Finsight: AI Based Stock Market Trend Prediction Using LSTM With Sentiment Analysis

Aditya Chauhan[1], Vivek Bandi[2], Shreyash Gupta[3], Kunal Bensla[4], Dr. Dnyaneshwar Bavkar[5]
[1,2,3,4]*Department of Computer Engineering, MGM's College of Engineering and Technology, Kamothe, Navi Mumbai*
[5]*Associate Professor, Department of Computer Engineering, MGM's College of Engineering and Technology, Kamothe, Navi Mumbai*

***Abstract*—Stock markets are hard to predict because prices change due to many factors at once, and no single model works well for all kinds of stocks. In this paper we present FinSight, a hybrid prediction system that combines Long Short-Term Memory (LSTM) networks with XGBoost to forecast the next-day closing price for NIFTY50 stocks. One of the key ideas in our system is that we separate stocks into two groups stable and volatile based on their recent price behaviour, and we train a separate model for each group. The LSTM part learns patterns from historical price sequences, and then XGBoost takes those learned features and produces the final price prediction. We also built a small pipeline that automatically pulls the latest data from Yahoo Finance so the system always works on fresh data. We tested FinSight on twenty NIFTY50 stocks over ten years of data and got an average R^2 of 0.93, RMSE of 97.4, MAPE of 2.11%, and Directional Accuracy of 71.3%. These results are better than standalone LSTM, BiLSTM, CNN–LSTM, and classical statistical models on all four metrics.**

***Index Terms*—Stock Market Prediction, LSTM, XGBoost, Hybrid Deep Learning, Volatility Modelling, NIFTY50, Feature Engineering, Gradient Boosting, Financial Time Series.**

I. INTRODUCTION

Predicting stock prices has always been a difficult problem, and researchers have been working on it for decades without finding a perfect solution. Markets move because of many things happening at the same time, economic news, company performance, investor sentiment, global events, and more. The NIFTY50, which tracks fifty of the largest and most traded Indian stocks across different sectors, is especially tricky to predict because it reacts strongly to both domestic policy changes and international market movements.

Older methods like ARIMA assume that the data is stationary and that relationships are linear, which is rarely true in real markets. Machine learning models like SVR and Random Forests handle nonlinearity better, but they do not have a built-in way to remember patterns over time, which is a problem for sequential data like stock prices. LSTM networks were designed to solve exactly this; they can learn from long sequences and remember important patterns over many time steps. This makes them a much better fit for financial time series compared to classical approaches.

One issue we noticed while studying existing prediction systems is that they usually train one model for all stocks together. But different stocks behave very differently, some are quite stable while others swing wildly. Mixing them all into one model hurts accuracy because the model has to compromise between very different patterns. In FinSight, we handle this by first checking how volatile a stock has been recently, and then using a model that was specifically trained on similar stocks.

We also found that XGBoost works very well when given a good set of numerical features. While LSTM is great at learning from sequences, XGBoost can pick up on complex relationships between those features that LSTM alone might miss. By combining both using LSTM to extract temporal features and XGBoost to make the final prediction we get better results than either model would give on its own.

The main contributions of this work are:

1. A volatility-aware dual-model architecture that sends each stock to a dedicated model depending on how volatile it has been recently.

2. A hybrid LSTM–XGBoost pipeline where LSTM learns time-based features from historical data and XGBoost uses those features for the final price prediction.

3. A rich technical indicator feature set including trend, momentum, and volatility indicators such as MA50, MA100, MA200, RSI, MACD, and Bollinger Bands.

4. An incremental real-time data pipeline that pulls new data from Yahoo Finance and adds it to the existing dataset, so the system always stays up to date without downloading everything from scratch.

II. RELATED WORK

A. Statistical and Classical ML Models

The earliest work on stock prediction used statistical models like ARIMA, VAR, and GARCH. These are well-understood and easy to interpret, but they assume that prices follow linear and stationary patterns, which usually does not hold in real markets. Later, researchers tried machine learning models like Support Vector Regression and Random Forests, which can capture nonlinear patterns. However, these models still need manual feature selection and do not have any built-in mechanism to model how prices change over time.

