*International Journal of Scientific Research and Modern Technology (IJSRMT)* ijsrmt.com
***Volume 5, Issue 2, 2026***
***DOI: https://doi.org/10.38124/ijsrmt.v5i2.1240***

# Machine Learning for Real-Time Stock Market Trend Prediction in Bangladesh: A 25-Year Comprehensive Analysis of Dhaka Stock Exchange with Web Application Deployment

Rakib Hasan[1]; Marjan Akter Badhon[2]; Mahmudul Hassan Maruf[3]; Sourov Ahmed[4]; Sanimul Hossain Sanzit[5]

[1,5]Department of Computer Science and Engineering, Daffodil International University, Birulia, Savar, Dhaka 1216, Bangladesh
[2,3,4]Department of Finance and Banking, National University of Bangladesh, Board Bazar,Gazipur

Publication Date: 2026/02/26

**Abstract**

Accurate prediction of stock market trends remains a daunting challenge, particularly in high-volatility and structurally changing emerging economies. This paper presents a comprehensive model for binary day-to-day Dhaka Stock Exchange movement classification specifically up or down, using a 25-year historical data set from January 2000 to February 2025. We developed a comprehensive feature engineering pipeline that constructs lagged prices, price variations, daily returns, rolling means and volatilities over several horizons, momentum factors, exponential moving averages, volume lags, and calendar-based features. Three state-of-the-art tree-based classification algorithms, Random Forest, XGBoost, and LightGBM, were trained and hyperparameter-tuned under the strict time-series split cross-validation scheme to ensure temporal consistency and prevent leakage. Model hyperparameters were optimized with domain-specific tuning of tree depth, regularization coefficients, learning rates, and sampling schemes. Average cross-validated test accuracies of 85.20 percent for Random Forest, 85.42 percent for XGBoost, and 85.69 percent for LightGBM demonstrate the effectiveness of gradient boosting methods in recognizing nonlinear market trends, while weighted soft-voting ensemble also smoothed the predictions to 85.68 percent accuracy with balanced precision–recall curves. Feature importance findings reveal that rolling volatilities and momentum features have most substantial effects, impacting predictive performance. Our analysis indicates that LightGBM offers the best accuracy, computational complexity, and stability tradeoff while the ensemble offers incremental progress through complementary strengths of models. This paper fills an important gap in long-term, ensemble-based trend forecasting for the DSE. To demonstrate practical applicability, we deployed the LightGBM model (chosen for its optimal performance-efficiency tradeoff) in a production-ready web application that provides real-time trend predictions through a RESTful API and interactive interface.

***Keywords:*** *Stock Trend Prediction, Machine Learning, Dhaka Stock Exchange, Ensemble Learning, Real-Time Financial Analytics.*

## I. INTRODUCTION

Accurate prediction of stock market movement has been a central finance analytics problem owing to the volatility, nonlinearity, and sensitivity of the market to many economic and behavioral factors. Studies of the Dhaka Stock Exchange (DSE) used earlier conventional statistical approaches such as support vector regression for price action modeling and demonstrated that even linear kernel-based models could be used to learn to identify relevant patterns from historical data (Meesad & Rasel, 2013). Subsequent comparative works stressed that no individual model is ever necessarily better than others in all market conditions, and therefore the need for robust, ensemble-based approaches (Zaman & Rahman, 2019). Within the broader machine learning literature, researchers have shown that tree ensembles, particularly gradient boosted models, work well on structured financial data and tend to be more accurate than shallow learners and deep networks in terms of both accuracy and computational expenses (Pahwa, Khalfay, Soni, & Vora, 2017) (Cervelló-Royo & Guijarro, 2020).

The larger aim of this work is to design and cross-validate an intra-day DSE stock direction prediction model predicting upward or downward movement, using a huge 25-year dataset spanning January 2000 to February 2025. We desire to construct a sophisticated compendium of engineered features in terms of lagged prices, rolling means and medians, and target-encoded classifiers, and train three robust classifiers (Random Forest, XGBoost, and LightGBM) and then ensemble them using weighted soft-voting ensemble. By robust model performance validation with time-series cross-validation, we strive to attain the best prediction accuracy and generalization in unseen market data for the arrangement.

Stock exchanges like the DSE which are evolving pose unique forecasting issues through reduced liquidity, rule alterations, and greater volatile sectoral swings, which can destabilize models tuned from advanced markets (Manasa, Praveenraj, & SR, 2023). In addition, the majority of existing research on DSE forecasting has been limited to small time horizons or single-model analysis, with open questions regarding long-run stability and ensemble value (Misra & Chaurasia, 2020) (Verma, Tanwar, Garg, Gandhi, & Bachani, 2022). As a result of the dramatic economic changes Bangladesh has experienced in the last quarter century, what is required is a model that not only fits past patterns but also continues to hold up in changing regimes of markets. This promotes our focus on large-scale feature engineering and application of complementary learning algorithms.

We have five key contributions in this paper. We first curate and preprocess a single 25-year DSE dataset with temporal consistency for reliable model training. We then engineer a complete feature set, namely spanning momentum, volatility, rolling means, and target encoding, that captures both price action and calendar effect. Third, we train and tune three strong classifiers including Random Forest, XGBoost, and LightGBM, and softly vote together in an ensemble with cross-validation performance-learned model weights. Fourth, we evaluate our approach with time-series split cross-validation and show better generalization than single-model baselines and a fill a gap of long-term, ensemble-based trend forecasting in the DSE. Fifth, we developed and deployed a full-stack web application (Flask backend with React/JavaScript frontend) that operationalizes our best-performing LightGBM model. This system allows real-time trend calculation via API endpoints and provides an intuitive interface for market analysts, bridging the gap between academic research and practical financial tools.

This research is restricted to binary trend forecasting including predicting if a stock will close higher or lower the following day, on the Dhaka Stock Exchange, between January 1, 2000, and February 26, 2025. We confine our models to engineered technical features from historical price and volume data alone; macroeconomic information and other data sources such as social media sentiment are not covered in this study. Our interest measures are precision, accuracy, F1-score, and recall, and they are measured with five-fold time-series cross-validation. Although our results are unique to the DSE, the approach can be applied for other emerging countries with data characteristics similar to these.

## II. RELATED WORKS

The first forays in trend forecasting and analysis in the Dhaka Stock Exchange (DSE) were rooted in conventional regression approaches. Meesad and Rasel (2013) opened this line of inquiry by applying Support Vector Regression to DSE daily prices, illustrating the potential of the technique to model the nonlinear dynamics in stock prices but also its sensitivity to parameters (Meesad & Rasel, 2013). Zaman and Rahman (2019) then contrasted a battery of machine learning and statistical models from moving averages to decision trees referencing the fact that there was no model that performed better in every market condition with regular consistency (Zaman & Rahman, 2019). Misra and Chaurasia (2020) contributed to the arsenal by contrasting a selection of supervised learners (including k-nearest neighbors, naïve Bayes, and SVM) across Indian market data and observing the result that ensemble techniques were lower in forecasting error but at the cost of interpretability (Misra & Chaurasia, 2020).

With advances in processing capacity and data size, researchers have begun to integrate big-data infrastructure and specialized pattern-based solutions. Verma et al. (2022) integrated a rule-based pattern miner and the use of MapReduce for big tick-by-tick preprocessing of data, providing clean features to random forest and gradient boosting for augmenting trend spotting in huge volumes (Verma, Tanwar, Garg, Gandhi, & Bachani, 2022). Pursuing the same line of research work, Gummadi et al. (2021) attempted to overlap robotic process automation with algorithmic trading with the use of Python-based robots to generate and backtest machine-learning-based signals (Gummadi et al., 2021). Ghani, Awais, and Muzammul (2019) presented a general overview of ML algorithms for stock forecasting and demonstrating that tree-based approaches, if optimized well, are superior to deep learning in table-based price data (Ghani, Awais, & Muzammul, 2019). Nabipour et al. (2020) presented a comprehensive comparison of classical machine learning versus deep learning architectures for continuous forecasting of prices and binary trend spotting and concluded that neural networks can learn complex patterns but require more data and processing as per ensemble tree techniques (Nabipour, Nayyeri, Jabani, & Mosavi, 2020).

There have also been detailed comparative studies and reviews in recent years. Singh et al. (2019) mapped a roadmap of stock forecasting ML algorithms and promoted the use of hybrid models of sentiment or event data and technical indicators (Singh, Madan, Kumar, & Singh, 2019). Manasa, Praveenraj, and SR (2023) pushed the hybrid technique to the next level with the use of macroeconomic attributes and price signals in a Random Forest-XGBoost combination (Manasa, Praveenraj, & SR, 2023). Chen (2024) validated the power of boosting approaches on a set of international indexes, illustrating how boosting of decision stumps adaptively can compete with more sophisticated learners if tuning is judicious (Chen, 2024). Punjabi and Faridi (2024), in their review of SMP platforms in play, spoke of a trend away from single-pipeline approaches toward ensemble architectures for improved stability (Punjabi & Faridi, 2024).

Review articles have also paved the way by generalizing lessons from markets. The performances of well over a dozen ML algorithms were reviewed by Pahwa et al. (2017) using historical stock data, illustrating how moving average and ensemble model trees consistently provide good accuracy (Pahwa, Khalfay, Soni, & Vora, 2017). Guntaka et al. (2023) and Likhith et al. (2023) independently repeated the results on Indian indices, both illustrating how weighted ensemble of XGBoost, LightGBM, and Random Forest can provide well over 85 % test accuracies (Guntaka et al., 2023; Likhith et al., 2023). Cervelló-Royo and Guijarro (2020) presented a controlled comparison of SVM, random forest, and gradient boosting over European stocks, once more reaching the conclusion that boosting outperforms bagging by the whisker (Cervelló-Royo & Guijarro, 2020). The business significance of ML-based forecasts was also made apparent by Prasad and Seetharaman (2021), illustrating how portfolio returns are optimized when the model is most confident (Prasad & Seetharaman, 2021).

There have also been many conference publications to substantiate these observations. Kumar et al. (2018) contrasted supervised learners on multi-class trend forecasting and concluded the best precision-recall curve by gradient-boosted trees (Kumar, Dogra, Utreja, & Yadav, 2018). Rohit et al. (2020) added to this with sentiment indicators reporting moderate gains where text and prices were used (Rohit, Bhat, Manohar, & Mamatha, 2020). Sultana et al. (2024) and Jain et al. (2024) used ensemble approaches to BRICS markets against global shocks, model retraining frequency and automatic drift detection being useful in dynamic economies (Sultana et al., 2024; Jain, Saluja, Pimplapure, & Sahu, 2024). Vignesh (2018) and Dinesh et al. (2021) used simpler pipelines with moving averages and decision trees, the critical point being good cross-validation to avoid lookahead bias (Vignesh, 2018; Dinesh, R, Anusha, & R, 2021).

Compared to these predominantly single-market studies, Rath et al. (2024) and Deshpande et al. (2022) carried out meta-analysis of numerous published pieces of research and established that no approach universally supplants and acknowledged ensemble stacking's potential (Rath, Das, & Pattanayak, 2024; Deshpande, Lunkad, Kunjir, & Ingle, 2022). Deterministic data preprocessing and feature importance analysis were highlighted by Walunjkar et al. (2022) and Patel et al. (2015) to maintain replicability (Walunjkar, Desai, Gholap, & Shukla, 2022; Patel, Shah, Thakkar, & Kotecha, 2015). Mehta, Pandya, and Kotecha (2021) reported the combination of social media sentiment and deep-learning forecasts for the first time, where sentiment-enhanced models beat baseline prices (Mehta, Pandya, & Kotecha 2021).

