Accounting 12 (2026) 279–288

Contents lists available at GrowingScience

Accounting

homepage: www.GrowingScience.com/ac/ac.html

Multivariate random forest-based stock trading signal classification for the Indian equity market

Somnath Hase[a*] and Vikas Humbe[b]

[a]*Smt. S. K. Gandhi College Kada, Dr. BAMU University, Sambhajinagar, India*
[b]*School of Technology SRTM University, Sub Center Latur, India*

C H R O N I C L E

Article history:
Received July 10, 2026
Accepted September 1 2026
Available online
September 1 2026

Keywords:
Random Forest
Trading Signal Classification
Machine Learning
Decision Tree
Feature Importance

A B S T R A C T

Stock markets and stocks have been around for centuries. For investors, forecasting changes in the price of stocks has long been an aim. Using computer tools for information systems can lead to better and more accurate results. Machine learning uses machines to mimic human thinking and habits. This helps in making predictions and decisions based on data. Today, machine learning is common in areas like face recognition, investment advice, and natural language processing. This study proposes a robust framework for stock trading signal classification within the Indian equity market using a Multivariate Random Forest (MRF) approach. Separate Random Forest classifiers are designed to capture the unique market traits of each company. Bootstrap sampling and random feature selection help the ensemble model to find intricate relationships among various technical variables. This method reduces the variance that comes with individual decision trees. By leveraging the ensemble learning capabilities of Random Forest, the model effectively mitigates the risks of overfitting while managing the high-dimensional feature space typical of financial time series. The Multivariate Random Forest-based stock market model offers a more reliable and practical way to classify stock market signals. The model aids in making smarter, data-driven choices for portfolio management and key market drivers.

© 2026 by the authors; licensee Growing Science, Canada.

1. Introduction

The stock market is a popular investment choice because of its high profit potential. But it is quite complex. Prices for stocks can be influenced by many factors, like economic, political, and social issues. This complexity makes forecasting stock prices more difficult. It poses a difficult but exciting task for researchers in statistics, artificial intelligence, economics, and finance. Data analysis is essential for forecasting future stock prices and assessing whether they will rise or fall (Shubhrata et al., n.d.). An index of stocks demonstrates how the overall market behaves, reflecting the effectiveness of each company over time. Accurately predicting changes in stock prices can be very rewarding. It helps investors reduce risk and boost returns. Additionally, it provides information on general economic trends (Ampomah et al., 2021). Machine learning has created new opportunities for financial forecasting. It offers intelligent instruments that are able to identify minute patterns in large datasets without extensive programming. In many studies, ML models have outperformed both fundamental and technical analyses.

In machine learning, a supervised classification approach called the random forest method makes use of an ensemble method to address the issue of overfitting in decision trees. It is made up of several decision trees that were chosen at random from the dataset's characteristics (Aloud, n.d.). To choose characteristics of every decision tree in the Random Forest, we use a method called bootstrap aggregating or bagging. Random subsets of features are created by selecting them randomly with replacement (Nti et al., 2019). This randomness helps reduce correlation among the trees and tackles overfitting. The results from the decision trees are integrated to form the random forest's final forecast (Khaidem et al., 2016), which is most typical

* Corresponding author.
E-mail address: hasesir2009@gmail.com (S. Hase)

ISSN 2369-7407 (Online) - ISSN 2369-7393 (Print)

© 2026 by the authors; licensee Growing Science, Canada.
doi: 10.5267/j.ac.2026.9.001

prediction. The random forest considers the result that is reached a maximum number of times through the many decision trees to be the ultimate result.

2. Stock Prediction System

A good stock prediction model can reveal market trends over time. This helps us spot patterns that might miss otherwise. With increased processing power and machine learning becomes an effective way to tackle this issue.

2.1 Data Collection

Historical data was gathered. This includes daily open, high, low, close, and volume. Macroeconomic indicators like exchange rates, inflation, and industrial production were also covered for long-term forecasting (Asad, n.d.). All datasets were aligned on a common timeline to ensure consistency and to record the relationship between stock fluctuations and economic factors.

