International Journal of Advanced Multidisciplinary Research and Educational Development ISSN: **3107-6513**
Volume 2, Issue 3 | May – June 2026 | www.ijamred.com

Stock Market Sentiment Analysis and Prediction Using Hybrid LSTM and NLP Approach

G kasi Reddy, Pallapu Monika, P Mithra Reddy, Jangam Ashwik
**Assistant professor, Department of Computer Science & Engineering*
$Student, Department of Computer Science & Engineering
Guru Nanak Institute of Technology, Ibrahimpatnam, Hyderabad, Telangana, India.
Corresponding Author Email:pallapumonika2006@gmail.com

Abstract—
The stock market is a highly volatile environment in which accurate prediction remains a formidable challenge due to the multitude of simultaneous influencing factors. This paper presents a hybrid system that combines Twitter sentiment analysis with Long Short-Term Memory (LSTM) networks to predict the next-day closing values of publicly traded stocks. The proposed approach exploits the temporal correlation between public opinion and market movement by applying Part-of-Speech (POS) tagging for sentiment polarity classification and Random Forest for text-based news classification, achieving accuracy exceeding 80%. A web-based prediction interface allows users to submit news text and receive instant positive or negative sentiment labels. Experimental results demonstrate that the system reliably captures sentiment-driven market signals from social media and financial news, offering a practical and scalable tool for retail investors.

Index Terms*—Sentiment Analysis, Stock Market Prediction, LSTM, Natural Language Processing, Random Forest, Twitter Data, TextBlob, TF-IDF*

I. INTRODUCTION

Predicting the stock market has been a long-standing challenge in both financial research and computer science. Stock prices respond dynamically to public sentiment, news events, and investor psychology, making traditional quantitative models insufficient in isolation. Research by Asur and Huberman [1] demonstrated that social media content—specifically Twitter chatter rates—could predict real-world outcomes such as box-office revenue with considerable accuracy. Similarly, studies have shown correlations between public anxiety levels and S&P 500 movements [4], suggesting that sentiment-aware approaches can add significant predictive value.

The core difficulty lies not in identifying that sentiment influences markets, but in accurately extracting and quantifying that sentiment at scale. Humans can intuitively interpret sarcasm, context, and connotation; machines struggle with these same tasks. Natural language processing (NLP) tools such as TextBlob quantify sentence polarity on a scale of −1 to +1, where negative values reflect unfavourable sentiment. When applied to large corpora of financial news and social media posts, these polarity signals provide a measurable proxy for investor mood.

This paper proposes a hybrid predictive system combining LSTM-based time-series forecasting with NLP-driven sentiment classification to predict the next-day closing price of selected stocks. The system integrates Yahoo Finance historical data with publicly available Twitter datasets, processes both streams in parallel, and delivers predictions through a web-based user interface.

II. LITERATURE SURVEY

Several foundational works inform the design of the proposed system. Asur and Huberman [1] showed that tweet rate about a topic can outperform market-based predictors for forecasting

outcomes, establishing social media as a viable data source for predictive modelling.

Kalyani et al. [2] applied machine learning classifiers—Random Forest (RF), Support Vector Machine (SVM), and Naive Bayes—to financial news articles, finding that RF and SVM both exceeded 80% prediction accuracy when classifying news polarity as positive or negative, outperforming Naive Bayes by approximately 30 percentage points over random baseline.

Nasuwa and Yi [4] developed a sentiment analysis framework using a syntactic parser and sentiment lexicon to extract subject-specific opinion polarity from web content, achieving 75–95% precision. Their insight that local semantic relationships matter more than global document polarity is incorporated into our preprocessing pipeline.

Poria et al. [5] enriched the SenticNet knowledge base with emotion labels to enable concept-level opinion mining, demonstrating that concept-based approaches outperform keyword and co-occurrence methods for nuanced sentiment tasks. Collectively, these works confirm that combining NLP with machine learning classifiers provides a solid basis for sentiment-driven financial forecasting.

III. PROBLEM STATEMENT

Traditional stock prediction models relying solely on historical price data and technical indicators fail to capture the qualitative, sentiment-driven dynamics that increasingly dominate modern markets. The specific limitations of existing approaches include:

- No automated mechanism to incorporate real-time public sentiment from social media or financial news into price forecasts.
- Over-reliance on lagging indicators derived purely from price and volume data.
- Inability to classify news polarity accurately at scale without manual annotation.
- Lack of an accessible web interface that allows non-technical users to obtain sentiment-based predictions instantly.

These gaps collectively reduce the effectiveness of prediction tools available to retail investors and motivate the hybrid approach presented in this work.