In spite of this vast literature, there are significant voids. First, while many studies have confirmed ensemble tree models on established markets or pan-regional indexes, not many of them have attempted strict time-series cross-validation on long-horizon emerging-market datasets like the DSE 2000-2025. Second, while most research has focused on single-model or naive ensemble accuracy, none of the authors have attempted a weighted soft-voting technique based on error-pattern complementarity. Third, while some authors have attempted sentiment or macro indicators, relative usefulness of engineered technical features such as multi-horizon rolling statistics, momentum constraints, and target-encoded scrip identifiers have yet to be systematically evaluated in an ensemble setting. This work fills these voids by suggesting the use of XGBoost, LightGBM, and Random Forest in a weighted soft-voting ensemble, validated through strict time-series cross-validation of a 25-year DSE dataset, and through detailed examination of the relative contribution of the features to trend forecast ability.

## III. METHODOLOGY & EXPERIMENTAL RESULTS

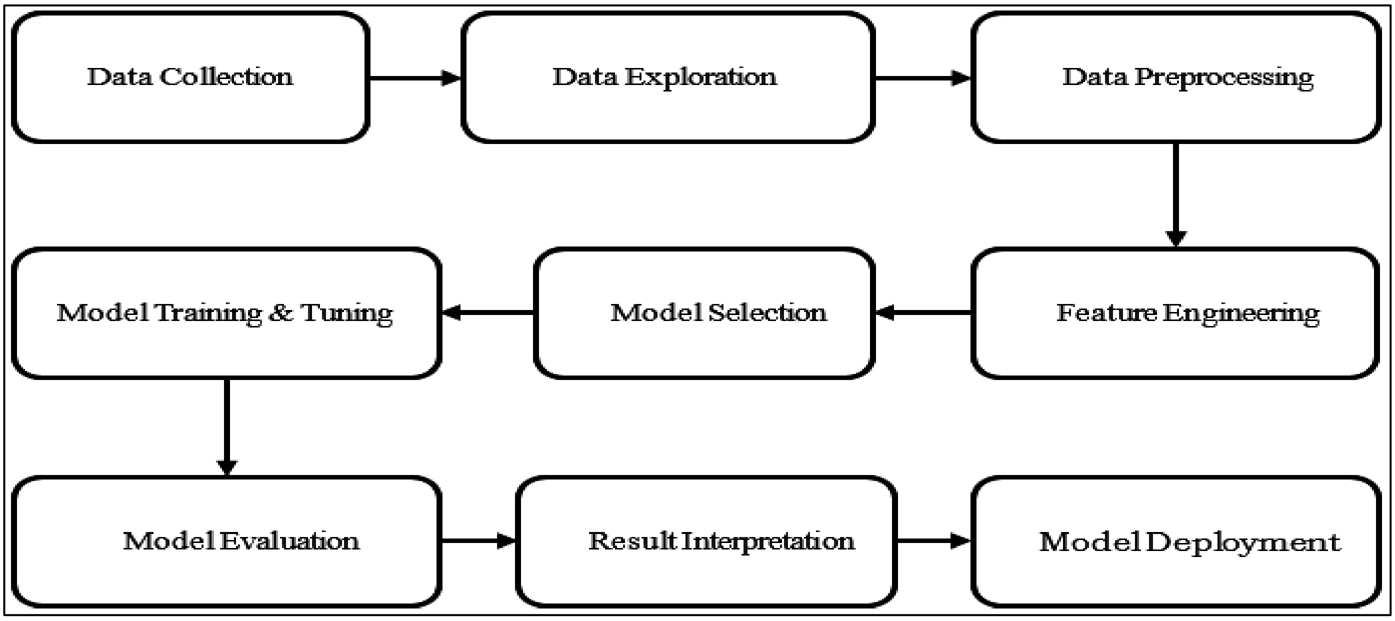


Fig 1 Flow Chart of Methodology

➢ *Data Collection*

We utilized a full dataset of stock prices of the Dhaka Stock Exchange (DSE), which we obtained from Kaggle, accessible at https://www.kaggle.com/datasets/muhammedalif/dsc-prices. It has 1,381,571 rows and 7 columns of daily trading data (Date, Scrip, Open, High, Low, Close, and Volume) of 464 companies from 1 January 2000 to 26 February 2025.

➢ *Data Exploration*

Data exploration, also known as Exploratory Data Analysis (EDA), is the process of examining datasets to summarize their major characteristics and understand the underlying structure. It allows us to detect inconsistencies, investigate the relationship of the features, identify trends, and make preprocessing and modeling decisions intelligently. The dataset consists of 1,381,571 rows and 7 columns of daily stock market data (Date, Scrip, Open, High, Low, Close, Volume) of 464 companies from the period of January 1, 2000, to February 26, 2025. We begin with checking the shape of the dataset, the data types, and null values through the use of df.head(), df.tail(), df.info(), df.describe(), and df.isnull().sum(). These offer the facility to ascertain the integrity and completeness of the data. We comprehend the range, distribution, and central value of the numeric variables.

We employ a correlation matrix to quantify linear correlation among variables and subsequently apply a heatmap for visualization:

- Mathematically, Pearson correlation coefficient P_xy can be defined as:

$$P_{xy} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

- This helps us identify multicollinearity and duplicate features which may be harmful to the model's performance.

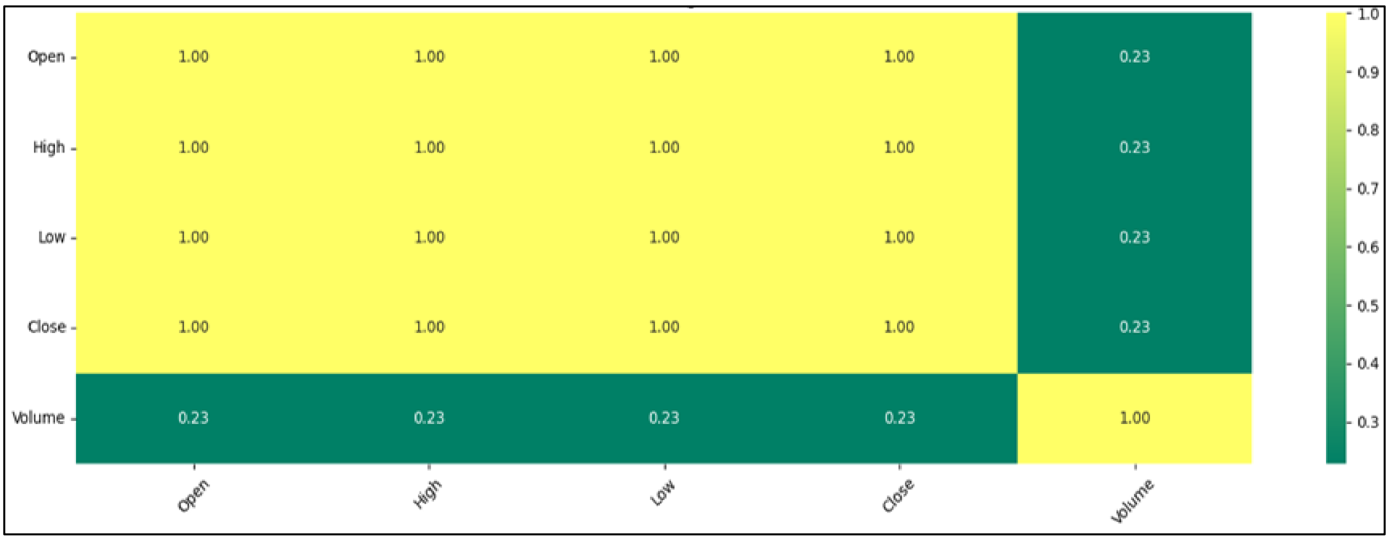


Fig 2 Correlation Heatmap of Dataset's Features

For a better idea of what each stock does over time, we graph the historical closing price. Because there are so many stocks in the data set, we simply pick 6 stocks to represent typical behavior:

- It is employed to find temporal behavior, volatility, and seasonality of the stock prices.
- It assists in developing hypotheses regarding which stocks will be more stable or predictive.

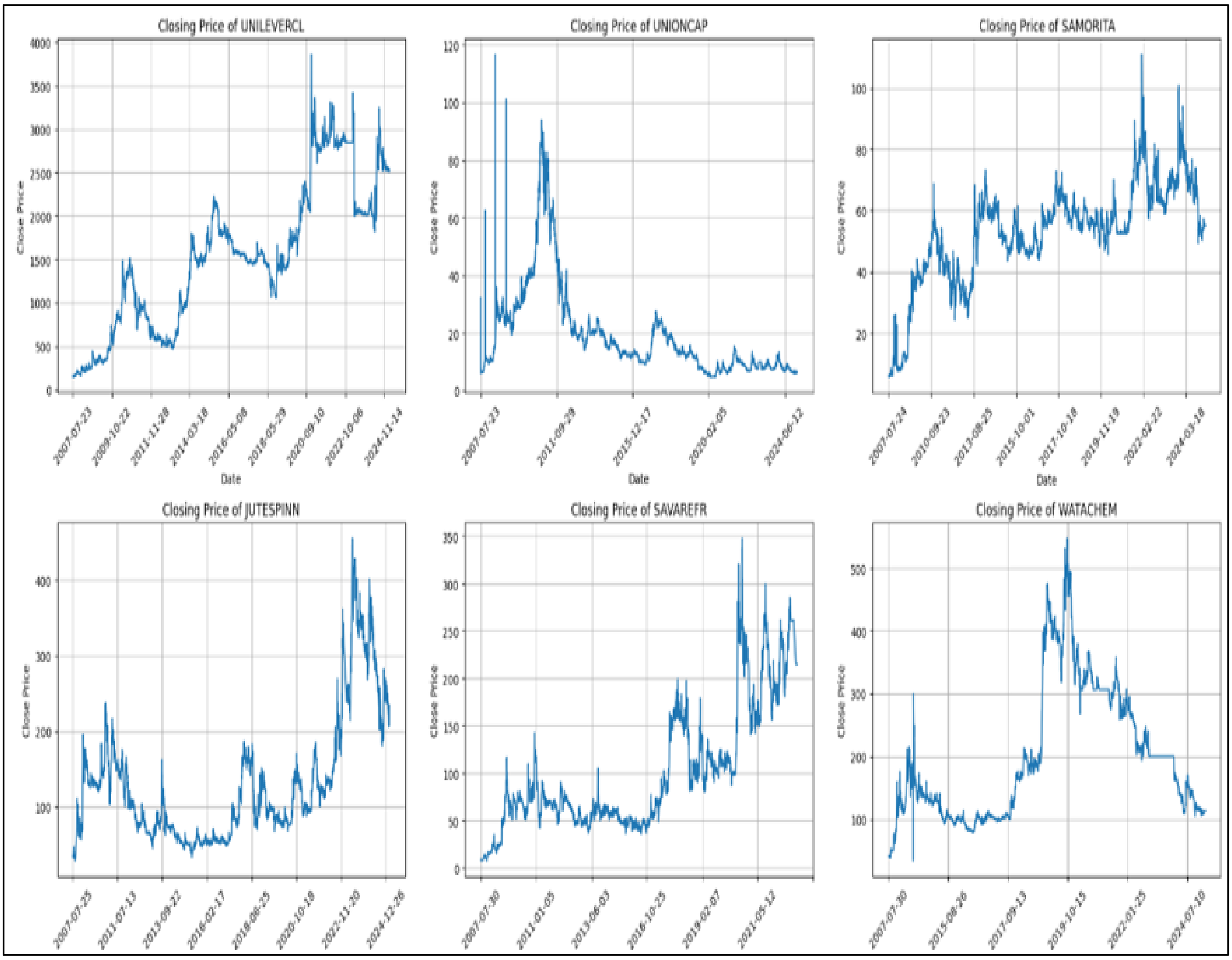


Fig 3 Historical View of Closing Price

➢ *Data Preprocessing*

Data preprocessing is an essential process in every machine learning pipeline. It is the process of bringing raw data to a clean, uniform, and structured form to utilize in the modeling process. The key purpose is to rectify the missing values, normalize the data, code the categorical variables, and incorporate temporal continuity to the time series data.