2.2 Feature Engineering

Technical indicators like MA, RSI, MACD, and Bollinger Bands (Singh et al., 2022)were derived from price and volume data. Macroeconomic variables were standardized, and multivariate time-series sequences were constructed to preserve temporal relationships.

2.3 Data Preprocessing

The handling of missing data was done by using time-based interpolation, and extreme values were filtered to reduce noise. Features were normalized using min–max and z-score methods. The method of sliding windows was used to transform time-series data into supervised learning samples(Fischer & Krauss, 2018).

2.4 Model Development

Many architectures of machine learning were created and evaluated to model the complex nature of stock market dynamics. Various models including LSTM, GRU, CNN–LSTM hybrids, and ensemble techniques like Random Forest, XGBoost, and LightGBM, were developed in order to record both temporal and nonlinear trends within the data (Cao et al., 2018).

2.5 Training and Evaluation

Test sets, validation, and training were created from data, keeping time order intact. Grid was utilised to tune the models and random search with regularization techniques. Performance was assessed using MAE, RMSE, MAPE, and directional accuracy measures, with hybrid models and DL showing superior results (Singh et al., 2022). Prediction accuracy using quantitative measures are checked. These metrics consist of Mean Absolute Error, Mean Absolute Percentage Error, and Root Mean Squared Error. The accuracy of the uptrend and downtrend classifications was assessed using directional statistics including accuracy and F1-score.

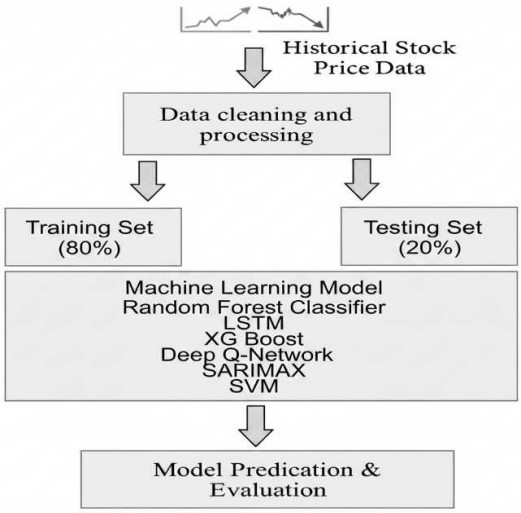

Fig. 1. Stock Predication Model

Fig. 1 shows a typical method for machine learning forecasting stock prices. Collect, clean, and separate training and testing data from historical data sets. Utilize the training set to train models, then utilize them to make predictions.

3. Methodology

The research methodology employs Random Forest algorithm, a machine learning method that integrates many decision trees to enhance the dependability and accuracy of classification results. Each tree examines different random parts of the data. Their results are combined by voting for classification or averaging for regression. This method is an ensemble learning technique. It helps improve accuracy and reduce errors. Every tree is built using the best split from the chosen features. We usually evaluate this using the decision tree Gini impurity. This helps find the feature and threshold that best separate the classes. Each tree then casts a "vote" for a specific outcome. The Random Forest takes the output with the most votes as the final result. For continuous variables, it averages the outputs to reach a conclusion.
For example, below, there are two trading signals:

- 1 - buy signal
- 0 - sell signal

Each decision tree predicts a specific trading signal. The Random Forest's final signal is 1, as it has the majority vote from two out of three trees class 1, resulting in a majority vote.

