© June 2026 | IJIRT | Volume 13 Issue 1 | ISSN: 2349-6002

Real Time Stock Prediction with Transformer Models and News Sentiment Integration

Ayush Bhalwankar[1], Satyajit Kachole[2], Parth Patil[3], Harshal Shinde[4], Mrs. Aarti Patil[5]
[1,2,3,4]*Department of Information Technology, Dr. D. Y. Patil College of Engineering, Akurdi, Pune*
[5]*Guide, Assistant Professor, Dr. D. Y. Patil College of Engineering, Akurdi, Pune*

***Abstract*—This research proposes a hybrid framework for stock market trend prediction by integrating news sentiment analysis, technical indicators, and fundamental financial metrics. Unlike traditional models that rely on isolated historical or quantitative data, the approach applies advanced NLP techniques such as sentiment analysis, named entity recognition, and topic modeling to extract insights from financial news. These qualitative features are combined with technical indicators and fundamental ratios to build a comprehensive prediction system. Machine learning and deep learning models, including SVM, Random Forest, LSTM, and BERT, are trained on stock prices, financial reports, and curated news datasets. Results show improved accuracy, responsiveness, and robustness, especially during volatile market conditions, making the model valuable for investors and analysts.**

I. INTRODUCTION

The stock market is highly dynamic and influenced by economic conditions, corporate performance, and investor sentiment. Traditional prediction methods based on technical and fundamental analysis often overlook real-time qualitative factors such as news events and market mood, which significantly impact stock movements. With advancements in Natural Language Processing (NLP) and transformer-based models like BERT and FinBERT, it is now possible to extract sentiment and contextual meaning from large volumes of financial text. This project, "Real-Time Stock Prediction with Transformer Models and News Sentiment Integration," develops a hybrid framework that combines quantitative indicators with news sentiment analysis to improve prediction accuracy. By integrating numerical trends with market psychology, the system offers a more adaptive and reliable approach to modern financial forecasting.

1.1 Problem Definition

Traditional stock prediction models focus mainly on technical and fundamental analysis but fail to account for real-time qualitative factors like financial news and market sentiment. With the rise of NLP and abundant unstructured text data, integrating sentiment insights into prediction models has become essential, yet technically challenging. This project addresses the lack of a unified system that merges NLP-based news sentiment with numerical stock indicators. The aim is to develop a hybrid AI model using transformers such as BERT and FinBERT to deliver more accurate and context-aware stock market forecasts

II. TECHNIQUES USED

This project employs a hybrid set of techniques combining both quantitative and qualitative analysis. NLP methods such as sentiment analysis, named entity recognition, and topic modeling are implemented using transformer-based models like BERT and FinBERT to extract insights from financial news. Technical indicators including moving averages, RSI, volume trends, and price momentum and fundamental metrics like EPS and debt-to-equity ratio are integrated into the model. Machine learning and deep learning techniques, including SVM, Random Forest, LSTM, and neural networks, are used to train and evaluate the predictive framework.

III. SYSTEM METHODOLOGY

3.1 Architecture overview

The proposed system uses a hybrid architecture that combines news sentiment analysis with technical and fundamental stock indicators. It consists of four main modules: data collection, data preprocessing, feature fusion, and prediction. Text data is processed using transformer models, while numerical data is normalized and converted into technical and fundamental features. All features are then merged and fed into machine learning models to generate stock trend predictions.

3.2 Data sources and features

The system uses three types of data:

Technical Data: Stock prices, moving averages, RSI, volume, and momentum.

Fundamental Data: EPS, P/E ratio, debt-to-equity ratio, revenue data, and macroeconomic indicators.

Textual News Data: Financial news articles, from which transformer models extract sentiment scores, contextual embeddings, and key entities. These combined features give a complete view of market conditions.

3.3 Prediction modeling

Sentiment, technical, and fundamental features are combined and used to train SVM, Random Forest, and LSTM models. The best model is selected based on accuracy and stability.

