Sentiment Analysis for Stock Price Prediction Using Machine Learning

Kalp Sheladiya, Kavisha Vaja, Driti Rathod, Jeet Gupta
Sardar Vallabhbhai National Institute of Technology
{U23AI001, U23AI016, U23AI036, U23AI056}

***Abstract*—This paper presents a comprehensive approach to sentiment analysis for stock price prediction by integrating social media sentiment with technical indicators. The system employs LightGBM and XGBoost machine learning models to analyze Twitter sentiment data combined with historical stock prices from the Angel One SmartAPI. Our methodology demonstrates the significant impact of public sentiment on stock market movements, achieving robust prediction accuracy through ensemble techniques and multi-source data fusion. The proposed system processes real-time data from Indian stock markets (NSE) and provides actionable insights for trading decisions. Experimental results show that incorporating sentiment features improves prediction accuracy by 8-12%, with the ensemble model achieving an R^2 score of 0.901.**

***Index Terms*—sentiment analysis, stock prediction, machine learning, Twitter analysis, LightGBM, XGBoost, financial forecasting**

I. Introduction

Stock market prediction has long been a challenging task in financial analytics, with traditional methods primarily relying on technical and fundamental analysis. However, the rise of social media platforms has introduced a new dimension to market analysis, where public sentiment can significantly influence stock prices and trading volumes.

The Efficient Market Hypothesis (EMH) suggests that stock prices reflect all available information. However, behavioral finance research has demonstrated that investor psychology and sentiment play crucial roles in market dynamics. Social media platforms, particularly Twitter, have become influential channels where investors share opinions, news, and analyses that can affect market sentiment and trading decisions.

This research integrates sentiment analysis from Twitter data with traditional technical indicators to create a hybrid prediction model. Our system processes real-time stock data from the Angel One SmartAPI platform, specifically focusing on Indian stock markets (NSE), while simultaneously analyzing social media sentiment to capture market psychology and investor behavior.

The proposed methodology employs advanced natural language processing techniques including VADER (Valence Aware Dictionary and sEntiment Reasoner) for sentiment extraction, combined with gradient boosting algorithms for prediction tasks. This multi-modal approach enables more accurate forecasting by considering both quantitative market data and qualitative sentiment indicators.

The main contributions of this work include: (1) Development of a hybrid prediction system combining sentiment analysis with technical indicators, (2) Implementation of ensemble machine learning models (LightGBM and XGBoost), (3) Integration of real-time data from Angel One SmartAPI and Twitter, (4) Comprehensive feature engineering pipeline incorporating over 50 technical indicators, and (5) Deployment of a production-ready Flask API for real-time predictions.

II. Background and Literature Survey

A. Sentiment Analysis in Finance

Sentiment analysis has emerged as a crucial component in financial prediction systems. Previous research has demonstrated strong correlations between social media sentiment and stock price movements, particularly during periods of high market volatility. The seminal work by Bollen et al. demonstrated that Twitter mood can predict changes in the Dow Jones Industrial Average with 87.6% accuracy. Similar studies have validated the predictive power of social media sentiment across various markets.

Research has shown that incorporating sentiment from financial news and social media improves stock return predictions for major companies. The relationship between Twitter sentiment and stock returns has been extensively studied, with findings indicating that message volume and sentiment are significantly associated with returns. The integration of sentiment features with traditional technical indicators has shown to improve prediction accuracy significantly.

B. Machine Learning for Stock Prediction

Recent advances in machine learning, particularly ensemble methods like XGBoost and LightGBM, have revolutionized time-series forecasting. XGBoost (Extreme Gradient Boosting) has become popular in financial applications due to its regularization capabilities and ability to handle missing data. LightGBM offers faster training speeds and lower memory usage while maintaining comparable accuracy, making it suitable for real-time applications.

Deep learning approaches have also gained traction. LSTM (Long Short-Term Memory) networks excel at capturing temporal dependencies and have been successfully applied to stock prediction. However, gradient boosting methods often outperform these approaches on tabular financial data due to their ability to handle heterogeneous features effectively.

Feature engineering plays a critical role in the success of machine learning models for financial prediction. Technical in-

dicators such as moving averages, RSI, MACD, and Bollinger Bands capture important price patterns and trends.

III. Proposed Methodology

A. System Architecture

The proposed system consists of four main components: (1) Data Collection Module for retrieving stock data from Angel One SmartAPI and Twitter data, (2) Feature Engineering Pipeline for extracting technical indicators and sentiment features, (3) Machine Learning Module for training and deploying LightGBM and XGBoost models, and (4) Prediction API as a Flask-based REST API for real-time predictions.

