International Journal of Science, Strategic Management and Technology
Volume **02** Issue **06** June-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Stock Market Prediction and Sentiment Analysis

Vilas Anil Bachhav, Aditya Avinash Sabnis, Piyush Vilas Dhabale, Akshay Ramdas Ambhore

Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Pune, India

https://doi.org/10.55041/ijsmt.v2i6.204

Cite this Article: Bachhav, V. A., Sabnis, A. A., Dhabale, P. V. & Ambhore, A. R. (2026). Stock Market Prediction and Sentiment Analysis. International Journal of Science, Strategic Management and Technology, *02*(6). https://doi.org/10.55041/ijsmt.v2i6.204

Abstract—

The increasing role of social media in financial discussions has created a need to understand how public sentiment can influence stock-related decision-making. This paper presents a Sentiment-Driven Stock Market Simulator, a web-based platform that demonstrates the relationship between user-generated tweets and simulated stock price movement. The system allows registered users to post stock-related tweets, which are processed through an external AI-based sentiment analysis service. The returned sentiment score is combined with engagement factors such as likes and comments to calculate an impact score. This impact score influences the simulated stock price, while a rule-based prediction engine generates a predicted price for comparison. The application uses React, Node.js, Express.js, MongoDB, Socket.IO, and Recharts to support full-stack development, database storage, real-time updates, and live chart visualization. The system also uses Mean Absolute Error to compare the actual simulated price with the predicted price. The proposed work is intended as an educational and demonstrative FinTech platform rather than a real-world trading or investment prediction tool.

Keywords— Sentiment Analysis; Stock Market Simulation; Real-Time Systems; Socket.IO; FinTech; Rule-Based Prediction

1. Introduction

Stock market behavior is influenced by several internal and external factors such as company performance, investor confidence, financial news, economic conditions, and public opinion. In recent years, social media platforms have become an important source of financial discussion, where users express opinions about companies, products, industries, and market expectations. These discussions can shape investor perception and may contribute to short-term market movement. As a result, sentiment analysis of social media content has become an important area of research in financial technology and market behavior studies [1], [2].

Traditional stock analysis methods mainly depend on historical price data, trading volume, technical indicators, and financial statements. Although these methods are useful for structured financial analysis, they may not clearly show the immediate effect of public opinion on stock behavior. In real-world markets, it is also difficult to isolate the effect of one tweet or one sentiment-based event because actual stock prices are influenced by many factors at the same time. Therefore, a controlled simulation environment can help students, researchers, and finance enthusiasts understand the relationship between public sentiment and stock price movement in a simple and observable way.

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **06** June-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Sentiment analysis is a natural language processing technique used to identify the emotional tone of text as positive, negative, or neutral. In financial applications, sentiment analysis can be applied to tweets, blogs, investor comments, and news articles to understand the general opinion about a company or market sector [4], [5]. However, sentiment analysis alone cannot guarantee accurate stock prediction because real-world financial markets are highly complex and uncertain. For this reason, the proposed system does not attempt to predict actual market prices. Instead, it uses sentiment analysis as an input for a controlled stock market simulation.

The proposed project is a real-time web application that simulates stock price fluctuations based on AI-analyzed sentiment from user-generated tweets. The system allows users to post stock-related tweets with a stock symbol. The backend sends the tweet content to a sentiment analysis service, receives a sentiment score, calculates an impact score, and updates the corresponding simulated stock price. At the same time, a rule-based prediction engine calculates a predicted price using recent trend, sentiment score, and impact score.

The main objective of this work is to provide an educational and demonstrative platform for understanding the effect of social media sentiment on simulated stock price movement. The system also calculates Mean Absolute Error to compare the actual simulated price with the rule-based predicted price [8]. This helps users observe how closely the prediction engine follows the simulated price movement within the controlled environment.

1.1 System Overview

The proposed Real-Time Sentiment-Driven Stock Market Simulator is a modular web-based platform that connects user-generated content, sentiment analysis, rule-based simulation, prediction logic, database storage, and real-time visualization within a single workflow. The application is organized into presentation, backend, processing, and database layers.

