International Research Journal of Multidisciplinary Scope (IRJMS), 2026; 7(1): 1175-1185 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2026.v07i01.07857

Comparative Evaluation of Deep Learning Models for Multi-Stock Prediction and Portfolio Optimization

Umesh Pratap Singh*, Manish Madhava Tripathi

Integral University, Computer Science & Engineering, Lucknow, India. *Corresponding Author's Email: usgemini@gmail.com

Abstract

The stock market is well-known for volatility and unpredictability, which can pose significant risks to investment strategies. This manuscript compares LSTM, GRU, CNN, and LSTM-CNN models for stock price prediction using standard performance metrics. A structured methodology is applied, including detailed data preprocessing, hyperparameter tuning, and splitting the data into training and test sets. The results show that each model has its own strengths and limitations in learning short- and long-term patterns, as well as differences in computational efficiency. Model performance is evaluated using RMSE, MAE, and R-squared metrics to support portfolio optimization strategies. All models are trained and tested under consistent data inputs, preprocessing steps, and experimental settings. This method helps maintain reliable and comparable results across different models. Practical applications include dynamic asset allocation, risk-based position sizing, sector-specific model deployment, and integration of LSTM forecasts into institutional portfolio strategies. The analysis examines model performance using datasets from six different companies. The LSTM model outperformed the other models due to its strong ability to learn long-term patterns in volatile stock price data. In comparison, CNN and GRU models showed weaker results as they were less effective at retaining long-range temporal information under changing market conditions. These findings provide practical guidance for investors in selecting appropriate deep learning models for real-world financial forecasting.

Keywords: Convolutional Neural Network, Gated Recurrent Unit, Long Short-Term Memory, Metrics, Multi-Stock Prediction Portfolio Optimization.

Introduction

Accurate multi-stock price prediction plays a vital role in modern investment decision-making by supporting risk management, asset allocation, and return optimization. Investors regularly seek to balance risk and return in rapidly changing financial markets. Economic indicators, market sentiment, and unexpected external events shape price movements. As a result, investors and analysts increasingly rely on data-driven forecasting models to build more robust and adaptive investment strategies. Among these approaches, deep learning architectures such as Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), and hybrid LSTM–CNN models have gained significant attention for time-series-based stock price prediction due to their ability to model complex, non-linear relationships in financial data. Each deep learning model has its own strengths and limitations in capturing time-based patterns and reacting to market volatility. Comparing the performance of these models is crucial for understanding their behaviour under different market conditions. Model effectiveness is assessed using widely accepted performance metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). Together, these metrics offer a clearer view of prediction accuracy and model stability across multiple stocks. When a model consistently demonstrates superior predictive performance, it may be integrated into investment decision frameworks to support portfolio construction and rebalancing strategies (1). Portfolio-level decision-making using learning-based approaches has also been explored through deep reinforcement learning frameworks, highlighting the importance of systematic evaluation across multiple assets (2). A systematic comparison of deep learning models is important because strong performance under stable market conditions does not necessarily translate into reliability during periods of high volatility or structural change. Understanding how different models respond to market changes enables investors and portfolio managers to select approa-

(Received 27th August 2025; Accepted 13th January 2026; Published 28th January 2026)

ches that align with their risk preferences and investment objectives. This supports better portfolio decisions and more effective risk management. Previous research shows that market information changes frequently and stock prices are inherently uncertain. These factors can affect prediction stability and highlight the importance of robust and flexible forecasting methods (3, 4).

Earlier studies have examined various machine learning and deep learning models to capture temporal dependence, non-linearity, and volatility in financial time-series. Comprehensive surveys have highlighted both the potential and the limitations of these approaches, underscoring the need for comparative studies that emphasize practical applicability across multiple stocks rather than isolated case analyses (5). In parallel, recent studies have examined deep reinforcement learning frameworks for trading and decision support, demonstrating the value of combining predictive modelling with data-driven decision-making systems (6).

The LSTM model captures long-term temporal dependencies in stock price data, enabling more accurate modelling of sequential market behaviour. By incorporating memory cells and gating mechanisms, LSTM networks selectively retain relevant historical information while filtering out noise, enabling them to model extended market trends and delayed responses more effectively than traditional recurrent networks (7–9). This capability is valuable in stock markets, as reliable prediction depends on identifying true trends beyond short-term fluctuations.

Alternative architectures offer complementary advantages. GRU model provides a simplified gating structure with fewer parameters, resulting in faster training and lower computational overhead while maintaining competitive predictive performance in many scenarios (10). These characteristics make GRUs attractive when computational efficiency or rapid model updates are required. CNN-based models, on the other hand, focus on extracting local and short-term patterns from time-series data and have demonstrated effectiveness in identifying price trends under certain market conditions (11). Hybrid architectures, such as LSTM–CNN, aim to combine the long-term dependency learning of LSTM with the local feature extraction capability of CNN, offering improved flexibility in multi-stock forecasting tasks (12).