The system architecture follows a modular design pattern that enables independent scaling of components. The data collection module operates asynchronously, fetching market data every minute during trading hours and social media data continuously throughout the day. This ensures that the prediction system always has access to the most current information available.

B. Data Collection and Processing

The system collects historical OHLCV (Open, High, Low, Close, Volume) data for major NSE-listed companies including RELIANCE, INFY, TCS, HDFCBANK, ICICIBANK, SBIN, WIPRO, HCLTECH, ITC, and BHARTIARTL. Data quality assurance includes handling missing values through forward-fill interpolation and detecting outliers using the Interquartile Range (IQR) method.

Twitter data is collected using the Twitter API v2 with search queries for stock symbols and relevant hashtags. The collection process implements rate limiting and exponential backoff to comply with API constraints. Tweet preprocessing includes removal of URLs, mentions, and special characters, followed by tokenization, stopword removal, and lemmatization. We maintain a rolling window of 30 days of tweet data, which provides sufficient historical context while keeping computational requirements manageable.

C. Sentiment Analysis

We employ a multi-method approach for sentiment analysis to capture different aspects of market sentiment:

VADER Sentiment Analysis: VADER provides compound scores ranging from -1 (most negative) to +1 (most positive). We enhanced VADER's lexicon with domain-specific financial terms including bullish terms (rally, surge, soar, gain, breakout) and bearish terms (crash, plunge, tumble, loss, downturn). This customization improved sentiment classification accuracy by approximately 15% compared to the base VADER implementation.

TextBlob Analysis: TextBlob provides polarity and subjectivity scores. Polarity indicates sentiment orientation, while subjectivity measures the degree of personal opinion versus factual information. High subjectivity scores often indicate speculative discussions, which we found to be particularly relevant during earnings announcements.

Combined Sentiment Score: The final sentiment score is computed as:

$$S_{combined} = 0.5 \cdot S_{VADER} + 0.5 \cdot S_{TextBlob} \tag{1}$$

Engagement-Weighted Sentiment: To account for influential tweets with higher engagement:

$$S_{weighted} = \frac{\sum_{i=1}^{n} S_i \cdot (likes_i + retweets_i + 1)}{\sum_{i=1}^{n} (likes_i + retweets_i + 1)} \tag{2}$$

This weighting scheme ensures that tweets from influential accounts or viral posts have proportionally greater impact on the overall sentiment score.

D. Feature Engineering

Our feature engineering pipeline creates over 50 features across multiple categories to capture different market dynamics:

Price-based Features: Daily returns, log returns, high-low spread, open-close spread, price range percentages, and intraday volatility. These features capture immediate price movements and market microstructure.

Technical Indicators: Moving Averages (MA-5, 10, 20, 50, 200), Exponential Moving Averages (EMA-12, 26), Bollinger Bands (20-period, 2), RSI (14-period), MACD (12, 26, 9), Stochastic Oscillator, ATR (Average True Range), ADX (Average Directional Index), and CCI (Commodity Channel Index). These indicators are standard in technical analysis and have proven predictive power.

Volume Features: Volume moving averages, volume ratio to average, volume-price correlation, On-Balance Volume (OBV), and Volume Rate of Change (VROC). Volume analysis helps identify the strength of price movements and potential trend reversals.

Sentiment Features: Daily average sentiment scores, sentiment volatility (7-day, 30-day rolling standard deviation), sentiment momentum (rate of change in sentiment), sentiment acceleration (second derivative), engagement-weighted sentiment, tweet count per day, positive/negative tweet ratio, and average subjectivity score. These features capture both the level and dynamics of public sentiment.

Lag Features: Lag features for 1, 2, 3, 5, and 10 periods are created for Close price, Volume, and sentiment scores to capture temporal dependencies. This allows the model to learn patterns across multiple time horizons.

All features are standardized using z-score normalization to ensure comparable scales across different feature types. Features with variance below a threshold of 0.01 are removed to eliminate uninformative variables.

E. Model Training

LightGBM Model: LightGBM uses gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) for efficient training. Configuration includes 1000 estimators, maximum depth of 6, learning rate of 0.1, subsample ratio of 0.8, and feature fraction of 0.8. The leaf-wise growth strategy

enables LightGBM to achieve lower loss compared to level-wise algorithms.

XGBoost Model: XGBoost implements a regularized gradient boosting framework with L1 and L2 regularization. Configuration includes 1000 estimators, maximum depth of 6, learning rate of 0.1, subsample ratio of 0.8, and column sampling of 0.8. The regularization parameters (alpha=0.1, lambda=1.0) help prevent overfitting on the training data.

Both models use early stopping with patience of 50 rounds to prevent overfitting. Hyperparameters were optimized using Bayesian optimization with 5-fold time-series cross-validation. The optimization process evaluated over 200 different hyperparameter configurations to find the optimal settings.