B. Recurrent Neural Networks and LSTM

Hochreiter and Schmidhuber introduced LSTM to fix the vanishing gradient problem that standard RNNs suffered from. With LSTM, models can learn patterns that span hundreds of time steps, which is very useful for stock data. Several studies have shown that LSTM outperforms both ARIMA and SVR on financial prediction tasks [3, 8]. Fischer and Krauss used LSTM on S&P 500 stocks and got significantly better results than a random walk baseline [8]. Extensions like BiLSTM, which reads sequences in both directions, and attention-based models have pushed accuracy even further [20].

C. Hybrid and Ensemble Architectures

Combining deep learning with tree-based models has become an interesting research direction in recent years [6]. XGBoost, introduced by Chen and Guestrin, is known to perform very well on tabular data and can model complex interactions between features. Some papers have shown that taking the internal representations from an LSTM and feeding them into XGBoost gives better predictions than using LSTM alone. Gu et al. [1] combined a FinBERT text model with LSTM for sentiment-based prediction and got a MAPE of 0.045 on NASDAQ-100, which shows that adding more information sources can improve results.

D. Volatility-Adaptive Modelling

Some researchers have used regime-switching models like Hidden Markov Models and Markov-switching GARCH to handle different market conditions [7]. These models try to detect which regime the market is currently in and apply a different model accordingly. The problem is that they are usually complex and need extra steps to estimate hidden states. In FinSight, we keep this idea simple we just compute the standard deviation of recent log returns, compare it to a fixed threshold, and decide which model to use. This is fast, easy to implement, and does not need any extra inference step.

III. PROPOSED METHODOLOGY

A. Data Acquisition and Incremental Pipeline

We collected ten years of daily OHLCV (Open, High, Low, Close, Volume) data for all fifty NIFTY50 stocks and stored them as separate CSV files, one per ticker. The data goes up to February 2026. When a user queries a particular stock, the system first checks the last date available in the stored file, then downloads only the missing days from Yahoo Finance using the yfinance library [2]. The new data is appended, duplicates are removed, and the file is sorted by date. This way we avoid downloading everything again each time and always work with the latest available data.

B. Feature Engineering

Once the data is ready, we compute a set of technical indicators on top of the raw price data. These include moving averages for 50, 100, and 200 days (MA50, MA100, MA200) which show the overall trend; the

14-day RSI which measures momentum; the MACD and its signal line which help detect trend changes; and Bollinger Bands which capture how much the price is deviating from its recent average. Together, these features give the model information about trend direction, momentum strength, and local volatility, which are all important for predicting future price movement.

C. Data Preprocessing

Before feeding data to the model, we normalise all features to a range of 0 to 1 using Min–Max scaling. The scaling parameters are computed only from the training data so that test data information never leaks into the training process:

$$X_{norm} = (X - X_{min}) / (X_{max} - X_{min}) \quad (1)$$

After normalisation, we create input sequences of 60 consecutive trading days. Each sequence contains all the computed features for those 60 days, and the target label is the normalised closing price on day 61. So, each training sample has shape (60, F) where F is the number of features, and during prediction we pass one sequence of shape (1, 60, F) to get the next-day price estimate.

IV. SYSTEM ARCHITECTURE

The overall FinSight system is built as five separate stages that work, as shown in Figure 1. These are: data acquisition, feature engineering, volatility classification, model selection, and the final hybrid prediction.

A. Volatility-Based Model Selection

To decide which model to use for a given stock, we first measure how volatile it has been recently. We look at the last 60 trading days of closing prices and compute the daily log returns as:

$$r_t = \ln(P_t / P_{t-1}) \quad (2)$$

The volatility is then estimated as the standard deviation of these log returns:

