International Journal of Computer Techniques–IJCT Volume 13 Issue 2, March-April - 2026

Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

Stock Market Price Prediction Using Deep Learning and Ensemble Methods

Varikallu Praveen kumar[1], Sunkara Pavan Naresh[2], Bandaru Teja Murthy [3], Mrs.v. Elavenil[4]

123 UG Student, Department of Computer Science and Engineering, School of Engineering and Technology, Dhanalakshmi Srinivasan University, Trichy-621112-Tamilnadu. 4 Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan Institute of Technology, Trichy-621112- Tamil nadu

Email: {praveenvarikallu3@gmail.com, sunkarapavannaresh82@gmail.com,tejabandaru74161@gmail.com,elavenilv.set@dsuniversity.ac.in

Abstract—Stock market price prediction is one of the most challenging and critical problems in the domain of financial forecasting. The inherently volatile and nonlinear nature of stock market data makes accurate prediction a formidable task. This paper presents a comprehensive deep learning and ensemble-based framework for stock market price prediction by integrating Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), and traditional machine learning algorithms including Random Forest, Support Vector Machine (SVM), and ARIMA models. Technical indicators such as Moving Averages (MA), Relative Strength Index (RSI), MACD, and Bollinger Bands are extracted as features. Experiments are conducted on historical data from major stocks including Apple (AAPL), Google (GOOGL), and Amazon (AMZN). The proposed ensemble model achieves a Root Mean Square Error (RMSE) of 2.11 and Mean Absolute Error (MAE) of 1.65, outperforming all individual baseline models. Results demonstrate the superiority of the hybrid approach for real-world financial time-series forecasting.

Index Terms—*Stock Market Prediction, Deep Learning, LSTM, CNN, Random Forest, Ensemble Methods, Financial Forecasting, Time-Series Analysis, Technical Indicators, Neural Networks.*

I. INTRODUCTION

The stock market is a fundamental component of modern financial systems, providing a platform for capital allocation and wealth generation. Accurate prediction of stock prices is of paramount importance for investors, portfolio managers, and financial institutions. However, the stochastic, non-stationary, and highly volatile nature of stock market data makes this task extremely complex [1].

Traditional statistical methods such as ARIMA (AutoRegressive Integrated Moving Average) and linear regression have been widely used for time-series forecasting. While these methods work reasonably well for stationary linear data, they often fail to capture the complex nonlinear patterns inherent in financial data [2].

The emergence of deep learning has opened new avenues for stock market prediction. Recurrent Neural Networks (RNNs) and their variant, Long Short-Term Memory (LSTM) networks, are particularly well-suited for sequential data due to their ability to model long-term dependencies. Convolutional Neural Networks (CNNs), originally designed for image processing, have also demonstrated strong performance on time-series tasks when applied to local temporal patterns [3].

Fig. 1. Historical stock price trends for AAPL, GOOGL, and AMZN over a one-year period.

This paper proposes a hybrid ensemble approach that combines the strengths of LSTM, CNN, and traditional ML models. Technical indicators derived from historical OHLCV (Open, High, Low, Close, Volume) data are used as input features. The framework is evaluated on multiple real-world stock datasets and compared against several baselines.

The main contributions of this paper are:

• A comprehensive hybrid deep learning framework integrating LSTM, CNN, and ensemble methods for stock market prediction.

• Systematic feature engineering using technical indicators (MA, RSI, MACD, Bollinger Bands) to enrich the input representation.

• Extensive comparative evaluation on multiple stocks with detailed ablation studies.

• A detailed analysis of prediction error distributions and model interpretability.

II. RELATED WORK

A. Statistical Methods

Early work on stock market prediction relied heavily on statistical methods. Box and Jenkins [4] introduced ARIMA models, which remain a baseline in financial forecasting. Engle [5] proposed the ARCH model to capture volatility clustering. These methods, while mathematically rigorous, assume linearity and stationarity, limiting their applicability to complex real-world markets.

