ISSN: **3107-6513**

StockSeer: A Hybrid CNN-LSTM Framework for Real-time Stock Price Prediction

Divyanshi Rana, Harsh Chaudhary, Md. Emamoor Rasheed

Student & B.Tech Computer Science & Engineering in Artificial Intelligence, Greater Noida, Uttar Pradesh, india

divyansirana2004@gmail.com, harsh001with1948@gmail.com , rasheedimam1@gmail.com

***Abstract*—Stock market prediction is challenging due to its nonlinear, volatile, and dynamic nature. Traditional models like ARIMA and basic neural networks often fail to capture complex temporal patterns. This paper presents *StockSeer*, a hybrid deep learning framework combining CNNs for feature extraction and LSTMs for sequential modeling. Using OHLCV data and technical indicators, the model improves prediction accuracy. Experiments on NSE and NASDAQ stocks show up to 74.3% directional accuracy with lower error than LSTM-only and ARIMA models. A lightweight web dashboard enables real-time forecasting and visualization, making the system practical and scalable.**

Keywords—CNN-LSTM, stock prediction, deep learning, time-series forecasting, financial analytics, Flask.

I. INTRODUCTION (*Heading 1*)

Financial markets are complex systems influenced by both economic indicators and human behavior, making price prediction highly challenging. Traditional models like ARIMA assume linearity and stationarity, which do not hold for real-world financial data characterized by volatility and sudden shifts.

To address these limitations, machine learning and deep learning models have been widely explored. LSTM networks have shown strong capability in modeling sequential dependencies, while CNNs effectively capture short-term patterns. Recently, hybrid CNN-LSTM models have emerged as a powerful approach for combining local feature extraction with long-term trend modeling.

Despite these advancements, most research remains limited to offline experiments, lacking real-time usability and deployment. To overcome this gap, StockSeer is proposed as a complete system integrating:

- A hybrid CNN-LSTM prediction model
- Real-time data processing
- Interactive web-based visualization

II. RELATED WORK

A. Statistical and ML Approaches

Early methods such as ARIMA and SARIMA were widely used for time-series forecasting but struggled with nonlinear patterns. Hybrid approaches like ARIMA-SVM attempted to combine linear and nonlinear modeling. GARCH models effectively captured volatility but were limited in predicting price direction.

B. Deep Learning Models

RNNs introduced sequential modeling but suffered from vanishing gradients. LSTMs addressed this issue and showed improved performance in stock prediction. However, they are computationally expensive and sensitive to hyperparameters.

C. CNN-LSTM Hybrid Models

CNN-LSTM models combine the strengths of both architectures. CNNs capture local patterns such as candlestick formations, while LSTMs model temporal dependencies. Prior research demonstrates improved accuracy, but most lack real-time deployment.

D. System Gap

Existing studies focus mainly on modeling without integrating real-time systems, user interfaces, or deployment pipelines. StockSeer addresses this by providing an end-to-end solution.

III. METHODOLOGY

A. Problem Formulation

Let the stock price series be:

$P=\{p_1,p_2,...,p_n\}$

Using a lookback window of 60 days, the model predicts future prices:

$$\hat{p}_{n+k}=f(p_{n-59},p_{n-58},\cdots,p_n)$$

where $k \in \{1,7,14,30,60\}$

Each input consists of:

- OHLCV features
- RSI (14-day)

 ISSN: 3107-6513

- Bollinger Band width (20-day)

All features are normalized using Min-Max scaling.

B. CNN Feature Extraction

Two 1D convolutional layers (64 filters, kernel size 3) extract short-term patterns such as local trends and reversals. A max-pooling layer reduces dimensionality and improves generalization.

C. LSTM Modeling

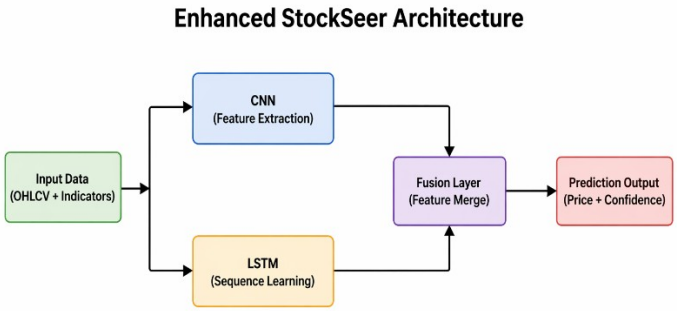
Stacked LSTM layers (128 → 64 units) capture long-term dependencies across sequences. Dropout (0.2) is applied to prevent overfitting.