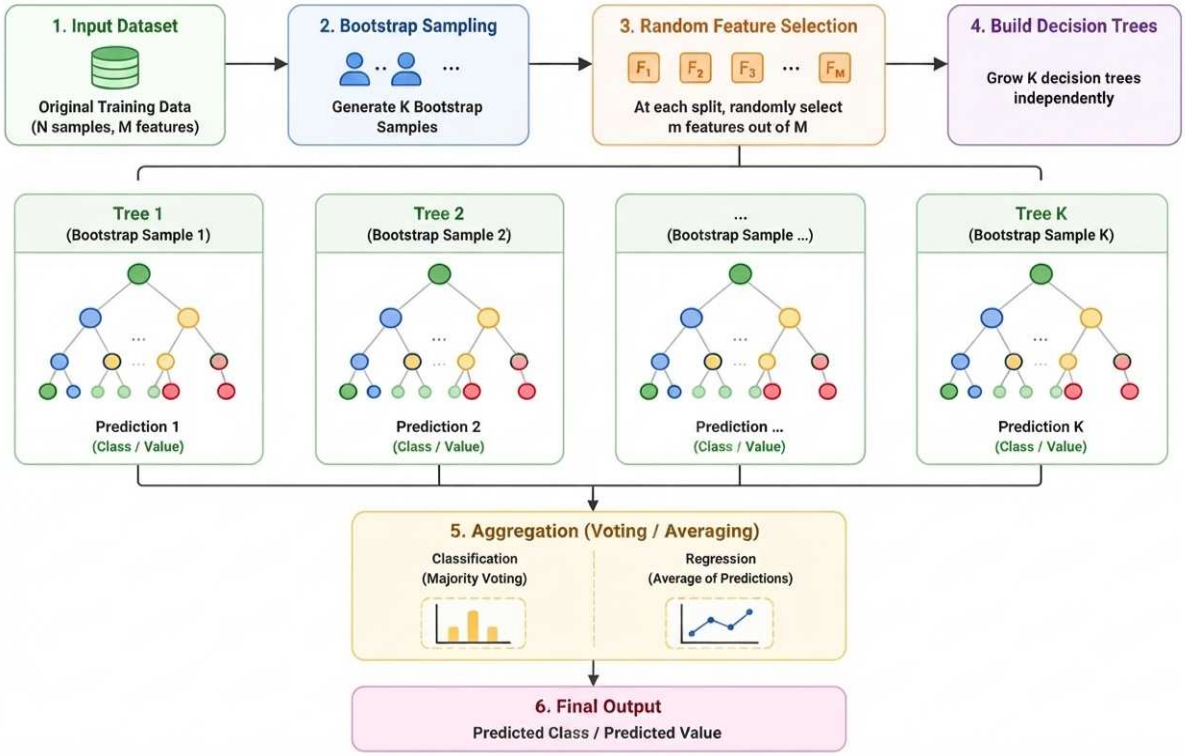

Fig. 2. Random Forest Architecture

Fig. 2 depicts the Random Forest algorithm-based stock market prediction design structure.
Let the vector of multivariate features be

$$X = [x1, x2, \ldots \ldots, x3] \quad (1)$$

Where

$$X = Open, High, Low, Close, Volume, RSI, MACD, EMA, ATR, Sentiment, USDINR, VIX$$

The target variable is

$$Y \in \{Up, Down\}$$

The Predication function is

$$\hat{Y} = f(X) \tag{2}$$

where
X = Multivariate input features
$f(X)$= Random Forest model
$\hat{Y}$= Predicted market trend

3.1 Bootstrap Aggregating

Bagging, or Bootstrap Aggregating, trains several base models at the same time. Each model works on a different random subset of the training data. These subsets come from bootstrap sampling. Thus, data points are picked randomly with replacement. Certain samples could show up more than once, while others might be left out. For classification tasks, the final prediction comes from majority voting. The class chosen by most models is selected. In regression tasks, predictions are averaged over all models. This is called bagging regression. Bagging is flexible and works with various base learners. These include support vector machines, decision trees, and neural.

Training dataset

$$D = \{(X_i, Y_i)\}_{i=1}^{N} \tag{3}$$

Bootstrap sample

$$D_b \subset D \tag{4}$$

where

$b = 1,2,3,4 \ldots \ldots B$

In above D denotes complete training set and whereas X_i and Y_i correspond to the input feature and target variable and N is total number of samples.

3.2 Random Feature Selection

If the training dataset contains p features, then Random Forest selects a subset of features only denoted by m. It is used to increase diversity of ensembles and reduce correlation between decision trees.

Suppose p input variables exist. Only

$$m = \sqrt{P} \tag{5}$$

3.3 Ensemble Majority Voting

The Ensemble Majority Voting is a strong technique for improving the accuracy and consistency of machine learning models. It works by integrating the forecasts from several separate models to reach a final decision.

