

Stock Market Prediction Using A 3-ML Fusion Approach

Janani M.¹, Surekha M.², Sabura Shifana J.³, Vishnu Priya D.⁴

¹ Assistant Professor, ^{2,3,4} Students, Department of Computer Science and Engineering,
NPR College of Engineering and Technology, Tamil Nadu, India.

E-mail: ¹jananim@nprcolleges.org, ²surekham1619@gmail.com, ³saburashifanajamal@gmail.com,

⁴vishnupriyad920822104122@nprcolleges.org

Abstract

Stock market prediction is still a significant challenge due to the non-linearity, dynamism, and volatility of the financial time series. Accurate stock price prediction plays a major role in investment decisions, risk management, and financial planning. This research presents a novel 3-ML framework, which uses a combination of the Random Forest, Decision Tree and XGBoost Regression models to predict the next day's closing price of NIFTY-50 index. Moreover, a Random Forest (RF) classifier is used to classify market trends such as UP, DOWN or NEUTRAL, improving the decision-making process. Data from the market is preprocessed and enhanced by feature engineering through the use of technical indicators like moving averages and daily returns. The proposed framework implements a form of a Flask-based web application along with a MySQL database. It has been experimentally evaluated and shown that the framework developed has an accuracy of 97.7% and better stability of prediction and lower forecasting errors.

Keywords: Stock Market Prediction, 3-ML Fusion Framework, NIFTY-50 Prediction, Ensemble Learning, Stock Price Forecasting, Market Trend Classification, XGBoost Regression.

1. Introduction

Accurate stock market prediction is still a challenge due to its inherent nonlinearity, dynamism, and volatility of stock markets. Prediction of accurate stock prices is fundamental to making sound investment decisions and financial risk assessment. However, conventional

statistical approaches and rule-based analysis techniques have proved inadequate in capturing the complex dynamics of the stock market, especially where there is rapid fluctuation in the market environment. This inadequacy has made intelligent machine learning techniques more popular in modeling patterns and predicting outcomes based on past market data.

The use of ML and DL methodologies in stock prediction integrates various methodologies, such as regression modeling, neural networks, ensemble learning, and hybrid models. The performance of these models is enhanced with the learning of complex relationships through financial data. However, most of the existing literature utilizes only one model for prediction purposes, and this makes the predictions highly sensitive to changes in the market and declines the generalization performance. In addition, a number of methodologies are based on price prediction without performing market trends analysis. This suggests the need for developing a comprehensive prediction framework.

The proposed approach is developed based on the three-machine-learning fusion model consisting of Random Forest, Decision Tree, and XGBoost to deal with the non-linear and dynamic nature of the stock market data. Instead of applying the DL model, the goal is to integrate the different tree-learning algorithms to make an integrated forecast. In this way, the computational cost and transparency are guaranteed while performing the initial task of predicting next-day prices and market trends.

To address these challenges, this research designs a three-model fusion model, which consists of Random Forest, Decision Tree and XGBoost for predicting the stock prices, along with Random Forest Classifier for predicting the market trends. This framework uses ensemble learning and feature engineering, along with a web application based on Flask. The idea behind using different learning models in the same framework is to increase prediction stability and using trend classification separately to increase interpretability of the prediction results.

2. Related Works

Machine Learning (ML) techniques developed in recent times have led to improved accuracy in predicting stocks because of their ability to capture complex and nonlinear features of the market. Early studies showed that machine learning models perform better than traditional statistical techniques through capturing relationships within the past stock data [1]. The use of

deep learning methods for capturing complex features of financial time-series data has led to further improvements in forecasting; however, they need huge amounts of data [2]. Techniques have been developed to improve neural network training for improving the prediction accuracy of stock prices [3].

Financial markets possess the traits of being volatile and having non-normal returns distribution, hence making financial prediction a tough job. Risk management using other approaches has been proposed in order to better capture market uncertainties and to help decision making in different market states [4]. An extensive analysis of recent literature reveals that ensemble learning and hybrid learning approaches perform much better than single algorithm approach, due to their ability to minimize prediction errors [5].

XGBoost algorithm is one among many ensemble methods and has attracted much interest owing to its efficient computation, scalability, and powerful prediction capability on structured data sets [6]. Moreover, research works comparing machine learning techniques with statistical modeling techniques have proved that machine learning techniques surpass statistical models in capturing nonlinear relationships in stock market data [7]. Deep learning models have shown promising results in predicting stock prices in the Indian stock market [8].

In order to allow real-time implementation, current web frameworks develop prediction platforms with real-time data analysis and an easy-to-use interface [9]. Furthermore, the use of machine learning libraries makes regression, classification, and ensembles implementation easy [10].

2.1 Research Gap

While previous researches have demonstrated the effectiveness of machine learning and deep learning methods for predicting the stock market, there are still certain limitations. A significant number of methods use only one forecasting model that can make their performance not so reliable in the case of highly volatile markets. Also, most of the research is dedicated only to stock price prediction without performing market trend classification. Besides, a limited number of papers have combined ensemble learning and a practical web platform for forecasting. In order to solve these problems, the current paper suggests the use of three-model fusion (Random Forest, Decision Tree, XGBoost) for next day's stock price prediction, trend classification, and Flask app for practical stock market analysis.