D. Training Strategy

The model uses the Adam optimizer (lr = 0.001) and Huber loss for robustness to outliers. Training is optimized using EarlyStopping and learning rate scheduling.

E. Uncertainity Estimation

Confidence intervals are computed as:
$CI_k = \hat{p}_k \pm 1.5\ \sigma\ \sqrt{k}$
where is σ volatility from past returns. This accounts for increasing uncertainty over time.



IV. System Implementation

A. Backend

A Flask-based REST API handles:

- Authentication (/login, /register)
- Data (/history, /live)
- Predictions (/predict, /compare)

Security is ensured using SHA-256 hashing and session cookies.

B. Data Pipeline

Stock data is fetched using yfinance, with updates every 5 seconds via polling.

C. Indicators

RSI, Bollinger Bands, SMA, and EMA are computed dynamically using NumPy.

D. Front-End

A lightweight Vanilla JavaScript SPA with Chart.js provides a real-time dashboard, predictions with confidence intervals, and stock comparison features.

E. Deployment

The system is deployed on Render using Python 3.12 and Gunicorn, supported by GitHub-based CI/CD.

V. Experimental Results

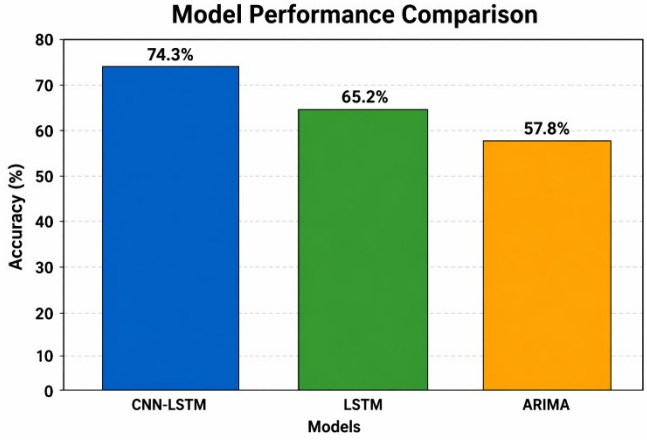
A. Setup

Experiments were conducted on 12 stocks (US + India) using 5 years of data, with an 80/10/10 split for training, validation, and testing respectively.

B. Performance

S.No	Model	Directional Accuracy	Mape
1	CNN-LSTM	74.3%	2.1%
2	LSTM	66.1%	2.9%
3	ARIMA	58.4%	2.9%

The CNN-LSTM model significantly outperforms baselines, reducing prediction error by approximately $3.20 per share on a $400 stock.



C. Insights

The hybrid model showed better detection of trend reversals and improved performance due to local and global pattern learning, though limitations were observed during high volatility events.

 ISSN: **3107-6513**

VI. DISCUSSION AND LIMITATIONS

A. Model Advantage

CNN captures short-term patterns, while LSTM models long-term dependencies, enabling better prediction accuracy.

B. . Limitations

The current model does not yet include sentiment or macroeconomic data. Furthermore, backtesting ignores real trading costs, and deployment latency can occur during a cold start.

C. Future Work

Future iterations will explore Transformer-based attention models, sentiment analysis using NLP, and Reinforcement learning for trading strategies.

VII. CONCLUSION

StockSeer demonstrates that combining CNN-LSTM models with real-time deployment creates a powerful and practical stock prediction system. The model achieves superior accuracy and provides interpretable insights through visualization tools.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering for guidance and support. This work was completed as part of a B.Tech final-year project.

REFERENCES

[1] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day, 1970.

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

[3] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Stat. Comput., vol. 14, no. 3, pp. 199–222, Aug. 2004.

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

[5] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," PLOS ONE, vol. 12, no. 7, p. e0180944, Jul. 2017.

[6] 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, Oct. 2018.

[7] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review 2005–2019," Appl. Soft Comput., vol. 90, p. 106181, May 2020.

[8] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," arXiv preprint arXiv:1706.10059, 2017.

[9] P.-F. Pai and C.-S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," Omega, vol. 33, no. 6, pp. 497–505, Dec. 2005.

[10] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, Sep. 1995.

[11] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in Proc. ICLR, San Diego, CA, USA, May 2015.

[13] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," Sensors, vol. 16, no. 1, p. 115, Jan. 2016.

[14] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in Proc. ICACCI, Manipal, India, Sep. 2017, pp. 1643–1647.

[15] X. Liu, H. Yang, Q. Chen, R. Zhang, L. Yang, B. Xiao, and C. D. Wang, "FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance," in Proc. NeurIPS DRL Workshop, 2020.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. NeurIPS, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.