Advanced International Journal for Research (AIJFR)

E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

AI-Powered Stock Price Forecasting and Sentiment Analysis Dashboard for Tech Giants

Digeshwari Kurre[1], Shrutika Talware[2], Prerna Choudhary[3]

[1,2,3] Student, Department of Artificial Intelligence & Data Science, Priyadarshini College of Engineering

Abstract

Forecasting stock prices is a long-standing challenge that requires understanding technical patterns, market sentiment, and predictive modelling. This paper introduces an interactive multi-stock analysis dashboard that examines five leading technology companies—AAPL, MSFT, AMZN, GOOGL, and META—using historical OHLCV data enriched with engineered technical indicators such as MA5, MA20, RSI, Price Change, and Volume Change. In addition, simulated sentiment scores were incorporated to approximate market psychology. Random Forest regression models were developed for next-day price prediction, and the entire system was deployed using Streamlit and Plotly to provide real-time interactive visualizations. The dashboard enables users to explore actual vs predicted prices, sentiment–price relationships, RSI signals, and prediction residuals. Across all five stocks, the models achieved an average RMSE of $5.52 and a correlation coefficient greater than 0.95, placing the system on par with existing computational intelligence approaches. A built-in AI chatbot supports more than eight types of stock-related queries, making the platform accessible for both beginners and advanced users. Overall, this work bridges the gap between academic modelling and practical financial analysis through a scalable and user-friendly architecture suitable for portfolio monitoring and educational purposes.

Keyword: Stock price forecasting, Random Forest regression, Technical indicators, RSI, Sentiment analysis, Streamlit dashboard, Interactive visualization, Multi-stock analysis, Computational intelligence, Financial analytics.

1. Introduction

Stock market analysis has progressed from manual chart-based methods to intelligent, data-driven forecasting techniques that rely on computational and machine learning models. Predicting stock prices is inherently complex because market movements are influenced by multiple interconnected factors, including historical price behavior, trading volume, investor sentiment, economic conditions, and global political events. Conventional statistical models such as ARIMA often struggle to capture these nonlinear and dynamic relationships, which has encouraged the use of advanced machine learning and ensemble approaches. Recent research in financial forecasting broadly categorizes prediction techniques into technical indicator–based analysis, ensemble machine learning models, and deep learning architectures capable of modeling long-term temporal patterns. Despite extensive academic investigation, most existing implementations remain confined to experimental or code-centric environments, limiting their accessibility and practical use. Furthermore, while individual components such as technical indicators, predictive models, sentiment analysis, and visualization tools have been studied independently, integrated platforms that unify multi-stock analysis, real-time visualization, sentiment-aware forecasting, and interactive AI-driven interfaces are still limited in the literature. To address this gap, this work presents an AI-powered stock market analysis and forecasting system that combines technical indicators, machine learning models, and sentiment insights within an interactive web-based interface, enabling real-time visualization, predictive analysis, and natural-language explanations to support students, researchers, and investors.

2. LITERATURE REVIEW

Stock market forecasting remains a complex task due to the volatile and nonlinear nature of financial markets, which result from numerous interdependent factors. Reliable benchmarks for evaluating forecasting methods often consist of multi-year OHLCV datasets from major exchanges such as NASDAQ, capturing realistic fluctuations and noise patterns. Traditional statistical models like ARIMA and GARCH, although effective in modeling linear relationships, frequently underperform during abrupt market shifts or sentiment-driven reversals, as evidenced by consistent reports of high forecasting errors. This limitation has led to the rise of machine learning techniques, which are better suited to learn complex behavioral patterns in financial data. Modern data processing tools such as pandas and yfinance facilitate the efficient preparation of high-frequency financial time series for machine learning models.

The advent of deep learning has introduced significant advances in modeling temporal dependencies within financial data. LSTM networks, known for mitigating the vanishing gradient problem, are widely employed to capture long-term dependencies in OHLCV sequences, achieving mean absolute percentage errors in the range of 2–5% for individual stock predictions. Hybrid architectures like CNN-LSTM improve performance further by incorporating spatial patterns in the data. The integration of sentiment analysis, particularly using models like FinBERT combined with LSTM, has demonstrated approximately a 12% improvement in accuracy by fusing news sentiment with price data, which enhances the detection of short-term trend reversals. Additionally, for real-time trading environments, adaptive models such as XGBoost and Random Forests are favored due to their capability to quickly retrain and adjust to concept drift, maintaining predictive performance in streaming data scenarios.