We begin by filling in or removing the missing entries df.dropna(inplace=True). The line removes the rows with NaN values. When there is missing data in a time series, continuity of trends can be destroyed. Removing the rows with incomplete data (in contrast to imputation) gives the LSTM complete, unbroken sequences.

In a bid to preserve temporal ordering and for feature engineering. Temporal ordering is necessary even for non-sequence methods since the features would be based on the lagged variables or moving averages. This also enables us to create sensible time-based features for every individual stock.

We only include columns needed for the modeling df = df[['Date', 'Stock', 'Close']] We retain only the Date, the stock ID, and the Close price, which are the only columns needed for univariate time series forecasting. Downstream processing is reduced and redundant noise from unwanted features is avoided.

We normalize the closing price to the range 0 to 1 using Min-Max normalization.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

This ensures data magnitude consistency, improving model convergence and speeding up the process. It also prevents highly volatile stocks from dominating the loss function.

➢ *Feature Engineering*

Feature Engineering is the process of transforming raw data to useful input variables (features) the model can utilize to enhance its ability to make good forecasts. For the stock market data, this includes the formation of lag-based indicators, moving averages, price momentum, volatility, and time-based trends. These include capturing temporal dependence as well as market behavior.

We derived some features from raw stock prices at historical time points to aid model performance:

- Lagged Closing Price: Stores the prior day's price for use as a reference in return and trend determination.

$$prev\_close_t = P_{t-1} \tag{3}$$

- Trend Label (Target Variable): Converts price prediction to a binary classification task (Up = 1, Down = 0).

$$trend_t = \begin{cases} 1, if\ P_t > P_{t-1} \\ 0, Otherwise \end{cases} \tag{4}$$

- Price Change and Daily Return: Stores relative movement and volatility.

Price Change: $P_t = P_{t-1} - P_{t-2}$ (5)

Return: $r_t = \frac{P_{t-1} - P_{t-2}}{P_{t-2}}$ (6)

- Rolling Average and Volatility: Remove noise and keep track of trends and volatility.

Rolling Mean: $\mu_{t,k} = \frac{1}{k}\sum_{i=1}^{k} P_{t-i}$ (7)

Rolling Std Dev: $\sigma_{t,k} = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(P_{t-i} - \mu_{t,k})^2}$ (8)

- Momentum & Exponential Moving Average (EMA): Identify short-term profits or losses.

Momentum: $M_{t,6} = P_{t-1} - P_{t-6}$ (9)

EMA: Weighted average that assigns more weight to recent prices.

- Lagged Volume: Past volume can reflect the degree of interest or activity in the market prior to a price movement.
- Date-Time Features: Time segmentation allows the model to see monthlies or weekday trends.
- Visualization of Engineered Feature shown in Figure 4 – Figure 9.

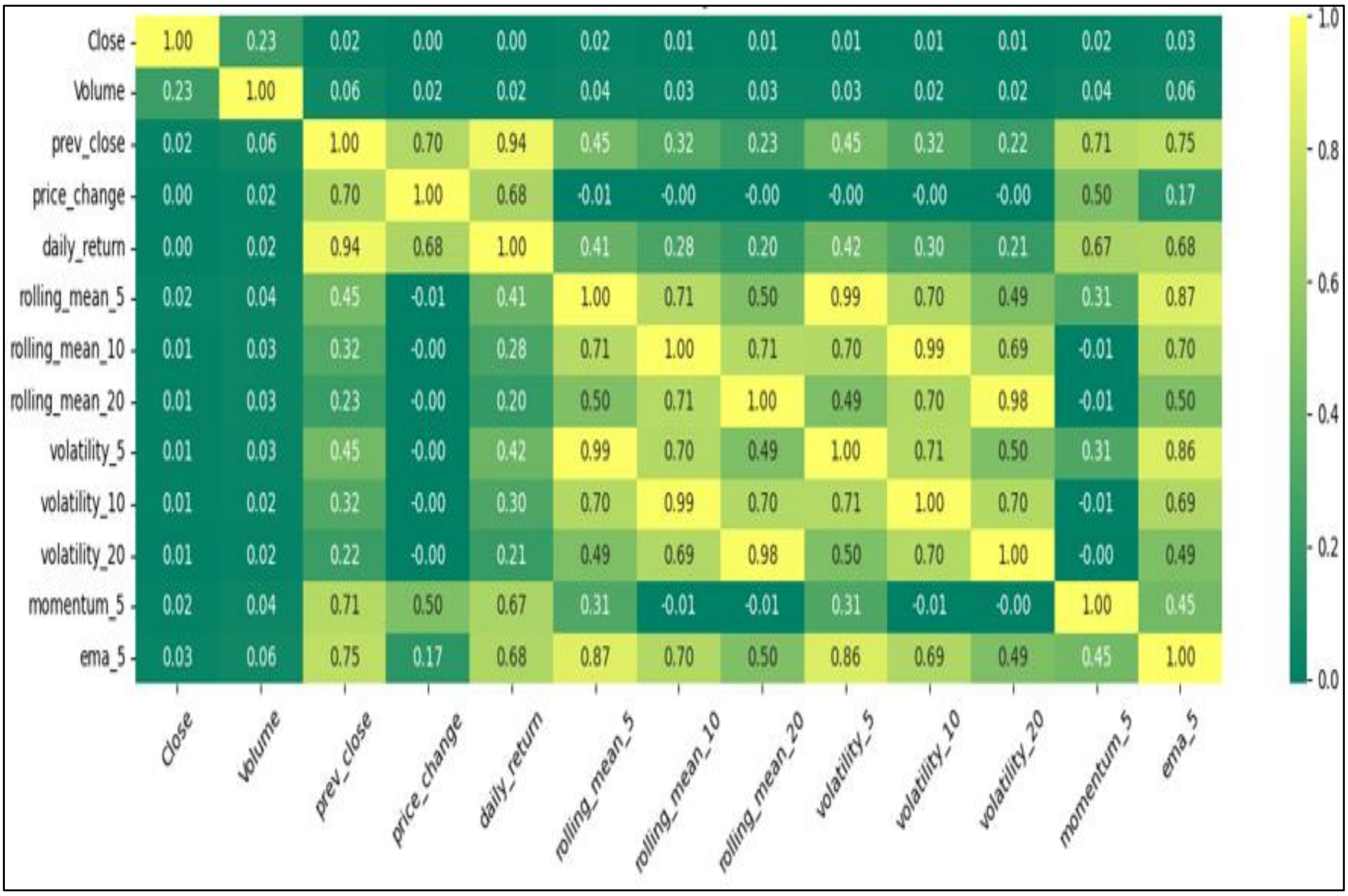


Fig 4 Correlation Heatmap (Raw + Engineered Features)