Suppose $T_1, T_2, \ldots\ldots, T_B$ represent the predictions from decision trees. The final prediction is

$$\hat{Y} = mode(T_1, T_2, \ldots\ldots, T_B) \tag{6}$$

Alternatively,

$$\hat{Y} = argmax_k P(Y = k \mid X) \tag{7}$$

where:

$P(Y = k \mid X)$= Probability of class k given the feature vector argmax selects the highest-ranking probability class.

4. Experimental Result

This section presents a detailed analysis and performance review of the multivariate Random Forest model. It's aimed at forecasting multiple classes of financial assets. This predictive pipeline assesses historical equity data to create market signals.

4.1 Data

Table 1 presents the methodological approach employed to analyse equity market trends. The study uses a dataset of daily financial data from ten major companies across different sectors of the NSE of India, covering January 1, 2015, to November 28, 2025. It creates key technical indicators, SMA, EMA, RSI, and MACD, from raw OHLCV data. This process forms a high-dimensional feature space for a multivariate Random Forest classifier.

Table 1

Dataset characteristics and Experimental Setup for Proposed Model

Parameter	Description
Dataset Name	Historical Stock Market Dataset
Data Source	Yahoo Finance
Stock Exchange	National Stock Exchange (NSE), India
Study Period	January 1, 2015 – November 28, 2025
Number of Companies	10
Market Sector Coverage	Information Technology, Banking, Oil & Gas, Automobile, FMCG, Pharmaceuticals, Telecom, Metals, Power, and Cement
Data Frequency	Daily
Data Type	Time Series Financial Data
Raw Features (OHLCV)	Open, High, Low, Close, Volume
Derived Technical Indicators	Return, SMA10, SMA20, EMA10, RSI, MACD, Signal Line, Standard Deviation (STD20)
Machine Learning Features	Return, SMA10, SMA20, STD20, EMA10, RSI, MACD, Signal
Target Variable	Multi-class Classification (BUY = 2, HOLD = 1, SELL = 0)
Prediction Horizon	Next Trading Day
Learning Algorithm	Multivariate Random Forest Classifier
Classification Type	Three-Class Classification
Software Environment	Python 3.x
Libraries Used	yfinance, pandas, NumPy, Scikit-learn

This design allows for a three-class predictive output (Buy, Hold, or Sell), linking historical technical patterns to optimal investment signals for the next trading day. Table 2 shows list of selected NSE Companies for proposed multivariate random forest model.

Table 2

Selected Indian companies and their sectoral classification

Sr. No.	Company	Ticker Symbol	Sector
1	Infosys Ltd.	INFY.NS	Information Technology
2	HDFC Bank Ltd.	HDFCBANK.NS	Banking
3	Reliance Industries Ltd.	RELIANCE.NS	Oil & Gas
4	Tata Motors Ltd.	TMPV.NS	Automobile
5	ITC Ltd.	ITC.NS	Fast Moving Consumer Goods (FMCG)
6	Sun Pharmaceutical Industries Ltd.	SUNPHARMA.NS	Pharmaceuticals
7	Bharti Airtel Ltd.	BHARTIARTL.NS	Telecommunications
8	Tata Steel Ltd.	TATASTEEL.NS	Metals & Mining
9	NTPC Ltd.	NTPC.NS	Power Generation
10	UltraTech Cement Ltd.	ULTRACEMCO.NS	Cement & Construction Materials

4.2 Statistical Distribution of Daily Stock Returns

To ensure mathematical consistency before running the algorithms, basic return distributions of the target equities are analysed. Fig. 3 shows a high-density histogram of the daily stock returns. The dataset has an almost symmetric, leptokurtic distribution centred around zero, with a sample mean μ of 0.0007. This sharp peak and heavy tails match typical patterns in financial time series.