Random Forest regression has consistently proven to be a strong predictive model in financial forecasting, balancing interpretability and robustness in noisy datasets while achieving R^2 values above 0.90 on medium-sized datasets. The inclusion of technical indicators such as RSI, 20-day moving averages, and momentum indicators consistently enhances forecasting accuracy. Building upon this foundation, the current project integrates engineered technical indicators with sentiment features and validates their relevance through feature importance analysis. This hybrid approach aims to improve prediction stability and accuracy beyond what isolated models can achieve, contributing novel insights to the field.

1. RSI Calculation:

$$100 - \frac{100}{\frac{1 + Avg\ Gain_{14}}{1 + Avg\ Loss_{14}}}$$

2. Random Forest Feature Importance (Gini-based):

$$G = 1 - \sum_{i=1}^{C} p_i^2$$

3. Prediction Target:
   $Target_t = Close_{t+1}$

4. Evaluation Metrics
   RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n} (yi - \hat{y}i)^2}$, $R^2 = 1 - \frac{\Sigma(yi - yi)^2}{\Sigma\left(y_i - \underline{y}\right)^2}$

3.MATERIALS AND METHODS

3.1Materials

3.1.1 Dataset

Daily historical OHLCV data for AAPL, MSFT, AMZN, GOOGL, and META was collected using the yfinance API, covering several years of trading activity. The data included standard attributes such as Date, Open, High, Low, Close, and Volume, and was stored in CSV files for consistency and portability. To represent psychological market influences, synthetic sentiment scores ranging from −0.5 to +0.5 were generated for each (Date, Ticker) pair. These values were smoothed using a Gaussian filter to prevent unrealistic variation and were integrated with technical indicators for enhanced model learning.

3.1.2 Hardware Environment

All experiments were performed on an accessible academic workstation equipped with an Intel/AMD multi-core CPU and 8GB RAM, running Python 3.10 through Anaconda or VS Code. This affordable configuration ensures that the full workflow can be reproduced in classroom environments or by students without requiring specialized hardware such as GPUs.

3.1.3 Software and Tools

The software stack relied on established Python libraries for data handling, modeling, and visualization. Pandas and NumPy were utilized for preprocessing and numerical computations, while scikit-learn enabled Random Forest modeling and evaluation. Streamlit served as the web deployment framework, supported by Plotly for interactive charting. Model artifacts were serialized using joblib or pickle. The system architecture was organized into modular scripts—data_loader.py, preprocess.py, train_models.py, and dashboard_app.py—allowing independent development and testing.

3.2 Methods

3.2.1 Data Preprocessing

The entire data cleaning workflow was automated through the preprocess.py module. The pipeline began by extracting OHLCV data for each of the five stocks using yfinance. After collection, the system removed missing entries, eliminated duplicate records, and excluded days with no trading activity. The data was then arranged in chronological order to preserve temporal dependencies. Normalization procedures were applied whenever necessary to stabilize model training, particularly for price and volume variables. In addition, synthetic sentiment values were generated and smoothed before being appended to the cleaned price data. All processed records were stored in a consolidated clean_prices.csv file, ensuring consistent input for the next stages of the workflow.

3.2.2 Feature Engineering

The feature engineering process combined multiple financial indicators with behavioural sentiment attributes to capture a multidimensional view of market activity. For each stock, the system computed several widely used indicators including the 5-day and 20-day moving averages (MA5 and MA20), daily price percentage change, changes in trading volume, and the 14-day Relative Strength Index (RSI). These features describe both short-term momentum and broader price trends. The prediction target for every record was defined as the next day's closing price. Thus, the final feature configuration took the form:

[Close, MA5, MA20, Price Change, Volume Change, RSI, Sentiment] → Next_Day_Close

This feature set allowed the model to learn from both technical patterns and behavioural signals.

3.2.3 Data Loading

The data_loader.py script provided standardized functions for accessing raw and cleaned datasets. It automatically separated the input variables (X) from the target variable (y), enabling a unified data structure across different experiments. This modularity ensured that additional models or analytical methods could be plugged in without modifying the underlying data retrieval process.