A. Core System Components

1) User Authentication and Access Management — Registered users can create accounts and log in securely. JSON Web Tokens are used for protected operations such as creating tweets, blogs, likes, and comments.

2) Tweet and Blog Management — The React frontend allows users to create stock-related tweets, view tweet details, manage profile information, like tweets or blogs, and comment on tweets. These interactions act as input events for the simulation workflow.

3) Sentiment Analysis Integration — The backend sends tweet text to an AI-based sentiment analysis module. The returned sentiment score is used to determine the direction and strength of user opinion.

4) Impact-Based Stock Simulation — The impact engine combines sentiment score and engagement information to calculate an impact score. The stock simulator uses this score with market noise to update the actual simulated stock price.

5) Rule-Based Prediction and Error Calculation — A prediction engine estimates the next simulated stock price using recent trend, sentiment score, and impact score. Mean Absolute Error is used to compare predicted price with actual simulated price.

6) Database and Live Output — MongoDB stores users, tweets, stocks, social profiles, and stock history. The frontend uses stored and updated stock data to display current price, simulated movement, predicted price, and event history.

B. Web Application Features

- Secure user registration and login using JWT-based authentication.
- Creation of stock-related tweets and comments through a React-based interface.
- Sentiment-driven stock simulation based on tweet content and engagement.
- Rule-based prediction for comparing expected movement with simulated movement.
- Database-backed storage of users, tweets, stocks, social profiles, and simulation history.
- Interactive visualization of actual simulated price and predicted price.

C. Backend Integration

The backend is implemented using Node.js and Express.js. It handles authentication, API routing, tweet processing, sentiment analysis requests, impact calculation, stock updates, and database communication. MongoDB is used as the storage layer, while the simulation services maintain the logical connection between tweet sentiment and stock price movement.

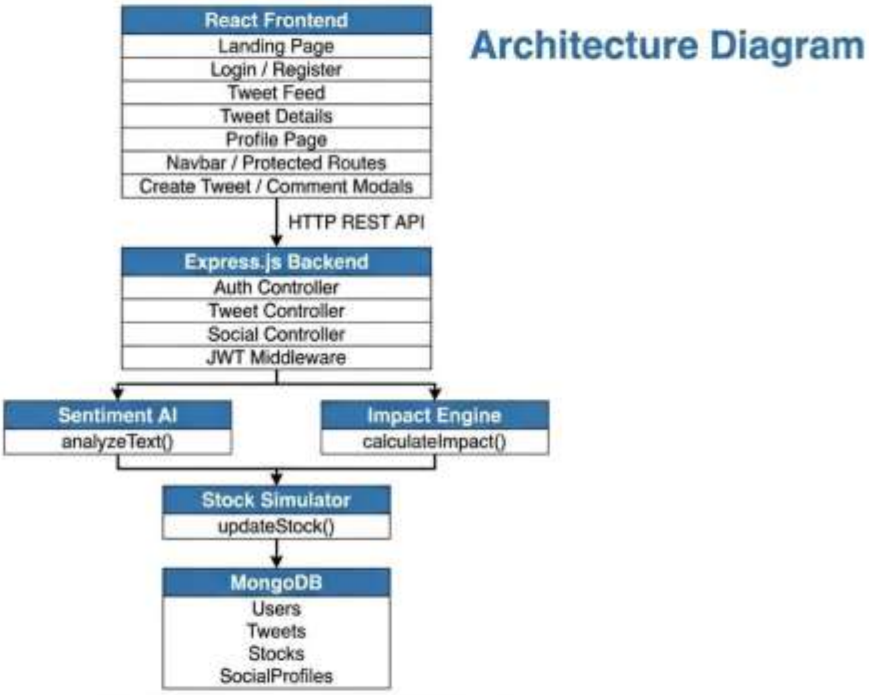

Figure 1. Architecture Diagram of the Sentiment-Driven Stock Market Simulator

2. Literature Survey

Bollen, Mao, and Zeng investigated whether collective mood extracted from Twitter could be used to predict movements in the Dow Jones Industrial Average. Their work demonstrated that public mood signals from social media can provide useful information for financial indicators, but their study focused on real market prediction using large-scale Twitter data and mood tracking tools [1].