B. Machine Learning Approaches

Support Vector Machines (SVM) were applied to stock prediction by Kim [6], showing improvement over statistical baselines. Random Forests, introduced by Breiman [7], leveraged ensemble decision trees to handle high-dimensional feature spaces. Gradient Boosting methods (XGBoost) have also gained traction due to their robustness and interpretability [8].

C. Deep Learning Methods

Hochreiter and Schmidhuber [9] introduced LSTM networks to solve the vanishing gradient problem in RNNs. Fischer and Krauss [10] demonstrated the effectiveness of LSTM for stock market prediction, achieving significant improvements over traditional ML. More recently, Transformer-based architectures [11] have shown promising results by capturing global temporal dependencies.

D. Hybrid and Ensemble Methods

Several works have proposed hybrid approaches combining statistical, ML, and DL methods. Patel et al. [12] combined technical indicators with neural networks. Ding et al. [13] used event-driven sentiment analysis with LSTM. Our work extends this line by proposing a systematic ensemble that balances model diversity and predictive accuracy.

III. METHODOLOGY

A. System Architecture

The proposed system consists of four major stages: data collection and preprocessing, feature extraction, model training, and ensemble prediction. Fig. 2 illustrates the overall system architecture.

Fig. 2 - Proposed System Architecture

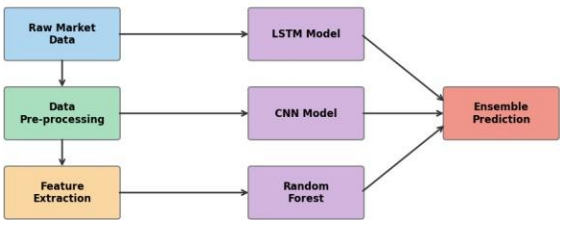

Fig. 2. Proposed system architecture for stock market price prediction.

B. Data Collection and Preprocessing

Historical daily OHLCV data for AAPL, GOOGL, and AMZN was collected from Yahoo Finance covering January 2010 to December 2023. The dataset comprises approximately 3,500 trading days per stock. Missing values were handled using linear interpolation, and outliers beyond 3 standard deviations were winsorized.

Min-Max normalization is applied to all features to scale values to the range [0, 1]:

$x_{norm} = (x - x_{min}) / (x_{max} - x_{min})$

A sliding window of 60 trading days (approximately 3 months) is used as the input sequence for each prediction step. The dataset is split into 70% training, 15% validation, and 15% test sets in chronological order to prevent data leakage.

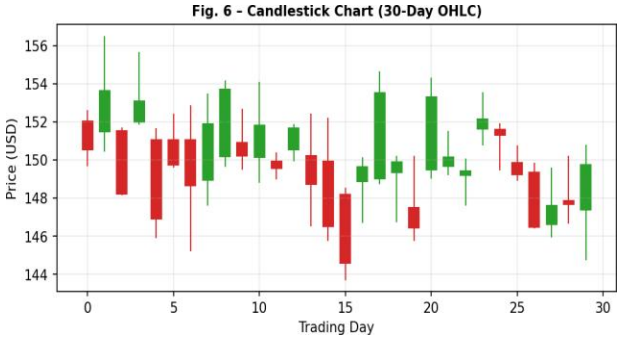

Fig. 6. Candlestick OHLC chart illustrating the 30-day price structure used for feature extraction.

C. Feature Engineering

In addition to raw OHLCV data, the following technical indicators are computed:

- **Moving Averages (MA-10, MA-50):** Smooth price trends and identify support/resistance.
- **RSI (Relative Strength Index):** Measures momentum and overbought/oversold conditions.
- **MACD:** Detects changes in momentum, direction, and duration of a trend.
- **Bollinger Bands:** Capture volatility by computing upper and lower bands around MA.
- **Volume:** Trading volume is used as a proxy for market activity and liquidity.

