	0.000573	0.019123	0.023938	14.176239	0.943122
ConvLSTM	0.002561	0.042281	0.050604	14.293955	0.745824
LSTM+Attention	0.010112	0.086244	0.100556	13.682608	-0.003658

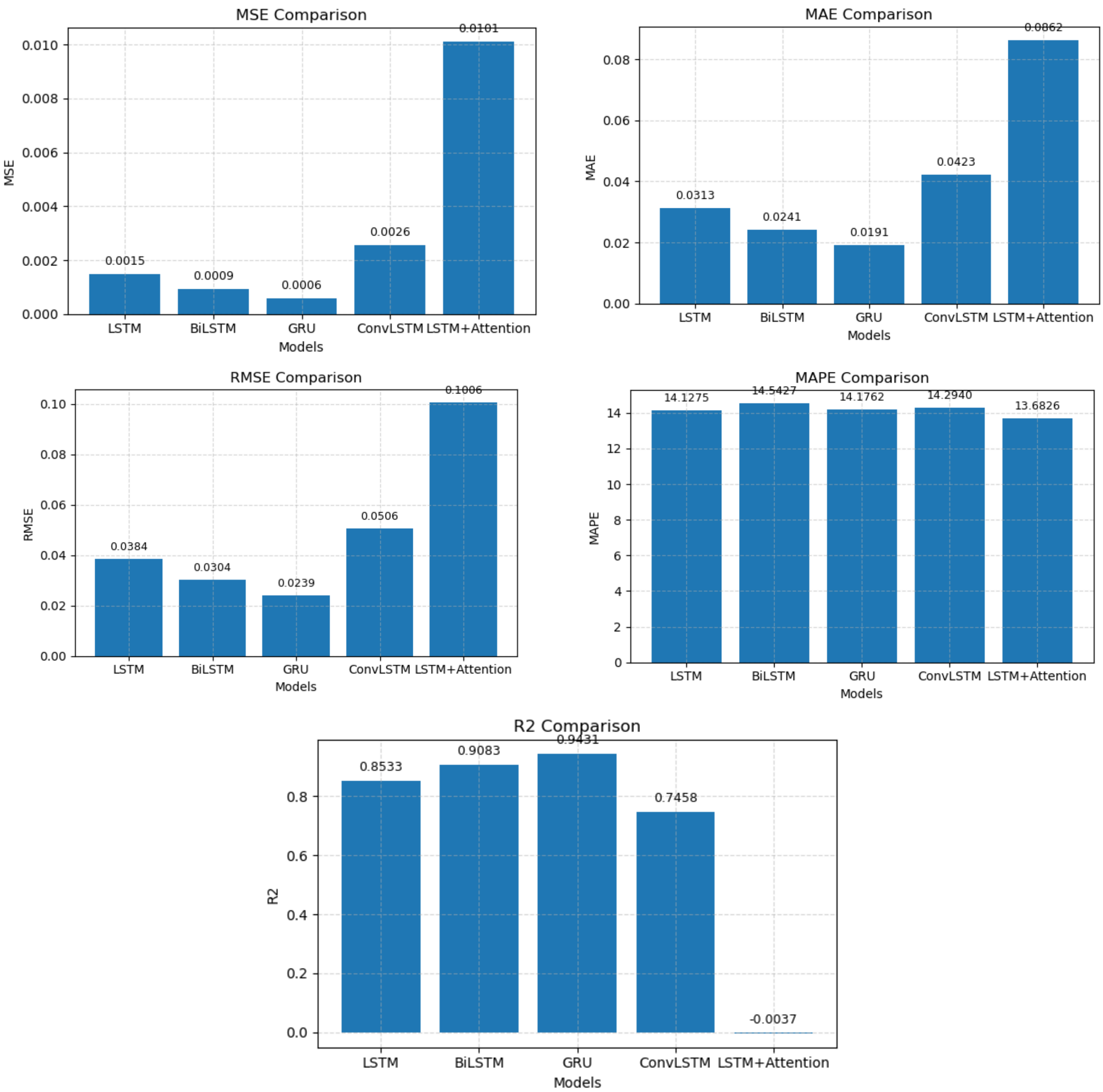


Fig. 8: MSE, MAE, RMSE, MAPE, R^2 of Nifty IT

C. Nifty Energy

Table 4: MSE, MAE, RMSE, MAPE and R^2 of different Model for Nifty Energy Indices

