Predicting Daily Stock Price Movements Using Data Mining Techniques: A Comparative Analysis of Logistic Regression, Decision Tree, Random Forest, and XGBoost on Yahoo Finance Time-Series Data

Soobia Saeed *

Posted Date: 26 November 2025

doi: 10.20944/preprints202511.2007.v1

Keywords: logistic regression; decision tree; random forest; XGBoost

 doi:10.20944/preprints202511.2007.v1

Article

Predicting Daily Stock Price Movements Using Data Mining Techniques: A Comparative Analysis of Logistic Regression, Decision Tree, Random Forest, and XGBoost on Yahoo Finance Time-Series Data

Soobia Saeed

Taylor's University. Malaysia; soobiasaeed1@gmail.com

Abstract

The study assesses the capability of various supervised machine learning approaches to predict the short-term movements in the stock market through historical financial time-series data. The Yahoo Finance dataset comprising 2018-2023, containing more than 1,200 daily trades, is the foundation for the research work, which seeks to determine if the closing price of the stock for the next day will be either higher or lower. To secure the quality of the data and to avoid temporal leakage, a thorough pre-processing procedure—missing value check, outlier smoothing, feature extraction with technical indicators like moving averages, normalization, and chronological splitting—was carried out. Four data mining models—Logistic Regression, Decision Tree, Random Forest, and XGBoost—were built, and their performance assessed in terms of accuracy, precision, recall, and F1-score, with a time-aware validation method through Time Series Split supporting this. Logistic Regression results indicated the highest recall (1.0) and F1-score (0.67) where it identified all price movements up, while the Random Forest and XGBoost have better precision (0.5248) and overall accuracy (0.5163) which means that a more balanced trade-off between false positives and false negatives has been indicated. The Decision Tree model was easy to interpret but was nonetheless the least effective in a highly fluctuating financial market setting because it was not able to generalize as much. In conclusion, the findings have shown the difficulties of predicting stock markets that are inherently volatile; however, it is still possible through the use of well-designed technical features and supervised learning to uncover patterns that have economic significance. The study finally recommends that the model should be retrained, the market regime should be adapted, multi-stock trading should be expanded, and testing frameworks that integrate back testing should be set up for the real-world applicability.

Keywords: logistic regression; decision tree; random forest; XGBoost

1. Background

Economy data Financial markets have increasingly relied on data in recent years. Subsequently, a great number of historical trading data has been accumulated, and thus investors and analysts have already taken the path of applying data mining and machine learning for the purpose of being able to discover the patterns, detect the trends, and identify the predictive signals in the stock markets. While the Efficient Market Hypothesis maintains that stocks are priced accurately at all times and incorporate all the available information, nevertheless, research and various applications of historical data and technical indicators have proven that sometimes short-term market trends can be predicted with very high accuracy [1].

The emergence of platforms such as Yahoo Finance or Kaggle in the last few years has greatly increased the access to high-quality financial data for students, researchers, and data experts. The datasets provided usually include opening and closing prices, trading volume, and the intraday high and low for each stock, which can be used to derive useful features for analysis. Through suitable

 doi:10.20944/preprints202511.2007.v1

data preprocessing and modeling, these features can assist us in making educated guesses about the future movement of stocks.

The data used for this research came from the "Yahoo Finance Dataset (2018-2023)" which is also known as the Kaggle dataset that contains over 1,200 daily trading records. This dataset has seven columns which include the date, opening, highest, lowest, closing, adjusted closing prices, and volume of trades. Each entry corresponds to a trading day and gives a comprehensive view of the stock on that particular day. They are provided in such a way that can be easily utilized for feature generation, the training of models, and their evaluation. This research practically corresponds to the Data Mining module, It promotes collaboration, and the self-taught learning of modeling techniques, and it holds the expectation for students to be ethical and responsible when handling real data. Besides, it is a wonderful chance to apply various fascinating data mining algorithms in a financial domain which is a hot and quickly growing area[2].

1.1. Research/Business Goal

The primary aim of this research is to create and evaluate models that can predict the stock market's closing price of a specific company for the current day as either going up or down. By doing so, we will treat this as a binary classification problem since the stock price's motion can be either upward or downward based on the dependent variable's prediction. There are various reasons for this objective to be significant. In the financial market, even a slight precision in predicting price direction can serve well to the investors for better trading actions. For example, purchasing the stock by the prediction of upward movement and selling or holding by predicting decreased price for the support of the sell side. These models do not promise profits, instead, they are support tools that can reduce the risk of uncertainty and enhance the planning of tactics. By pre-processing the data set and engineering the features, it will be possible to achieve this. Newly created features will include moving averages, percent returns, volatility indicators, and momentum metrics. Simultaneously, various machine learning models such as Logistic Regression, Decision Tree, Random Forest, and XGBoost will be trained and tested. Their performance will be evaluated based on commonly used classification metrics such as accuracy, precision, recall, and F1-score. If a model is able to outperform a simple baseline (like always predicting the most common class) and at the same time illustrate interpretable features that are related to the dynamics of stock prices, then it will be termed as successful. To gain a deeper understanding of the model's decisions, we will employ feature importance scores and visualizations. Ethical considerations will also be discussed, including generalization vs. overfitting, handling imbalanced data, and making it clear that the model is not intended for live trading. To sum up, this research work can be seen as a case where data mining has been applied on the real financial data to create models of prediction that are ready for deployment, interpretable, and responsible. It unites theoretical and practical skills and shows how much data-driven decisions can count in the finance sector[3–5].