Building on these developments, the present study addresses limitations observed in prior work by conducting a comprehensive comparative evaluation of four deep learning architectures—LSTM, GRU, CNN, and LSTM–CNN using stock price data from six major Indian companies spanning multiple sectors. Earlier studies often focused on individual stocks or short time horizons. In contrast, this work uses a long-term dataset spanning approximately 17 years and reports company-wise performance using multiple evaluation metrics. By linking predictive performance to potential portfolio-level applications, the study provides a broader and more application-oriented perspective on deep learning–based stock market forecasting (13).

Recent studies have shown that hybrid and comparative deep learning architectures, including LSTM–CNN models, improve robustness and predictive reliability under volatile market conditions (14, 15). Several review studies have also examined deep learning models for financial time-series forecasting, highlighting their strengths, limitations, and practical challenges across different market settings (16-18). In the Indian context, hybrid machine learning approaches have been applied to analyze trends in major indices such as the Nifty-50 (19).

Methodology

A structured experimental workflow was adopted to ensure reliable and reproducible results. First, raw stock market data were collected from Investing.com. The dataset was then divided into training and testing subsets. Next, four deep learning models—GRU, CNN, LSTM, and the hybrid LSTM–CNN were trained and evaluated separately for each company. Model performance was assessed using RMSE, MAE, and R-squared as evaluation metrics.

Data Collection and Pre-Processing

Data Description

The data were collected from Investing.com covering the period from 1 January 2008 to 10 February 2025. The six well-known firms have a massive market share in the Indian economy. The stock dataset includes the following features: Date, Symbol, Series, Open, High, Low, Close, Volume, and Change (%). The selected companies are Reliance, GAIL, SBIN, IOC, WIPRO, and Tata Steel. Before model training, the dataset was checked for missing values and consistency. Missing records were removed, and numerical features were normalized to ensure comparable input scales and stable model training. A sample of the Reliance stock dataset is shown in Table 1.

Table 1: Sample Dataset of Reliance

Date	Symbol	Series	Open	High	Low	Close	Vol.	Change (%)
01-01-2008	RELIANCE	EQ	324.88	325.9	318.71	320.19	9.73M	-1.19
02-01-2008	RELIANCE	EQ	320.95	324.21	316.04	321.71	25.18M	0.47
03-01-2008	RELIANCE	EQ	320.61	329.38	318.17	326.42	27.88M	1.46
06-02-2025	RELIANCE	EQ	1273.7	1288	1270.35	1281.55	9.96M	0.26
07-02-2025	RELIANCE	EQ	1276.15	1283.7	1262	1266.7	8.76M	-1.16
10-02-2025	RELIANCE	EQ	1264.5	1266.5	1245.55	1253.65	6.97M	-1.03

Training and Testing Dataset Ratio

Out of 4240 records, the final 740 records were used for testing and the remaining were used to train the models. Time-series forecasting models were trained on past data, which is important because of temporal dependencies in financial data. The split between training and testing data influences how well models perform. A fixed train–test split was used to preserve the time order of stock data and avoid information leakage. Although financial time-series are inherently non-stationary, this setup reflects real-world forecasting scenarios and enables a fair comparison across models.

Training of Models

LSTM, GRU, CNN, and hybrid LSTM–CNN models were trained using the training dataset to learn temporal and spatial patterns in the data. Iterative optimization was applied to reduce over fitting and under fitting during the training process.

Testing the Models

All four models—GRU, CNN, LSTM, and the hybrid LSTM–CNN were evaluated using RMSE, MAE, and R^2 metrics. RMSE and MAE were used to measure prediction error, while R^2 indicates how well each model explains the variability in stock prices relative to a mean baseline. Although an R^2 value of 1 represents a perfect fit, it may become negative for highly volatile financial time-series data (20). The detailed comparative results are discussed in the Results section. The mathematical definitions of the evaluation metrics are provided in equations [1-3].

$$\text{MAE} = (1/n) \sum_{i=1}^{n} |(P_i - O_i)| \quad [1]$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (P_i - O_i)^2} \quad [2]$$

$$R^2 = 1 - \frac{\sum(O_i - P_i)^2}{\sum(O_i - \bar{O})^2} \quad [3]$$

Here, P_i and O_i denote the predicted and observed values, respectively, $\bar{O}$ represents the mean of observed values, and n is the total number of observations.