Model	MSE	MAE	RMSE	MAPE	R^2
LSTM	0.002378	0.039556	0.048763	22.218385	0.873953
BiLSTM	0.001098	0.026226	0.033134	21.871459	0.941802
GRU	0.000564	0.018763	0.023741	21.333780	0.970122
ConvLSTM	0.001421	0.031559	0.037694	21.136911	0.924683
LSTM+Attention	0.018080	0.110581	0.134461	26.103858	0.041614

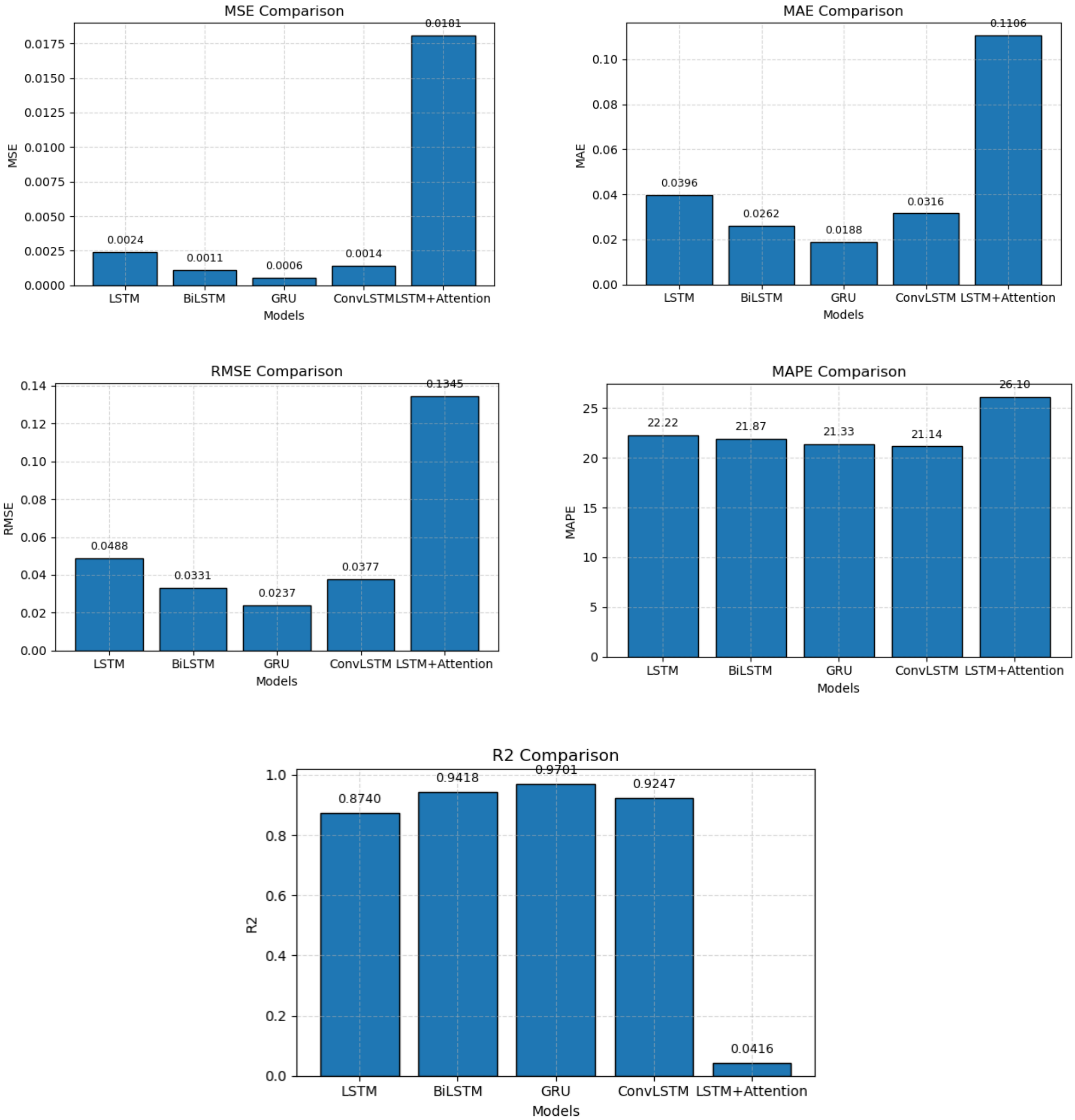


Fig. 9: MSE, MAE, RMSE, MAPE, R^2 of Nifty Energy

D. Nifty Pharma

Table 5: MSE, MAE, RMSE, MAPE and R2 of different Model for Nifty Pharma Indices

Model	MSE	MAE	RMSE	MAPE	R^2
LSTM	0.001207	0.027570	0.034745	10.421085	0.791454
BiLSTM	0.001198	0.026781	0.034611	10.766425	0.793061
GRU	0.000458	0.017043	0.021409	10.207033	0.920822
ConvLSTM	0.002129	0.038381	0.046137	10.938715	0.632282
LSTM+Attention	0.012625	0.093490	0.112362	14.621101	-1.180966