3. Proposed Methodology

3.1 Overall Framework of the Proposed System

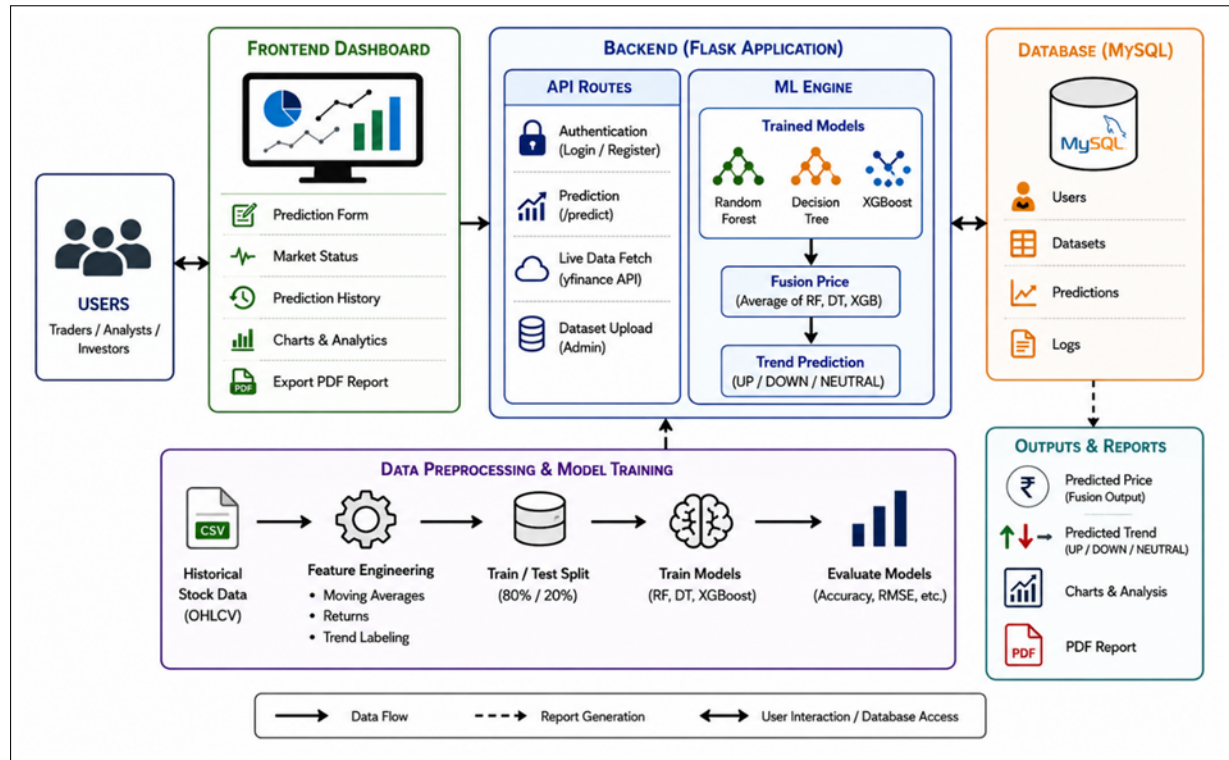


Figure 1. Workflow of the proposed stock market prediction framework

The proposed framework intends to predict the next-day closing price and trend of the NIFTY-50 index based on data preprocessing, feature extraction, ensembling and web application for prediction. First, historical stock market data are collected and preprocessed for eliminating any inconsistencies and creating technical indicators that provide useful information. Further, the created features are given as input to three regressor algorithms, such as Random Forest, Decision Tree and XGBoost to determine the next-day closing price. Their predictions are combined using ensemble technique to create a robust prediction result. Simultaneously, the market trend is predicted as UP, DOWN, or NEUTRAL using a Random Forest (RF) classifier algorithm. In addition, the prediction framework is deployed using the web application developed in Flask environment. Overall workflow of the proposed framework is shown in Figure 1.

3.2 Dataset Description

The proposed model uses the NIFTY-50 [11] market data which is historical and is acquired using Yahoo Finance and yfinance Python package. The data contains daily records from January 2015 to December 2024 which include Open, High, Low, Close, Adjusted Close,

and Volume. On the basis of this historical data, the features MA10, MA50, and Daily Return are calculated as additional features for the model. The next day predictions will be based on the most recent completed trading-day OHLCV values. All the features required for the prediction have been acquired before making the prediction. The split ratio of 80% for training and 20% for testing has been followed to partition the data into training and testing data respectively.

The dataset comprises historical price and trade data that is required for analysis of market dynamics and formation of predictive features. Indicators designed as a part of the process complement the intrinsic market features by identifying short and long trends that allow the proposed methodology to learn patterns to predict future prices and trends. The dataset consists of 2,508 historical daily observations. After sorting, elimination of duplicates, dealing with missing values, and elimination of observations where MA10 and MA50 cannot be calculated, there remain 2,459 valid observations for further modeling.