Mao, Counts, and Bollen compared survey data, news, Twitter data, and search engine data for financial market prediction. Their study highlighted the value of online behavioral signals in understanding market movement, but it also showed that prediction results depend strongly on the selected data source, sentiment measurement method, and market indicator [2].

Schumaker and Chen proposed the AZFinText system, which used financial news articles for stock market prediction. Their work demonstrated the importance of textual information in financial forecasting, especially when market reactions are influenced by breaking news. However, their approach was based on financial news rather than user-generated social media content [3].

Pagolu et al. applied sentiment analysis to Twitter data for predicting stock market movements. Their work supports the idea that public opinion expressed in tweets can be useful for financial analysis. However, such systems require carefully selected datasets and validation when used for real-world prediction [4].

Ranco et al. studied the effects of Twitter sentiment on stock price returns and found that sentiment may have stronger relationships with abnormal returns during high-volume Twitter events. This supports the need to observe sentiment-driven changes over time and to track the source of events that influence price behavior [5].

Sprenger et al. studied stock microblogs and showed that user-generated financial messages can contain information useful for market-related analysis. Their work supports the use of microblog content as a financial signal, but the proposed system differs by using such content inside a controlled simulation instead of claiming direct real-market prediction [6].

The reviewed works show that sentiment analysis, financial text mining, and social media signals have strong research relevance in financial technology. However, many existing systems focus on real-market prediction and require large datasets, advanced models, and complex validation. The proposed system addresses a different gap by creating a controlled, real-time simulator that visually demonstrates the relationship between tweet sentiment, impact score, simulated stock price, prediction error, and event history.

Paper/System	**Technique / Approach**	**Domain**	**Strength**	**Limitation**
Bollen et al. [1]	Twitter mood analysis	Market index prediction	Shows relation between public mood and market indicators	Requires large-scale Twitter data and mood tracking tools
Mao et al. [2]	Survey, news, Twitter, and search data comparison	Financial market prediction	Compares multiple online data sources	Results depend on selected data source and indicator
Schumaker and Chen [3]	Financial news text mining	Stock prediction	Uses textual information from breaking news	Focuses on news, not user-generated tweets
Pagolu et al. [4]	Twitter sentiment and machine learning	Stock movement analysis	Uses tweet sentiment for market movement study	Requires real datasets and validation
Ranco et al. [5]	Twitter sentiment and event study	Stock return analysis	Highlights impact during high-volume events	Not designed as an educational simulator
Proposed System	Sentiment-driven simulation and rule-based prediction	FinTech simulation	Provides real-time visualization and event history	Simulated output, not real market prediction

Table 1. Comparative Analysis of Reviewed Literature

3. Methodology and System Integration

The proposed system follows a structured pipeline that begins with user-generated content and ends with live stock visualization. The methodology includes tweet collection, sentiment analysis, impact calculation, stock simulation, rule-based prediction, database storage, and result visualization.

A) Tweet Processing and Data Collection — A logged-in user creates a stock-related tweet by entering tweet content and a stock symbol. The frontend sends this data to the backend through an HTTP REST API. The backend validates the request, stores the tweet, and starts the simulation process.

B) Sentiment Analysis Pipeline — The tweet text is passed to the AI sentiment analysis module using an analyzeText-style operation. The sentiment score returned by this module represents the tone of the tweet and becomes an important input for the impact calculation stage.

C) Impact Score Calculation — The impact engine converts sentiment and engagement information into an impact score. Positive sentiment creates upward pressure in the simulator, while negative sentiment creates downward pressure. Engagement factors such as likes and comments help represent the strength of the event.