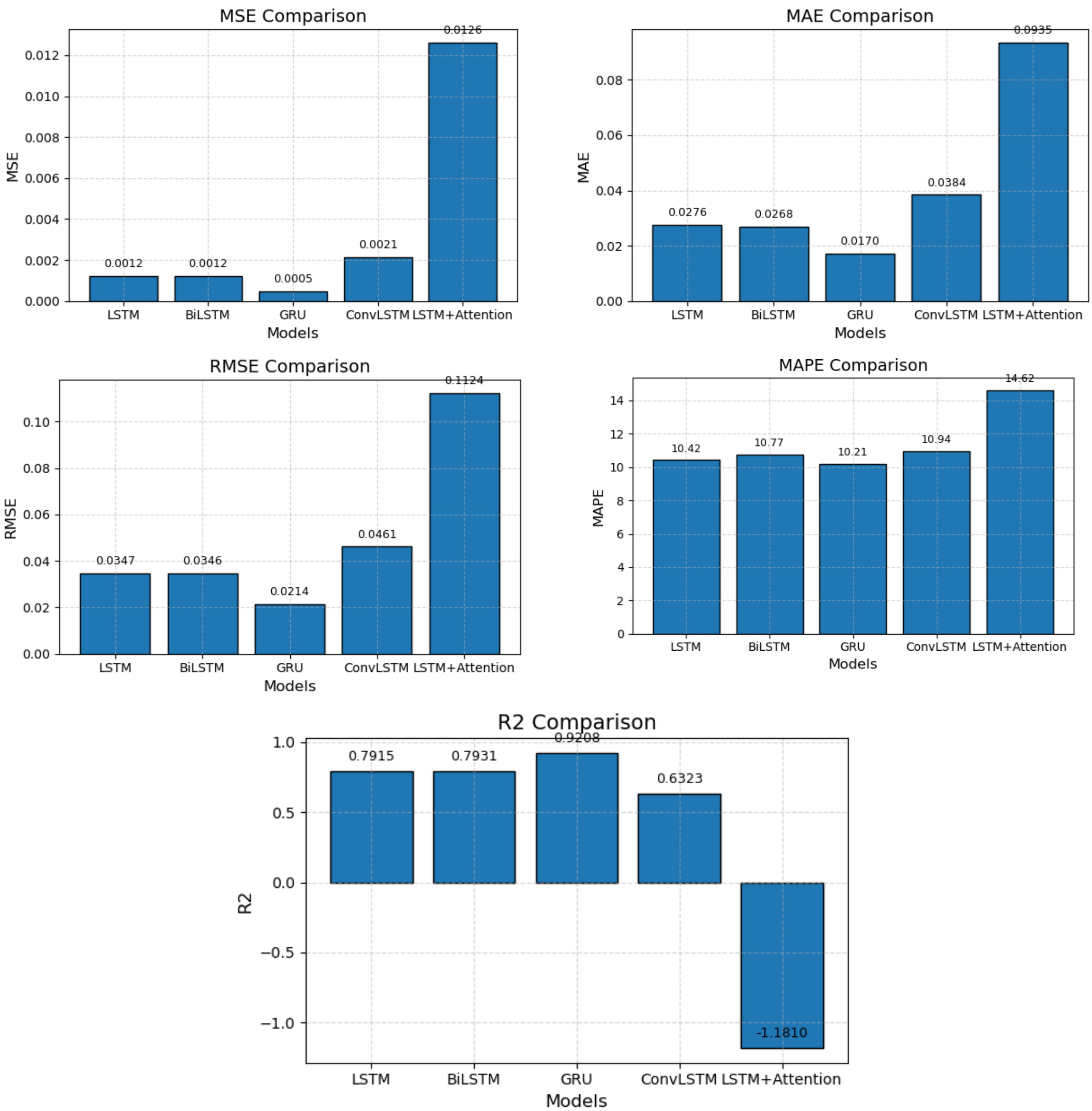


Fig. 10: MSE, MAE, RMSE, MAPE, R^2 of Nifty Pharma

5. Discussion

The comparative analysis reveals several key strengths and limitations of each RNN-based architecture for stock market prediction. Among all models tested across Nifty indices, GRU consistently demonstrated the best balance between predictive accuracy and training efficiency. It achieved high values and low error metrics with relatively quick training time, making ideal for real-time or resource-constrained environments. Interestingly, while the LSTM with Attention Mechanism was expected to outperform all others due to its ability to focus on important time steps, it underperformed on all indices except Nifty Bank. Its poor results, particularly in Nifty IT and Nifty Pharma, may be attributed to overfitting or ineffective attention weight distribution in relatively stable datasets. This emphasizes that attention mechanisms must be carefully tuned and may not universally guarantee better results.

The ConvLSTM, although originally designed for spatial-temporal data, showed decent performance, especially in indices like Nifty Energy, suggesting that temporal trends with some periodicity or structure may benefit from convolutional feature extraction. However, its larger training time and complexity reduces its practicality in some use cases. Classic LSTM and BiLSTM offered moderate performance, with BiLSTM showing marginal improvements in capturing bidirectional dependencies resulting in longer model convergence times. This indicates that while bidirectionality is theoretically beneficial, in stock data where future context is unavailable at inference, its impact may be limited.

Overall, model accuracy strongly correlates with the characteristics of the specific dataset (e.g., volatility, trend, noise). No single architecture proved superior across all sectors, reaffirming the importance of tailoring the model choice to the financial instrument being analyzed.