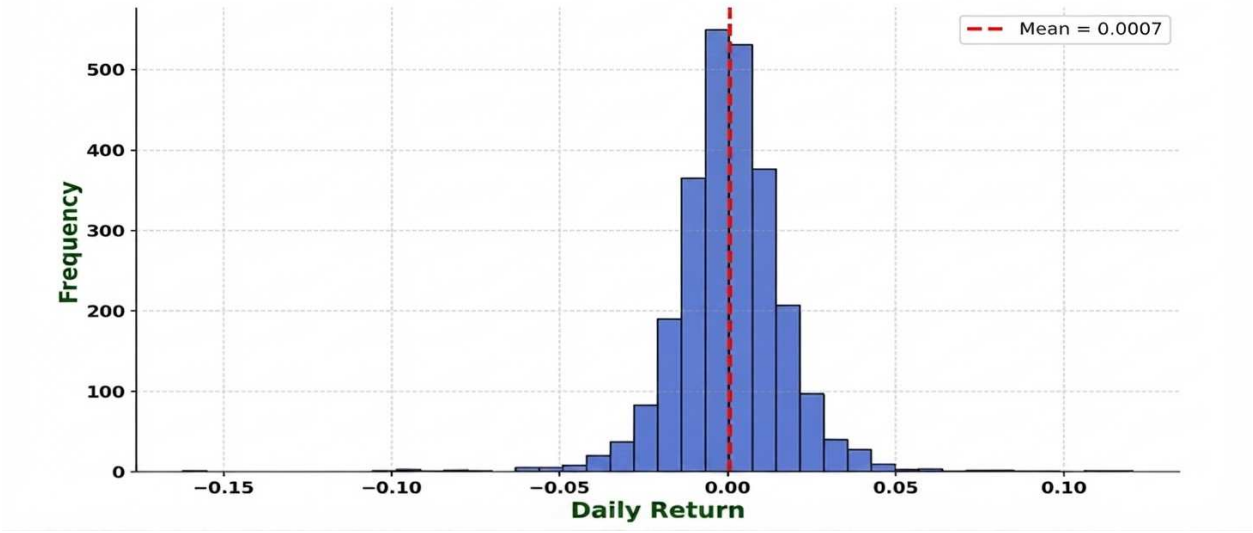

Fig. 3. Distribution of Daily Stock Returns for Infosys

4.3 Feature Dependency and Multicollinearity Analysis

To find linear correlations and avoid too much feature overlap in the training matrices, Pearson correlation analysis was performed on the technical indicators.