Applied Algorithm

LSTM (Long Short-Term Memory)

Long Short-Term Memory (LSTM) is a recurrent neural network architecture designed to learn long-term dependencies in sequential data, making it suitable for time-series forecasting tasks. As shown in Figure 1, LSTM controls information flow through a set of gating mechanisms. LSTM networks have been widely used for financial market prediction and have demonstrated strong performance in modeling sequential price movements (21). The forget gate determines which information from the previous cell state is retained, as specified in Equation [4]. The input gate regulates the amount of new information added to the cell state, as defined in Equation [5]. A candidate memory state is then generated to represent newly learned information, as described in Equation [6]. The cell state is then updated by combining past and current information, as described in Equation [7]. The output gate controls the information passed to the next hidden state, as defined in Equation [8]. The final hidden state is computed based on the updated cell state, as shown in Equation [9].

$$f_t = \sigma(W_f[h_t - 1, x_t] + b_f) \quad [4]$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad [5]$$
$$g_t = tanh\ (W_g[h_{t-1}, x_t] + b_g) \quad [6]$$
$$C_t = f_t \odot C_{t-1} + i_t \odot g_t \quad [7]$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad [8]$$
$$h_t = o_t \odot tanh\ (C_t) \quad [9]$$

Here, $x_t$denotes the input vector, $h_t$represents the hidden state, $C_t$is the cell state, $\sigma(\cdot)$denotes the sigmoid function, and $\odot$represents element-wise multiplication.

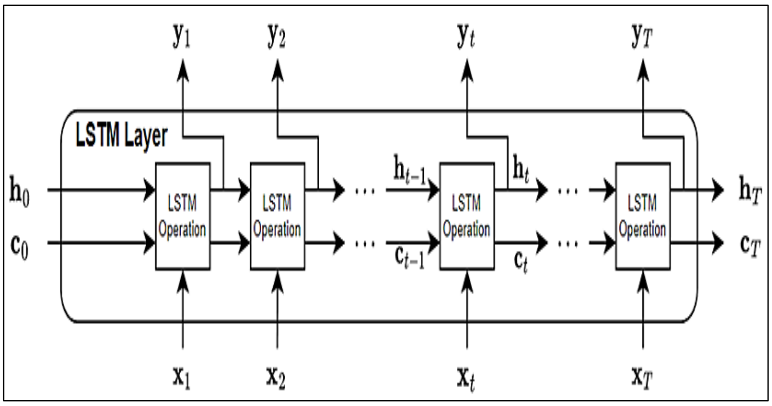

Figure 1: LSTM Architecture

GRU (Gated Recurrent Unit)

The Gated Recurrent Unit (GRU) is a streamlined alternative to LSTM that reduces computational complexity while maintaining strong predictive performance. GRU has a simpler structure and performs efficiently even with limited training data. As shown in Figure 2, the GRU combines the forget and input gates into a single update gate, thereby reducing the number of parameters. The update gate, as defined in Equation [10], controls the amount of past information retained. The reset gate, defined in Equation [11], regulates the influence of previous states. Based on the reset gate, the candidate hidden state is computed as given in Equation [12]. The final hidden state is then obtained by combining the previous and candidate states, as expressed in Equation [13].

$$z_t = \sigma(W_z[h_t - 1, x_t]) \quad [10]$$
$$r_t = \sigma(W_r[h_{t-1}, x_t]) \quad [11]$$
$$\tilde{h}_t = tanh\ (W[r_t \odot h_{t-1}, x_t]) \quad [12]$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad [13]$$

Here, $x_t$denotes the input at time step t, $h_t$represents the hidden state, $z_t$and $r_t$are the update and reset gates, respectively, $\sigma(\cdot)$is the sigmoid activation function, and $\odot$denotes element-wise multiplication.

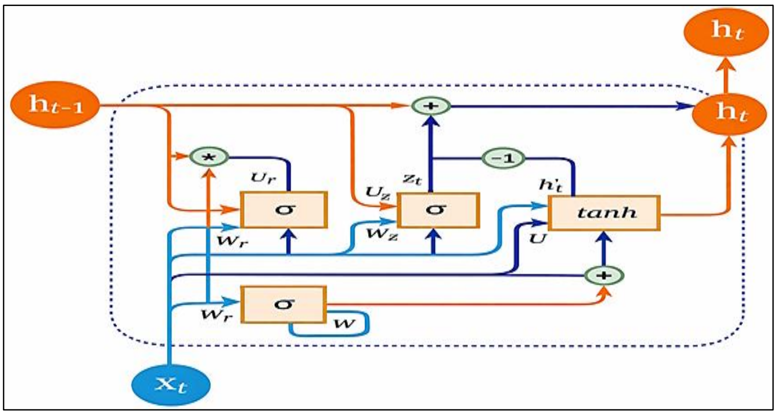

Figure 2: GRU Architecture

CNN (Convolutional Neural Network)