6. Conclusion

This study provides a detailed comparison of five deep learning architectures—Classic LSTM, BiLSTM, GRU, ConvLSTM, and LSTM with Attention—applied to stock market prediction using daily historical data from various Nifty sectoral indices. The findings highlight that:

- GRU achieved the most consistent and robust performance across all indices.
- LSTM with Attention, despite its theoretical advantages, underperformed in most sectors and demands more refined tuning.
- ConvLSTM showed promise for indices with structured temporal patterns but require higher computational resources.
- Classic LSTM and BiLSTM remain solid choices, with BiLSTM offering slight improvements for trend-heavy data.

The choice of architecture must therefore be guided by data characteristics, resource constraints, and prediction horizon. These results contribute to a more informed strategic selection of neural network architectures for financial time-series analysis.

7. Future Work

To further improve the predictive capabilities of deep learning in stock market forecasting, future research can explore:
Multimodal Data Integration: The system ingests three external data streams: news sentiment (VADER-processed), macroeconomic indicators (FRED API), and social media trends (Twitter API)
High-Frequency Trading Analysis: Extending the models to work with intraday (tick-level) data, where volatility and noise are more pronounced.
Advanced Architectures: Investigating Transformer-based models such as BERT, Informer, or Temporal Fusion Transformers, which have shown superior performance in sequential learning tasks.
Hybrid Approaches: Combining statistical models with deep learning or ensembling multiple architectures to leverage their individual strengths.
Such advancements can make AI-driven stock market prediction more accurate, interpretable, and applicable across various trading scenarios.