Ensemble Prediction: Final predictions use simple averaging:

$$P_{final} = \frac{P_{LightGBM} + P_{XGBoost}}{2} \tag{3}$$

We also experimented with weighted averaging and stacking approaches, but found that simple averaging provided the best balance between performance and computational efficiency.

F. Model Evaluation

Models are evaluated using time-series cross-validation to respect temporal ordering of data. Evaluation metrics include Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R^2 Score, Mean Absolute Percentage Error (MAPE), and directional accuracy. Directional accuracy measures the percentage of correctly predicted price movement directions (up/down), which is particularly relevant for trading applications.

IV. Implementation Details

A. System Requirements and Deployment

The complete system is implemented in Python 3.8+ using scikit-learn, pandas, numpy, and the respective XGBoost and LightGBM libraries. The Flask API is containerized using Docker, enabling easy deployment across different environments. The production deployment runs on a cloud infrastructure with 4 CPU cores and 16GB RAM, which provides sufficient resources for real-time prediction with sub-second latency.

B. Data Storage and Management

Historical market data and tweet archives are stored in a PostgreSQL database with appropriate indexing for time-series queries. The database schema is optimized for fast retrieval of recent data, with partitioning by date to improve query performance. Trained model artifacts are versioned and stored using MLflow, enabling easy rollback to previous versions if needed.

C. Challenges and Solutions

Several challenges were encountered during implementation:

Data Quality: Missing data during market holidays and weekends required careful handling. We implemented forward-fill strategies with maximum gap limits to avoid propagating stale information too far into the future.

API Rate Limits: Twitter API rate limits constrained data collection. We implemented distributed collection across multiple API keys and intelligent caching to minimize redundant requests.

Real-time Processing: Maintaining low latency for predictions required optimization of the feature computation pipeline. We implemented incremental feature updates that only recompute changed values rather than recalculating all features from scratch.

Model Drift: Financial markets evolve over time, causing model performance degradation. We implemented automated weekly retraining and performance monitoring with alerts for significant accuracy drops.

V. Results and Discussion

A. Dataset Description

The dataset comprises 100-200 days of historical stock data per symbol (January 2023 - June 2023), averaging 150-300 tweets per day per stock, with 52 engineered features. Training used 80% of data, validation 10%, and testing 10%, totaling approximately 15,000 data points across all stocks. The dataset exhibits natural class imbalance with approximately 52% up days and 48% down days, reflecting realistic market conditions.

B. Model Performance

Table I shows performance metrics aggregated across all stock symbols.

TABLE I
Model Performance Comparison

Model	RMSE	MAE	R^2	MAPE
LightGBM	2.34	1.87	0.892	2.41%
XGBoost	2.51	1.95	0.878	2.58%
Ensemble	**2.28**	**1.82**	**0.901**	**2.35%**
Linear Reg.	3.87	3.12	0.745	4.12%
Random Forest	2.94	2.35	0.832	3.08%
LSTM	2.62	2.08	0.865	2.73%

The ensemble model achieved the best performance with an R^2 score of 0.901 and directional accuracy of 68.5%. Predictions are typically within 2-3% of actual prices. The ensemble significantly outperforms baseline methods, showing 20.9% improvement over Linear Regression and 4.2% over LSTM in R^2. The superior performance of gradient boosting methods validates their suitability for financial time-series prediction with heterogeneous features.

C. Feature Importance Analysis

SHAP (SHapley Additive exPlanations) analysis revealed feature importance distribution: lag features (28%), moving averages (18%), sentiment features (16%), volume features (12%), technical indicators (14%), and other features (12%).

Among sentiment features, engagement-weighted sentiment contributed 6.2%, sentiment momentum 4.1%, and tweet count 2.8%. This distribution confirms that while sentiment is valuable, it complements rather than replaces traditional technical features.

Interestingly, the 5-day lag of close price was the single most important feature (8.3

D. Impact of Sentiment Analysis

Table II quantifies the contribution of sentiment features to prediction accuracy.

TABLE II
IMPACT OF SENTIMENT FEATURES

Configuration	R^2	RMSE
Technical indicators only	0.821	2.58
Sentiment features only	0.612	3.87
Combined (full model)	**0.901**	**2.28**

Including sentiment features improved R^2 from 0.821 to 0.901 (approximately 9.7% improvement), validating that public sentiment contains valuable predictive information beyond what is captured by technical indicators alone. This improvement was statistically significant (p ¡ 0.01) based on paired t-tests across different stocks and time periods. Notably, sentiment features alone achieve only moderate performance, confirming that they are most valuable when combined with traditional features.