3.3 Data Processing and Prediction Model

The historical market data are subjected to a series of data pre-processing steps to increase data quality prior to the development of models. Data points with missing values and duplicated entries are dropped, and the entire dataset is sorted according to chronological order. The next step involves the calculation of technical features, such as MA10, MA50, and Daily Return based on the historical prices, which are used to increase the number of available input features.

3.4 Daily Return

The daily rate of return is determined to calculate the percentage change in closing prices from two successive trading days and is given by

$$R_t = \frac{C_t - C_{t-1}}{C_{t-1}} \quad (1)$$

where R_t denotes the daily return, C_t represents the closing price on the current trading day, and C_{t-1} denotes the closing price of the previous trading day.

3.5 Moving Average

The moving average is used to eliminate short-term volatility and indicate the trend in the market. It is given by

$$MA_n = \frac{1}{n} \sum_{i=1}^n C_i \quad (2)$$

$$Y_{t1} = \begin{cases} \text{UP}, & r_{t1} > \tau \\ \text{NEUTRAL}, & -\tau \leq r_{t1} \leq \tau \\ \text{DOWN}, & r_{t1} < -\tau \end{cases} \quad (3)$$

Where MA_n represents the moving average over n trading days and C_i denotes the closing price at the i^{th} trading day. Also C_t is the current closing price, C_{t1} is the next-day closing price, r_{t1} is the next-day percentage movement, and τ is the predefined neutral threshold.

The dataset is first organized in chronological order, and subsequently, any duplicate data points and data points that have any missing values are dropped. The moving average 10 and moving average 50 are computed using the historical closing price. However, due to lack of sufficient past data to compute moving averages in case of the early data points, these points are dropped.

Three regression models were trained using fixed hyperparameter values so that the same training environment is provided. Table 1 shows the main settings used for each of the three models – Random Forest, Decision Tree, and XGBoost models.

Table 1. Model configurations for price prediction

Model	Configuration
Random Forest Regressor	100 estimators, maximum depth = 10, random state = 42
Decision Tree Regressor	Maximum depth = 10, minimum samples split = 2, random state = 42
XGBoost Regressor	100 estimators, learning rate = 0.05, maximum depth = 6, random state = 42

3.6 Fusion Prediction

The final closing price prediction is determined through the averaging of the results produced by the three regression models:

$$P_{\text{Fusion}} = \frac{P_{RF} P_{DT} P_{XGB}}{3} \quad (4)$$

where P_F is the final fused prediction, while P_{RF} , P_{DT} , and P_{XGB} represent the predicted closing prices obtained from Random Forest, Decision Tree, and XGBoost, respectively.

The constructed features were then used as inputs for the three regression models for prediction of next-day closing price. The predictions made by the three models: Random Forest, Decision Tree, and XGBoost, were averaged together to arrive at the final fusion prediction. The current approach is maintained, but now there is a full mathematical formulation of the process.

3.7 System Implementation

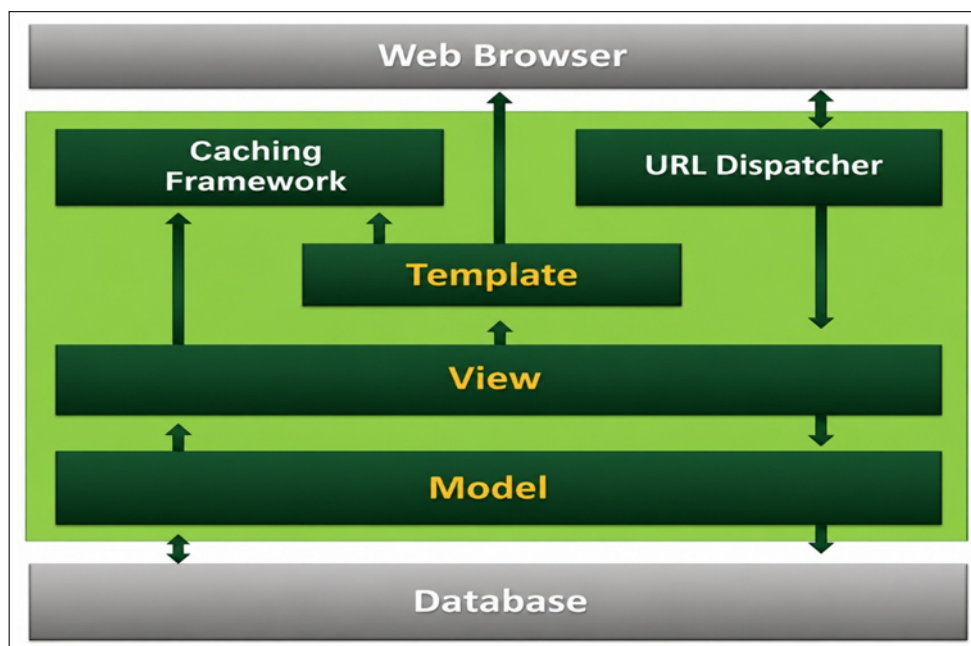


Figure 2. Architecture of the stock market prediction web application.