Acknowledgement

We express our sincere gratitude to our colleagues and mentors at Visvesvaraya National Institute of Technology (VNIT), Indian Institute of Information Technology (IIITN), for their invaluable support and guidance throughout this research. We are thankful to the academic community for providing access to essential resources and datasets, including Yahoo Finance for the historical stock index data.

References

[1] Sahu, S.K., Mokhade, A. and Bokde, N.D., 2023. An overview of machine learning, deep learning, and reinforcement learning-based techniques in quantitative finance: recent progress and challenges. Applied Sciences, 13(3), p.1956.
[2] Zheng A, Jin J (2017) Using AI to make predictions on stock market. pp. 1–6.
[3] Li Q, Chen Y, Wang J, Chen Y, Chen H (2018) Web media and stock markets: A survey and future directions from a big data perspective. *IEEE Transactions on Knowledge and Data Engineering* 30(2): 381–399. https://doi.org/10.1109/TKDE.2017.2763144.
[4] Saini A, Sharma A (2019) Predicting the unpredictable: An application of machine learning algorithms in Indian stock market. *Annals of Data Science*. https://doi.org/10.1007/s40745-019-00230-7.
[5] Weng B, Ahmed M A, Megahed F M (2017) Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications* 79: 153–163. https://doi.org/10.1016/j.eswa.2017.02.041.
[6] Paspanthong, A., Tantivasadakarn, N. and Vithayapalert, W., 2019. Machine learning in intraday stock trading. Computer Science Department, Stanford University.
[7] Chakole J, Kurhekar M (2020) Trend following deep Q-learning strategy for stock trading. *Expert Systems* 37(4): 1–16. https://doi.org/10.1111/exsy.12514.
[8] Chakole J B, Kolhe M S, Mahapurush G D, Yadav A, Kurhekar M P (2021) A Q-learning agent for automated trading in equity stock markets. *Expert Systems with Applications* 163: 113761. https://doi.org/10.1016/j.eswa.2020.113761.
[9] Park H, Sim M K, Choi D G (2020) An intelligent financial portfolio trading strategy using deep Q-learning. *Expert Systems with Applications* 158: 113573. https://doi.org/10.1016/j.eswa.2020.113573.
[10] Patel J, Shah S, Thakkar P, Kotecha K (2015) Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications* 42(1): 259–268. https://doi.org/10.1016/j.eswa.2014.07.040.
[11] Tan Z, Yan Z, Zhu G (2019) Stock selection with random forest: An exploitation of excess return in the Chinese stock market. *Heliyon* 5(8): e02310. https://doi.org/10.1016/j.heliyon.2019.e02310.
[12] Ananthi M, Vijayakumar K (2020) Stock market analysis using candlestick regression and market trend prediction (CKRM). *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-020-01892-5.
[13] Sahu S K, Mokhade A S, Agrawal P K, Kalyani K (2025) Utilizing supervised machine learning and technical indicators for quantitative trading to improve stock market trading decisions. *SN Computer Science* 6(5). https://doi.org/10.1007/s42979-025-04025-x.
[14] Joshi S, Mahanthi B L, Pavithra G, Pokkuluri K S, Ninawe S S, Sahu R (2025) Integrating LSTM and CNN for stock market prediction: A dynamic machine learning approach. *Journal of Artificial Intelligence and Technology* 5: 168–179. https://doi.org/10.37965/jait.2025.0652.
[15] Abdulghani A, Alhammadi S (2025) Enhancing stock market prediction using attention-based LSTM models and classical models. 2(4): 25–32.
[16] Talhartit I, et al. (2025) Machine learning for stock price prediction on the Casablanca Stock Exchange: A comparative study of ANN and LSTM approaches. HAL Id: hal-05063012.
[17] Mehtarizadeh H, Mansouri N, Mohammad B, Zade H, Hosseini M M (2025) Stock price prediction with SCA-LSTM network and statistical model ARIMA-GARCH. *Springer US*. https://doi.org/10.1007/s11227-024-06775-6.