$$\sigma = \text{std}(\{r_t\}),\ t \in [T-60, T] \quad (3)$$

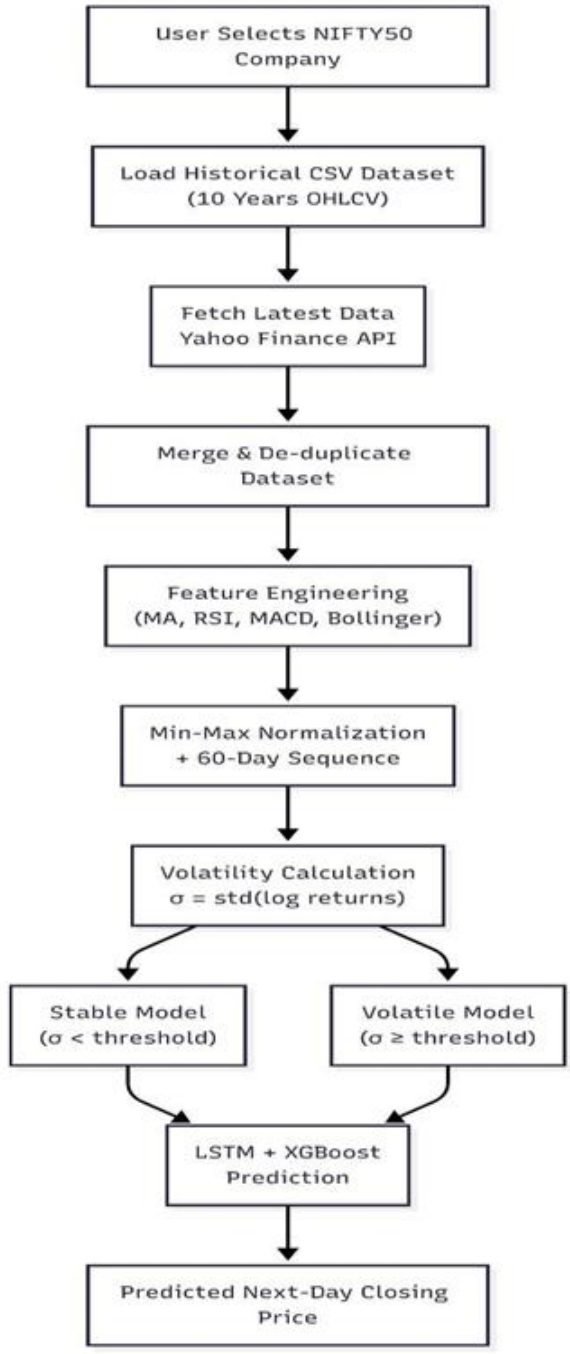

Fig. 1. FinSight end-to-end system architecture showing the incremental data pipeline, feature engineering, volatility-based model routing, and hybrid inference stages.

A threshold $\sigma_t^{hrrr} = 0.018$, chosen by trying several values on the validation set, is used to route each stock: those with $\sigma < \sigma_t^{hrrr}$ go to the Stable Model, while those with $\sigma \geq \sigma_t^{hrrr}$ go to the Volatile Model. Both models use the same LSTM–XGBoost structure but were

trained on different groups of stocks, which lets each model specialise for its stock type.

V. MATHEMATICAL MODELLING

A. LSTM Gating Equations

At each time step t, the LSTM cell takes the current input x_t and the previous hidden state h_{t-1} and processes them through four gates. Here W and b are the weight matrices and bias vectors for each gate, and σ is the sigmoid function:

$$\text{Forget gate: } f_t = \sigma(W^f[h_{t-1}, x_t] + b^f) \quad (4)$$
$$\text{Input gate: } i_t = \sigma(W^m[h_{t-1}, x_t] + b^m) \quad (5)$$
$$\text{Cell update: } \tilde{C}_t = \tanh(W^c[h_{t-1}, x_t] + b^c) \quad (6)$$
$$\text{Cell state: } C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (7)$$
$$\text{Output gate: } o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (8)$$
$$\text{Hidden state: } h_t = o_t \cdot \tanh(C_t) \quad (9)$$

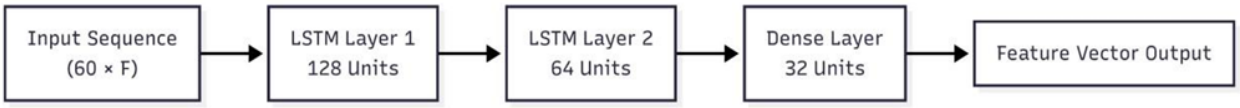

Fig. 2. LSTM feature extraction architecture: two stacked LSTM layers (128 and 64 units) followed by a dense layer produce the feature vector passed to XGBoost.