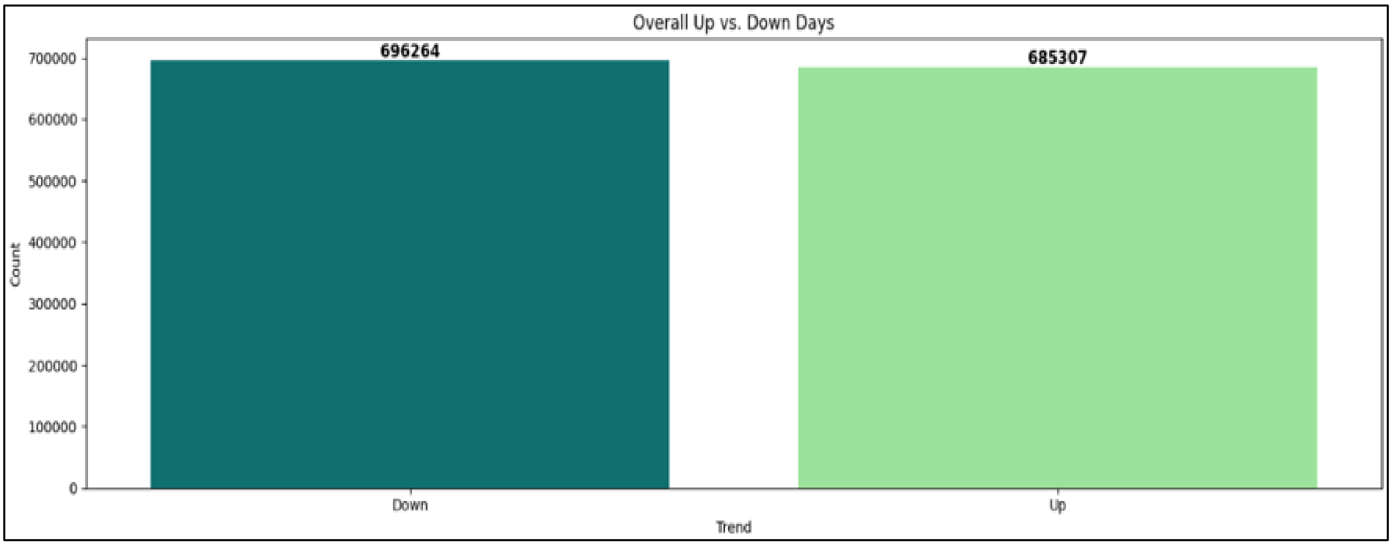

Fig 5 Monthly Up-Rate Bar Chart

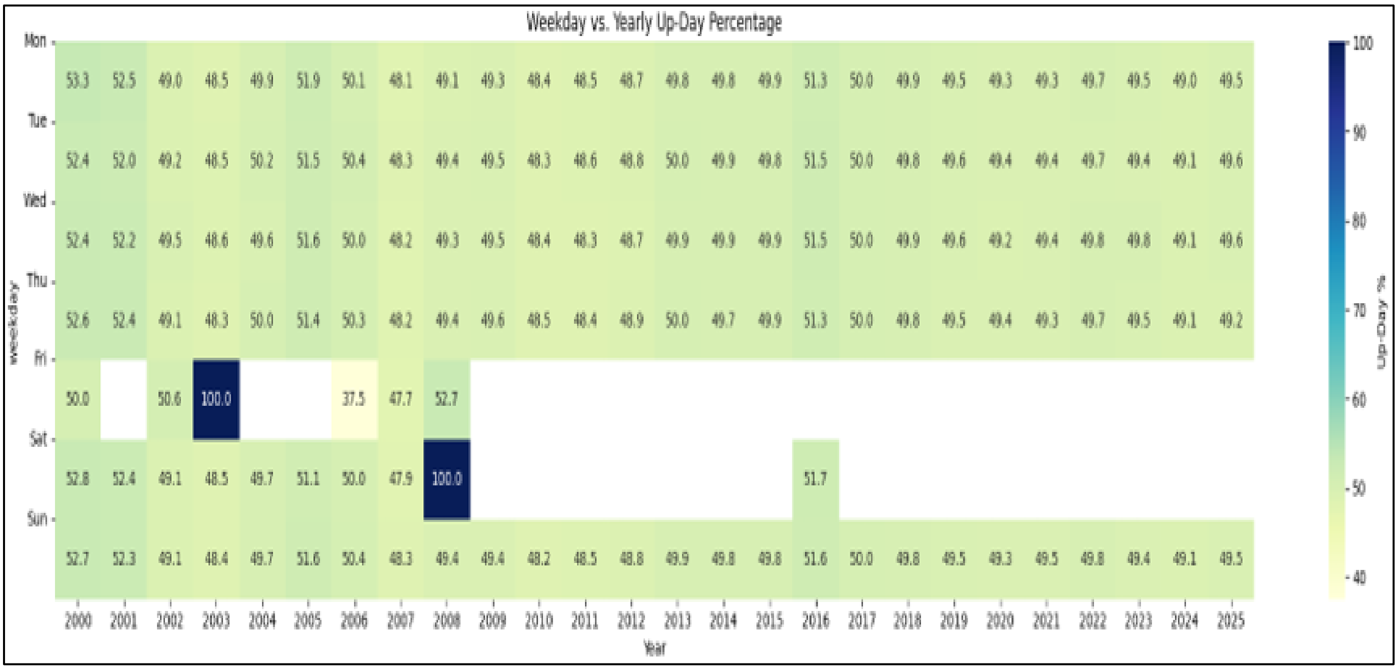

Fig 6 Weekday Heatmap of Trend Frequency

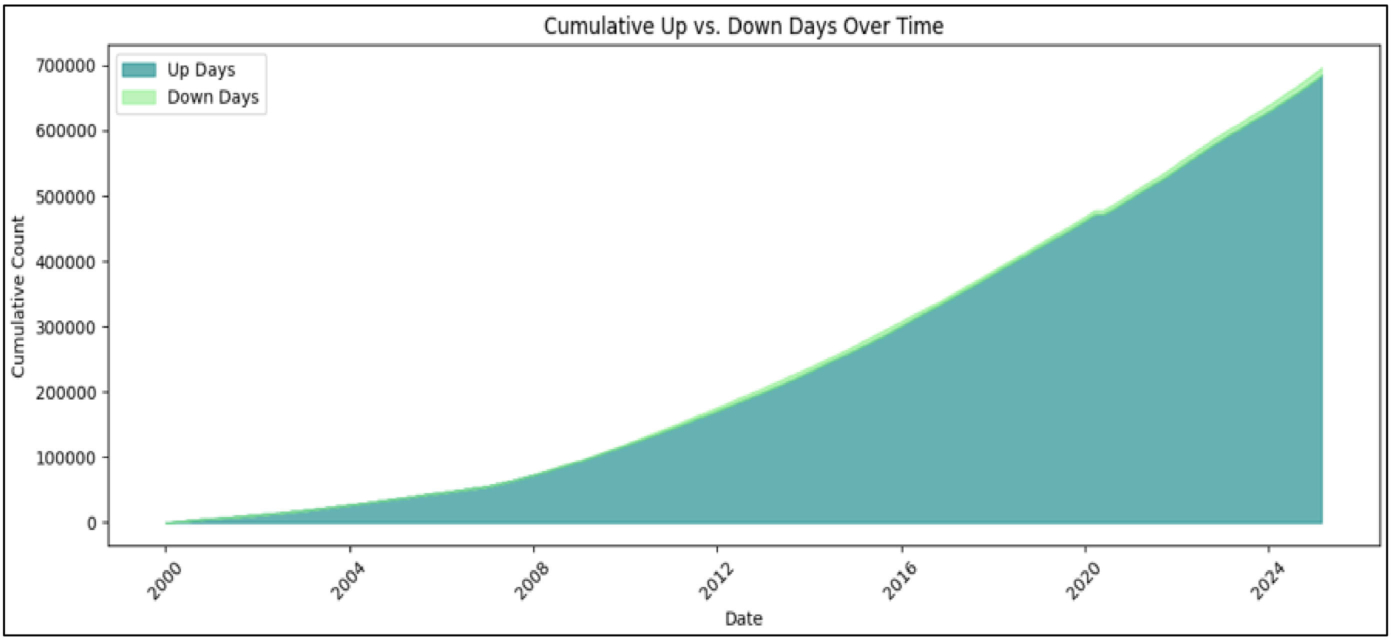

Fig 7 Stacked Area Chart – Cumulative Up/Down Days

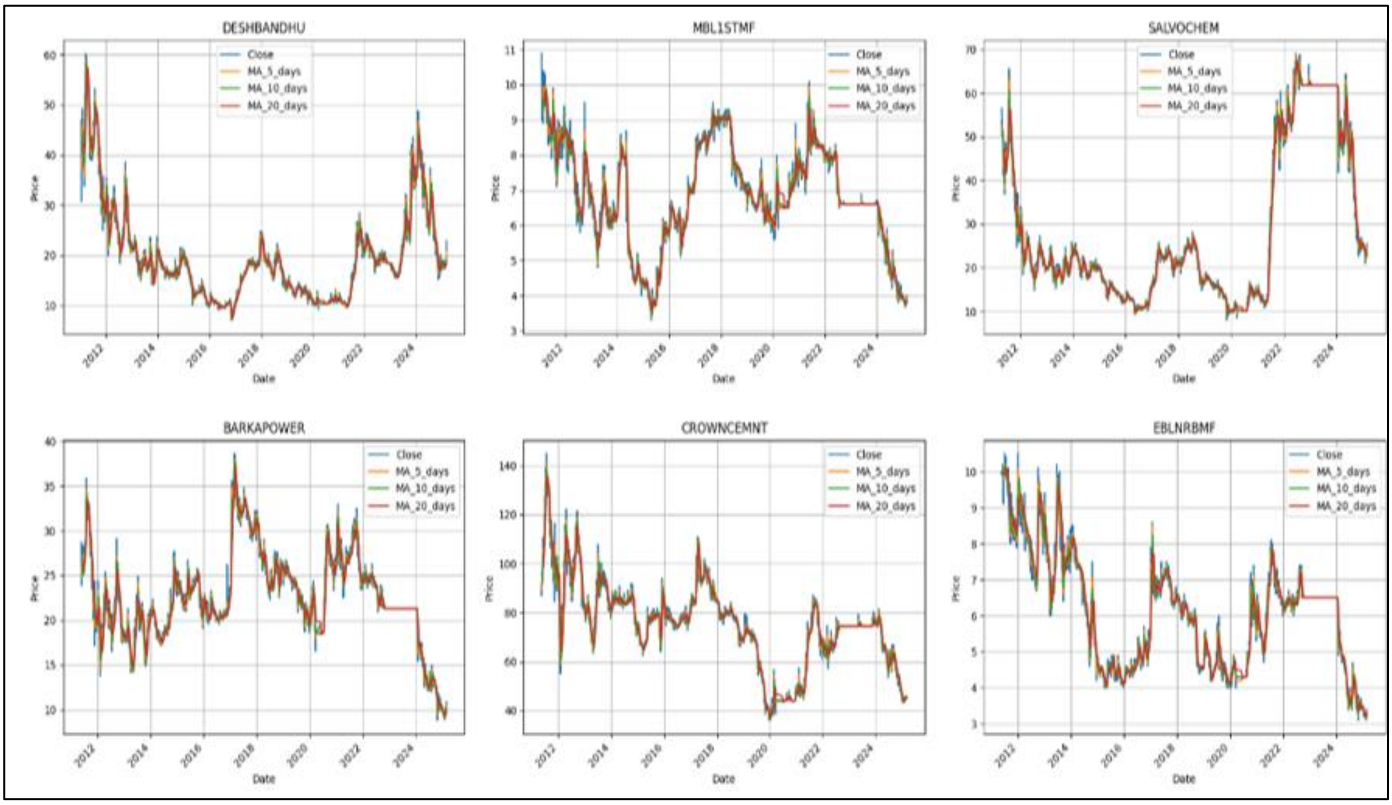

Fig 8 Moving Averages Plot (5, 10, 20 days)

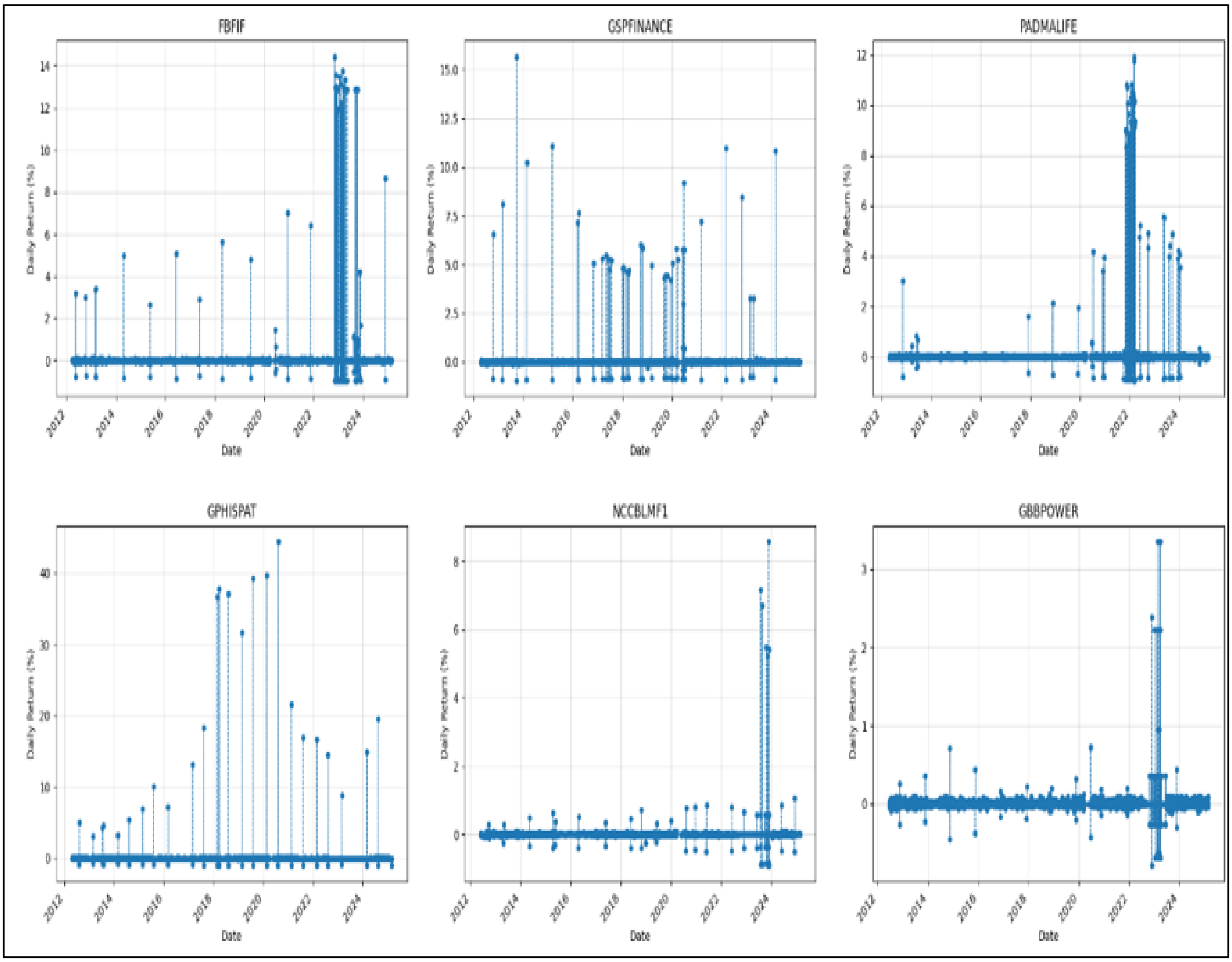

Fig 9 Average Daily Return Per Stock

➢ *Model Selection*

Model selection is an important stage in every machine learning process, most significantly in financial time series forecasting, which is volatile, non-stationary, and susceptible to noise. We sought a combination of good-performing, independent classifiers with complementary abilities where their combination in an ensemble would yield improvement. After rigorous testing, we chose three individual models including XGBoost, LightGBM, and Random Forest and combined them in the ultimate ensemble with the help of Soft Voting. These models were chosen on account of their stability, ability to handle tabular data, and proven success in a broad scope of structured forecasting tasks.