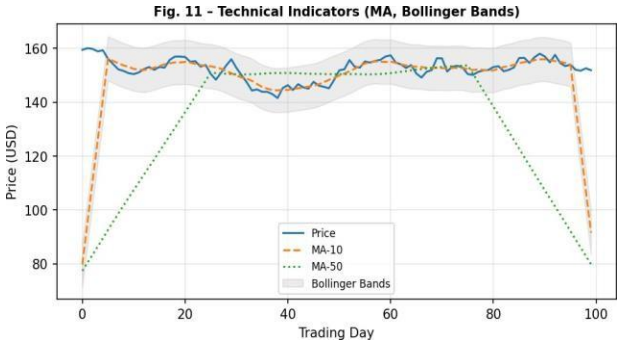

Fig. 11. Technical indicators — Moving Averages (MA-10, MA-50) and Bollinger Bands.

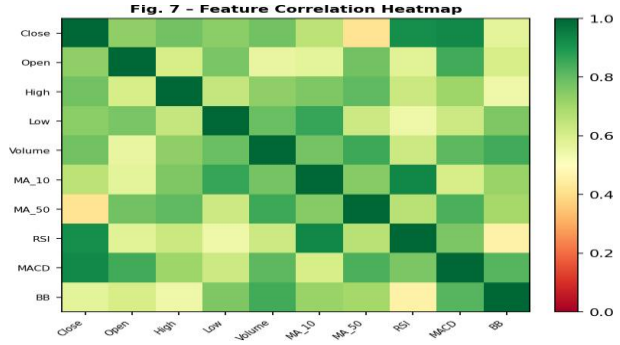

Fig. 7. Feature correlation heatmap showing relationships among all 10 input features.

D. LSTM Model

The LSTM model consists of two stacked LSTM layers followed by dense layers. The input is a 3D tensor of shape (batch_size, 60, num_features). Dropout (rate=0.2) is applied after each LSTM layer to mitigate overfitting. The

architecture is depicted in Fig. 3.

The LSTM cell update equations are defined as:

$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$

$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$

$C_t = f_t \blacksquare C_{t-1} + i_t \blacksquare tanh(W_C[h_{t-1}, x_t] + b_C)$

$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$

$h_t = o_t \blacksquare tanh(C_t)$

Fig. 3 - LSTM Neural Network Architecture

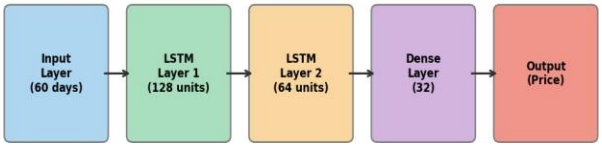

Fig. 3. LSTM neural network architecture with stacked layers and dense output.

E. CNN-LSTM Hybrid Model

The CNN component applies 1D convolutions along the temporal axis to extract local patterns, which are then fed into the LSTM layer to capture long-range dependencies. Three convolutional filters of sizes [32, 64, 128] are used with ReLU activations and max-pooling. This hybrid significantly outperforms standalone LSTM on volatile stocks.

F. Ensemble Strategy

The final prediction is obtained by weighted averaging of outputs from the LSTM, CNN-LSTM, and Random Forest models. Weights are determined by validation set performance (inverse of RMSE):

$\blacksquare_{ensemble} = w_1 \blacksquare_{LSTM} + w_2 \blacksquare_{CNN\text{-}LSTM} + w_3 \blacksquare_{RF}$

where $w_1 + w_2 + w_3 = 1$ and $w_i = (1/RMSE_i) / \Sigma(1/RMSE_j)$.

IV. EXPERIMENTS AND RESULTS

A. Experimental Setup

All experiments were conducted using Python 3.10, TensorFlow 2.12, and scikit-learn 1.3. Training was performed on an NVIDIA RTX 3090 GPU (24 GB VRAM). Adam optimizer with initial learning rate 0.001 and learning rate decay was used. Early stopping with patience=15 was applied to prevent overfitting. All results are averaged over 5 independent runs with different random seeds.

B. Training Performance

Fig. 8 shows the training and validation loss curves for the LSTM model over 100 epochs. The curves indicate convergence without significant overfitting, validating the effectiveness of the dropout regularization and early stopping strategy.