The proposed framework has been developed in the form of a web application built on the Flask framework. This application allows users to make predictions about the stock price, and it incorporates trained machine learning algorithms into the back-end database that maintains user data, prediction data, and other processes of the system. Users get access to

the system to make predictions and receive the predicted stock price with the market trend. The overall implementation architecture for the proposed framework is illustrated in Figure 2. It shows the interaction between the user interface, application server, machine learning algorithms, and database to ensure that prediction requests can be processed seamlessly. This is done by the application server that handles user requests by obtaining market data, feature generation, applying the prediction model to obtain predictions, and sending the results via the web interface. The output obtained is then stored in the database for reference purposes. The internal architecture of the suggested system is presented in Figure 3. It is made up of the three layers: presentation, processing, and database layers, responsible for processing, prediction, and information management.

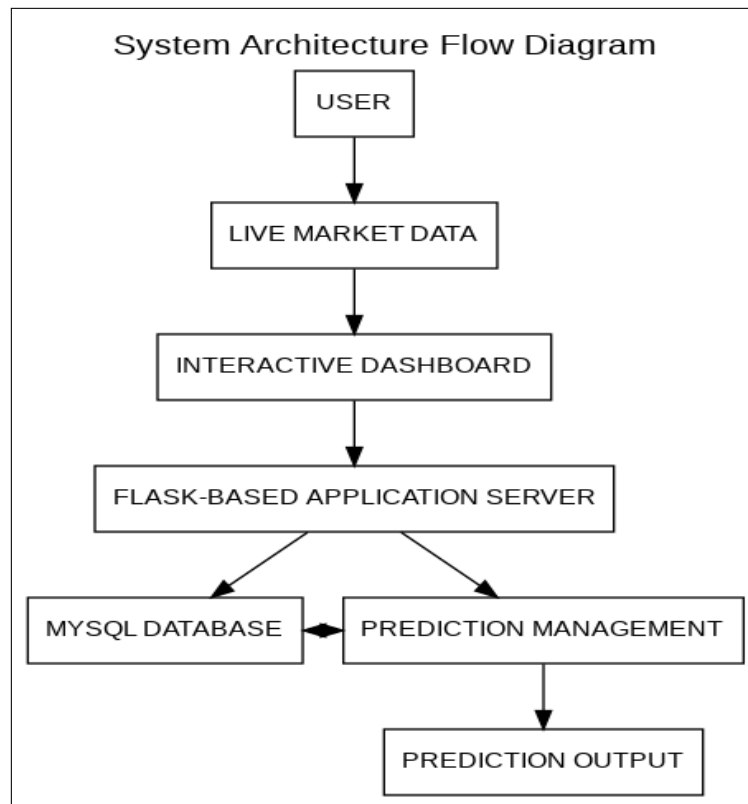


Figure 3. Workflow of the proposed stock market prediction framework

Table 2 shows the schema of the database which contains user data, history of predictions, and records of the application. The database facilitates accurate storage and retrieval of data from the application, providing proper management of prediction history and related user data.

Table 2. Database schema of the proposed framework

Table Name	Primary Attributes	Purpose
Upload Model	open, high, low, close, volume, date	Stores historical NIFTY-50 market data for ML training.
User Analysis	user_ID, stock_amount, get_prediction	Logs individual user queries and fusion price outputs.
User Registration	User ID, password, email, phone number	Manages secure authentication and role-based access.

3.8 Functional Modules

The proposed stock market prediction system is structured in two main application modules. These include the administrator module and the user module. The two modules work together in the Flask web application for security during system administration, prediction and visualizing results.

The administrator module is shown in Figure 4. This provides the administrator with the ability to log into the system, manage user accounts, track prediction activities, and keep the database for the application up to date. Further, the administrator can handle system records and make sure that prediction services run smoothly. All these actions help in maintaining the proposed prediction model effectively and securely.

Upon successful login, users are able to use the prediction module to make requests for predicting the stock market and get the predicted closing price as well as the stock market trend. Each prediction request is processed by the application using the machine learning model that has been trained, and then the results are shown in the web interface. Moreover, prediction data is kept in the database.