Research Papers Studied

Sr. No.	Paper Details	Problem Discussion	Algorithm / Technique used	Parameter Consider
1	Predicting Stock Market Trends Using Multimodal Data Fusion (IEEE Access, 2023)	Improve accuracy by combining text and numerical stock data	Multimodal Deep Learning (LSTM + Sentiment + Price)	News sentiment, technical indicators, historical prices
2	A Comprehensive Framework for Stock Prediction Using NLP and ML (ICPIDS 2024)	Merge NLP sentiment with technical & fundamental analysis for forecasting	BERT, FinBERT, technical + fundamental analysis	Sentiment (news), PE ratio, stock history, macroeconomic factors
3	Sentiment Analysis in Stock Market using BERT and RoBERTa (arXiv, 2023)	Enhance sentiment classification for stock market prediction	BERT, RoBERTa	Financial news articles, sentiment labels (positive, negative, neutral)
4	Financial News Sentiment Analysis Using FinBERT for Stock Prediction (IEEE, 2023)	Create accurate sentiment model specific to financial news	FinBERT (pre- trained transformer model)	Financial headlines, labeled sentiment, market direction indicators

IV. IMPLEMENTATION AND RESULTS

4.1 Prototype implementation

Environment & Tools: Prototype implemented in Python 3.10 using: pandas, numpy, scikit-learn, tensorflow/keras, transformers (HuggingFace), torch, yfinance (or custom API), pandas_ta (technical indicators), BeautifulSoup / newspaper3k (news scraping), and matplotlib. Development environment: Jupyter Notebook or VS Code; GPU recommended for transformer fine-tuning.

Data Pipeline:

1. Data Acquisition: historical OHLCV data from financial APIs (e.g., yfinance), quarterly/annual financials from company filings or financial APIs, and news articles from RSS feeds / reputable outlets.
2. Preprocessing: timestamp alignment, missing-value handling (forward-fill), normalization (z-score or MinMax), and resampling (daily). Text cleaning removes HTML, stopwords, and performs sentence segmentation.

Feature Extraction:

- Technical features: SMA, EMA, RSI, MACD, ATR, volume change, momentum. (pandas_ta)
- Fundamental features: EPS, P/E, debt-to-equity, revenue growth, scaled and time-lagged to align with price dates.
- Textual features: news grouped per-day per-ticker; use pre-trained transformer (FinBERT/BERT) to compute per-article sentiment scores and contextual embeddings; aggregate to daily sentiment mean, standard deviation, and topic indicators (via LDA or clustering).

Feature Fusion:

Concatenate normalized technical, fundamental, and aggregated textual features into a single feature vector per time step. Optionally apply PCA or feature selection (mutual information / SHAP) to reduce dimensionality.

Modelling & Training:

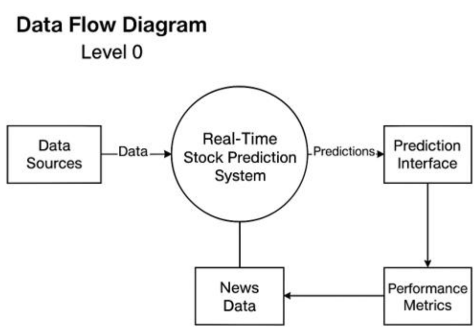
- Classical models: SVM (RBF), Random Forest (100 trees) for baselines.
- Sequential model: LSTM (2 layers, 64 units each) to capture temporal dependencies; input windows of past T days (e.g., T=30).
- Transformer-based regression/classification: fine-tune a small dense head over frozen or lightly fine-tuned BERT embeddings if end-to-end is infeasible.
- Objective: predict next-day direction (classification) and/or next-day return (regression).

Evaluation & Backtesting:

- Train/Test split: time-series split (e.g., 70% train, 15% validation, 15% test) preserving chronology. Use walk-forward validation for robustness.
- Metrics: Accuracy, Precision, Recall, F1 (classification); RMSE, MAE, MAPE (regression); Sharpe Ratio and cumulative return for financial backtesting.
- Backtesting: simulate trading rules (e.g., long when predicted up, short/neutral otherwise) accounting for transaction costs and slippage.