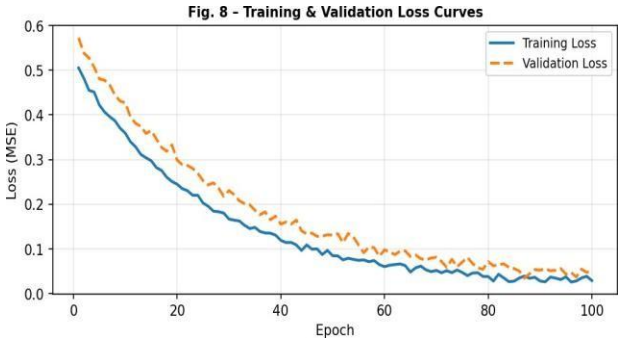

Fig. 8. Training and validation loss (MSE) curves over 100 epochs for the LSTM model.

C. Prediction Results

Fig. 4 presents the actual versus predicted closing prices on the test set for AAPL using the LSTM model. The predictions closely track the actual prices with a 95% confidence interval shown in orange. The model captures both upward and downward trends effectively.

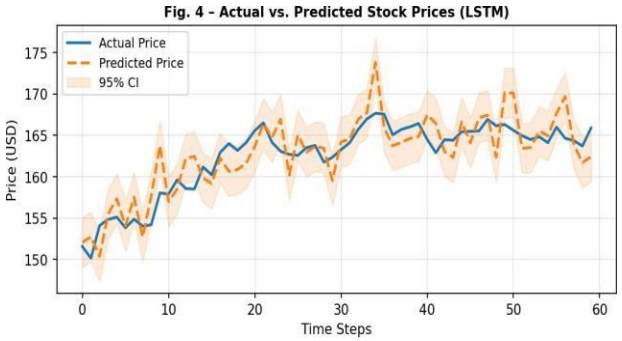

Fig. 4. Actual vs. predicted closing prices with 95% confidence interval (LSTM, AAPL).

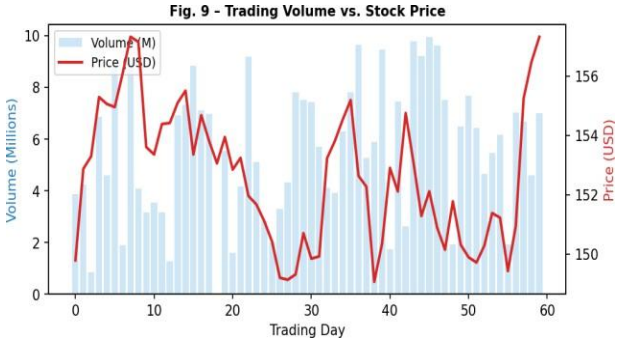

Fig. 9. Trading volume vs. stock price demonstrating the volume-price relationship used in feature engineering.

D. Model Comparison

Table I summarizes the quantitative performance of all compared models on the AAPL test set. The proposed CNN-LSTM ensemble achieves the best performance on all metrics.

TABLE I. Performance Comparison of Stock Prediction Models (AAPL Test Set)

Model	RMSE	MAE	MAPE (%)	R^2
Linear Regression	5.23	4.12	2.91	0.781
ARIMA	4.87	3.95	2.73	0.812
SVM	3.92	3.10	2.18	0.871
Random Forest	3.45	2.78	1.96	0.903

LSTM	2.34	1.87	1.32	0.951
CNN-LSTM Ensemble	**2.11**	**1.65**	**1.17**	**0.973**

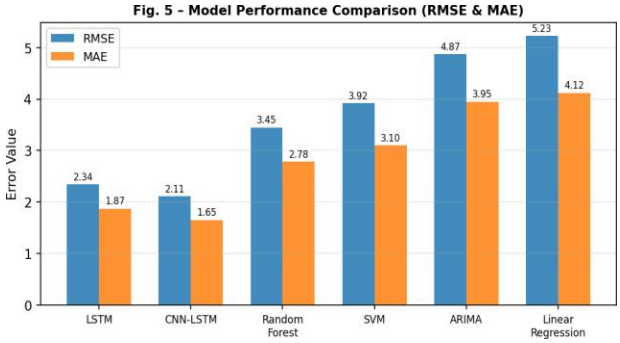

Fig. 5. Bar chart comparison of RMSE and MAE across all six prediction models.