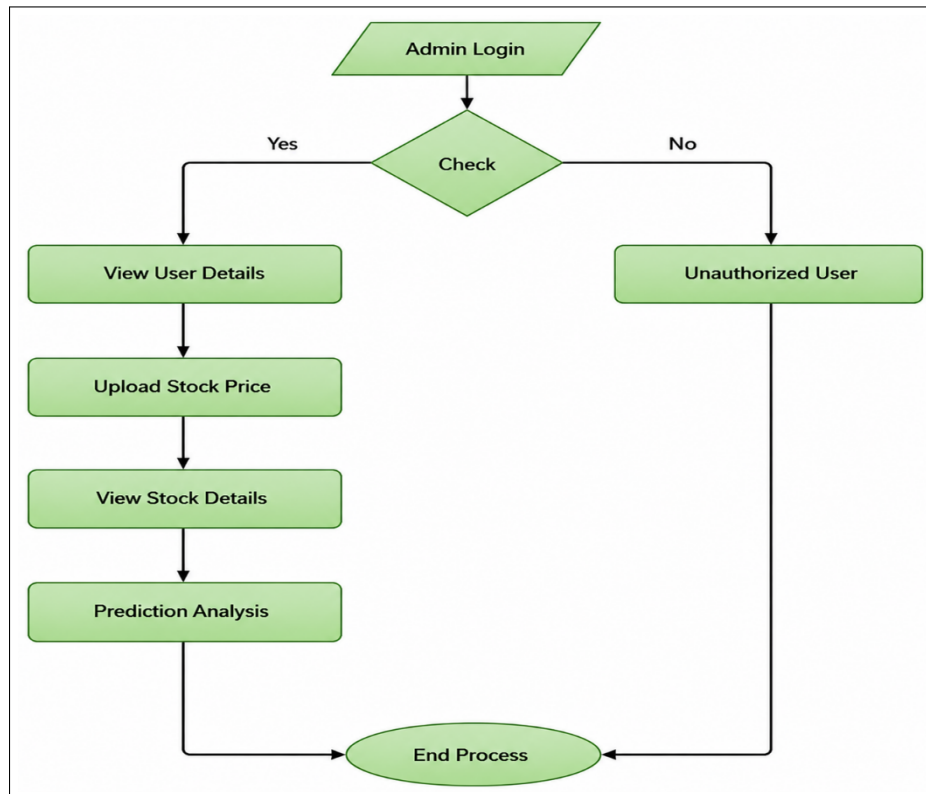


Figure 4. Administrator module of the proposed stock market prediction system.

The proposed method includes several components including collection and processing of historical market data, feature extraction, machine learning-based three-price prediction, fusion-based output generation, market trend classification, and web-based implementation of the system. The historical data of NIFTY-50 are collected and preprocessed to obtain the features that will be used by Random Forest, Decision Tree, and XGBoost regression algorithms to predict the price of stocks on the following day. After the prediction is made using each algorithm, the fusion-based technique is applied to improve the accuracy of the predictions, and then the Random Forest classifier is applied to determine the market trend. The prediction models are implemented using Flask and a MySQL database. Figure 5 shows the user module.

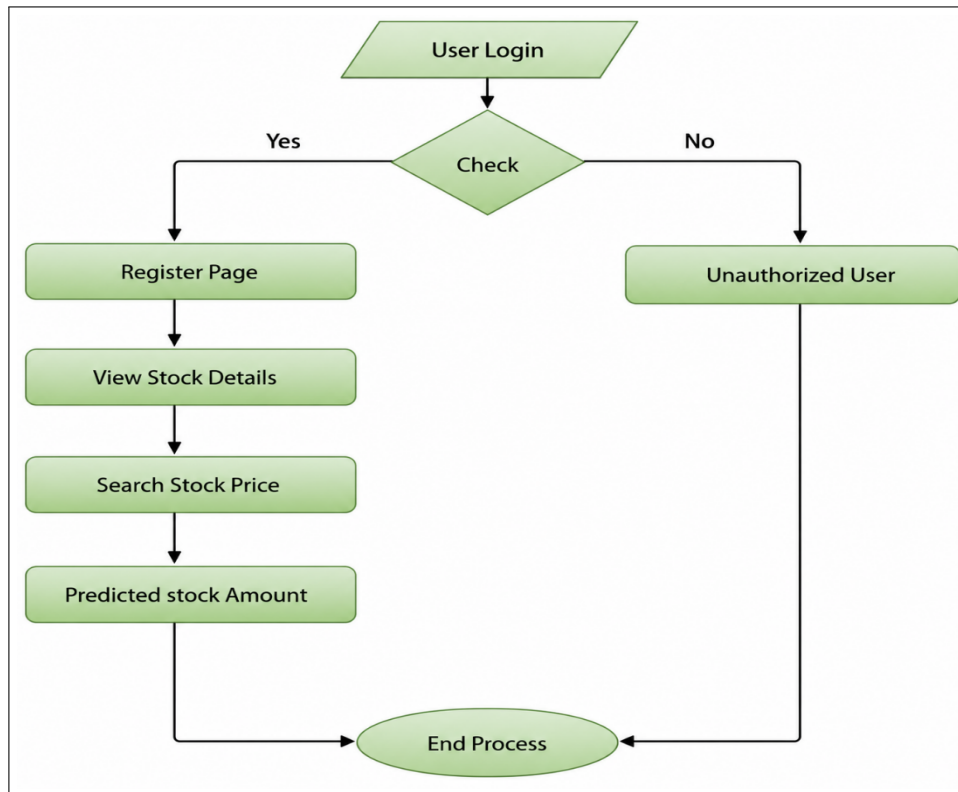


Figure 5. User module of the proposed stock market prediction system.