D) Stock Simulation Logic — The stock simulator updates the actual simulated stock price using the impact score and controlled market noise. This keeps the stock movement dynamic while allowing the effect of sentiment-based events to remain visible.

E) Rule-Based Prediction and MAE Calculation — The prediction engine estimates a predicted price based on recent trend, sentiment score, and impact score. The system compares the predicted price with the actual simulated price using Mean Absolute Error. This metric helps show how close the rule-based prediction is within the simulation environment.

F) Data Storage and Visualization — MongoDB stores stock history, tweet data, user data, and related records. The frontend displays the current stock state, actual simulated price, predicted price, and event history in a user-friendly form.

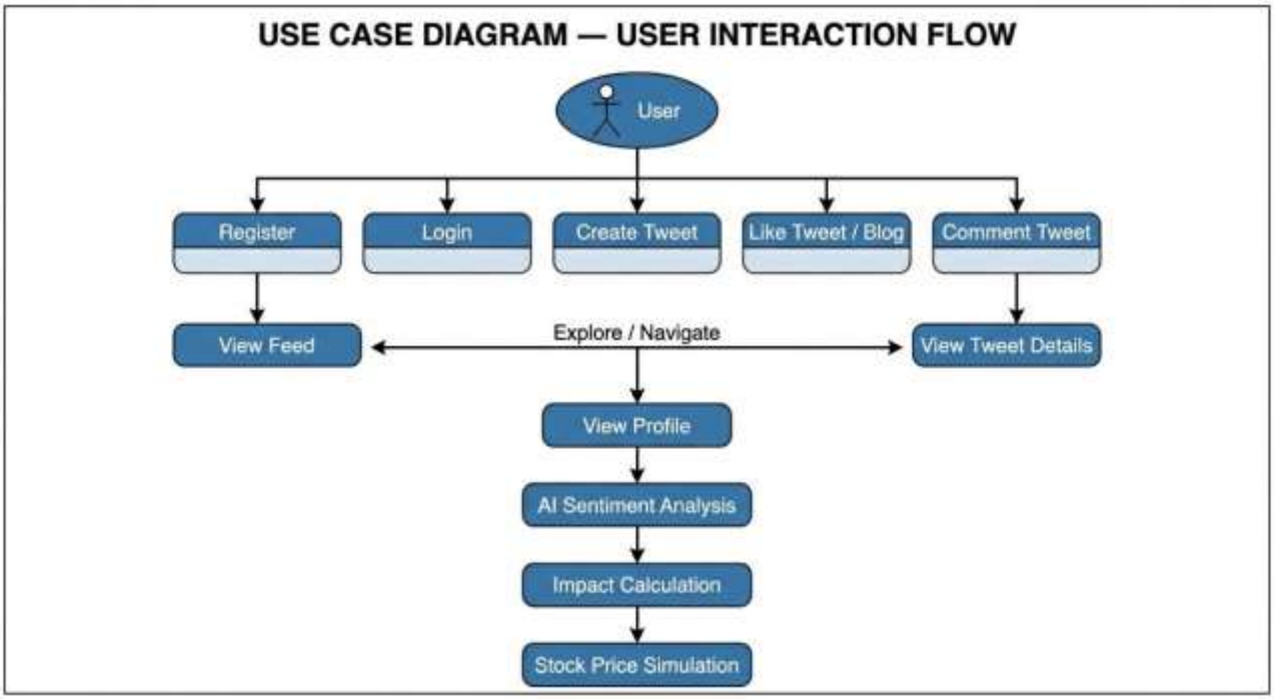

Figure 2. Use Case Diagram - User Interaction Flow

4. Application Workflow

A) User Interaction — The user accesses the application, registers or logs in, views the feed, creates tweets, likes tweets or blogs, comments on tweets, and views profile or tweet details. These actions create meaningful user interaction data for the platform.

B) Tweet Submission and Sentiment Processing — When the user creates a tweet, the backend receives the tweet content and associated stock symbol. The tweet content is analyzed by the sentiment AI module, which returns a sentiment score for further processing.