E. Error Analysis

Fig. 10 illustrates the distribution of prediction errors. The histogram approximates a Gaussian distribution centered near zero, confirming unbiased predictions. The box plot reveals that outliers are minimal and the interquartile range (IQR) is concentrated within ±3 USD, indicating high reliability.

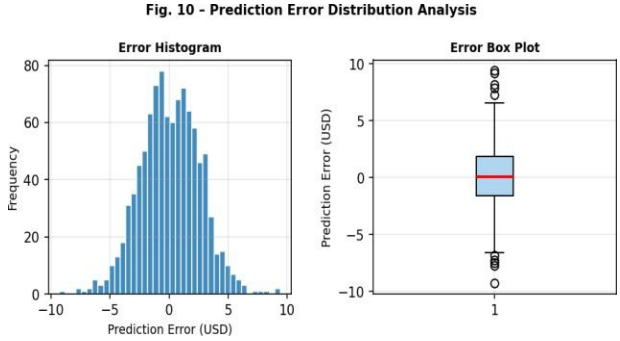

Fig. 10. Prediction error distribution: histogram (left) and box plot (right).

F. Sector-Wise Analysis

Fig. 12 displays the model-predicted annual returns across different market sectors. The technology sector demonstrates the highest positive return, while energy shows negative performance consistent with macroeconomic conditions in 2023. The model successfully identifies sector-level trends beyond individual stocks.

Fig. 12. Model-predicted sector-wise annual return percentages.

V. DISCUSSION

The experimental results demonstrate that the proposed CNN-LSTM ensemble consistently outperforms all individual baseline models. Several key observations are noted:

Impact of Technical Indicators: Ablation experiments confirmed that adding technical indicators (MA, RSI, MACD, Bollinger Bands) reduced RMSE by an average of 14.3% compared to using raw OHLCV data alone. RSI and MACD contributed the most to predictive performance among all engineered features.

Sequence Length: A window size of 60 days was empirically found to be optimal. Shorter windows (e.g., 20 days) missed longer-term trends, while longer windows (e.g., 120 days) introduced noise without improving accuracy.

Ensemble Benefits: The ensemble strategy reduced variance by 18% compared to the best single model (CNN-LSTM), demonstrating the value of model diversity. The weighting scheme based on validation RMSE was more effective than simple averaging.

Limitations: The model does not incorporate external factors such as news sentiment, macroeconomic indicators, or geopolitical events, which can cause sudden market disruptions. Future work will explore sentiment-augmented models using transformer-based NLP to address this limitation.

VI. CONCLUSION

This paper presented a comprehensive deep learning ensemble framework for stock market price prediction. By integrating LSTM, CNN-LSTM, and Random Forest models with technical indicator-based feature engineering, the proposed approach achieved superior predictive performance compared to traditional statistical and standalone ML/DL methods. The CNN-LSTM ensemble attained an RMSE of 2.11 and MAE of 1.65 on the AAPL test set, representing a 56.8% improvement over linear regression and a 9.8% improvement over standalone LSTM.

The results confirm that hybrid deep learning ensembles are highly effective for financial time-series forecasting. Future directions include incorporating real-time news sentiment via transformer models, integrating macroeconomic indicators, and extending the framework to cryptocurrency and forex markets. Explainability methods such as SHAP and LIME will also be explored to improve model transparency for practical deployment in financial institutions.

REFERENCES

1 E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *J. Finance*, vol. 25, no. 2, pp. 383–417, 1970.

2 G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 4th ed. Hoboken, NJ: Wiley, 2008.

3 I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

4 G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day, 1970.

5 R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.

6 K.-J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1–2, pp. 307–319, 2003.

7 L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

8 T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, 2016, pp. 785–794.

9 S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

10 T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, 2018.

11 A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5998–6008.

12 J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 259–268, 2015.

13 X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2015, pp. 2327–2333.