1.2. Introduction

Over the years, financial markets have transformed themselves into inquisitive data-driven environments wherein even the slightest traders' activities are monitored and recorded for centuries, thus allowing analysts, researchers, and investors to recognize significant trends in stock price movements. The Efficient Market Hypothesis, which claims that market prices reflect all the information available at any time and, as a result, cannot be predicted consistently, nevertheless, has empirical studies and practical applications that confirm the existence of short-term market trends that sometimes can be predicted. The advance of open-source tools like Yahoo Finance and Kaggle has brought the high-quality financial databases to the broad public, and thus, the students, researchers, and practitioners of the financial market have started to engage in the stock price prediction through data mining and machine learning methods proactively [6–8].

 doi:10.20944/preprints202511.2007.v1

The data sets used in the stock market usually comprise the most critical attributes like opening prices, closing prices, daily maximums and minimums, adjusted closing prices, and volumes of transactions. If these variables are treated right, they can be turned into very powerful explanatory features that can be used for predictive modeling [9,10]. Through the application of preprocessing, feature extraction, and supervised learning techniques, it is possible to bring forth predictions regarding price changes that are well-informed thus granting the investors a decision-support tool that is very precious. The models in question do not promise a certain profit or do away with the risk involved but they can still be of great assistance in directing investors' decisions through probabilistic insights into the behavior of the market since they can provide the investor with such insights [11]

In this research, the Yahoo Finance Dataset (2018–2023) from Kaggle is used, which contains daily trading records of over 1,200 and seven main financial characteristics. The research work is in line with the Data Mining module that emphasizes collaborative learning, methodological rigor, and ethical awareness in the handling of the real-world data. It also demonstrates through the systematic preprocessing of data, model development, and comparative evaluation that machine-learning techniques can be applied in a financial context to discover predictive patterns and enable decision-making based on the overall understanding [12]. Thus, this research work is a practical demonstration of the data mining techniques in modern finance, reflecting how the theoretical concepts can be turned into actual analytical tools for comprehending the market dynamics.

2. Data Set Description

The dataset that is being referred to in the present research paper is the “Yahoo Finance Dataset (2018-2023)” which had been taken from Kaggle. This means that it is a rich and complete source of five-year daily stock price data for S&P 500 companies which can be used for financial trend analysis and further predictive modeling. The dataset is in Excel format (yahoo_data.xlsx) and thus easy to access for preprocessing and analyzing in a Python-based environment. Containing 1,257 rows of data where each row corresponds to a single day of trading, it also has seven financial attributes as key columns. In the case of the Date column, all other variables are numerical, which ensures that they can be easily integrated into machine learning workflows. The features are trading date (later being changed into datetime format) opening price, highest and lowest prices of the day, closing price, adjusted closing price that considers stock splits and dividends, and total trading volume. The daily record provides a complete picture of how the stock behaved in the market on that day, which is a good enough granularity for the generation of technical indicators like moving averages, Bollinger Bands, and the Relative Strength Index (RSI).

Due to the nature of the dataset being highly structured and mostly consisting of numerical data, it is open for a wide range of machine learning techniques such as classical time-series forecasting models like ARIMA or even deep learning architectures like Long Short-Term Memory (LSTM) networks. Its consistency highly reduces the need for extensive data cleaning and allows more focus to be put on feature engineering and model development. In general, the Yahoo Finance dataset has been a great pillar for stock price behavior analysis, development of predictive models, and also for performance evaluation of different data mining techniques using real-world financial data.

3. Data-Related Issues & Preprocessing

Prior to implementing any predictive model or even conducting exploratory data analysis (EDA) we must necessarily go through a step of comprehensive data processing so that we can rely on analysis that is consistent and easy to apply to our data. The Yahoo Finance dataset at first glance might look to be well-structured and numerically consistent but in practice, financial time series often consist of hidden anomalies, subtle irregularities and structural inconsistencies that are not directly observable. The problems of missing values, sudden spikes in trade volume, outliers in closing prices or feature rescaling whenever they occur can greatly limit the capability and power of machine learning algorithms to generalize if they are not properly treated.

 doi:10.20944/preprints202511.2007.v1

In the case of financial time series forecasting, the ensuing predictions are dependent on the timeline's continuity and the errors made at each step can multiply and have a forecasting cascade phenomenon that goes along with blaming. Therefore, a properly built preprocessing pipeline is imperative before any deep learning architectures (LSTMs) or statistical models take in data.