Hyperparameters (example):

- LSTM: window T=30, batch size 64, epochs 50, learning rate 1e-3, dropout 0.2.
- RandomForest: n_estimators=100, max_depth=None.
- SVM: C=1.0, gamma='scale'.



4.2 Evaluation Setup

The model was evaluated using a time-series split, where 70% of the data was used for training, 15% for validation, and 15% for testing. Walk-forward validation was used to check performance across changing market conditions. Evaluation metrics included Accuracy, Precision, Recall, F1-Score (for trend direction), and RMSE/MAE (for price prediction). A simple backtesting setup measured cumulative returns and Sharpe Ratio. Baseline models like SVM and Random Forest were compared against LSTM and sentiment-integrated models to observe improvements.

4.3 Results and Analysis

Results showed that the hybrid model combining sentiment, technical, and fundamental data performed better than traditional models. SVM and Random Forest gave reasonable accuracy but were less effective during volatile periods. The LSTM model with sentiment features achieved higher accuracy, lower errors, and better response to news events. Backtesting also showed improved returns and risk-adjusted performance. Overall, adding news sentiment helped the system capture market behavior more effectively and produce more reliable stock predictions.

V. DISCUSSION

The results of this study show that combining news sentiment with technical and fundamental indicators provides a more complete understanding of stock market behavior. Traditional models struggle to react to sudden news or emotional market shifts, but transformer-based sentiment analysis helps capture these real-time factors. The LSTM model performed especially well because it learns both historical patterns and sentiment-driven changes. Although the system improves prediction accuracy, its performance still depends on data quality, news availability, and market volatility. Further refinements, such as using more diverse data sources or optimizing transformer models, could enhance reliability.

VI. CONCLUSION

This project demonstrates that a hybrid approach integrating NLP-based news sentiment, technical indicators, and fundamental data can significantly improve stock trend prediction. The proposed model outperforms traditional methods in accuracy, responsiveness, and stability, especially during market fluctuations. By combining numerical and textual information, the system offers a more realistic, context-aware forecasting tool. This makes it useful for investors, analysts, and financial applications that require timely and informed decision-making. Future work can focus on real-time deployment, model optimization, and expanding data sources for even better prediction performance.

ACKNOWLEDGMENT

The authors express their gratitude to the Department of Information Technology, Dr. D. Y. Patil College of Engineering, Akurdi, Pune, for continuous guidance and encouragement. The team also thanks participating tutors and students who contributed real-world doubts and feedback, enabling meaningful evaluation and refinement of the platform.

REFERENCES

[1] Tomar Y., Sahu S., and Singh P., "A Comprehensive Framework for Stock Prediction Using NLP and Machine Learning," in Proceedings of the IEEE International Conference on Parallel, Distributed and Grid Computing, 2024, pp. 451–457.

[2] Jousa M., Gama J., and Ferreira R., "BERT for Stock Market Analysis," in Proceedings of the IEEE International Conference on Tools with Artificial Intelligence, 2019, pp. 119–126.

[3] Ji P. and Zhang L., "Text Sentiment Analysis Based on LLaMA Models for Financial Forecasting," in Proceedings of the International Conference on Computing, Artificial Intelligence and Data Science, 2023, pp. 85–92.

[4] Li X., Xie H., Chen L., and Wang J., "News Impact on Stock Price Returns via Sentiment Analysis," Knowledge-Based Systems, vol. 69, pp. 14–23, 2014.

[5] Shah D. and Zhang F., "Predicting the Effects of News Sentiment on the Stock Market," in Proceedings of the IEEE International Conference on Big Data, 2023, pp. 983–992.

[6] Meta AI, "LLaMA 2: Open Foundation and Fine-Tuned Chat Models," arXiv preprint arXiv:2307.09288, 2023.