Convolutional Neural Networks are commonly used to capture local patterns and have been increasingly applied to financial time-series forecasting. However, when used independently, CNNs have limited ability to model long-term temporal dependencies, as shown in Figure 3. The historical stock price data are arranged as a two-dimensional time–feature matrix, where rows denote time steps and columns represent financial features. One-dimensional convolution is applied along the temporal axis to extract short-term patterns, as defined in Equation [14]. Non-linear activation is introduced using the ReLU function, as described in Equation [15]. To reduce dimensionality while retaining key information, max pooling is applied in Equation [16]. The final prediction is obtained by passing the resulting features through fully connected layers, as expressed in Equation [17].

$$FeatureMap(t) = (\textstyle\sum_k \ Input\ (t+k).Filter(k)) \qquad [14]$$

$$ActivatedFeatureMap(t) = max\ (0, FeatureMap(t)) \qquad [15]$$

$$PooledFeature(t) = max\ (ActivatedFeatureMap(t\colon t + pool\_size)) \qquad [16]$$

$$Prediction = Activation(\textstyle\sum_i \ Weight_i \cdot PooledFeature_i + Bias) \qquad [17]$$

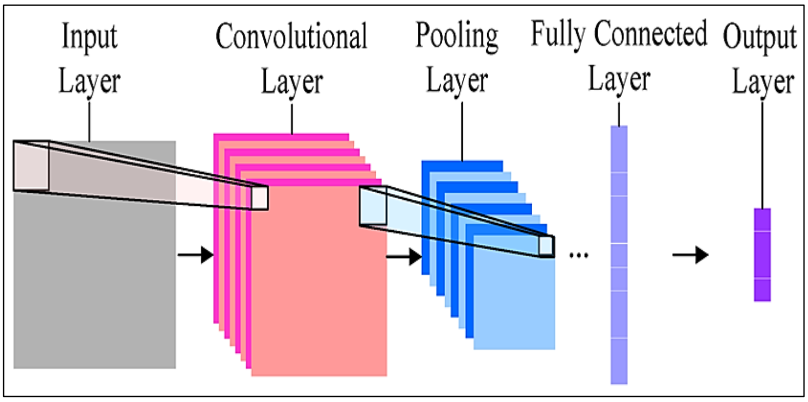

Figure 3: CNN Architecture

Recent studies have expanded deep learning–based stock price prediction by examining a range of recurrent neural network architectures and their comparative performance. A hybrid modelling approach and optimized ensemble learning frameworks have been explored further to enhance prediction accuracy (22-24).

Model Architecture and Hyper Parameter Summary

This section presents the architectures of the four deep learning models considered in the paper: LSTM, GRU, CNN, and the hybrid LSTM–CNN. Figures 4–7 illustrate these architectures, showing the arrangement of layers, the use of dropout, and the design of the output layers.