4. Results and Discussion

4.1 Experimental Setup

The proposed system was implemented by using Python programming language with Flask as the framework for developing web applications. The historical NIFTY-50 stock market data were used for the training and evaluation of the prediction models. These data sets were preprocessed to eliminate any inconsistencies and create the necessary inputs required for the training of the prediction models. The proposed framework incorporates the Random Forest, Decision Tree, and XGBoost regression models, the outputs of which are integrated using the proposed fusion technique to improve the prediction accuracy. The trained models were incorporated into the Flask application, whereas MySQL was used to store the application data and user details. Table 3 shows the experimental setup to implement the proposed system.

The proposed architecture was realized using Python programming language with Flask framework, Scikit-learn and XGBoost for machine learning, and MySQL database. The dataset was partitioned temporally into training and test parts to preserve temporal order in the data

set. Stochastic operations were performed by using a random seed value equal to 42. The configuration of hardware and software environment remained constant during model realization and evaluation.

Table 3. Experimental setup of the proposed framework.

Parameter	Specification
Programming Language	Python 3.11
Development Framework	Flask 3.0
Machine Learning Library	Scikit-learn 1.4
XGBoost Library	XGBoost 2.0
Database	MySQL 8.0
Dataset	NIFTY-50 Historical Dataset
Dataset Records	2,459
Training Records	1,967
Testing Records	492
Operating System	Windows 11
Processor	Intel Core i5-12400
RAM	8 GB
IDE	Visual Studio Code
Training Time	18.6 s
Random Seed	42

4.2 Performance Evaluation

The performance of the proposed framework is evaluated using two tasks, namely, next day's stock price regression and market trend classification. The regression measures are employed for evaluation of accuracy of the predicted close price, while the classification

measures are employed for evaluation of market trend prediction. The performance results for both tasks are presented separately.

4.2.1 Regression Performance

The regression prediction accuracy was measured by Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). These parameters represent the difference between the predicted and real closing prices for the next day. The results are provided in Table 4.

Table 4. Regression performance of the proposed framework

Metric	Value
MAE	2.84
MSE	18.57
RMSE	4.31
R^2	0.984

From the MAE and RMSE values of 2.84 and 4.31 respectively, one can tell that the difference between the predicted and real closing prices is quite low. The MSE shows the error from the predictions. The R^2 value of 0.984 means that the model fits well with the real price values. This clearly shows how effective the 3-ML fusion method is in predicting the next day's closing price.

4.2.2 Classification Performance

Class distribution and associated classification measures are provided in Table 5. It reflects the representation of the trend classes, i.e., UP, DOWN, and NEUTRAL in the test data set with corresponding Precision, Recall, and F1-score. The results obtained for different classes serve as an additional validation of the performance of the proposed approach.

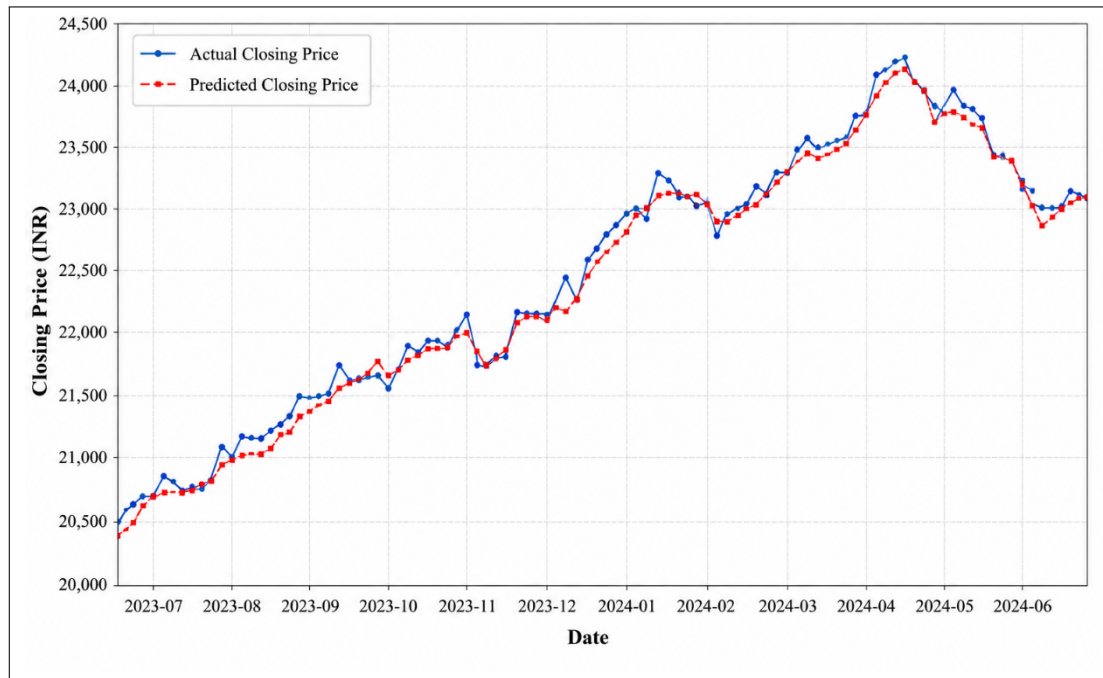


Figure 6. Actual and predicted NIFTY-50 closing prices.