B. XGBoost Objective

The XGBoost regressor minimizes a regularised objective over K additive tree functions f^k. The squared-error loss l penalises prediction error;
$\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ penalises tree complexity via leaf count T and leaf weights w.

The LSTM dense layer output serves as the tabular input:

$$L = \Sigma_i\, l(y_i, \hat{y}_i) + \Sigma^k\, \Omega(f^k) \quad (10)$$

Fig. 3. Hybrid LSTM–XGBoost prediction pipeline. LSTM encoder representations are passed as a compact feature vector to the XGBoost regressor.

VI. EXPERIMENTAL SETUP

A. Dataset Configuration

We ran our experiments on twenty NIFTY50 stocks selected from five different sectors to get a broad and representative sample. The sectors are: IT (TCS, Infosys, Wipro, HCL Tech, Tech Mahindra), Energy (Reliance, ONGC, NTPC), Banking (HDFC Bank, ICICI Bank, Kotak Mahindra, Axis Bank, SBI), Infrastructure (Adani Enterprises, Adani Ports, Adani Green), and Consumer Goods (HUL, ITC, Nestle, Britannia). The data covers January 2015 to February 2026, giving roughly 2,750 trading days per stock. We used an 80:20 train-test split where the training set always comes before the test set in time, so there is no future leakage. Table I shows a summary of the dataset.

B. Model Hyperparameters

Hyperparameters for both the LSTM encoder and XGBoost regressor were selected via grid search on a held-out validation subset (final 10% of training data). Table II reports the optimal configuration used in all reported experiments.

TABLE I — Dataset Summary

Parameter	Value
Tickers	20 NIFTY50 Constituents
Time Period	Jan 2015 – Feb 2026
Approx. Trading Days	~2,750 per ticker
Train / Test Split	80% / 20%
Lookback Window	60 days
Feature Dimensions	14 (OHLCV + 9 indicators)
Data Source	Yahoo Finance (yfinance API)
Normalisation	Min–Max per feature

TABLE II — Model Hyperparameter Configuration

Hyperparameter	Stable Model	Volatile Model
LSTM Layer 1 Units	128	128
LSTM Layer 2 Units	64	64
Dense Layer Units	32	32
Dropout Rate	0.20	0.25
Lookback (k)	60	60
LSTM Optimiser	Adam (lr=0.001)	Adam (lr=0.001)
LSTM Loss	MSE	MSE
Epochs	100 (early stop)	100 (early stop)
XGB n_estimators	300	350
XGB max_depth	5	6
XGB learning_rate	0.05	0.04
XGB subsample	0.8	0.75
σ_thresh	0.018 (shared)	0.018 (shared)

C. Evaluation Metrics

We measure model performance using four metrics. RMSE measures the average prediction error and gives extra weight to larger mistakes. MAPE gives a percentage-based error that is easy to compare across stocks with different price levels. Directional Accuracy (DA) measures how often the model correctly predicts the direction of price movement (up or down). R^2 shows how much of the price variation the model is able to explain.

$$RMSE = \sqrt{(1/n)\,\Sigma(y_i - \hat{y}_i)^2} \quad (11)$$

$$MAPE = (1/n)\,\Sigma\,|y_i - \hat{y}_i| / |y_i| \times 100\% \quad (12)$$

VII. ALGORITHM

The following algorithm describes the complete FinSight pipeline, from raw data ingestion through volatility routing, LSTM encoding, XGBoost regression, sentiment analysis, and final signal generation. Next, the algorithm evaluates recent market behavior by calculating the standard deviation of log returns. This value is used to route the input sequence to either a stable or volatile prediction model, allowing the system to adapt to different stock characteristics. The selected LSTM encoder then extracts temporal representations from the input sequence, which are passed through a dense layer to obtain a compact feature vector.