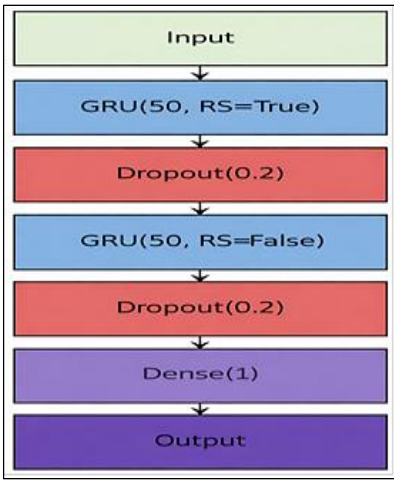

Figure 4: GRU Model Architecture

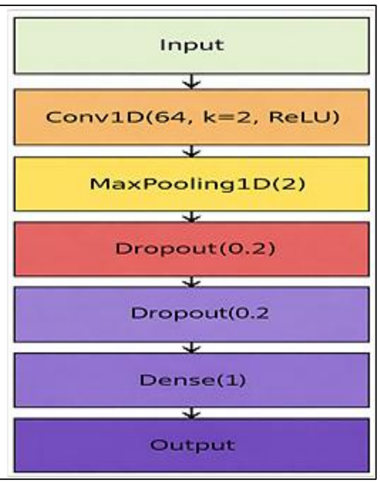

Figure 5: CNN Model Architecture

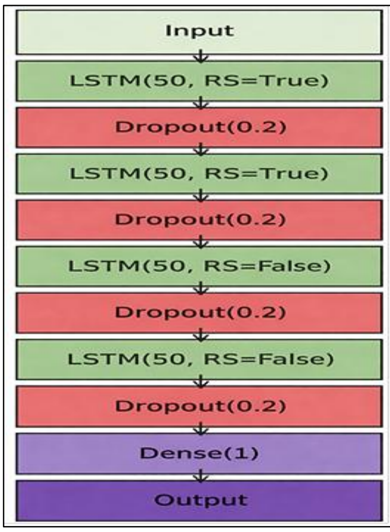

Figure 6: LSTM Model Architecture

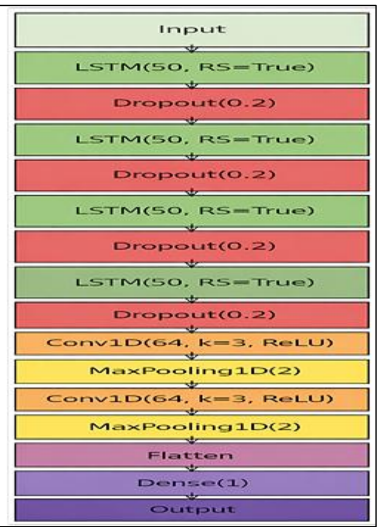

Figure 7: LSTM-CNN Model Architecture

Hyper parameter values were chosen based on prior studies and initial trial experiments, as summarized in Table 2. Dropout was introduced to reduce over fitting, and the Adam optimizer was used due to its stable performance on financial time-series data. A batch size of 64 provided a good balance between training stability and computational efficiency, while 100 training epochs were sufficient to capture temporal patterns without excessive model complexity. Early stopping and cross-validation were not employed to maintain consistent training settings across all models and to preserve the chronological structure of the data.

Table 2: Training Hyper Parameter Summary

Hyper Parameter	**Value**
Batch size	64
Epochs	100
Optimizer	Adam
Loss function	Mean Squared Error (MSE)
Dropout rate	0.2
Activation function (Conv1D)	ReLU
Pooling method	MaxPooling1D (pool size = 2)
Kernel size (Conv1D)	2 (CNN), 3 (LSTM–CNN)

Results and Discussion

The results are summarized using tables and figures. This research focuses on four models (LSTM, GRU, CNN, and LSTM-CNN) selected to forecast the stock prices of six companies. The dataset spanned from 1 January 2008 to 10 February 2025. The models were evaluated using three metrics: R-squared, RMSE, and MAE. Together, these metrics provide a practical basis for comparing the predictive performance of different models.

Table 3: RMSE Comparison of LSTM, GRU, CNN and LSTM–CNN Models

Name Of Company	Model Name			
	LSTM	GRU	CNN	LSTM-CNN
RELIANCE	49.53	328.13	293.81	266.85
GAIL	31.39	32.41	39.30	27.10
SBIN	61.39	136.14	114.63	98.06
IOC	27.80	25.39	20.60	34.10
WIPRO	37.41	59.08	65.97	64.37
TATASTEEL	17.00	18.94	17.18	23.21
AVERAGE	**37.42**	**100.02**	**91.91**	**85.62**

Table 3 summarizes the RMSE values obtained for each model across the evaluated companies. Among the models, LSTM achieved the lowest average RMSE (37.42), indicating superior predictive accuracy. The hybrid LSTM–CNN model ranked second with an average RMSE of 85.62. Overall, lower RMSE values reflect better alignment between predicted and actual stock prices.

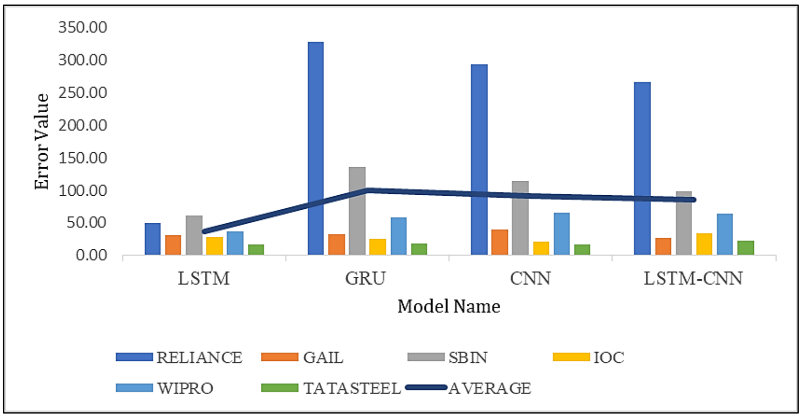

Figure 8: Comparison of RMSE Values for LSTM, GRU, CNN, and LSTM–CNN Models Across Six Companies

Figure 8 presents the RMSE comparison of four models across six companies. Overall, the LSTM model achieves the lowest RMSE, indicating superior predictive accuracy on average. However, performance varies at the individual stock level. For instance, CNN yields lower RMSE and MAE values for IOC and Tata Steel, suggesting that convolutional-based feature extraction is more effective for stocks with distinct price patterns. These findings reveal clear company-specific performance variations, indicating that no single model consistently dominated across all evaluated stocks. While CNN and the hybrid LSTM–CNN models also demonstrate competitive performance, LSTM remains the most robust model overall.