C) Impact and Simulation — The impact engine calculates the influence of the tweet using sentiment and engagement signals. The stock simulator then updates the corresponding simulated stock value using the calculated impact score.

D) Prediction and Error Tracking — The rule-based prediction engine generates a predicted stock price. The difference between the actual simulated price and the predicted price is stored as prediction error and summarized using Mean Absolute Error.

E) Database Storage and Display — MongoDB stores users, tweets, stocks, social profiles, and stock history. The frontend displays the latest stock state and allows users to understand how tweet-based events affected the simulated stock movement.

Step	Operation	Output
1	User registers or logs in	Authenticated user session
2	User creates a stock-related tweet	Tweet content and stock symbol
3	Backend processes the tweet	Stored tweet and simulation trigger
4	Sentiment AI analyzes text	Sentiment score
5	Impact engine calculates effect	Impact score
6	Stock simulator updates price	Actual simulated price
7	Prediction engine estimates price	Predicted price and error
8	Database stores history	Time-series event record
9	Frontend displays output	Live stock visualization and history

Table 2. Application Workflow of the Proposed System

4.1 Feasibility and Scope

A) Practical Feasibility — The system is practical for educational demonstrations, academic projects, and FinTech learning because it does not require real trading accounts or live exchange access. Users can observe sentiment impact safely within a simulated environment.

B) Economic Feasibility — The project uses widely available web technologies and open-source frameworks such as React, Node.js, Express.js, MongoDB, Socket.IO, and Recharts. This reduces development cost and makes the system suitable for student-level implementation.

C) Technical Feasibility — The system architecture separates frontend, backend, sentiment processing, impact calculation, stock simulation, prediction, and database storage. This modular structure makes the project easier to maintain and extend.

D) Scope — The current scope is limited to simulated stock movement based on user-generated tweets, engagement metrics, and rule-based prediction. Future versions can add real market APIs, trained forecasting models, backtesting, user influence scoring, and advanced technical indicators.

Layer	Technology	Purpose

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **06** June-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Frontend	React	User interface, tweet feed, profile, and stock visualization
Backend	Node.js and Express.js	REST API, controllers, middleware, and business logic
Database	MongoDB	Storage of users, tweets, stocks, social profiles, and history
Authentication	JWT	Protected access for registered users
AI/Sentiment Layer	Sentiment AI module	Tweet sentiment score generation
Processing Layer	Impact engine and stock simulator	Impact calculation and simulated price update
Prediction Layer	Rule-based prediction engine	Predicted price and error calculation
Visualization	Chart-based frontend output	Display of actual and predicted stock movement

Table 3. Technology Stack Used in the Proposed System

5. Results and Discussion

The proposed system was evaluated based on functional behavior, data flow, simulation output, and visualization capability. When a registered user creates a stock-related tweet, the system processes the tweet, analyzes sentiment, calculates impact, updates the simulated stock value, stores the generated history, and presents the result through the frontend interface. This confirms that the system successfully links user-generated sentiment with simulated stock movement.

The major output of the system is the comparison between actual simulated price and predicted price. The actual simulated price is generated through the stock simulator using impact score and controlled market noise, while the predicted price is produced through rule-based logic. The difference between both values is represented using Mean Absolute Error, which provides a simple numerical indication of prediction difference inside the simulator.

The result should be interpreted as a simulation-based output and not as a real-world stock prediction result. The simulator is useful because it makes the abstract relationship between social media sentiment and price movement visible to learners. It also provides transparency by maintaining history records for tweets, sentiment scores, impact scores, stock changes, and prediction error.