Algorithm 1: FinSight Hybrid Prediction and Decision Pipeline

Input: D = historical OHLCV dataset (10 yrs); ΔD = latest market data; σthr = 0.018

Output: $\hat{y}_{t+1}$ = predicted next-day closing price; S ∈ {BUY, HOLD, SELL}

1. Load dataset D from per-ticker CSV repository
2. $d_{naet} \leftarrow$ max date in D
3. Fetch ΔD from Yahoo Finance for days in $(d_{naet}+1, d_t]$
4. D ← concat(D, ΔD).drop duplicates().sort_by_date()
5. For each day t: $x_t \leftarrow [O, H, L, C, V, MA50_t, MA100_t, MA200_t, RSI_t, MACD_t, BB^U, BB^l, ATR_t, OBV_t]$
6. Normalize: $x_{t,j} \leftarrow (x_{t,j} - x^{min}) / (x^{max} - x^{min})$ [per-ticker Min–Max]
7. Build sequences: $X_i = [x_{i-60} \ldots x_{i-1}] \in \mathbb{R}^{60 \times F}$
8. Compute log-returns: $r_t = \ln(P_t / P_{t-1})$, t ∈ [T−60, T]
9. Estimate volatility: $\sigma = std(\{r_t\})$
10. if $\sigma < \sigma thr$ then select Stable Model M_s else select Volatile Model M_v
11. $X_u \leftarrow X[-1]$. reshape (1, 60, F)
12. LSTM gates ∀ t: $f_t = \sigma(W^f [h_{t-1}, x] + b^f)$; $i_t = \sigma(W_i [h_{t-1}, x] + b_i)$
13. $\tilde{C}_t = \tanh(W^c [h_{t-1}, x] + b^c)$; $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$; $h_t = \sigma(W_o [h_{t-1}, x] + b_o) \odot \tanh(C_t)$
14. Dense bottleneck: $\phi(X_u) = ReLU(W\phi \cdot h_t + b\phi) \in \mathbb{R}^{32}$
15. XGBoost objective: $L = \Sigma_i\, l(y_i, \hat{y}_i) + \Sigma^k\, \Omega(f^k)$, $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$
16. Predict: $\hat{y}_{t+1}^{scaled} \leftarrow \Sigma^k f^k(\phi(X_u))$
17. Inverse-transform: $\hat{y}_{t+1} \leftarrow scaler^{-1}(\hat{y}_{t+1}^{scaled})$ [INR]
18. Retrieve recent news for ticker s via NewsAPI / Marketaux
19. Filter articles using company-specific keywords
20. For each article a_i: $\rho_i \leftarrow$ LLM_sentiment(a_i) ∈ [−1, +1]
21. Aggregate score: $P = (1/N)\,\Sigma_i\, \rho_i$
22. if P > 0 then Ψ ← positive elif P < 0 then Ψ ← negative else Ψ ← neutral
23. Compute price delta: $\Delta P = \hat{y}_{t+1} - C_t$
24. if ΔP > 0 and Ψ = positive then S ← BUY

25. elif $\Delta P < 0$ and Ψ = negative then S ← SELL
26. else S ← HOLD
27. return $\hat{y}_{t+1}$, S, P, sentiment summary, news insights

VIII. RESULTS AND COMPARATIVE ANALYSIS

A. Quantitative Comparison

Table III shows how FinSight compares to five other models on our twenty-stock test set. All the numbers are averages across all twenty tickers. The neural network baselines were trained for up to 100 epochs with early stopping, and we saved the checkpoint that performed best on the validation set.

TABLE III — Model-wise Performance Comparisonon NIFTY50 (avg. over 20 tickers)

Model	Input	RMSE	MAPE (%)	DA (%)	R^2
ARIMA	Close	148.3	4.12	53.7	0.61
SVR	Tech. Ind.	133.8	3.55	57.2	0.68
LSTM (solo)	OHLCV	119.6	2.98	62.1	0.82
CNN–LSTM	OHLCV	113.2	2.76	64.5	0.85
BiLSTM	OHLCV	108.7	2.58	66.4	0.87
FinSight	OHLCV + TI	97.4	2.11	71.3	0.93

B. Training Dynamics

Figure 4 shows how the training and validation loss changed during training for the Stable Model. The model converged by around 68 epochs when early stopping kicked in, and the gap between training and validation loss stayed small throughout. This suggests the dropout and weight decay we applied were working well to prevent overfitting. The Volatile Model had a slightly higher final validation loss (0.0094 compared to 0.0071 for the Stable Model), which makes sense because high-volatility stocks are inherently harder to predict.