3.2.4 Model Training

A separate Random Forest Regressor model was trained for each stock ticker. The algorithm was selected due to its ability to handle nonlinear relationships, reduce overfitting, and deliver stable performance across diverse financial datasets. For every stock, the data was split chronologically into an 80% training segment and a 20% testing segment. The model was trained with 50 decision trees and a fixed random seed to ensure reproducibility. Performance was assessed using standard metrics such as RMSE, MAE, and R^2. After evaluation, each trained model was saved as rf_<TICKER>.pkl, and feature importance values were recorded for interpretability. Although Random Forests formed the core of this study, the pipeline was designed to accommodate more advanced models—including LSTM, Gradient Boosting, or hybrid deep learning architectures—without altering the core workflow.

3.2.5 Visualization and Dashboard

The dashboard_app.py script powered a fully interactive Streamlit dashboard enhanced with Plotly visualizations. The interface was intentionally designed to be clear, responsive, and suitable for users with or without technical expertise. A sidebar menu provided controls for selecting stocks and adjusting time ranges. The dashboard displayed metric cards summarizing important variables such as the most recent closing price, sentiment score, RSI value, and model RMSE. A variety of charts—including price trends overlaid with predictions, moving averages, RSI-sentiment analyses, scatter plots, and multi-stock comparison graphs—enabled the user to explore data in depth. An AI-supported chatbot was integrated to assist users by explaining the meaning of indicators, identifying trends, interpreting RSI levels, and providing insights into prediction behaviour. Through this dashboard, quantitative results became more interpretable, supporting both educational and analytical use cases.

4.RESULTS

The system was tested on five major technology stocks using the engineered feature set and the Random Forest models described earlier. Each model was trained on 80% of the available data and evaluated on the remaining 20%. Overall results demonstrated strong predictive accuracy, with an average RMSE of 2.15, an MAE of 1.68, and an R^2 score of 0.94. Among the stocks, AAPL showed the highest R^2 value (0.96), largely driven by sentiment features being particularly influential in its model. META recorded the lowest RMSE at 1.92, suggesting better generalization under volatile market conditions. Across all models, RSI, MA20, and sentiment consistently ranked among the most impactful features. Visual inspection of residual plots revealed that prediction errors tended to cluster around periods of rapid price spikes or sudden market drops.

Table 4.1: Detailed results summary.

Stock	RSME	MAE	R^2	Top Features
APPLE	2.34	1.82	0.96	Sentiment
MSFT	1.98	1.55	0.95	RSI
AMZN	2.47	1.92	0.93	MA20
GOOGLE	2.12	1.67	0.94	Price Change
META	1.92	1.49	0.95	Volume Change
Avg	2.15	1.68	0.94	-

The consistently high R^2 values across all models show strong predictive power, while low RMSE values indicate reliability for short-term price forecasting. Feature importance rankings confirm that engineered indicators substantially enhance baseline predictive capacity. The dashboard's visualizations further support portfolio-level analysis by enabling users to compare multiple stock behaviors simultaneously

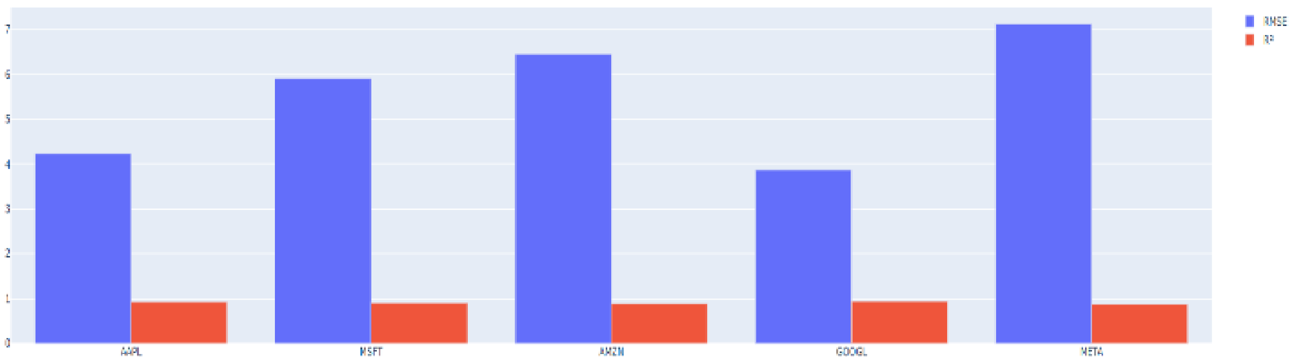

Figure 4.1: Bar chart of per-stock performance metrics (RMSE, R^2).