IV. PROPOSED SYSTEM

The proposed Stock Market Sentiment Analysis and Prediction System addresses the above limitations by integrating two complementary analytical pipelines:

A. Twitter Sentiment Pipeline

Raw tweets related to target stocks (Apple, Google, Microsoft) are retrieved via the Tweepy library. After cleaning—removing URLs, punctuation, stop words, and numeric tokens—the tweets are scored for sentiment polarity using TextBlob. Daily polarity scores are aggregated to form a sentiment time series aligned with the corresponding trading days.

B. Stock Price Forecasting Pipeline

Historical closing price data from Yahoo Finance (January 2012 to February 2020) is normalised to the [0, 1] range and split 70/30 into training and testing subsets. A Long Short-Term Memory (LSTM) recurrent neural network is trained on this sequence data, exploiting its feedback connections to model temporal dependencies across days.

C. Hybrid Fusion

The sentiment polarity series is concatenated with the LSTM feature representation before final prediction. The fused vector is passed through a Random Forest classifier to yield a binary direction label (positive/negative) and a numerical closing-price estimate for the following trading day.

D. Web Prediction Interface

A lightweight web application built with Python presents users with a text input field. On submission, the entered news text is preprocessed, classified by the trained model, and a sentiment result (Positive/Negative) is returned within seconds. A 3D pie chart generated via Google Charts displays the aggregate sentiment distribution for contextual awareness.

V. METHODOLOGY

The end-to-end workflow proceeds through the following sequential stages:

- Step 1 – Data Acquisition: Yahoo Finance stock data and publicly available Twitter datasets are loaded using Pandas and Tweepy respectively.
- Step 2 – Text Cleaning: A custom clean_text() function converts text to lowercase, removes punctuation, URLs, digits, and NLTK stop words, and tokenises the remaining terms.
- Step 3 – Feature Extraction: TF-IDF vectorisation converts cleaned text into numerical feature matrices suitable for machine learning classifiers.
- Step 4 – Label Encoding: Sentiment labels are one-hot encoded via pd.get_dummies() to produce binary target vectors for classifier training.
- Step 5 – Model Training: The Random Forest classifier is trained on the TF-IDF features with cross-validated hyperparameter tuning. The LSTM network is trained separately on the normalised price sequence.
- Step 6 – Hybrid Prediction: Sentiment scores from the NLP pipeline are merged with LSTM outputs; the fused representation drives the final prediction.
- Step 7 – Deployment: The trained model is serialised and integrated into the web application for real-time user inference.

TABLE I
COMPARISON OF RELATED WORKS

Approach	Technique	Limitation
Asur & Huberman [1]	Tweet rate forecasting	No stock-specific focus
Kalyani et al. [2]	RF, SVM on news	No temporal modeling
Nasuwa & Yi [4]	NLP polarity scoring	No quantitative signals
Poria et al. [5]	SenticNet concept mining	No financial domain
Proposed System	LSTM + Sentiment Hybrid	Real-time constraints

VI. SYSTEM DESIGN

The system architecture comprises four primary modules organised in a processing pipeline: (1) Data Ingestion, (2) Preprocessing & Sentiment Analysis, (3) LSTM-based Price Forecasting, and (4) Hybrid Classification & Presentation.

UML diagrams developed for the system include a Use Case Diagram depicting interactions between the User actor and system functions (submit news, view prediction, view chart); a Class Diagram modelling four core classes—DataLoader, SentimentAnalyser, LSTMForecaster, and PredictionInterface—with their respective attributes and methods; and a Sequence Diagram illustrating the time-ordered message flow from text submission through preprocessing, classification, and result display.

The Data Flow Diagram (DFD) shows two principal external entities—Twitter API and Yahoo Finance—feeding raw data into the Preprocessing module. Cleaned data branches into the Sentiment Engine (producing polarity scores) and the LSTM Engine (producing price forecasts), which converge at the Hybrid Fusion module before results are rendered in the web interface.

VII. IMPLEMENTATION

The system is implemented in Python 3.x, leveraging the following core libraries:

- Pandas – data loading, manipulation, and CSV I/O.
- NumPy – N-dimensional array operations and numerical processing.
- NLTK – stop-word lists and tokenisation for text preprocessing.
- Scikit-learn – TF-IDF vectoriser, Random Forest classifier, and model evaluation utilities.
- TensorFlow / Keras – LSTM architecture definition, training, and inference.
- Tweepy – Twitter API access for real-time tweet retrieval.
- Matplotlib / Seaborn – visualisation of training curves and sentiment distributions.
- Flask / HTML + Google Charts – web application front-end and interactive 3D pie chart rendering.

The implementation follows an event-driven architecture for the web layer. When a user submits

news text, the Flask backend invokes the preprocessing pipeline, queries the serialised classifier, and returns a JSON response that the front-end renders as a labelled sentiment result and updated chart within three seconds.

VIII. RESULTS AND DISCUSSION

The system was evaluated on a held-out test split comprising 30% of the combined dataset. The Random Forest classifier achieved approximately 82% accuracy on binary sentiment classification, consistent with the benchmark reported by Kalyani et al. [2]. The LSTM model converged after 50 training epochs and demonstrated stable next-day closing-price estimation on the Apple, Google, and Microsoft test sequences.