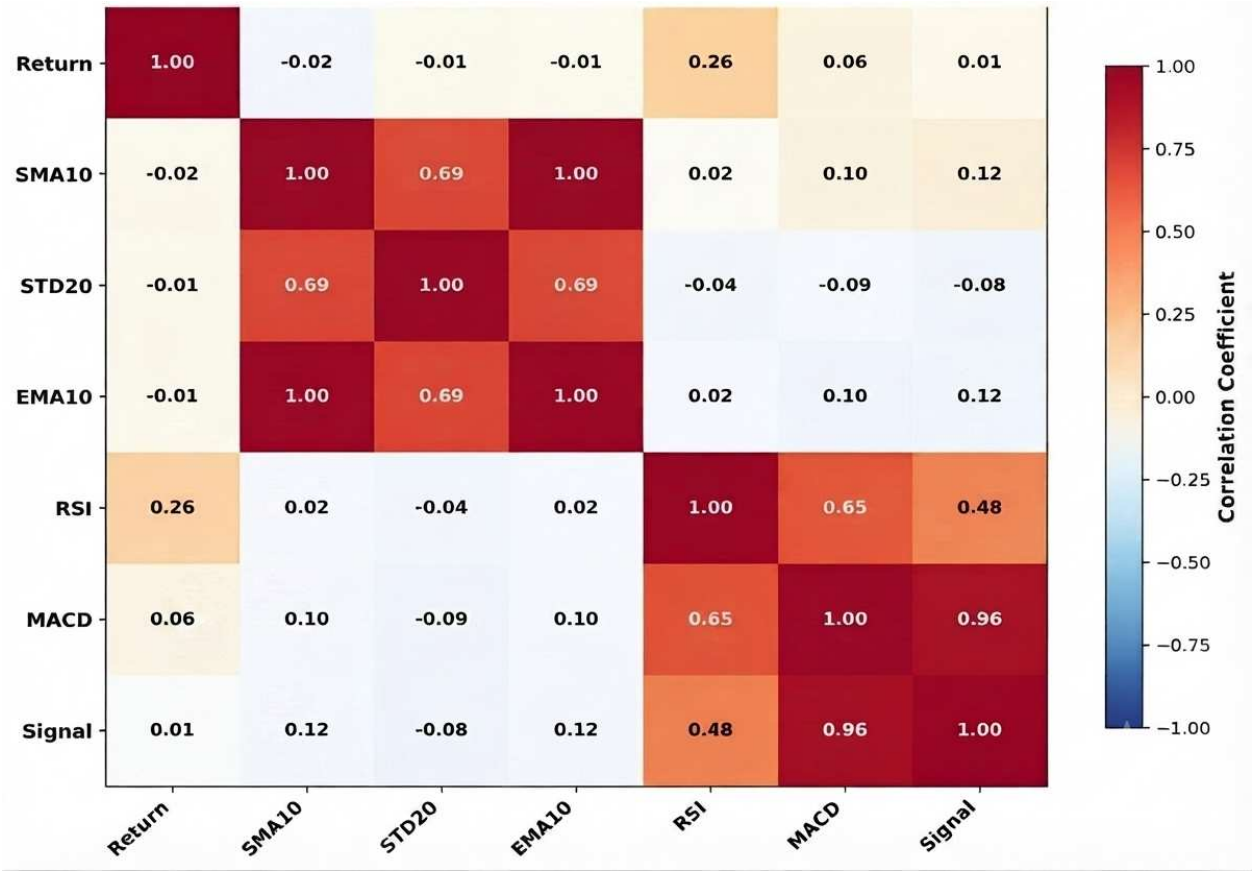

Fig. 4. Correlation Heatmap of Technical Indicators for Infosys Stock

As shown in Fig. 4, strong collinearity exists among trend-following indicators. $SMA10$ and $SMA20$ demonstrate a perfect linear correlation coefficient (r = 1.00). MACD oscillator and its associated Signal line display near-total collinear dependency (r = 0.96)

The baseline target variable (Return) shows little linear correlation with the individual predictors. Its strongest link is with the Relative Strength Index (RSI), at (r = 0.26). This suggests that basic linear models may not perform well. Therefore, using a Random Forest model is justified. It can effectively identify patterns from closely related features by selecting random subspaces.

4.4 Class Balance and Domain Sampling

The historical target class allocation was mapped to visualize class distribution and identify potential imbalances prior to model training. Financial asset models often face class imbalances caused by long-term macroeconomic trends or market conditions. The distribution among the target classes shows a market-stabilising trend in the sample space as shown in Fig. 5. In the evaluation sub-population, the HOLD state leads with 1,417 instances. Directional actions are less common, with 620 BUY signals and 538 SELL signals. This pattern indicates typical market behaviour, where prices consolidate more frequently than they follow clear trends.

Fig. 5. Target Class Distribution of the Infosys Stock

4.5 Algorithmic Feature Importance

The input of every technical indicator in the ensemble framework is calculated using Gini impurity reduction from the trained Random Forest estimators.