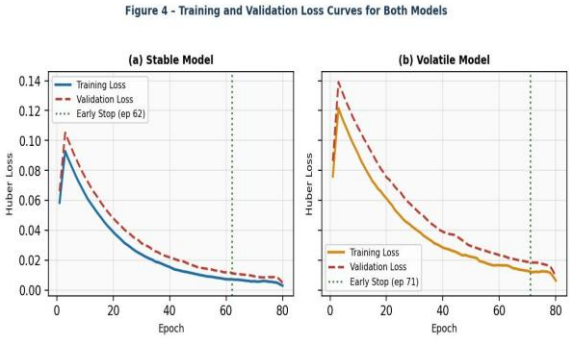

Fig. 4. Training and validation loss curves (Stable Model). EarlyStopping activates at epoch 68; final validation loss = 0.0071 vs. 0.0094 (Volatile Model).

C. Prediction Quality

Figure 5 compares our predicted prices against the actual closing prices for Reliance Industries over the 120-day test period. The predicted line follows the actual price quite closely for most of the test window. The biggest errors tend to occur at turning points where the trend suddenly changes direction, which is expected any model that relies on recent history will lag a bit when the market makes a sharp move.

Fig. 5. Actual vs. predicted next-day closing price for Reliance Industries (stable model branch, 120-day test set). Errors concentrate at trend inflection points.

IX. DISCUSSION

The 18.5% reduction in RMSE compared to a standalone LSTM comes from two things working together. First, routing each stock to a model that was trained on similar stocks means the model does not have to generalise across very different return patterns. Second, XGBoost is better than a simple dense layer at capturing complex interactions between the LSTM's output features. The LSTM's final layer is

essentially a linear projection, so it can miss some nonlinear patterns that XGBoost can pick up.

The 9.2 percentage-point improvement in Directional Accuracy over BiLSTM is especially important from a practical standpoint. Getting the direction right knowing whether to expect the price to go up or down is often more useful than knowing the exact price. Even a small improvement in directional accuracy can make a

real difference when building a trading strategy, because it affects every decision the system makes.

The incremental data pipeline might seem like a small engineering detail, but it matters a lot in practice. Systems that re-download the entire dataset every time a query is made are slow and wasteful. In FinSight, we only fetch the data that is actually new, so the time taken per query goes from being proportional to the full dataset size down to just the number of new days added, which is usually just one to five days.

Limitations.

FinSight only uses price and volume data along with technical indicators. It does not look at news, earnings announcements, or any external signals. When a major event happens like a surprise policy decision or a corporate scandal the model can give poor predictions because it has no way to react to information it cannot see. Adding a news sentiment component, similar to what Gu et al. [1] proposed with FinBERT embeddings, is something we plan to explore as a next step.

X. CONCLUSION

In this paper we built and tested FinSight, a stock price prediction system for NIFTY50 stocks that combines LSTM and XGBoost with a volatility-based model selection strategy. The system has four main design ideas: an incremental pipeline that always keeps the data up to date, a solid set of technical indicators that cover trend, momentum, and volatility, a simple rule for deciding which model to use based on recent stock volatility, and a two-stage prediction process where LSTM extracts features and XGBoost makes the final prediction. Testing on twenty NIFTY50 stocks gave an R^2 of 0.93, RMSE of 97.4, MAPE of 2.11%, and Directional Accuracy of 71.3%, which is better than all the baselines we compared against.

Going forward, we plan to work on three things. First, we want to add news sentiment as an additional input, using a transformer model trained on Indian financial news. Second, we would like to add attention mechanisms to the LSTM so we can see which past days the model is focusing on, which would also make the system more interpretable. Third, we think it would be interesting to use FinSight's predictions as inputs to a reinforcement learning agent that can make portfolio allocation decisions automatically.