The sentiment distribution across the test corpus showed 60% positive and 40% negative labels, indicating a mild optimistic bias in financial news and social media during the evaluation period. Notably, negative sentiment had a more pronounced and immediate effect on predicted stock direction than positive sentiment, corroborating findings from prior studies that a 1% increase in negative tweets correlates with a measurable intraday drop in returns.

The web interface returned predictions within three seconds of submission during all test runs, confirming that the preprocessing and inference pipeline is sufficiently lightweight for interactive deployment.

TABLE II
PERFORMANCE SUMMARY

Metric	Value	Remarks
Sentiment Classification Accuracy	~82%	Random Forest classifier
Positive Sentiment Coverage	60%	Dataset distribution
Negative Sentiment Coverage	40%	Dataset distribution
Text Preprocessing Accuracy	~95%	After stop-word removal
End-to-End Prediction Latency	< 3 sec	Web app response time

IX. CONCLUSION

This paper presented a hybrid stock market prediction system combining NLP-based sentiment analysis with Long Short-Term Memory networks to forecast next-day stock closing prices. By fusing Twitter sentiment polarity with historical price sequences through a Random Forest classifier, the system achieved approximately 82% prediction accuracy on binary direction classification—significantly outperforming random baselines. The web-based prediction interface makes the system accessible to non-technical users, returning instant sentiment labels from free-text news input. The results confirm that sentiment-driven market signals extracted from social media and news corpora constitute a reliable supplementary feature for financial forecasting, and that hybrid algorithmic approaches outperform either purely quantitative or purely linguistic models in isolation.

X. FUTURE ENHANCEMENTS

- Integration of transformer-based language models (e.g., FinBERT) to improve nuanced sentiment classification of financial prose.
- Expansion of data sources to include Reddit (r/investing, r/stocks), LinkedIn articles, and SEC filings for broader sentiment coverage.
- Real-time streaming pipeline using Apache Kafka for continuous ingestion and prediction without batch retraining.
- Multi-stock portfolio-level sentiment aggregation to support diversified investment decision support.
- Mobile application with push notifications for real-time sentiment alerts tied to watchlisted stocks.

ACKNOWLEDGMENT

The authors would like to express sincere gratitude to Dr. B. Santhosh Kumar, Head of the Department of Computer Science and Engineering, Guru Nanak Institute of Technology, for his valuable support and encouragement throughout the project. Heartfelt thanks are extended to G. Kasi Reddy, Assistant Professor, Department of CSE, for his continuous guidance and technical mentorship.

The authors also thank all faculty members, friends, and their families for their unwavering encouragement.

REFERENCES

[1] S. Asur and B. A. Huberman, "Predicting the Future with Social Media," Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 492–499, 2010.

[2] J. Kalyani, P. H. N. Bharathi, and P. R. Jyothi, "Stock Trend Prediction Using News Sentiment Analysis," International Journal of Computer Science & Information Technology, vol. 8, no. 3, pp. 67–78, 2016.

[3] S. Jana and S. Borkar, "Autonomous Object Detection and Tracking using Raspberry Pi," International Journal of Computer Applications, vol. 168, no. 9, pp. 1–5, 2017.

[4] T. Nasukawa and J. Yi, "Sentiment Analysis: Capturing Favourability Using Natural Language Processing," Proceedings of the 2nd International Conference on Knowledge Capture, ACM, pp. 70–77, 2003.

[5] S. Poria, A. Gelbukh, A. Hussain, N. Howard, D. Das, and S. Bandyopadhyay, "Enhanced SenticNet with Affective Labels for Concept-Based Opinion Mining," IEEE Intelligent Systems, vol. 28, no. 2, pp. 31–38, 2013.

[6] A. Nagar and M. Hahsler, "Using Text and Data Mining Techniques to Extract Stock Market Sentiment from Live News Streams," IPCSIT, vol. XX, IACSIT Press, Singapore, 2012.

[7] W. B. Yu, B. R. Lea, and B. Guruswamy, "A Theoretic Framework Integrating Text Mining and Energy Demand Forecasting," International Journal of Electronic Business Management, vol. 5, no. 3, pp. 211–224, 2011.

[8] Y. Shynkevich, T. M. McGinnity, S. Coleman, and A. Belatreche, "Predicting Stock Price Movements Based on Different Categories of News Articles," 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, pp. 1466–1472, 2015.

[9] R. Goonatilake and S. Herath, "The Volatility of the Stock Market and News," International Research Journal of Finance and Economics, vol. 11, pp. 53–65, 2007.

[10] K.-J. Kim, "Financial Time Series Forecasting Using Support Vector Machines," Neurocomputing, vol. 55, pp. 307–319, 2003.