E. Performance Across Market Conditions

Analysis across different market conditions showed varying model performance: high volatility periods (R^2 = 0.875), low volatility periods (R^2 = 0.918), trending markets (R^2 = 0.932), and range-bound markets (R^2 = 0.891). Performance degraded slightly during high volatility, consistent with increased market unpredictability during these periods. The model performed best in trending markets where momentum patterns are clearer.

F. Stock-Specific Performance

Performance varied across different stocks, with technology stocks (INFY, TCS, WIPRO) showing higher R^2 scores (0.915-0.925) compared to banking stocks (0.885-0.900). This difference may be attributed to higher social media engagement and more active retail trading in technology stocks, providing richer sentiment signals.

G. API Deployment and Production Performance

The Flask API demonstrates production readiness with average response time of 420ms for single-stock predictions and 1.2s for multi-stock batch predictions. The system maintains throughput of 15-20 requests per second with 99.2% uptime over three months of operation. Support for 12+ NSE stocks enables comprehensive market coverage for traders. Automatic weekly model retraining ensures adaptation to evolving market conditions without manual intervention.

VI. CONCLUSION AND FUTURE WORK

This paper presented a comprehensive sentiment analysis system for stock price prediction that successfully integrates social media sentiment with technical indicators. The hybrid approach using LightGBM and XGBoost ensemble methods achieved strong predictive performance with an R^2 score of 0.901 and directional accuracy of 68.5%.

Key findings include: (1) Sentiment analysis significantly improves prediction accuracy by 8-12% when combined with technical indicators, (2) Ensemble methods consistently outperform individual models and traditional baselines, (3) Real-time integration is feasible for production deployment with sub-second prediction latency, (4) The system scales effectively to multiple stock symbols while maintaining performance, and (5) Engagement-weighted sentiment provides more predictive power than simple sentiment averaging.

The practical implications of this research extend to both individual investors and institutional traders. The system can provide decision support for entry and exit timing, help identify sentiment-driven price movements, and enable more informed risk management strategies.

Future work will focus on several directions: (1) Incorporating additional data sources including news articles, financial reports, and earnings call transcripts, (2) Implementing advanced deep learning architectures such as Transformers for sentiment analysis and attention mechanisms to weight different information sources dynamically, (3) Expanding to multiple markets beyond NSE including global exchanges and cryptocurrency markets, (4) Developing automated trading strategies based on predictions with proper risk management and position sizing, and (5) Investigating causal relationships between sentiment and price movements to better understand market dynamics.

Additionally, we plan to explore adversarial robustness of the sentiment analysis component against market manipulation attempts and develop methods to detect and filter coordinated social media campaigns. Integration with fundamental analysis data such as earnings reports and financial ratios could further enhance prediction accuracy.

ACKNOWLEDGMENT

The authors would like to thank Sardar Vallabhbhai National Institute of Technology for providing the necessary resources and infrastructure for this research. We also acknowledge the valuable feedback from anonymous reviewers that helped improve the quality of this paper.

REFERENCES

[1] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," Journal of Computational Science, vol. 2, no. 1, pp. 1-8, 2011.

[2] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[3] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems, 2017, pp. 3146-3154.

[4] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, no. 1, 2014.

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

[6] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news," ACM Transactions on Information Systems, vol. 27, no. 2, pp. 1-19, 2009.

[7] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," Journal of Finance, vol. 25, no. 2, pp. 383-417, 1970.

[8] R. J. Shiller, "From efficient markets theory to behavioral finance," Journal of Economic Perspectives, vol. 17, no. 1, pp. 83-104, 2003.

[9] T. O. Sprenger et al., "Tweets and trades: The information content of stock microblogs," European Financial Management, vol. 20, no. 5, pp. 926-957, 2014.

[10] X. Ding et al., "Deep learning for event-driven stock prediction," in Proceedings of the 24th International Joint Conference on Artificial Intelligence, 2015, pp. 2327-2333.

[11] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," Journal of Finance, vol. 62, no. 3, pp. 1139-1168, 2007.

[12] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654-669, 2018.

[13] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001.

[14] J. J. Murphy, Technical Analysis of the Financial Markets. New York Institute of Finance, 1999.

[15] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[16] L. Prechelt, "Early stopping-but when?" in Neural Networks: Tricks of the Trade, Springer, 2012, pp. 53-67.

[17] J. Bergstra et al., "Algorithms for hyper-parameter optimization," in Advances in Neural Information Processing Systems, 2011, pp. 2546-2554.

[18] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," International Journal of Forecasting, vol. 22, no. 4, pp. 679-688, 2006.

[19] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, 2017, pp. 4765-4774.

[20] D. Bertsimas and A. W. Lo, "Optimal control of execution costs," Journal of Financial Markets, vol. 1, no. 1, pp. 1-50, 1998.