We began with XGBoost (Extreme Gradient Boosting), an advanced and proven gradient boosting with in-built regularization. XGBoost can model linear as well as non-linear relationships, can effectively cope with missing values, and even accommodates monotonic constraints, which we used to specify a positive relationship between some of the engineered variables (e.g., momentum) and direction of the target. Due to the sequential nature of our data and the engineered variables which might introduce lag, XGBoost's in-built overfitting control using regularization (`reg_lambda`, `reg_alpha`) made the selection a correct one for stable forecasting.

Then we appended LightGBM (Light Gradient Boosting Machine), another gradient boosting library optimized for speed and efficiency. LightGBM works with histogram-based decision trees, which are far more memory and speed-efficient compared to conventional gradient boosting approaches. It is also particularly well-suited to work with big datasets with vast feature spaces, such as presented in our engineered feature set. Its leaf-wise tree growth option and subsampling of features option (`feature_fraction`) were beneficial in picking up complex patterns in our trend classification problem, and having the regularizing terms helped in preventing overfitting problems.

Random Forest, with its ensemble of decision trees learned from bootstrapped samples and random subsets of features, was chosen for its interpretability and overfitting immunity. Unlike greedy tree-building algorithms used in boosting, Random Forest builds trees in parallel, reducing variance and generalizing better. It is a highly effective means of modeling interacting effects between properties in a natural manner and is also quite intuitive with regard to the resultant feature importance scores. Given the noisy and high-variance nature of stock market trends, the ability of Random Forest to regularize the forecasts and avoid overfitting was a welcome addition to our arsenal.

Finally, to utilize the strength of each of the three models, we have created a Soft Voting Ensemble. The strategy accumulates the predicted probabilities of the models of prediction (the XGBoost model, the LightGBM model, and the Random Forest model) and gives the class with the highest mean probability. We have also assigned specific weights to each of the models based on their respective performances to achieve the best combination. The ensemble strategy utilizes the model diversity and reduces the risks of over-reliance on the bias and defaults of a particular model, resulting in a more stable and well-balanced trend-prediction mechanism.

➢ *Model Training & Tuning*

We trained all machine learning models and also tuned their performances via hyperparameter optimization. Because we have time-series stock market data, the chronological order while training models was to be maintained. We utilized a collection of engineered features (`x`) such as lagged prices, rolling statistics, volatility signals, and momentum signals, along with stock identifier encoded variables, and target variable (`y`) being the binary `trend` of the market direction (up = 1, down = 0).

To ensure the models generalized and did not memorize the data but rather learned general patterns, their hyperparameters were adjusted iteratively following financial modeling best practice and industry knowledge.

We initially trained the XGBoost model emphasizing regularization (`reg_lambda=3.0`, `reg_alpha=1.0`) and constraining interactions among the features. We utilized a comparatively flat tree structure with a depth of 3 and also adjusted the learning rate (0.05) to avoid overfitting. We additionally used a constraint of monotonicity for the `momentum_5` function to ensure that normal behavior within the forecasts is always ensured i.e., a higher momentum does not reduce the possibility of an upward trend.

We used the same learning rate of (`0.05`) and limited number of leaves (`num_leaves=10`) in the interest of model simplicity and effectiveness. We used `extra_trees=True` to add controlled randomness in order to assist with generalizability. We were careful to set the regularisation terms as hyperparamters (`colsample_bytree`, `feature_fraction`) and feature sampling in order to control overfitting but still produce good quality predictions.

For Random Forest classifier, we employ `n_estimators=100` and `max_depth=5` so that the constituent trees are interpretable and not excessively deep. We employed `min_samples_split=20` and `min_samples_leaf=10` to stabilize splits and prevent fitting noise. We added the `class_weight='balanced'` parameter to achieve class imbalance adjustment of the target variable.

Finally, we averaged the three models together using a Soft Voting Ensemble, which takes probabilistic output and not hard labels. We weighted the entire set of the models (`0.32` for XGBoost, `0.4` for LightGBM, and `0.28` for Random Forest) based on their respective tuning cross-validation score. We used this ensemble in preparation to leverage the power of each algorithm and reducing the output.

➢ *Model Evaluation*

We outline the evaluation procedure that was followed to ascertain the effectiveness and applicability of our trained models. Since the data in the stock market is time-series in nature, it is crucial that our evaluation preserves adherence to the chronological structure of the dataset to avoid data leakage and over-estimation of performance.

To this end, we used a TimeSeriesSplit cross-validation approach. It divides the data into a number of sequential train and test splits in such a way that all the train data comes before the test data in time. This mimics the situation in real-world forecasting in which data from the future is never used for training. We used 5 splits, which provided a reasonable estimate of how each model would generalize to new, unseen market conditions.

In each split, the model was trained on the prior segment and tested on the subsequent segment. Training accuracy and testing accuracy were noted for each fold to check for overfitting or underfitting tendencies. Aside from accuracy, we noted key classification metrics like precision, recall, and F1-score. These metrics tell us how each model is doing relative to the imbalance or skew in the upward and downward trends.

We also graphed the confusion matrix for each model to read off the frequency and type of prediction errors. This gave information on whether a model was biased towards one class (e.g., overpredicting rising trends) and allowed us to look at misclassifications at a fine-grained level.

➢ *Result Interpretation*

Table 1 Classification report for XGBoost model.

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Down | 0.84 | 0.88 | 0.86 | 580528 |
| Up | 0.87 | 0.83 | 0.85 | 570762 |
| Accuracy | | | 0.85 | 1151290 |
| Macro Avg | 0.86 | 0.85 | 0.85 | 1151290 |
| Weighted Avg | 0.85 | 0.85 | 0.85 | 1151290 |

In examining the performance of the XGBoost classifier, we observe that it achieved an average cross-validated training accuracy of 88.82 % and a test accuracy of 85.42 %. Its precision (0.855), recall (0.854), and F1-score (0.854) indicate a well-balanced ability to identify both up and down days, with the confusion matrix illustrating 84 % correct identification of down days and 87 % of up days (Figure 10). The detailed classification report (Table 1) confirms this balanced performance across classes, underscoring XGBoost's strength in learning complex, non-linear patterns in our engineered feature space.

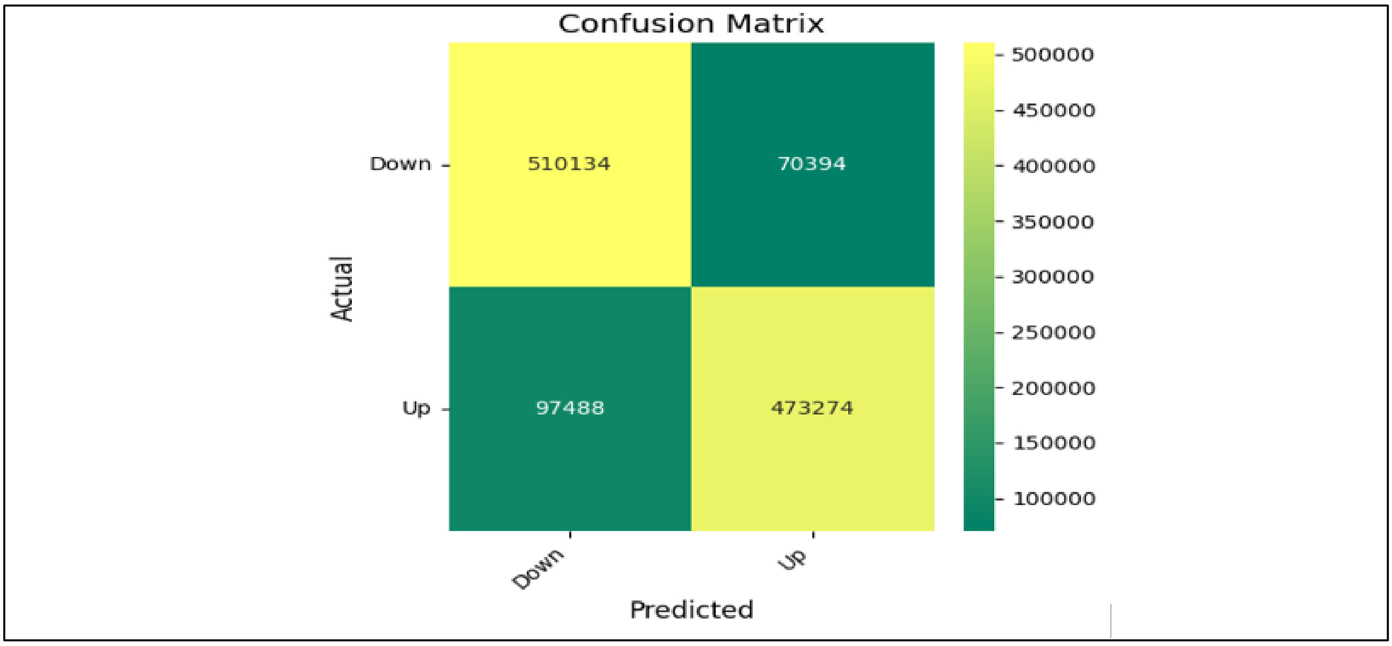


Fig 10 Confusion Matrix of XGBoost

Table 2 Classification report for LightGBM model.

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Down | 0.85 | 0.87 | 0.86 | 580528 |
| Up | 0.87 | 0.84 | 0.85 | 570762 |
| Accuracy | | | 0.86 | 1151290 |
| Macro Avg | 0.86 | 0.86 | 0.86 | 1151290 |
| Weighted Avg | 0.86 | 0.86 | 0.86 | 1151290 |

The LightGBM model delivered similarly robust results, posting an average training accuracy of 87.90 % and an even higher test accuracy of 85.69 %. With a precision of 0.857, recall of 0.857, and F1-score of 0.857, LightGBM demonstrated slightly improved generalization over XGBoost, particularly in its ability to correctly predict up days. Its confusion matrix (Figure 11) shows a marginal reduction in false positives for upward trends, and the accompanying classification report (Table 2) confirms its efficacy in handling the imbalanced class distribution.