Table 4: MAE Comparison of LSTM, GRU, CNN, and LSTM–CNN Models

NAME OF COMPANY	MODEL NAME			
	LSTM	GRU	CNN	LSTM-CNN
RELIANCE	39.70	317.92	283.28	255.64
GAIL	24.46	23.84	33.07	20.26
SBIN	55.57	128.87	108.93	90.08
IOC	22.86	20.49	16.41	26.82
WIPRO	31.67	51.44	60.57	59.21
TATASTEEL	15.47	17.75	15.53	22.33
AVERAGE	**31.62**	**93.38**	**86.30**	**79.06**

Table 4 summarizes the Mean Absolute Error (MAE) obtained for each company across the evaluated models. Lower MAE values indicate better predictive accuracy. Among the tested approaches, the LSTM model achieved the lowest MAE of 31.62, followed by the hybrid LSTM–CNN model with an MAE of 79.06.

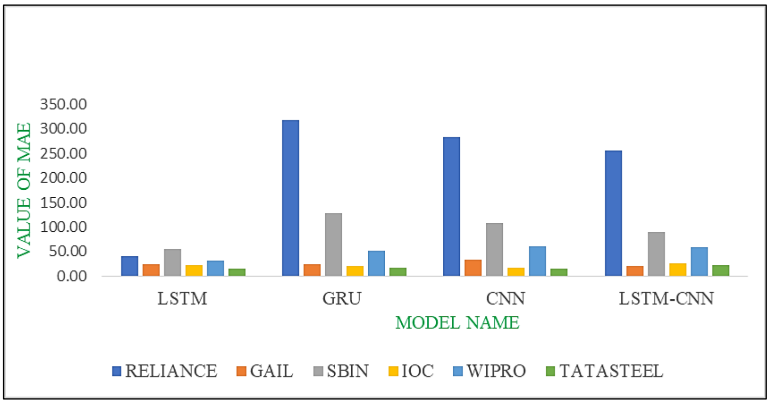

Figure 9: MAE Comparison of LSTM, GRU, CNN, and LSTM–CNN Models Across Companies

Figure 9 compares the mean absolute error (MAE) of four models across six companies. Among the evaluated methods, the LSTM model consistently records lower MAE values, with bars closest to zero for most stocks, indicating superior predictive accuracy. While the CNN and hybrid LSTM–CNN models also achieve reasonable performance, their errors remain higher than those of the standalone LSTM model.

It is also observed that average error metrics are affected by highly volatile stocks, particularly RELIANCE, which tend to inflate overall error values. As a result, company-wise performance analysis provides a more reliable basis for comparing model effectiveness than aggregate averages alone.

Table 5: R^2 Comparison of LSTM, GRU, CNN, and LSTM–CNN Models

NAME OF COMPANY	MODEL NAME			
	LSTM	GRU	CNN	LSTM-CNN
RELIANCE	0.88	-4.10	-3.09	-2.37
GAIL	0.60	0.57	0.37	0.70
SBIN	0.77	-0.15	0.18	0.40
IOC	0.49	0.58	0.72	0.24
WIPRO	-0.03	-1.57	-2.20	-2.05
TATASTEEL	0.43	0.30	0.42	-0.05
AVERAGE	**0.52**	**-0.73**	**-0.60**	**-0.52**

Table 5 reports the R^2 values obtained for all models across multiple datasets. Among the four deep learning models, LSTM achieves the highest R^2 value (0.52), indicating comparatively better explanatory power, followed by the hybrid LSTM–CNN model, while the remaining models show lower or negative values. Negative R^2 scores observed for some models indicate weak predictive performance on certain datasets, rather than computational errors. Hence, R^2 is interpreted in conjunction with RMSE and MAE to provide a more balanced evaluation of model performance.

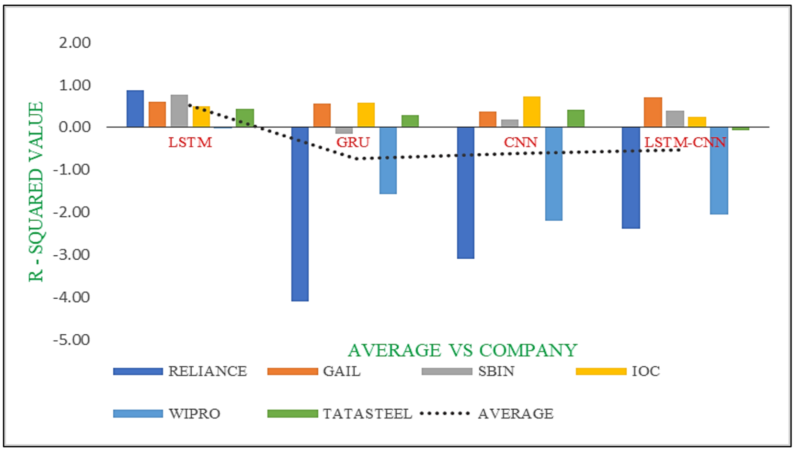

Figure 10: R^2 Performance Comparison of Deep Learning Models Across Six Companies

