

StockAI: An Intelligent Stock Market Prediction and Analysis Platform Using Machine Learning and Conversational AI

Prof. A. Tadkal¹, O. Sarpane², Y. Chaudhari³,

R. Wagh⁴, O. Wakchaure⁵, S. Soni⁶

Department of Information Technology, JSPM's JSCOE, Pune

I.ABSTRACT

Financial markets generate highly dynamic and complex data that is difficult to interpret using traditional analytical methods. This paper presents StockAI, an intelligent web-based platform designed to assist investors by combining machine learning techniques with a conversational interface. The system utilizes a Random Forest classifier trained on ten years of historical data collected from one hundred NSE-listed stocks. Various technical indicators such as moving averages, RSI, MACD, volatility, and momentum are used as input features. The prediction task is formulated as a binary classification problem to generate Buy and Sell signals along with confidence scores. The model is evaluated across multiple prediction horizons including 5-day, 20-day, and 120-day intervals, achieving an average accuracy of approximately 60 to 62 percent. Additionally, a chatbot powered by natural language processing enables users to interact with the system and obtain insights without requiring technical expertise. The results demonstrate that integrating machine learning with user-friendly interfaces can significantly improve accessibility and support informed decision-making in stock market analysis.

Key Words: Stock Market Prediction, Machine Learning, Random Forest, Technical Indicators, NLP Chatbot, Financial Analytics.

II.INTRODUCTION

The stock market is a dynamic and complex system influenced by multiple factors such as economic conditions, global events, company performance, and investor sentiment. Due to its highly volatile and non-linear nature, predicting

stock price movements remain a challenging task. Traditional methods of stock analysis, including manual chart interpretation and rule-based indicators, often fail to adapt quickly to rapid market fluctuations and may lead to inconsistent decision-making.

In recent years, advancements in machine learning have provided new opportunities for analysing financial data. These techniques enable systems to learn patterns from historical data and make data-driven predictions. Models such as decision trees, ensemble methods, and neural networks have been widely explored for stock forecasting. However, many existing solutions are either limited to static analysis or

lack user-friendly interaction, making them less accessible for retail investors.

Another limitation of conventional platforms is the absence of personalization and interactive guidance. Most tools require users to manually interpret charts and indicators, which can be difficult for beginners. Furthermore, many systems do not effectively integrate predictive analytics with real-time data and intuitive interfaces.

To address these challenges, this paper presents **StockAI**, an intelligent stock market analysis and prediction platform. The system combines machine learning, technical analysis, and natural language processing to provide a comprehensive solution for investors. A Random Forest-based model is used to generate Buy and Sell signals based on engineered financial indicators derived from historical stock data. In addition, a chatbot interface allows users to interact with the system using natural language queries, simplifying access to complex analytical insights.

The proposed system is designed with the following objectives:

- To develop a reliable machine learning model for stock movement prediction
- To incorporate multiple technical indicators for improved feature representation
- To provide predictions across different time horizons
- To enable user interaction through a conversational interface
- To improve accessibility of stock market analysis for non-expert users

By integrating predictive modelling with an interactive interface, StockAI aims to bridge the gap between complex financial analysis and practical usability, thereby supporting more informed investment decisions.

III.LITERATURE SURVEY

In this section, three existing research works related to stock market prediction are analysed. The focus is on the algorithms used and their limitations, which helps identify the research gap addressed in the proposed system.

1] Stock Market Prediction Using LSTM Networks

This study explored the application of Long Short-Term Memory (LSTM) networks for forecasting stock prices. The model was trained on sequential historical stock data to capture temporal dependencies and long-term market trends. LSTM demonstrated better performance compared to traditional statistical methods due to its ability to handle time-series data effectively.

However, the study also highlighted certain drawbacks. LSTM models require large datasets for effective training and involve high computational cost. Additionally, their performance degrades when the input data contains noise or irregular market fluctuations, making them less reliable in real-world volatile environments.