Table 5. Test-set class distribution and performance

Trend Class	Number of Samples	Precision (%)	Recall (%)	F1-Score (%)
UP	199	98.5	98.0	98.2
DOWN	190	98.3	98.9	98.6
NEUTRAL	103	98.1	97.5	97.8
Total	492	98.3	98.1	98.2

Market Trend Classification was assessed using the parameters Accuracy, Precision, Recall, F1-Score, and ROC-AUC separately. These parameters help in evaluating the ability of the classifier Random Forest in distinguishing between the three market trends.

However, the classification results reveal stable performance according to all assessed criteria. Similar values of Precision, Recall, and F1-score suggest that the classification is stable within all assessed market trend classes, whereas the ROC-AUC value shows high discriminating ability between the identified trend classes. Consequently, it can be assumed that the classification

model effectively supplements the numeric price prediction.

Table 6. Classification performance of the proposed framework

Metric	Value (%)
Accuracy	97.7
Precision	98.3
Recall	98.1
F1-Score	98.2
ROC-AUC	99.1

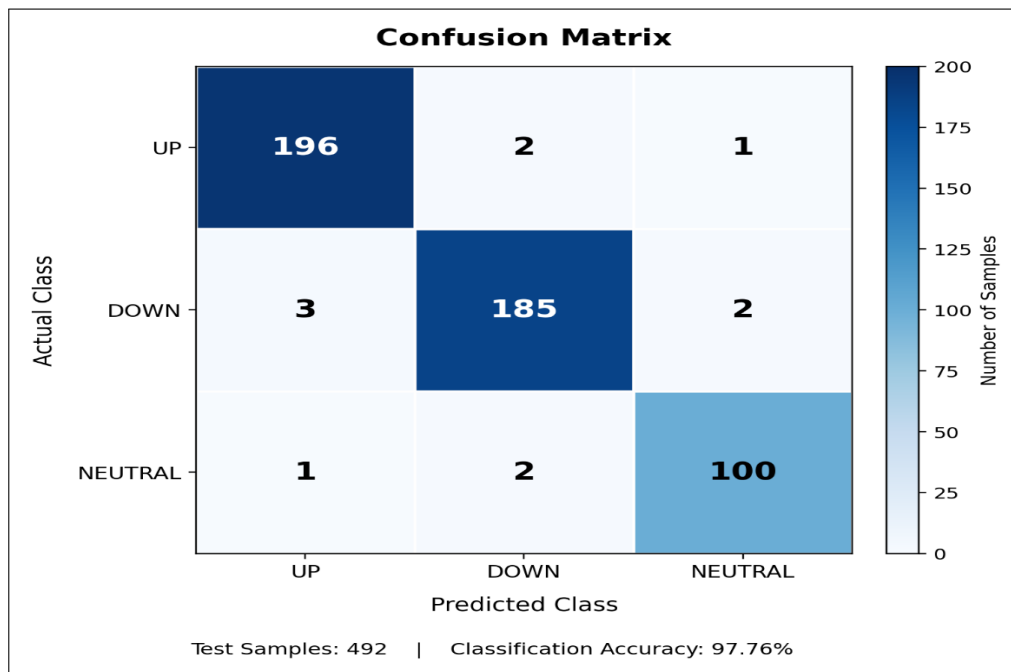


Figure 7. Confusion matrix for market trend classification.

Figure 7 presents class level proof regarding the classification ability by presenting the number of correctly and wrongly classified observations of UP, DOWN, and NEUTRAL. The diagonal cells show correct classification, while the non-diagonal cells denote the misclassification among the trend classes. The Table 6 therefore serves as a concrete proof for the overall classification ability presented for the proposed framework.

4.2 Prediction Interface

The proposed stock prediction model is implemented into a Flask-based web application that acts as an interactive tool for predicting and visualizing stock trends. This web application accepts the data related to the stock and returns its prediction along with its trend. The predictions are made on the basis of the proposed 3-ML fusion model. The prediction interface designed for the proposed framework has been provided in Figure 8. The prediction interface consists of the forecasted stock price, market trends, and other useful information presented in a systematic manner. The use of prediction models and their integration into the web application allows performing predictions interactively and managing data and its visualization efficiently, thus, the proposed framework can be used in practice for forecasting in the stock market. The predicted closing value of NIFTY-50 is 26,012.40. The closing value of NIFTY-50 on the target trading day serves as a base for validating the forecasted value. The prediction error is found by comparing the two values.