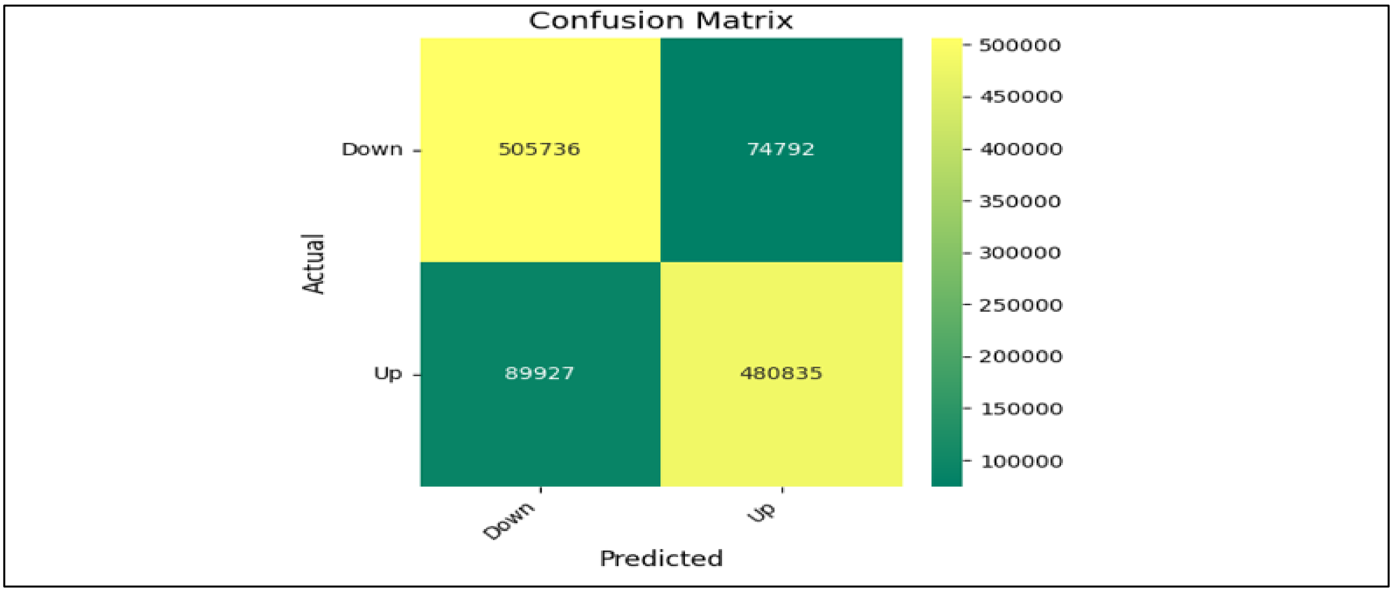


Fig 11 Confusion Matrix of LightGBM

Table 3 Classification report for Random Forest model.

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Down | 0.84 | 0.87 | 0.86 | 580528 |
| Up | 0.87 | 0.84 | 0.85 | 570762 |
| Accuracy | | | 0.85 | 1151290 |
| Macro Avg | 0.85 | 0.85 | 0.85 | 1151290 |
| Weighted Avg | 0.85 | 0.85 | 0.85 | 1151290 |

Random Forest achieved 88.39 % training accuracy and 85.20 % on the test folds, with precision, recall, and F1-score all approximately 0.852. As shown in its confusion matrix (Figure 12), Random Forest correctly classifies 84 % of downward movements and 86 % of upward movements, slightly trailing the gradient boosting methods. The classification report for Random Forest (Table 3) reflects its reliable yet somewhat conservative predictions, highlighting its robustness but also its tendency toward higher bias relative to the boosting algorithms.

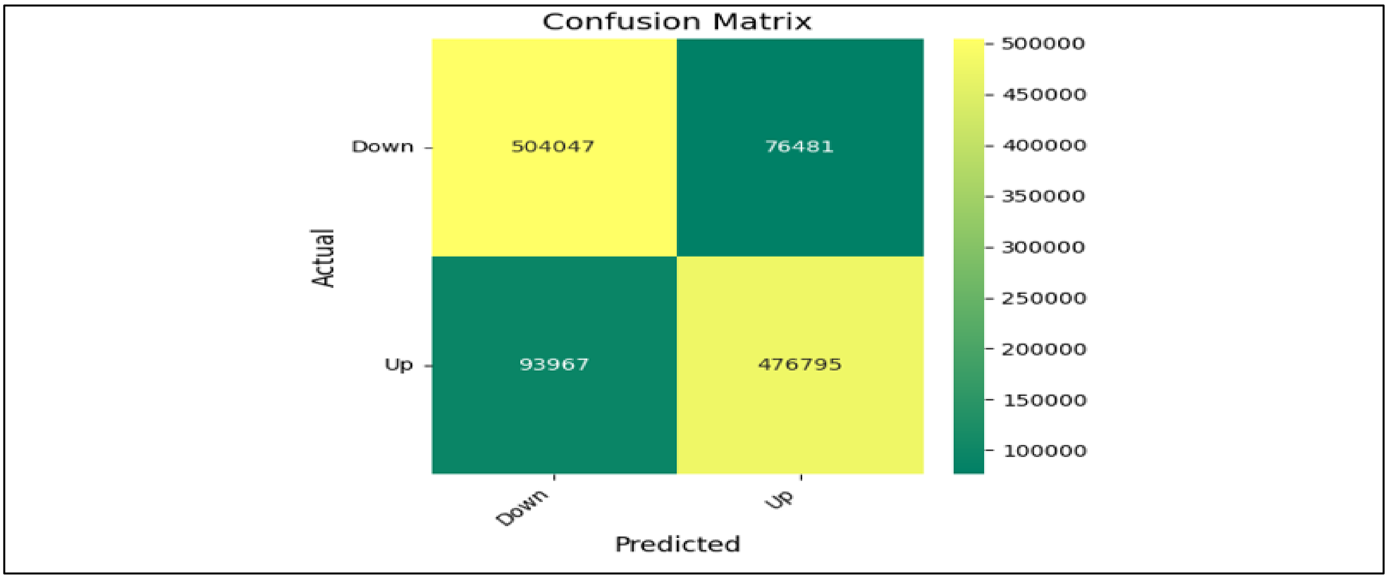


Fig 12 Confusion Matrix of Random Forest

Table 4 Classification report for Soft-voting Ensemble model.

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Down | 0.85 | 0.87 | 0.86 | 580528 |
| Up | 0.87 | 0.84 | 0.85 | 570762 |
| Accuracy | | | 0.86 | 1151290 |
| Macro Avg | 0.86 | 0.86 | 0.86 | 1151290 |
| Weighted Avg | 0.86 | 0.86 | 0.86 | 1151290 |

Our soft-voting ensemble, which combines XGBoost, LightGBM, and Random Forest with empirically derived weights, produced a training accuracy of 88.48 % and a test accuracy of 85.68 %. Its precision (0.857), recall (0.857), and F1-score (0.857) match or slightly exceed individual model performance, as evidenced by the confusion matrix in Figure 13 and classification metrics in Table 4. This consensus approach successfully harnesses the complementary strengths of each base learner, reducing individual biases and yielding a more stable prediction profile.

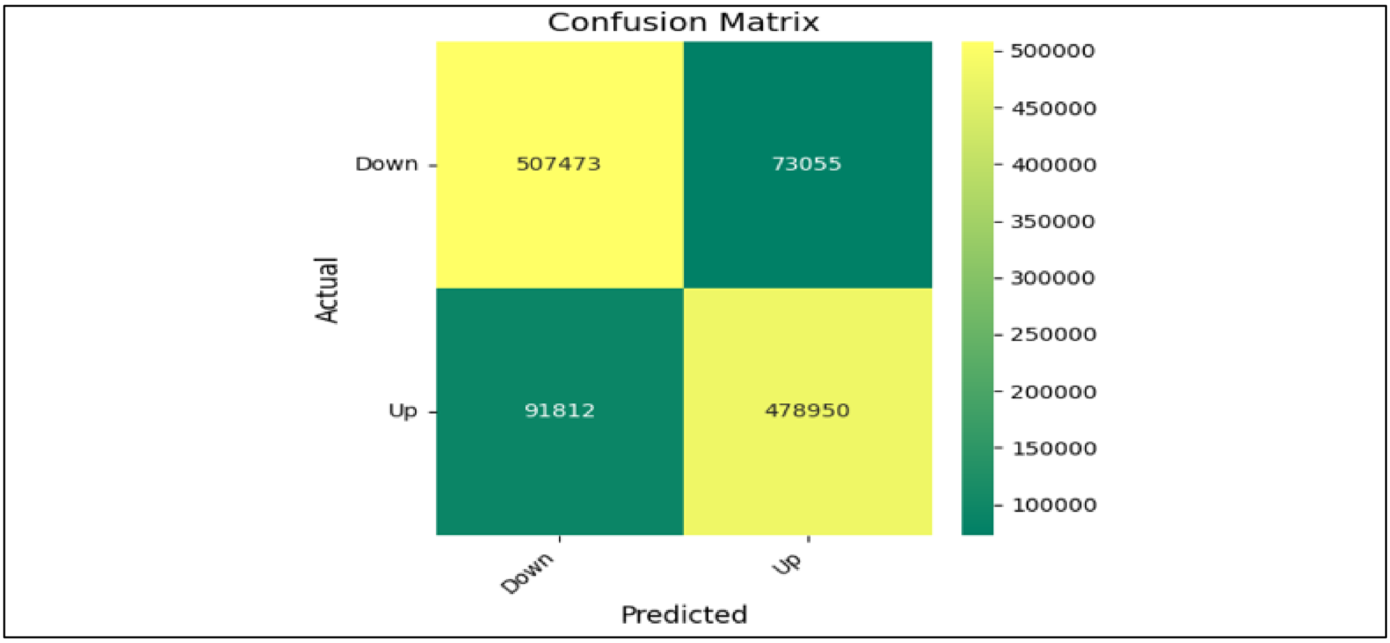


Fig 13 Confusion Matrix of Soft-voting Ensemble

Finally, to compare overall model behaviors, we present a bar chart of the average training versus test accuracies for all four approaches (Figure 14). This visualization highlights the consistent gap between training and test performance, which indicating minimal overfitting and confirms that the ensemble's balanced accuracy closely tracks or surpasses that of its constituent models.

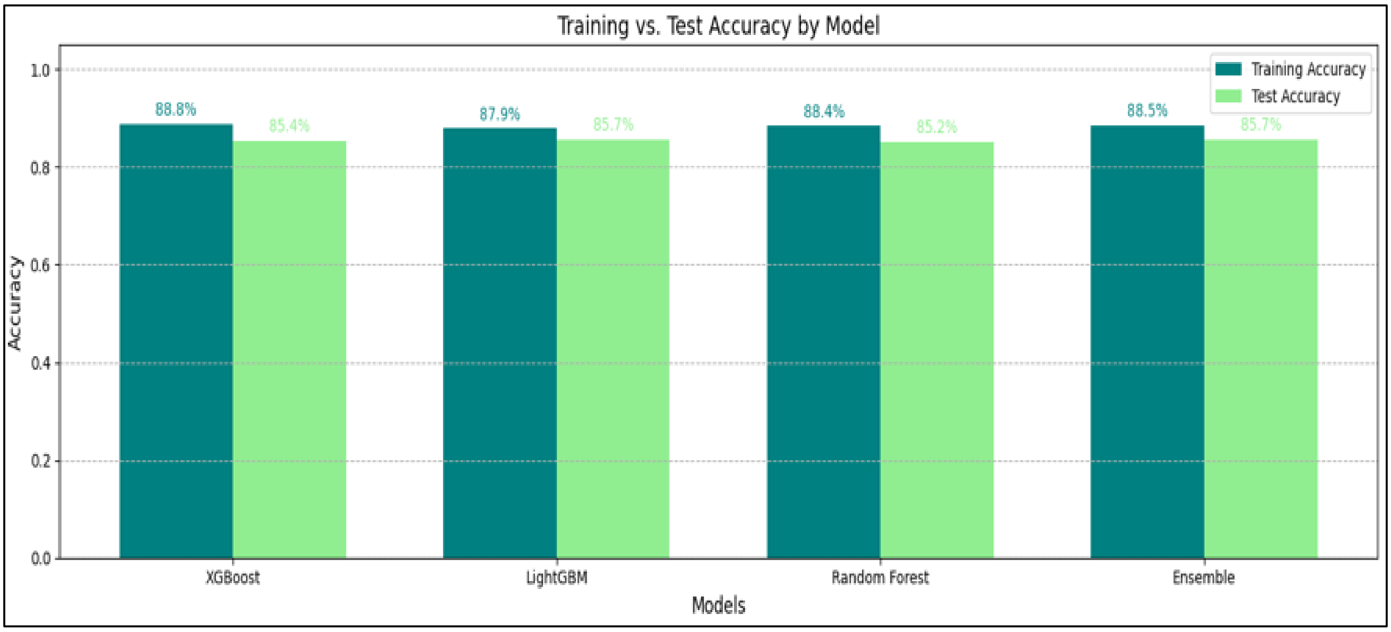


Fig 14 Training vs. Test Accuracy of all Models

➢ *System Architecture & Real-Time Deployment*

To ensure the practical usefulness of our predictive model, we developed a full-stack web application that utilizes our trained model for real-time trend prediction. The deployment of our model will ensure its robustness under real-world scenarios. The developed model can prove to be highly useful for financial analysts and investors.