Figure 10 compares the R^2 performance of four models across six companies. The LSTM model consistently achieved higher R^2 values, often close to unity, while the hybrid LSTM–CNN followed as the second-best performer. No formal statistical significance tests or confidence intervals were applied; therefore, differences in RMSE, MAE, and R^2 should be interpreted as indicative rather than conclusive. The stronger performance of LSTM can be attributed to its ability to retain long-term temporal information through memory cells and gating mechanisms. Stock price movements are often shaped by extended trends and market cycles rather than short-term fluctuations alone. In comparison, GRU and CNN models showed weaker results for certain stocks, likely due to architectural constraints. GRU's simplified gating may limit long-term information retention in volatile markets, while CNN primarily captures local patterns without explicitly modelling long-range temporal dependencies. As a result, these models may struggle during regime changes or periods of high volatility.

Negative R^2 values observed for several models in Table 5 indicate that, for some stocks, prediction errors exceeded the variance of the target series, leading to performance below a mean-based baseline. This outcome reflects market noise and changing data patterns rather than numerical instability. Overall, LSTM achieved the best average performance (MAE = 31.62, R^2 = 0.52), although results varied across individual stocks, suggesting that model suitability is stock-specific.

Earlier research has examined deep learning models in conjunction with traditional econometric and machine learning approaches. These studies often focus on unstable market conditions and assess performance using a variety of evaluation measures (25-27).

Conclusion

This study evaluated the performance of LSTM, GRU, CNN, and hybrid LSTM–CNN models for stock price forecasting using data from six major Indian companies. Among the evaluated models, LSTM consistently produced the most reliable results, as reflected by RMSE, MAE, and R-squared metrics. While GRU and CNN models were able to capture complex data patterns and showed competitive performance under certain market conditions, the hybrid LSTM–CNN architecture demonstrated potential for improving prediction accuracy for selected stocks.

The results show that deep learning–based forecasting models can support the identification of unusual market behaviour and contribute to more inform financial decision-making. In particular, LSTM-based predictions show promise for applications related to portfolio analysis, such as portfolio rebalancing, sector-level risk management, and asset allocation. Future work may extend this study to portfolio-level applications and trading simulations. Recent studies suggest that reinforcement learning and graph-based approaches can further support portfolio optimization and cross-stock dependency modeling.

Abbreviations

CNN: Convolutional Neural Network, DL: Deep Learning, GRU: Gated Recurrent Unit, LSTM: Long Short-Term Memory, MAE: Mean Absolute Error, RMSE: Root Mean Square Error.

Acknowledgement

I am grateful for the guidance and support offered by my supervisor during this research journey. This manuscript has been submitted under communication number IU/R&D/2025-MCN0003956, as allotted by Integral University, Lucknow.

Authors Contributions

Umesh Pratap Singh: carried out the data collection, model implementation, experiments, and result analysis, prepared the manuscript draft, Manish Madhava Tripathi: supervised the research work, provided methodological guidance, reviewed the manuscript for technical accuracy and clarity. Both authors read and approved the final manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Declaration of Artificial Intelligence (AI) Assistance

The authors declare that limited AI-assisted tools were used only for language and grammar refinement. No AI tools were used for scientific content formation, data analysis or interpretation.

Ethics Approval

Not Applicable.

Funding

This research received no external funding.