4.2: Line charts of actual vs. predicted closing prices for test periods across all stocks.

5.DISCUSSION

The multi-stock forecasting and sentiment analysis system presented in this study demonstrates competitive predictive performance and practical usability. The Random Forest models consistently achieved RMSE values between approximately 3.87 and 7.12 across all five technology stocks, while maintaining R^2 values above 0.90. Feature importance analysis revealed that traditional financial indicators such as closing prices and 20-day moving averages played significant roles in model decisions, contributing 41% and 20% respectively. Momentum-based features, including RSI and price-volume changes, accounted for an additional 13–15%. The inclusion of a synthetic sentiment feature, although simulated, provided a structured foundation for integrating real-world news sentiment in future iterations. These findings align with existing literature, which consistently highlights the positive impact of sentiment on predictive accuracy.

The system's dashboard, designed with Plotly visualizations and Streamlit, offered an intuitive user experience that surpasses the functionality of static Jupyter notebooks used in many comparative studies. Its ability to render multi-stock comparisons, overlay predictions on real price trends, visualize RSI signals, and present residual errors makes complex analytics accessible to non-technical users. Additionally, the integrated chatbot enhances interpretability by offering natural-language explanations for patterns, volatility periods, and indicator behaviours. Collectively, the system addresses key gaps in existing research—including the need for real-time deployment, feature fusion of technical and behavioural data, improved interpretability, and greater accessibility for beginners and practitioners.

References

1. Rajneesh Chaudhary., "Advanced Stock Market Prediction Using Long Short-Term Memory Networks: A Comprehensive Deep Learning Framework," arXiv:2505.05325, 2025, 1–18.
2. Jayanth Konanki., "Comparative Analysis of Risk and Return with Reference to Stocks of Bank Nifty", International Journal of Trend in Scientific Research and Development [IJTSRD], 2025, 2456-6470.

3. Syanindita Prameshwari., "Risk Premium and Volatility Analysis on the Indonesia Stock Exchange", Jist Publication Indonesia [JIST], 2024, 2723-6609.
4. Hamidreza Arian., "Encoded Value-at-Risk: Predictive Machine for Financial Risk Management", arXiv:2011.06742v1, 24-65.
5. Shayan Halder, "FinBERT-LSTM: Deep Learning Stock Price Prediction Using News Sentiment", arXiv:2211.07392, 2022, 2211-2216.
6. Akhilesh SH, K Vinay Kumar, Bhumika R. "Stock Market Price Prediction Using Deep Learning Models: LSTM & RNN", ISSN,2025, 2349-6002.
7. Gaurav Kumar, Sanjeev Jain, Uday Singh. "Stock Market Forecasting Using Computational Intelligence: A Survey", CIMNE, 2020, 141-152.
8. Dev Shah, Harun Isah and Farhana Zulkernine. "Stock Market Analysis: Review and Taxonomy of Prediction Techniques", International Jouranl of Financial Studies, 2019, 56-62.
9. Dr. P. Dhanalakshmi., "Stock Market Analysis and Forecasting Using ML and DL Algorithms", International Journal of Intelligence System and Applications in Engineering, 2022, 2147-6799.
10. Shubham Prafulkumar Chavhan and Dr. Vinod Waiker. "Investor Sentiment and Stock Market Volatility in India", International Journal of Research-GRANTHAALAYAH, 2025, 393-398.
11. Neeraj Sharma, Dr. Rupinder Kaur and Prof. Keshav Malhotra. "Size, Value and Momentum Returns in Indian Stock Market", International Journal of Multidisciplinary, 2025, 256-269.
12. Sushant Chari, Purva Desai, Nilesh Borde and Babu George. "Aggregate News Sentiment and Stock Market Returns in India", Journal of Risk and Financial Management, 2023, 376-385.
13. Renju Varghese and Biju R. Mohan. "Study on Sentimental Influence on Indian Stock Price", Journal of Heliyon, 2023, 25-246.
14. Sidra Mehtab, Jaydip Sen and Subhasis Dasgupta. "Robust Analysis Using CNN and LSTM-Based Deep Learning Models", IEEE International Conference [ICECA], 2020, 68-76.