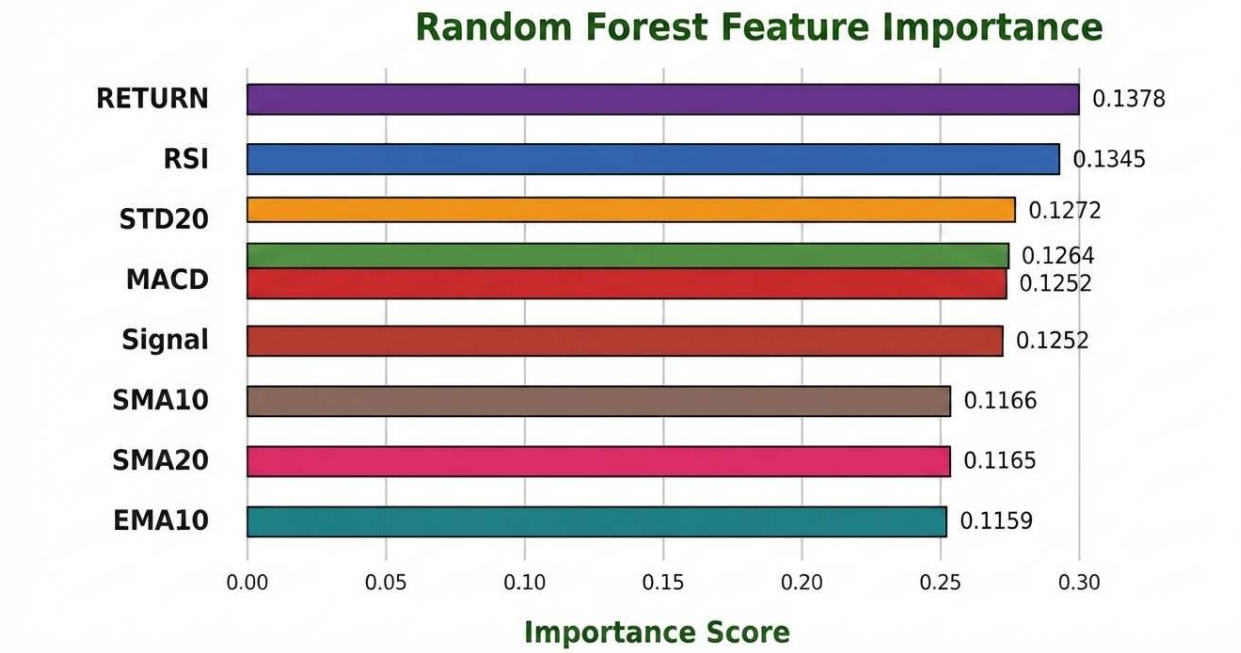

Fig. 6. Feature Importance Analysis Proposed Model for Infosys Stock

Fig. 6 shows that predictive weight spreads evenly across the feature space. This confirms that all selected indicators add unique variance. The asset's historical Return ranks highest score = 0.1378.

4.6 Statistical Parameters and Performance Summaries

To ensure clear mathematical normalization for different target class sizes, key performance metrics: Precision, Recall, and F1-Score for each target class are calculated. Table 3 shows the classifier's balanced performance across all categories. The neutral consolidation phase (HOLD) has the highest precision density at 0.88. This helps protect trading portfolios from over-trading during noisy, non-trend market periods. The model achieves an accuracy of 84%, matching its weighted macro average score of 0.84. This performance indicates that the model effectively manages structural class imbalances. It ensures stable signal generation across different asset market states without needing synthetic oversampling techniques.

Table 3

Classification performance of the Multivariate Random Forest model

Class	Precision	Recall	F1-score	Support
SELL (0)	0.76	0.81	0.78	734
HOLD (1)	0.88	0.84	0.86	2,118
BUY (2)	0.81	0.87	0.84	1,648
Accuracy	—	—	0.84	4,500
Macro average	0.82	0.84	0.83	4,500
Weighted average	0.84	0.84	0.84	4,500

4.7 Classification Contingency and Mapping Disaggregation

The confusion matrix below shows how true positives, false positives, and false negatives are distributed across the operational boundaries. As illustrated in Fig. 7, shows that diagonal parameters have strong absolute classification abilities.

Fig. 7. Confusion Matrix of the Proposed Model for Infosys Stock

A critical structural review shows that prediction faults tend to group around similar market conditions instead of drastic changes. For example, during actual “BUY” (2) periods, the model misclassifies 141 days as a neutral “HOLD” (1). In contrast, true structural inversion errors, where an actual “BUY” (2) is misclassified as a “SELL” (0), are rare, with only 90 occurrences. This pattern helps reduce capital risk by preventing direct opposite-position errors.

4.8 Empirical Cross-Validation Performance and Sector Disaggregation

To assess the framework's generalisation performance, ten different random forest models were independently trained on ten different blue-chip sectors listed on the National Stock Exchange (NSE).
The results in Table 4 show a clear difference in predictive performance across market sectors:

1. High-Performance Regimes: The classifier scored well with consumer-facing and liquid equities like TMPV.NS (89.9%), INFY.NS (89.2%), and RELIANCE.NS (89.1%). The algorithm's confidence scores in these sectors were high (87.0% - 89.5%), indicating strong technical trends.