References

1. Sahu SK, Mokhade A, Bokde ND. An overview of machine learning, deep learning, and reinforcement learning-based techniques in quantitative finance: Recent progress and challenges. Applied Sciences. 2023;13(3):1956.
2. Rezaei M, Nezamabadi-Pour HA taxonomy of literature reviews and experimental study of deep reinforcement learning in portfolio management. Artificial Intelligence Review. 2025;58(3):94-108.
3. Zhang C, Sjarif NNA, Ibrahim R. Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020-2022. WIREs Data Mining and Knowledge Discovery. 2023;14(4):e1519. https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1519
4. Audrino F, Sigrist F, Ballinari D. The impact of sentiment and attention measures on stock market volatility. International Journal of Forecasting. 2020;36(2):334-57.
5. Singh UP, Tripathi MM. A critical review of the effectiveness of machine learning and deep learning approaches in forecasting stock market trends. International Journal on Recent and Innovation Trends in Computing and Communication. 2023;11(9):3797-801.
6. Liu P, Zhang Y, Bao F, *et al*. Multi-type data fusion framework based on deep reinforcement learning for algorithmic trading. Applied Intelligence. 2023; 53(2):1683-706.
7. Hassan MA, Youssif AA, Imam O, *et al*. On the impact of news for reliable stock market predictions: an LSTM-based ensemble using FinBERT word-embeddings. WSEAS Transactions on Computers. 2022;21:294-303.
8. Fathali Z, Kodia Z, Ben Said L. Stock market prediction of Nifty 50 index applying machine learning techniques. Applied Artificial Intelligence. 2022;36(1):2111134.
9. Bhanja S, Das A. A Black Swan event-based hybrid model for Indian stock markets' trend prediction. Innovations in Systems and Software Engineering. 2024;20(2):121-35.
10. Bansal M, Goyal A, Choudhary A. A comparative analysis of k-nearest neighbour, genetic, support vector machine, decision tree, and long short-term memory algorithms in machine learning. Decision Analytics Journal. 2022;3:100071.
11. Guan B, Zhao C, Yuan X, *et al*. Price prediction in China stock market: an integrated method based on time series clustering and image feature extraction. The Journal of Supercomputing. 2024;80(7):8553-91.
12. Chiang JK, Chi R. A novel stock price prediction and trading methodology based on active learning surrogated with CycleGAN and deep learning and system engineering integration: A case study on TSMC stock data. FinTech. 2024;3(3):427-59.
13. Kalra R, Singh T, Mishra S, *et al*. An efficient hybrid approach for forecasting real-time stock market indices. Journal of King Saud University - Computer and Information Sciences. 2024; 36(8):102180. https://www.sciencedirect.com/science/article/pii/S1319157824002696
14. Gao Z. Enhancing Stock Market Forecasting: A hybrid machine learning approach integrating LSTM and GRU models. In: Proceedings of the 2024 International Conference on Cloud Computing and Big Data. 2024:254-9. https://dl.acm.org/doi/fullHtml/10.1145/3695080.3695125
15. Thakkar A, Chaudhari KA comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. Expert Systems with Applications: An International Journal. 2021;177(C):114800. https://dl.acm.org/doi/10.1016/j.eswa.2021.114800
16. Sezer OB, Gudelek MU, Ozbayoglu AM. Financial time series forecasting with deep learning: A systematic literature review: 2005-2019. Applied Soft Computing. 2020;90:106181.
17. Giantsidi S, Tarantola C. Deep learning for financial forecasting: a review of recent trends. International Review of Financial Analysis. 2025:104719.

https://www.sciencedirect.com/science/article/pii/S1059056025008822
18. Lim B, Zohren S. Time-series forecasting with deep learning: A survey. Philosophical Transactions of the Royal Society A. 2022;379(2214). https://royalsocietypublishing.org/rsta/article/379/2194/20200209/41189
19. Manjunath C, Marimuthu B, Ghosh B. Analysis of Nifty 50 index stock market trends using a hybrid machine learning model in quantum finance. International Journal of Electrical and Computer Engineering. 2023;13(3):3549-60.
20. Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research. 2018;270(2):654–669.
21. Dong J, Liang S. Hybrid CNN–LSTM–GNN neural network for A-share stock prediction. Entropy. 2025;27(8):881.
22. Shahi TB, Shrestha A, Neupane A, *et al*. Stock price forecasting with deep learning: A comparative study of LSTM and GRU models. Mathematics. 2020;8(9):1441. https://www.mdpi.com/2227-7390/8/9/1441
23. Orra A, Bhambu A, Choudhary H, *et al*. Dynamic reinforced ensemble using Bayesian optimization for stock trading. In: Proceedings of the 5th ACM International Conference on AI in Finance. 2024:361-9. https://dl.acm.org/doi/10.1145/3677052.3698595
24. Barua M, Kumar T, Raj K, Roy AM. Comparative analysis of deep learning models for stock price prediction in the Indian market. FinTech. 2024;3(4):551-68. https://www.mdpi.com/2674-1032/3/4/29
25. Nsengiyumva E, Mung'atu JK, Ruranga C. Hybrid GARCH-LSTM forecasting for foreign exchange risk. FinTech. 2025;4(2):22. https://www.mdpi.com/2674-1032/4/2/22
26. Zhong X, Enke D. Forecasting daily stock market return using dimensionality reduction and machine learning techniques. Expert Systems with Applications. 2017;67:126-39.
27. Sen J, Dutta A, Mehtab S. Stock portfolio optimization using a deep learning LSTM model. In: IEEE Mysore Sub Section International Conference. 2021:263-71. https://ieeexplore.ieee.org/abstract/document/9641662/?casa_token=R_lIoy_ymt0AAAAA:dcZEge3teVScQgkIlwu8pFtSbIquE06sGUFSqsms7x70G4-nfEI-pAKEX6vBLPZglI2DPi9PJpw

How to Cite: Singh UP, Tripathi MM. Comparative Evaluation of Deep Learning Models for Multi-Stock Prediction and Portfolio Optimization. Int Res J Multidiscip Scope. 2026; 7(1): 1175-1185.
DOI: 10.47857/irjms.2026.v07i01.07857