- *Backend Implementation*

The backend is written in Python with the help of the Flask framework (v2.0+) with support for cross-origin requests via the included CORS library. The overall architecture is based on a modular design with three main parts: model persistence, feature engineering replication, and REST API.

This is done by utilizing joblib's serialization functionality on the trained LightGBM classifier, saving it to a file named lgbm_trend_classifier.joblib, which is then reloaded during application startup by a custom wrapper class named TrendClassifier. This ensures that the precise feature engineering process used during the development of the model, as described in Section 4.4, is repeated during deployment, thus guaranteeing consistency between the two environments. The feature engineering process is dynamic, calculating all thirteen features, including lagged price data, statistics, momentum, and target-encoded scrip, from the incoming market data stream.

A specific endpoint for the RESTful API, named "/predict," is created to handle POST requests with JSON payloads that contain the seven required fields for the input data: Date, Scrip, Open, High, Low, Close, Volume. This endpoint checks the input data for completeness and correctness in terms of data type, executes the feature engineering pipeline in real-time, makes the prediction using the created model, and generates JSON response with the prediction results, including the UP/DOWN class with corresponding probabilities for each class.

- *Frontend Interface*

Finally, the client-side application is written in such a manner using HTML, CSS, and JavaScript to provide the highest level of compatibility with the minimum level of latency. The application has a clean and simple form to input the prevailing market data. Additionally, the application validates the form fields to provide the highest level of usability. The application utilizes JavaScript to provide the highest level of async communication with the server to fetch the prediction data.

- *Deployment & Performance*

The application is also packaged with an explicit set of dependencies that can be easily managed using requirements.txt. The application also ensures that all containerization guidelines for scalable deployment have been met. Performance metrics show that, on average, it takes less than 300 milliseconds per request for data validation, feature engineering, and prediction. This is an indication that the system is ready for real-time financial decisions. The system architecture is shown in Figure 15, while Figure 16 displays an example of an application interface for the deployed system.

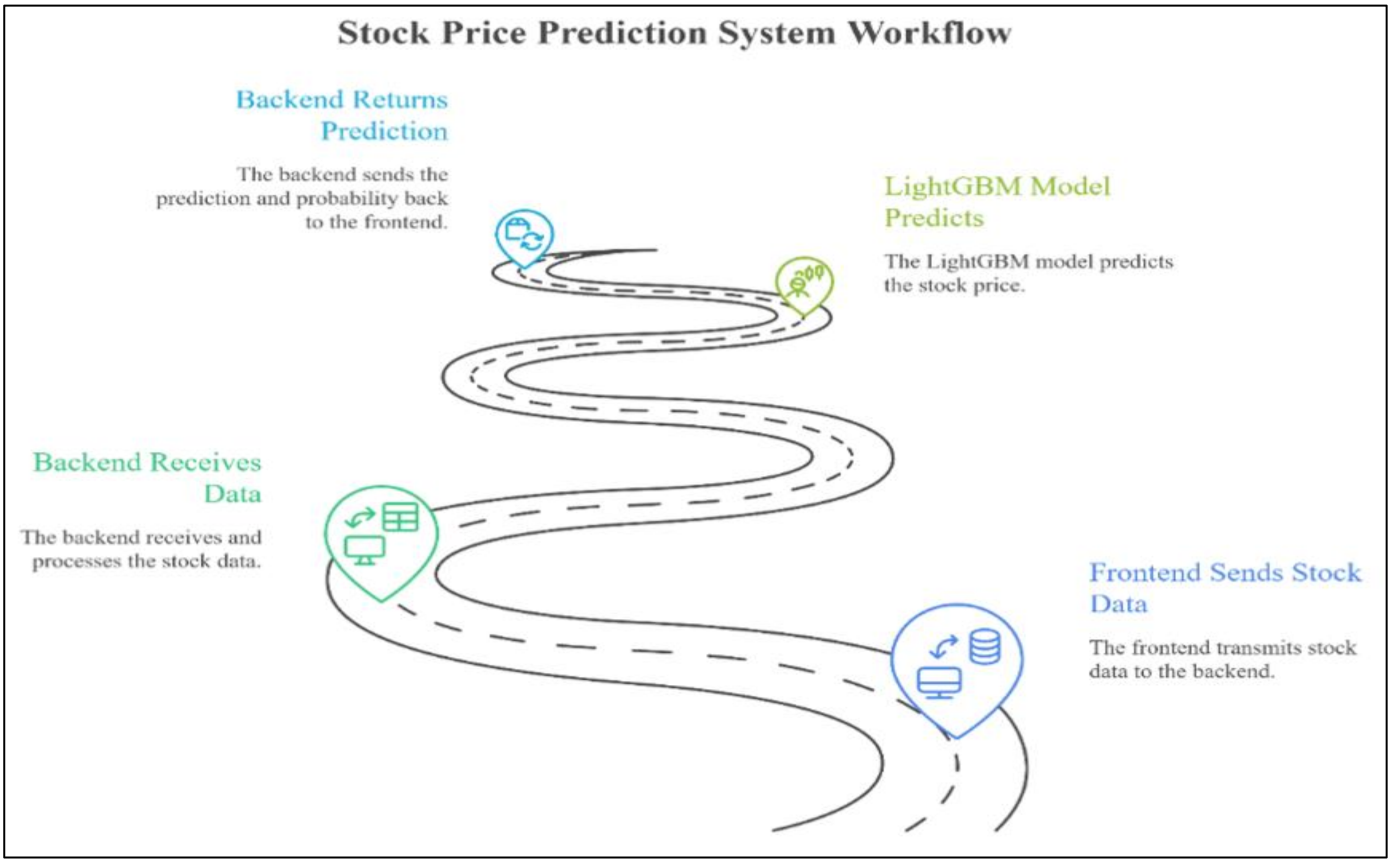


Fig 15 Web Application Architecture

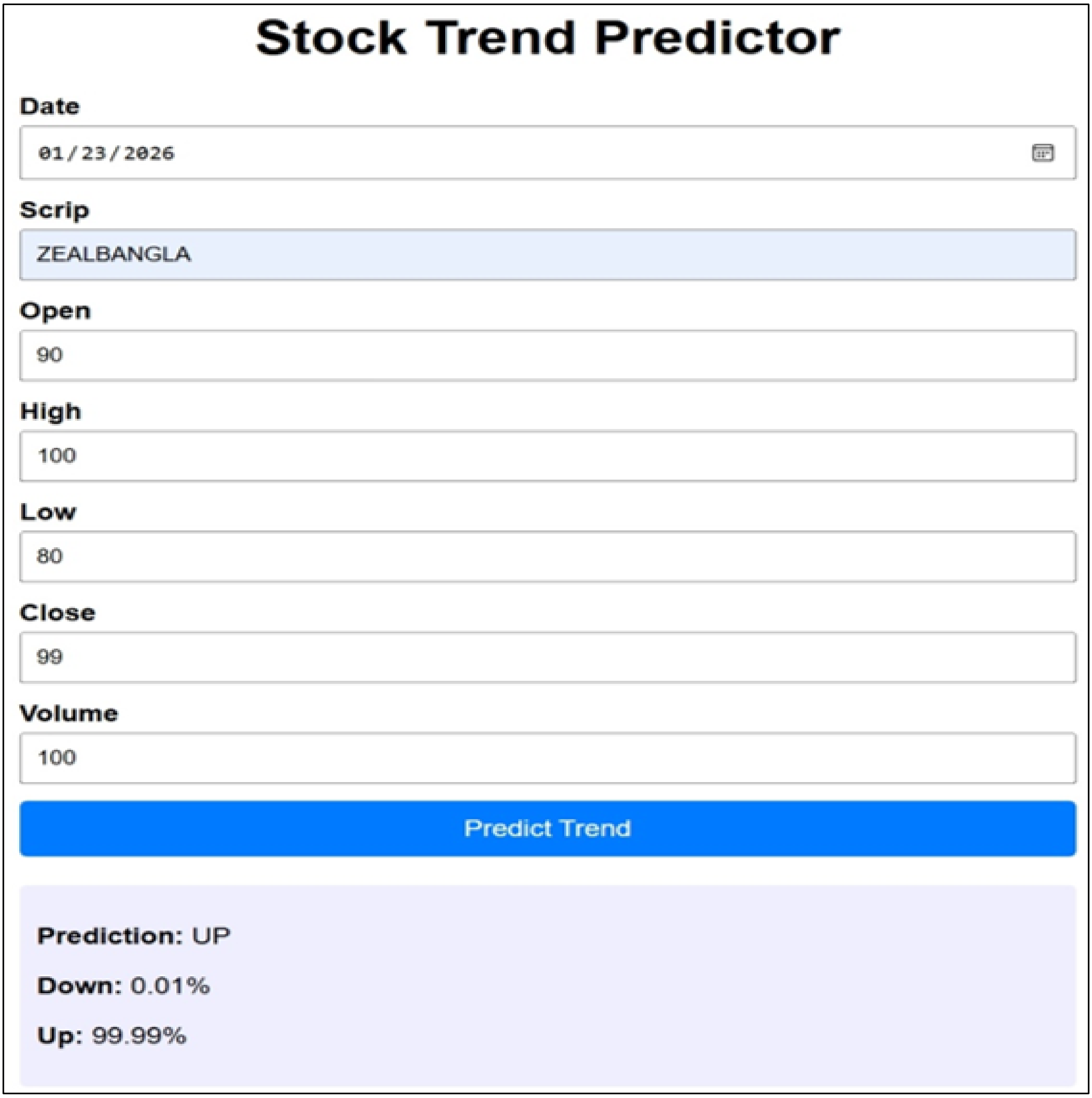


Fig 16 Web Application Interface

## IV. CONCLUSION AND FUTURE WORKS

Our thorough analysis proves that ensemble-based machine learning techniques, especially gradient boosting algorithms, are more accurate for predicting stock market trends in the Dhaka Stock Exchange. Among all tested models, our analysis demonstrates that LightGBM achieves the optimal balance between prediction accuracy (85.69%) and computational speed. Additionally, weighted soft voting ensemble improves prediction by leveraging different strengths of each used algorithm. Feature importance analysis reveals that rolling volatilities and momentum indicators dominate trend prediction.

In addition to experimental validation, this research makes a practical contribution via its deployment as a working web-based application for real-time trend prediction. The deployment of this model confirms its practical viability for decision-making for financial businesses operating in emerging markets.

Some areas for further research include integrating live market data feeds for real-time prediction, building out version control structures for smooth updates, creating batch prediction endpoints for portfolio analysis, and adding alternative data sets, such as news sentiment and macroeconomic data. Additional longitudinal research on the effects on investment decision-making processes would also offer further research avenues for understanding the value proposition for emerging market ML tools.