Module	**Output Produced**	**Discussion**
Authentication	Registered and logged-in user access	Protects user actions and platform features
Tweet Processing	Tweet content with stock symbol	Acts as the primary sentiment event
Sentiment Analysis	Sentiment score	Converts text opinion into a numerical signal
Impact Engine	Impact score	Represents the strength and direction of tweet influence
Stock Simulator	Actual simulated stock price	Shows how sentiment affects simulated stock movement
Prediction Engine	Predicted price and error	Allows comparison with simulated movement

Database	Stored users, tweets, stocks, and history	Maintains traceable event records
Visualization	Live stock chart and event information	Improves understanding through graphical output

Table 4. Functional Results and Discussion

MAE = (1 / n) Σ |Actual Simulated Price - Predicted Price|

The use of MAE is suitable for this simulator because the system compares two numerical values: actual simulated stock price and predicted stock price. A smaller MAE indicates that the rule-based predicted price is closer to the simulated price, while a larger MAE indicates a greater difference.

6. Advantages and Limitations

A) Advantages —

- Provides a simple and controlled way to understand sentiment-driven stock movement.
- Uses user-generated tweets to create observable simulation events.
- Combines sentiment score, engagement data, impact calculation, and stock simulation.
- Compares actual simulated price with rule-based predicted price using MAE.
- Stores event history for transparency and later analysis.
- Uses a modular full-stack architecture that can be extended in future work.

B) Limitations —

- The system is a simulator and is not connected to real stock exchanges or live market APIs.
- The prediction engine is rule-based and not a trained machine learning forecasting model.
- The sentiment analysis module is treated as an external or independent AI component.
- The output should not be used for real investment decisions or trading advice.
- The current simulation does not consider wider financial factors such as company fundamentals, economic events, and real trading volume.
- Large-scale deployment would require optimization of real-time events, database indexing, and stock update processing.

7. Conclusion and Future Work

The Real-Time Sentiment-Driven Stock Market Simulator demonstrates how user-generated social media content can influence simulated stock price movement in a controlled environment. The system connects tweet creation, sentiment analysis, impact calculation, stock simulation, rule-based prediction, database storage, and visual output into a complete full-stack web application.

The project provides a useful educational platform for understanding the relationship between sentiment and stock price behavior without depending on actual stock market data or real trading activity. By comparing actual simulated price and predicted price using Mean Absolute Error, the system allows users to observe the behavior and limitations of a rule-based prediction engine inside the simulator.

Future work can improve the system by replacing the rule-based prediction engine with a trained forecasting model such as ARIMA or LSTM. The sentiment analysis module can also be integrated directly into the main application. Additional factors such as user influence, follower count, company news, real market APIs, technical indicators, and backtesting can be added to make the simulator more realistic and analytically useful.

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] H. Mao, S. Counts, and J. Bollen, "Predicting financial markets: Comparing survey, news, Twitter and search engine data," arXiv preprint arXiv:1112.1051, 2011.

[3] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The AZFinText system," ACM Transactions on Information Systems, vol. 27, no. 2, Article 12, 2009.

[4] V. S. Pagolu, K. N. R. Challa, G. Panda, and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," arXiv preprint arXiv:1610.09225, 2016.

[5] G. Ranco, D. Aleksovski, G. Caldarelli, M. Grčar, and I. Mozetič, "The effects of Twitter sentiment on stock price returns," PLoS ONE, vol. 10, no. 9, e0138441, 2015.

[6] T. O. Sprenger, A. Tumasjan, P. G. Sandner, and I. M. Welpe, "Tweets and trades: The information content of stock microblogs," European Financial Management, vol. 20, no. 5, pp. 926-957, 2014.

[7] B. Liu, Sentiment Analysis and Opinion Mining. San Rafael, CA, USA: Morgan & Claypool Publishers, 2012.

[8] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error over the root mean square error in assessing average model performance," Climate Research, vol. 30, no. 1, pp. 79-82, 2005.

[9] Express.js, "Express.js Documentation: Node.js Web Application Framework," OpenJS Foundation. [Online]. Available: https://expressjs.com/

[10] Socket.IO, "Socket.IO Documentation," Socket.IO. [Online]. Available: https://socket.io/docs/

[11] MongoDB, "MongoDB Manual," MongoDB Inc. [Online]. Available: https://www.mongodb.com/docs/

[12] Recharts, "Recharts Documentation," Recharts. [Online]. Available: https://recharts.org/