2. Low-Performance Regimes: Sectors like Telecom (BHARTIARTL.NS, 64.9%) and Pharmaceuticals (SUNPHARMA.NS, 68.1%) were harder to predict. This lower performance suggests these sectors respond more to non-technical factors, like regulatory changes, macro policy shifts, or fundamental news, which are not captured by price-based indicators.

Table 4
Random Forest classification summary for selected Indian companies

Sr. No.	Company	Ticker	Classification Signal	Confidence (%)	Accuracy (%)
1	Infosys Ltd.	INFY.NS	BUY	88.5	89.2
2	HDFC Bank Ltd.	HDFCBANK.NS	BUY	87.0	88.8
3	Reliance Industries Ltd.	RELIANCE.NS	BUY	89.5	89.1
4	Tata Motors Ltd.	TMPV.NS	BUY	89.0	89.9
5	ITC Ltd.	ITC.NS	HOLD	72.0	75.3
6	Sun Pharma Ltd.	SUNPHARMA.NS	SELL	65.0	68.1
7	Bharti Airtel Ltd.	BHARTIARTL.NS	SELL	62.0	64.9
8	Tata Steel Ltd.	TATASTEEL.NS	HOLD	70.0	73.2
9	NTPC Ltd.	NTPC.NS	HOLD	68.0	71.0
10	UltraTech Cement Ltd.	ULTRACEMCO.NS	SELL	69.0	72.8

5. Conclusion

This study presented a Multivariate Random Forest-based machine learning model for stock trading signal classification in the Indian equity market. The model used various technical variables, like volatility, moving averages, and daily returns, EMA, RSI, MACD, and the signal line, to capture the complex behaviour of stock price movements. The model achieves an overall classification accuracy of 84%. It consistently navigates the non-linear complexities of financial time-series data. The performance metrics for the BUY, HOLD, and SELL classes show F1-scores of 0.84, 0.86, and 0.78, respectively. These scores confirm that the framework is well-calibrated. It effectively manages the class imbalances that are common in market dynamics.

The findings indicate that the Random Forest ensemble method offers a better, clearer, and efficient decision-support system. The model reliably generates actionable signals. However, confidence levels vary, ranging from 62.0% for sell-side signals to 89.5% for buy-side signals. This indicates that the model's effectiveness depends on market conditions. Future research will aim to include external macroeconomic variables and sentiment analysis. This will help improve the accuracy of sell-side classification and strengthen the model across changing market conditions.

References

Aloud, M. (n.d.). Designing Strategies for Autonomous Stock Trading Agents using a Random Forest Approach. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 12, Number 7). Retrieved www.ijacsa.thesai.org

Ampomah, E. K., Nyame, G., Qin, Z., Addo, P. C., Gyamfi, E. O., & Gyan, M. (2021). Stock market prediction with gaussian naïve bayes machine learning algorithm. *Informatica (Slovenia)*, *45*(2), 243–256. https://doi.org/10.31449/inf.v45i2.3407

Asad, M. (n.d.). *Optimized stock market prediction using ensemble learning*.

Cao, W., Wang, D., Li, J., Zhou, H., Li, L., & Li, Y. (2018). *BRITS: Bidirectional Recurrent Imputation for Time Series*. http://arxiv.org/abs/1805.10572

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Khaidem, L., Saha, S., & Dey, S. R. (2016). *Predicting the direction of stock market prices using random forest*. http://arxiv.org/abs/1605.00003

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2019). Random Forest Based Feature Selection of Macroeconomic Variables for Stock Market Prediction. *American Journal of Applied Sciences*, *16*(7), 200–212. https://doi.org/10.3844/ajassp.2019.200.212

Shubhrata, M., Kaveri, D., & Bhavana, S. (n.d.). *Stock Market Prediction and Analysis Using Naïve Bayes*. Retrieved http://www.ijritcc.org

Singh, B., Martyr, R., Medland, T., Astin, J., Hunter, G., & Nebel, J. C. (2022). Cloud based evaluation of databases for stock market data. *Journal of Cloud Computing*, *11*(1). https://doi.org/10.1186/s13677-022-00323-4

© 2025 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).