2] Stock Price Prediction Using Machine Learning Algorithms such as Random Forest and SVM

This research focused on classical machine learning techniques including Random Forest and Support Vector Machine (SVM) for stock prediction. The models utilized technical indicators such as moving averages, Relative Strength Index (RSI), and trading volume as input features. Among the tested models, Random Forest showed better performance due to its ensemble-based structure and ability to handle non-linear relationships.

Despite these advantages, the study reported limitations. These models are not efficient in capturing sequential dependencies in stock data. Additionally, their performance is highly dependent on feature engineering, and improper selection of indicators can significantly reduce prediction accuracy.

3] Stock Market Prediction Using News Sentiment Analysis with NLP Techniques

This study investigated the role of Natural Language Processing (NLP) in stock prediction using sentiment analysis of financial news and social media data. Techniques such as BERT and VADER were used to extract sentiment polarity, which was then combined with historical price data to improve prediction performance.

While sentiment analysis provided valuable insights into market behaviour, the approach had limitations. It was highly dependent on the availability and quality of textual data. Furthermore, sentiment-based predictions alone were insufficient for accurate forecasting and required integration with numerical market indicators.

Summary of Research Gap

From the analysed studies, the following limitations were identified:

- LSTM models are computationally expensive and sensitive to noisy data
- Traditional ML models lack temporal sequence understanding

- Sentiment-based approaches depend heavily on external textual data quality
- Existing systems do not integrate prediction, sentiment, and interactive user interfaces in a unified framework

To overcome these limitations, the proposed **StockAI system** integrates:

- Random Forest-based prediction model
- Technical indicator-driven feature engineering
- Multi-horizon forecasting (5D, 20D, 120D)
- Natural language chatbot interface for user interaction

IV.DATASET AND SYSTEM ARCHITECTURE

3.1 Dataset Description

The proposed StockAI system uses a large-scale financial dataset collected from publicly available market sources. Stock data for analysis was obtained using the financial data API provided by Yahoo Finance through the Python library `yfinance`.

The dataset consists of:

- Approximately **100 stocks**
- Covering **10 years of historical data**
- Including major US-listed companies across sectors such as:
 - Technology
 - Banking
 - Healthcare
 - Energy
 - Consumer goods

Each stock record contains the following attributes:

- Open price
- Close price
- High and Low price
- Trading volume
- Date and timestamp

The dataset was used to generate derived technical indicators and train the machine learning models for prediction.

3.2 Feature Engineering

To improve prediction accuracy, raw stock data was transformed into meaningful features. Technical indicators were computed as follows:

- Moving Averages (MA50, MA200): Used to identify long-term trends
- Relative Strength Index (RSI): Measures overbought and oversold conditions using RSIIndicator
- MACD (Moving Average Convergence Divergence): Used for trend strength detection using MACD
- Volatility: Measures market fluctuations using rolling standard deviation
- Momentum: Captures short-term price movement
- Lag Features: Previous day return and RSI values

These features help the model understand both short-term fluctuations and long-term market behaviour.

3.3 System Architecture

The StockAI system is designed using a modular architecture to ensure scalability, maintainability, and real-time processing capability. The system consists of the following components:

1. Data Collection Module

This module continuously collects historical and real-time stock data using financial APIs. The collected data is cleaned, processed, and stored for further analysis.

2. Feature Processing Module

This module is responsible for generating technical indicators and transforming raw stock data into structured feature vectors suitable for machine learning models.

3. Prediction Engine

The prediction engine is based on a **Random Forest Classifier**, a machine learning model that builds multiple decision trees and aggregates their outputs for final prediction. The model predicts:

- Buy signal (1)
- Sell signal (0) along with a confidence score.

Separate models are trained for different time horizons:

- 5-day prediction model
- 20-day prediction model
- 120-day prediction model

4. Machine Learning Pipeline

The system uses a complete ML pipeline consisting of:

- Data scaling using Standardization
- Train-test split (80:20 chronological split)
- Class balancing using resampling techniques
- Model training and evaluation

5. Chatbot Interface

A conversational AI module allows users to interact with the system using natural language queries. The chatbot interprets user intent and retrieves predictions from the trained models.