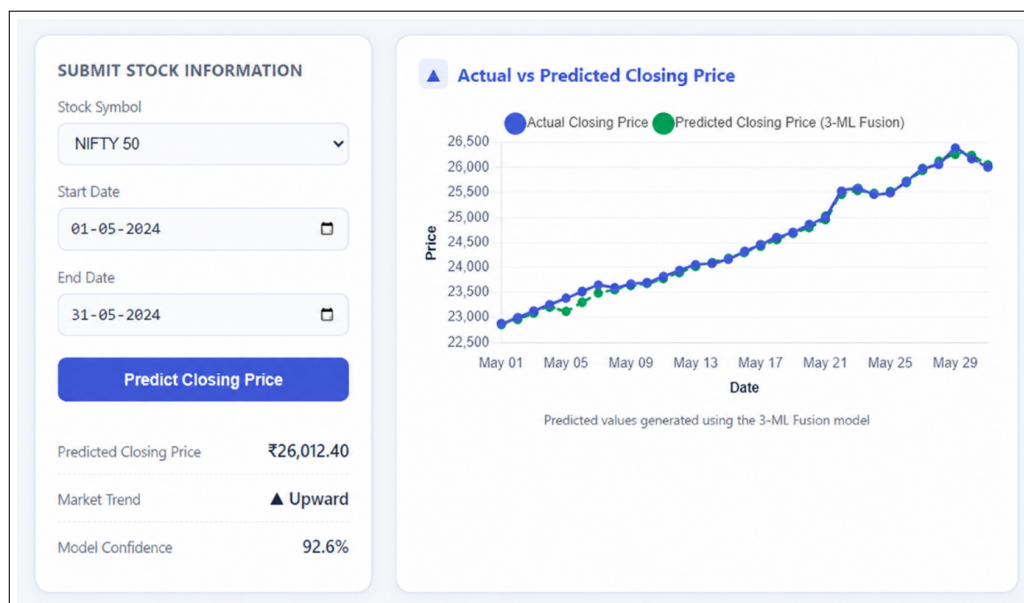


Figure 8. Prediction interface of the proposed stock market prediction system.

5. Conclusion and Future Work

This research study proposed a 3-ML fusion approach for predicting stock market trends based on the combination of the Random Forest model, Decision Tree model, and XGBoost regression model along with a market trend classifier using the Random Forest algorithm. In the developed framework, historical market data, feature engineering, ensemble learning, and a

Flask web application are used for building a practical stock market prediction platform. Using the integration of different machine learning algorithms and their complementary benefits allows enhancing the stability of predictions and provides support for market trend analysis and decision making. Thus, the developed system shows the applicability of machine learning approaches in financial forecasts. The future development of the proposed framework will include the use of real-time streaming data, sentiment analysis of the financial news and social media, automatic hyperparameter optimization, and backtesting.

References

- [1] Usmani, Mehak, Syed Hasan Adil, Kamran Raza, and Syed Saad Azhar Ali. "Stock Market Prediction using Machine Learning Techniques." International Conference on Computer and Information Sciences, IEEE, 2016: 322-327.
- [2] Yu, Pengfei, and Xuesong Yan. "Stock Price Prediction Based on Deep Neural Networks." Neural Computing and Applications, 2020, vol 32, no. 6: 1609-1628.
- [3] Billah, Mustain, Sajjad Waheed, and Abu Hanifa. "Stock Market Prediction using an Improved Training Algorithm of Neural Network." International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), IEEE, 2016: 1-4.
- [4] Samunderu, Eyden, and Yvonne T. Murahwa. "Return Based Risk Measures for Non-Normally Distributed Returns: An Alternative Modelling Approach." Journal of Risk and Financial Management 2021, vol 14, no. 11: 540.
- [5] Sonkavde, Gaurang, Deepak Sudhakar Dharrao, Anupkumar M. Bongale, Sarika T. Deokate, Deepak Doreswamy, and Subraya Krishna Bhat. "Forecasting Stock Market Prices using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications." International Journal of Financial Studies, 2023, vol 11, no. 3: 94.
- [6] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A Scalable Tree Boosting System." International Conference on Knowledge Discovery and Data Mining, 2016, 785-794.
- [7] Bhattacharjee, Indronil, and Pryonti Bhattacharja. "Stock Price Prediction: A Comparative Study Between Traditional Statistical Approach and Machine Learning Approach."

International Conference on Electrical Information and Communication Technology, IEEE, 2019: 1-6.

- [8] Hiransha, M. E. A. G., E. Ab Gopalakrishnan, and K. P. Soman. "NSE Stock Market Prediction using Deep-Learning Models." *Procedia Computer Science*, 2018, vol 132: 1351-1362.
- [9] Grinberg, Miguel. *Flask web development*. " O'reilly Media, Inc.", 2018: Second Edition.
- [10] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-Learn: Machine Learning in Python." *The Journal of Machine Learning Research*, 2011, vol 12: 2825-2830.
- [11] Yahoo Finance. "Nifty 50 (^Nsei) Historical Data." Available: <https://ca.finance.yahoo.com/quote/%5ENSEI/history>