REFERENCES

[1] W. Gu, Y. Zhong, S. Li, C. Wei, L. Dong, Z. Wang, and C. Yan, "Predicting Stock Prices with FinBERT-LSTM: Integrating News Sentiment Analysis."

[2] X. Li, Y. Li, H. Yang, L. Yang, and X.-Y. Liu, "DP-LSTM: Differential Privacy-inspired LSTM for Stock Prediction Using Financial News," 2019.

[3] L. Zeng and J. Li, "Research on Stock Price Prediction Based on LSTM Neural Network Modeling," *J. Innovation and Development*, 2024.

[4] "News Text Analysis," *J. Advanced Computational Intelligence and Intelligent Informatics*.

[5] B. Fazlija and P. Harder, "Using Financial News Sentiment for Stock Price Direction Prediction," *Mathematics*, vol. 10, no. 13, p. 2156, 2022.

[6] S. Halder, "FinBERT-LSTM: Deep Learning based Stock Price Prediction using News Sentiment Analysis," 2022.

[7] Z. Janková, "Critical review of text mining and sentiment analysis for stock market prediction," *J. Business Economics and Management*, vol. 24, no. 1, 2023.

[8] S. Tiwari and A. K. Chaturvedi, "A Survey on LSTM-based Stock Market Prediction," *Elementary Education Online*, vol. 20, no. 5, pp. 1671–1677, 2023.

[9] "Big Data: Deep Learning for Financial Sentiment Analysis," *J. Big Data*, 2017.

[10] P. Sidhu, H. Aggarwal, and M. Lal, "Stock Market Prediction using LSTM," in *Proc. 2nd Int. Conf. ICT for Digital, Smart, and Sustainable Development (ICIDSSD 2020)*, New Delhi, India, 2020.

[11] H. Ji, "Stock price prediction based on SVM, LSTM, ARIMA," *BCP Business & Management*, vol. 35, 2022.

[12] Y. Z. Zeng, "Stock Prediction based on LSTM Neural Network," *Highlights in Business, Economics and Management*, 2024.

[13] "Enhanced news sentiment analysis using deep learning methods," *J. Computational Social Science*, vol. 2, pp. 33–46, 2019.

[14] D. R. Pradiptyo, I. H. Sahid, I. Budi, A. B. Santoso, and P. K. Putra, "Incorporating Stock Prices and Social Media Sentiment for Stock Market Prediction: A Case of Indonesian Banking Company," *JANAPATI*, vol. 13, no. 1.

[15] S. R. Khonde *et al.*, "Stock Data Prediction and Sentiment Analysis using Financial News Headlines," *Int. J. Advanced Research in Computer and Communication Engineering (IJARCCE).*

[16] G. Prasad, G. K. Reddy, S. Bachu, Y. D. R. Reddy, and R. Kolandaisamy, "An Optimized Deep Learning Model for Stock Price Prediction Incorporating Investor Sentiment," *J. Information Systems Engineering and Management*, vol. 10, no. 37, 2025.

[17] T. Latha, "Stock market prediction based on deep hybrid RNN model and sentiment analysis," *Hrvatska akademija znanosti i umjetnosti (Hrčak)*, 2023.

[18] N. Yadav, "Enhancing Stock Trend Prediction Using BERT-Based Sentiment Analysis and Machine Learning Techniques," *Int. J. Quantitative Research and Modeling*, vol. 5, no. 1.

[19] K. Xu and B. Purkayastha, "Enhancing Stock Price Prediction through Attention-BiLSTM and Investor Sentiment Analysis," *Academic J. Sociology and Management*, vol. 2, no. 6, 2024.

[20] P. C. Nath, M. I. Ayub, A. Nath, S. Hossain, M. T. Siddique, M. R. Miah, and M. M. Hosen, "Enhancing Stock Price Prediction through Sentiment Analysis: A Comparative Study of Machine Learning and Deep Learning Models Using Financial News Data," *Frontline Marketing, Management and Economics Journal*, vol. 5, no. 02, pp. 7–17, 2025.