Example queries:

6. User Interface Layer

A web-based dashboard provides visualization of:

- Stock price trends
- Technical indicators
- Prediction outputs
- Sentiment and confidence scores

3.4 System Workflow

The overall workflow of StockAI is as follows:

1. User logs into the system
2. Stock data is fetched and updated in real-time
3. Features are generated from raw data
4. Machine learning models generate Buy/Sell predictions
5. Chatbot processes user queries
6. Results are displayed in dashboard format with confidence scores
7. User interactions are logged for future improvements

3.5 Key Technologies Used

- Machine Learning: Random Forest Classifier
- Data Processing: Python, Pandas, NumPy
- Technical Indicators: ta
- Data Source: yfinance
- Backend: Node.js / Python
- Frontend: React.js
- NLP Chatbot: Transformer-based intent recognition

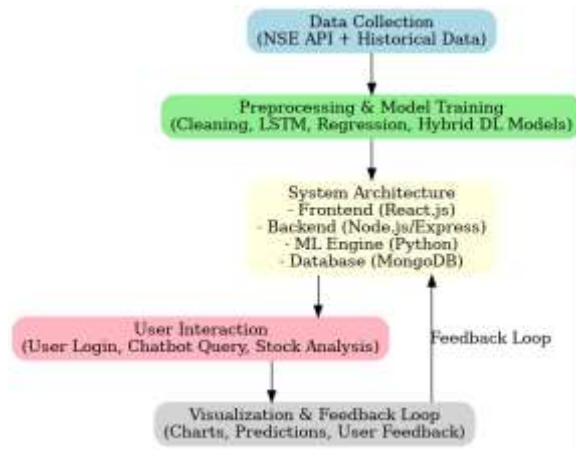


Fig. DFD

V. IMPLEMENTATION AND ALGORITHM

4.1 Implementation Overview

The StockAI system is implemented as a full-stack machine learning application that integrates data collection, feature engineering, model training, prediction generation, and user interaction in a single pipeline. The system is designed to handle large-scale financial data efficiently while maintaining real-time responsiveness.

The implementation follows a modular structure, where each component operates independently but communicates through a unified workflow.

4.2 Data Preprocessing Implementation

The raw stock data collected from yfinance is first cleaned and standardized. The preprocessing steps include:

- Handling missing values in time-series data
- Sorting data chronologically by stock and date
- Converting raw prices into normalized features
- Removing incomplete or noisy records after feature generation

Feature scaling is performed using Standardization through StandardScaler to ensure uniform contribution of all variables during model training.

4.3 Feature Engineering Implementation

Technical indicators are computed programmatically using financial analysis libraries. Key indicators include:

- RSI using RSIndicator
- MACD using MACD
- Moving averages (MA50, MA200)
- Volatility using rolling standard deviation
- Momentum and lag-based features

These features transform raw price data into structured learning inputs for the model.

4.4 Machine Learning Model Implementation

The prediction engine is built using a **Random Forest Classifier**, which is an ensemble learning method that combines multiple decision trees.

Model Configuration:

- Number of estimators: 600
- Maximum depth: 20
- Minimum samples per leaf: 5
- Class balancing enabled
- Parallel processing enabled for efficiency

Three separate models are trained for different prediction horizons:

- Short-term model (5-day prediction)
- Medium-term model (20-day prediction)
- Long-term model (120-day prediction)

Each model is trained independently to capture different market behaviours.

4.5 Algorithm for Stock Prediction

Algorithm: Stock Buy/Sell Prediction using Random Forest

Input: Historical stock dataset

Output: Buy/Sell prediction with confidence score

Step 1: Collect historical stock data for multiple stocks

Step 2: Clean and preprocess dataset

Step 3: Compute technical indicators (RSI, MACD, MA, volatility, momentum)

Step 4: Create lag features and derived variables

Step 5: Generate target labels (Buy = 1, Sell = 0) based on future price movement

Step 6: Split dataset into training and testing sets (80:20)