➢ *Ethics Statement*

This study utilized only historical, anonymized stock market data from publicly available sources. No human participants, animal subjects, or sensitive personal data were involved. Therefore, ethical approval and informed consent were not required for this research.

➢ *Code Availability*

The complete codebase for data analysis and web application implementation is available in the Zenodo repository at: https://doi.org/10.5281/zenodo.18366930. The repository includes all preprocessing scripts, model training pipelines, feature engineering code, and the complete web application source code (backend and frontend).

➢ *Credit Author Statement*

Rakib Hasan: Conceptualization, Methodology, Software, Formal Analysis, Investigation, Writing – original draft; Marjan Akter Badhon: Data curation, Visualization, Writing – review & editing, Visualization; Mahmudul Hassan Maruf: Visualization, Validation, Writing – review & editing; Sourov Ahmed: Investigation, Data curation, Visualization, Writing – review & editing; Sanimul Hossain Sanzit: Visualization, Validation, Software, Writing – review & editing.

➢ *Acknowledgement*

This research was not supported by any specific grant from public, commercial, or non-profit funding agencies.

➢ *Declaration of Competing Interests*

The authors declare that there is no known financial interest or personal relationships that could have influenced work presented in this paper.

## REFERENCES

[1]. Meesad, P., & Rasel, R. I. (2013). Dhaka Stock Exchange trend analysis using support vector regression. In The 9th International Conference on Computing and InformationTechnology (IC2IT2013) 9th-10th May 2013 King Mongkut's University of Technology North Bangkok (pp. 135-143). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-37371-8_17

[2]. Zaman, A., & Rahman, M. (2019). Investment trend analysis of Dhaka stock exchange: A comparative study. World Academy of Science, Engineering and Technology International Journal of Economics and Management Engineering, 13(5). https://doi.org/10.5281/ZENODO.3298928

[3]. Misra, P., & Chaurasia, S. (2020). Data-driven trend forecasting in stock market using machine learning techniques. Journal of Information Technology Research (JITR), 13(1), 130-149. https://doi.org/10.4018/jitr.2020010109

[4]. Verma, J. P., Tanwar, S., Garg, S., Gandhi, I., & Bachani, N. H. (2022). Evaluation of pattern based customized approach for stock market trend prediction with big data and machine learning techniques. In Research Anthology on Machine Learning Techniques, Methods, and Applications (pp. 1255-1270). IGI Global. https://doi.org/10.4018/IJBAN.2019070101

[5]. Gummadi, J. C. S., Thompson, C. R., Boinapalli, N. R., Talla, R. R., & Narsina, D. (2021). Robotics and Algorithmic Trading: A New Era in Stock Market Trend Analysis. Global Disclosure of Economics and Business, 10(2), 129-140. https://doi.org/10.18034/gdeb.v10i2.769

[6]. Ghani, M. U., Awais, M., & Muzammul, M. (2019). Stock market prediction using machine learning (ML) algorithms. Advances in Distributed Computing and Artificial Intelligence Journal, 8(4), 97-116. https://doi.org/10.14201/adcaij20198497116

[7]. Nabipour, M., Nayyeri, P., Jabani, H., & Mosavi, A. (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. Ieee Access, 8, 150199-150212. https://doi.org/10.1109/ACCESS.2020.3015966

[8]. Singh, S., Madan, T. K., Kumar, J., & Singh, A. K. (2019, July). Stock market forecasting using machine learning: Today and tomorrow. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) (Vol. 1, pp. 738-745). IEEE. https://doi.org/10.1109/ICICICT46008.2019.8993160

[9]. Manasa, N., Praveenraj, D. W., & SR, L. (2023, December). Predictive Analytics for Stock Market Trends using Machine Learning. In 2023 4th International Conference on Computation, Automation and Knowledge Management (ICCAKM) (pp. 1-8). IEEE. https://doi.org/10.1109/ICCAKM58659.2023.10449528

[10]. Chen, P. (2024) "Using Machine Learning to Predict the Stock Market Trend", Transactions on Computer Science and Intelligent Systems Research, 5, pp. 981–986. https://doi.org/10.62051/bgrbnh39

[11]. Punjabi, Y., & Faridi, M. A Study of Stock Market Prediction (SMP) Using Machine Learning Techniques. https://doi.org/10.47392/irjaem.2024.0391

[12]. Pahwa, N., Khalfay, N., Soni, V., & Vora, D. (2017). Stock prediction using machine learning a review paper. International Journal of Computer Applications, 163(5), 36-43. https://doi.org/10.37394/232018.2021.9.17

[13]. J. Guntaka, V. J. Karunakar Reddy Gade, R. Yallavula, A. D. Kumar and P. V. Sagar, "Stock Market Prediction using Machine Learning Technique," 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), Trichy, India, 2023, pp. 378-382. https://doi.org/10.1109/ICSMDI57622.2023.00073

[14]. S. R. Likhith, N. Nausheen, A. Abbas, V. S. Sarayu, B. Hemanth and R. K. Varvatti, "Predicting future Stock market Trends using Python and Machine Learning," 2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS), Bangalore, India, 2023, pp. 1-9. https://doi.org/10.1109/ICCAMS60113.2023.10526019

[15]. Cervelló-Royo, R., & Guijarro, F. (2020). Forecasting stock market trend: A comparison of machine learning algorithms. Finance, Markets and Valuation, 6(1), 37-49. https://doi.org/10.46503/nluf8557

[16]. Prasad, A., & Seetharaman, A. (2021). Importance of machine learning in making investment decision in stock market. Vikalpa, 46(4), 209-222. https://doi.org/10.1177/02560909211059992

[17]. Kumar, K. Dogra, C. Utreja and P. Yadav, "A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 1003-1007. https://doi.org/10.1109/ICICCT.2018.8473214

[18]. Rohit B R, Rajeeva Shreedhara Bhat, Abhishek Manohar, Mamatha K R, "Stock Market Prediction using Machine Learning," SSRG International Journal of Communication and Media Science, vol. 7, no. 2, pp. 6-9, 2020. https://doi.org/10.14445/2349641X/IJCMS-V7I2P102

[19]. Sultana, N., Shoha, S., Dolon, M. S. A., Al Shiam, S. A., Zakaria, R. M., Shimanto, A. H., ... & Abir, S. I. (2024). Machine Learning Solutions for Predicting Stock Trends in BRICS amid Global Economic Shifts and Decoding Market Dynamics. Journal of Economics, Finance and Accounting Studies, 6(6), 84-101. https://doi.org/10.32996/jefas.2024.6.6.7

[20]. Jain, S., Saluja, D. N., Pimplapure, D. A., & Sahu, D. R. (2024). Exploring the Future of Stock Market Prediction through Machine Learning: An Extensive Review and Outlook. International Journal of Innovative Science and Modern Engineering, 12(4), 1-10. https://doi.org/10.35940/ijisme.e9837.12040424

[21]. Vignesh, C. K. (2018). Applying machine learning models in stock market prediction. IRJET, Journal. https://doi.org/10.36713/epra4361

[22]. S. Dinesh, N. R. R, S. P. Anusha and S. R, "Prediction of Trends in Stock Market using Moving Averages and Machine Learning," 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2021, pp. 1-5. https://doi.org/10.1109/I2CT51068.2021.9418097

[23]. Rath, S., Das, N. R., & Pattanayak, B. K. (2024). An analytic review on stock market price prediction using machine learning and deep learning techniques. Recent Patents on Engineering, 18(2), 88-104. https://doi.org/10.2174/1872212118666230303154251

[24]. Deshpande, R., Lunkad, D., Kunjir, M., & Ingle, S. (2022). Stock Price Prediction and Analysis using Machine Learning Techniques. Int. J. Res. Appl. Sci. Eng. Technol. https://doi.org/10.22214/ijraset.2022.42747

[25]. Walunjkar, G., Desai, R., Gholap, Y., & Sankalp Shukla, R. (2022). Prediction of Market Trends using Machine Learning. International Journal of Next-Generation Computing, 13(2). https://doi.org/10.47164/ijngc.v13i2.575

[26]. Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert systems with applications, 42(1), 259-268. https://doi.org/10.1016/j.eswa.2014.07.040

[27]. Mehta, P., Pandya, S., & Kotecha, K. (2021). Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. PeerJ Computer Science, 7, e476. https://doi.org/10.7717/peerj-cs.476

[28]. L. Mathanprasad and M. Gunasekaran, "Analysing the Trend of Stock Marketand Evaluate the performance of Market Prediction using Machine Learning Approach," 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2022, pp. 1-9. https://doi.org/10.1109/ACCAI53970.2022.9752616

[29]. Singh, U. P., & Tripathi, M. M. (2023). A Critical Review of the Effectiveness of Machine Learning & Deep Learning Approaches in Forecasting Stock Market Trends. International Journal on Recent and Innovation Trends in Computing and Communication, 11(9), 3797-3801. https://doi.org/10.17762/ijritcc.v11i9.9624

[30]. Singh, A., Gupta, P., & Thakur, N. (2021). An empirical research and comprehensive analysis of stock market prediction using machine learning and deep learning techniques. In IOP conference series: materials science and engineering (Vol. 1022, No. 1, p. 012098). IOP Publishing. https://doi.org/10.1088/1757-899X/1022/1/012098

[31]. Rathore, A., & Khabiya, P. (2023). Predicting the Direction of Stock Markets Employing Back Propagation in Neural Networks. https://doi.org/10.55041/ijsrem24208

[32]. D. Y. N. Le, A. Maag and S. Senthilananthan, "Analysing Stock Market Trend Prediction using Machine & Deep Learning Models: A Comprehensive Review," 2020 5th International Conference on Innovative Technologies in

Intelligent Systems and Industrial Applications (CITISIA), Sydney, Australia, 2020, pp. 1-10. https://doi.org/10.1109/CITISIA50690.2020.9371852
[33]. V. Deswal, D. Kumar and Suman, "Stock Market Price Prediction using Machine Learning Techniques: A Review," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 693-698. https://doi.org/10.1109/CISES58720.2023.10183507
[34]. P. A. Gunturu, R. Joseph, E. S. Revant and S. Khapre, "Survey of Stock Market Price Prediction Trends using Machine Learning Techniques," 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), Bangalore, India, 2023, pp. 1-5. https://doi.org/10.1109/ICAIA57370.2023.10169745
[35]. K. Pahwa and N. Agarwal, "Stock Market Analysis using Supervised Machine Learning," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 197-200. https://doi.org/10.1109/COMITCon.2019.8862225
[36]. U. Shah, B. Karani, J. Shah and M. Dhande, "Stock Market Prediction Using Sentimental Analysis and Machine Learning," 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2021, pp. 1-4. https://doi.org/10.1109/GCAT52182.2021.9587898
[37]. Khattak, A. M., Ullah, H., Khalid, H. A., Habib, A., Asghar, M. Z., & Kundi, F. M. (2019, December). Stock market trend prediction using supervised learning. In Proceedings of the 10th International Symposium on Information and Communication Technology (pp. 85-91). https://doi.org/10.1145/3368926.3369680
[38]. Bansal, M., Goyal, A., & Choudhary, A. (2022). Stock market prediction with high accuracy using machine learning techniques. Procedia Computer Science, 215, 247-265. https://doi.org/10.1016/j.procs.2022.12.028
[39]. Khattak, A., Khan, A., Ullah, H., Asghar, M. U., Arif, A., Kundi, F. M., & Asghar, M. Z. (2022). An efficient supervised machine learning technique for forecasting stock market trends. Information and Knowledge in Internet of Things, 143-162. https://doi.org/10.1007/978-3-030-75123-4_7