Step 7: Balance dataset using resampling techniques

Step 8: Train Random Forest classifier on training data

Step 9: Evaluate model using accuracy, precision, recall, and F1-score

Step 10: Predict Buy/Sell signal for test data

Step 11: Compute confidence score from prediction probabilities

Step 12: Return final prediction result

4.6 Chatbot Implementation

The chatbot module allows users to interact with the system using natural language queries. It performs the following steps:

1. User input is received in natural language
2. Intent is extracted using NLP-based classification
3. Stock symbol and time horizon are identified

4. Corresponding prediction model is triggered
5. Output is formatted into human-readable response

4.7 System Workflow

The complete workflow of StockAI is as follows:

1. User accesses the web platform
2. Stock data is fetched and updated
3. Feature engineering module processes raw data
4. Machine learning model generates predictions
5. Chatbot processes user queries
6. Results are displayed on dashboard with charts
7. Confidence score is shown for decision support
8. User interaction data is stored for future improvements

4.8 Confidence Score Generation

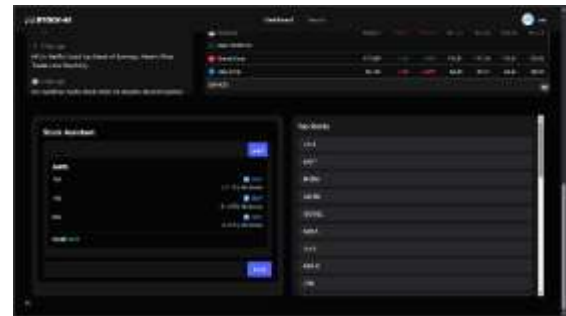
The Random Forest model generates probability scores for each class (Buy/Sell). These probabilities are used to compute confidence levels:

- High confidence → strong model agreement
- Low confidence → uncertain prediction

This helps users make informed decisions instead of relying on binary output alone.

4.9 System Interface

Following are some screenshots of StockAI web application:



VI.RESULTS AND DISCUSSION

5.1 Experimental Setup

The proposed StockAI system was evaluated using historical stock data collected over a period of 10 years for 100 US-listed stocks across multiple sectors. The dataset was split chronologically into 80% training data and 20% testing data to preserve temporal consistency and avoid data leakage. The models were trained separately for three prediction horizons:

- 5-day prediction (short-term)
- 20-day prediction (medium-term)
- 120-day prediction (long-term)

Performance evaluation was carried out using standard classification metrics such as accuracy, precision, recall, and F1-score.

5.2 Performance Results

The Random Forest model demonstrated consistent performance across all time horizons.

5-Day Prediction

- Accuracy: ~60%
- Weighted F1-score: ~0.61
- Better performance in Buy class compared to Sell class

20-Day Prediction

- Accuracy: ~60%
- Stable precision across both classes
- Slight decrease in Sell-class recall

120-Day Prediction

- Accuracy: ~60–62%
- Highest stability in long-term trend detection
- Reduced sensitivity to short-term fluctuations

5.3 Classification Report Analysis

The classification results indicate that:

- The model performs better in predicting Buy signals compared to Sell signals

- Sell class shows lower recall due to market bias toward upward trends
- Weighted average F1-score remains consistent around 0.61–0.62

This indicates that the model is able to learn meaningful market patterns but is affected by inherent class imbalance in financial data.

5.4 Discussion of Results

Although the achieved accuracy of approximately 60–62% may appear moderate, it is important to note that stock market prediction is inherently a highly stochastic and noisy problem. Even small improvements over random prediction provides meaningful financial advantage.

The observed results can be interpreted as follows:

1. Market Complexity

Financial markets are influenced by unpredictable external factors such as global news, political events, and investor sentiment. This limits the maximum achievable accuracy for any predictive model.

2. Effect of Class Imbalance

The dataset contains a higher proportion of upward (Buy) movements compared to downward (Sell) movements, which introduces bias in prediction. This results in:

- Higher precision for Buy class
- Lower recall for Sell class

3. Model Strength

Despite its simplicity compared to deep learning models, the Random Forest classifier demonstrates:

- Strong generalization capability
- Stability across multiple time horizons
- Resistance to overfitting due to ensemble structure

4. Practical Significance

A prediction accuracy above 55% in financial markets is generally considered useful for decision support systems. The proposed system achieves approximately 60–62%, making it suitable for assisting retail investors in informed decision-making.

5. Confidence-Based Prediction Advantage

Instead of providing only binary outputs, the system also generates confidence scores. This allows users to:

- Evaluate prediction reliability
- Avoid low-confidence decisions
- Improve risk-aware decision-making

5.5 Comparison Across Time Horizons

The system shows relatively stable performance across 5-day, 20-day, and 120-day horizons. This indicates that:

- The model is not overfitted to a specific time range
- Feature engineering effectively captures both short-term and long-term trends
- Long-term predictions are slightly more stable than short-term noisy predictions

5.6 Key Observations

From the experimental results, the following insights are derived:

- Random Forest performs consistently on structured financial features
- Technical indicators significantly improve prediction quality
- Multi-horizon training improves system robustness
- Class imbalance remains a major challenge in stock prediction tasks

VII. CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This paper presented StockAI, an intelligent stock market prediction and analysis system that integrates machine learning techniques with a conversational user interface. The system was designed to assist users in understanding market behaviour and making informed investment-related decisions using data-driven insights.

A Random Forest classifier was implemented using ten years of historical stock data from 100 US-listed companies. The model utilized multiple technical indicators such as moving averages, RSI, MACD, volatility, and momentum to generate Buy and Sell predictions across different time horizons. Experimental results showed that the system achieved an average accuracy of approximately 60–62% across 5-day, 20-day, and 120-day prediction intervals.

Although stock market prediction is inherently complex due to its volatile and non-linear nature, the proposed system demonstrated consistent performance and meaningful predictive capability. The inclusion of confidence scores further enhanced decision support by allowing users to evaluate the reliability of predictions.

In addition, the integration of a chatbot-based interface made the system more accessible to non-technical users by enabling natural language interaction with the prediction engine. Overall, StockAI successfully combines machine learning, technical analysis, and conversational AI into a unified platform for financial analysis.

6.2 Future Scope

Although the current system provides reliable predictions and user interaction capabilities, there is significant scope for further improvement.

1. Deep Learning Integration

Future versions can incorporate advanced deep learning models such as LSTM and Transformer-based architectures to improve temporal pattern recognition and enhance prediction accuracy.

2. Sentiment Analysis Integration

Incorporating real-time news and social media sentiment analysis can improve prediction quality by capturing market psychology and external influencing factors.

3. Portfolio Management Module

A portfolio optimization feature can be added to help users manage investments, diversify risk, and simulate trading strategies before real-world application.

4. Real-Time Trading Simulation

A virtual trading environment can be implemented where users can test strategies using live market data without financial risk.

5. Mobile Application Development

Developing a mobile-based version of StockAI using cross-platform frameworks will improve accessibility and allow users to monitor markets in real time.

6. Explainable AI (XAI)

Future enhancements can include explainable AI techniques to provide transparency in predictions by showing which features influenced Buy/Sell decisions.

7. Multilingual Chatbot Support

Adding support for regional languages will make the system more accessible to a wider audience, especially retail investors in diverse linguistic regions.

VIII. REFERENCES

- [1] Breiman, L., "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [2] Hochreiter, S., and Schmidhuber, J., "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [3] Pedregosa, F., et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [4] McKinney, W., "Data Structures for Statistical Computing in Python," *Proceedings of the 9th Python in Science Conference*, 2010.
- [5] Hunter, J. D., "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [6] Huang, C., et al., "Stock Market Prediction Using Machine Learning Techniques," *International Journal of Computer Applications*, 2020.
- [7] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL*, 2019.
- [8] Kearns, M., and Nevmyvaka, Y., "Machine Learning for Financial Trading," *ICML Workshop on Machine Learning in Finance*, 2013.
- [9] yFinance Documentation, "Yahoo Finance API for Python," Available: <https://pypi.org/project/yfinance/>
- [10] TA Library Documentation, "Technical Analysis Library in Python," Available: <https://technical-analysis-library-in-python.readthedocs.io/>