The detailed methodology of the data set preprocessing is thoroughly explained in this section. This entails missing values detection and substitution, outliers' detection and clipping, and feature engineering of the most relevant application-specific features (e.g., moving averages). Normalization is also applied in the preprocessing step, mainly for the purpose of preparing the data for modeling with sequences and scaling the numeric columns. All this endeavor is backed up with theoretical justifications drawn from best data science practices and comes with actual python code that you can implement in your reserach works, which is written in the widely used libraries of pandas, numpy, scikit-learn, and also scipy.

Eventually, turning raw historical stock data into a structured and model-ready format that accurately represents the market dynamics and effectively reduces the risks of overfitting, data mining bias, and leakage is the aim of the preprocessing step. The quality of preprocessing is a direct determinant of the robustness of any subsequent models thereby supporting the entire forecasting system's reliability.

3.1. Missing Values and Data Consistency

The data set, after looking great on visual examination, was still checked programmatically and verified using dfitraining.isnull().sum() to find out if there were any missing values in the columns. No explicit nulls were found, but as a precaution, a forward fill was applied. This method is particularly beneficial for time series data as it keeps the order of the data intact and does not introduce any statistical bias. The forward filling ensures that sudden jumps in the input features (for example, due to market holidays or input data formatting discrepancies) do not result in noise being added to the sequence of the input features fed to the model during training.

3.2. Outlier Detection and Smoothing

Data contamination in the stock market is due to factors like earnings releases, economic shocks, and unusual trades results in the emergence of outliers. Even though the actual markets might experience such turning points, they can destabilize the model. Thus, we performed Z-score analysis on the Volume feature to detect the extreme outlier, and the 95th percentile was used to set the limit. This nudging method keeps the data's diversity intact and reduces the impact of the one-time spikes that could otherwise control the loss functions during model training.

3.3. Feature Engineering – Moving Averages

A novel signal MA20 (20-day moving average), is brought in to more accurately depict the inflation trend. In technical analysis, Simple Moving Averages are employed to reduce the impact of price fluctuations and to determine the direction of the trend. The incorporation of historical dependencies will thus boost the predictive power of models. Using technical indicators like MA20 not only results in better model performance but also enables more openness in decision-making concerning the interpretation of market dynamics.

3.4. Normalization and Scaling

One of the main factors why scaling of numeric features was done through Min-Max Scaling was that practically all the machine learning algorithms (especially, neural networks and SVM) are influenced by input data's scale. This normalization is to [0, 1] range and it will likely condition learning and speed up convergence in deep networks.

 doi:10.20944/preprints202511.2007.v1

3.5. Chronological Train/Test Split

In standard classification scenarios records are completely interchangeable, so that we can randomly split the instances except for time series data which the order must be maintained to avoid data leakage. Thus, there was an 80/20 split based on time order. This resembles a real-life scenario where models trained on historical price data are applied for future prices.

3.6. Preparing Sequential Inputs for LSTM

Input data for deep learning models like the LSTM needs to be in the form of 3D tensors (samples, timesteps, features). A sliding window (with a sequence length of 30days) was implemented to forecast the Close price for the next day. This transformation is crucial to capture temporal dependencies which a simple regression analysis may not model.

4. Methodology

The research utilized a systematic approach purposed to prepare financial time-series data, engineer insightful features and assessed a variety of machine learning algorithms in their ability to predict daily stock price direction. Five main stages made up the process: data preprocessing, feature engineering, dataset splitting, model training, and time-aware validation.

The first step in the workflow was the preprocessing phase which was extensive, and this ensured that the Yahoo Finance Dataset (2018-2023) was reliable and consistent. Even though df.isnull().sum() did not show any missing values, still a forward-fill technique was applied just to be on the safe side and to keep the temporal continuity. Then, outlier detection was done with the help of z-score analysis, and this analysis was particularly focused on the Volume attribute where very extreme spikes were capped at the 95th percentile to ensure that the model training would not be affected by distortion. After that, Min-Max scaling was used to normalize the data so that all numerical features would be in the range [0, 1]. This was done so that attributes with larger magnitudes like Volume would not overpower the learning process.

Feature engineering was utilized to increase the dataset's value by subtracting the necessary financial indicators. A 20-day moving average (MA20) was calculated to act as a trend-smoothing feature which is usually present in technical analysis. Other transformations like percent changes and volatility-based indicators were debated to include temporal dependencies and create short-term market momentum. The input for LSTM models was made three-dimensional by using a sliding window of 30 days, thus allowing the model to learn from the sequential patterns that were previously not captured by other algorithms.

To prevent data leaks, the split was made in a way that 80% of the data was used for training and 20% for testing in a chronological way, and therefore only the past data was used for predicting future values. The target variable was formed by applying the function .shift(-1) to classify the next day's closing price as either an increase (1) or a decrease (0). At no point was shuffling done so the time-series structure was maintained.

The preprocessed dataset was used to train four supervised learning models—Logistic Regression, Decision Tree, Random Forest, and XGBoost. These models were chosen in such a way that they would share the same level of interpretability, computational efficiency, and at the same time, have the same predictive power. The performance of each model was assessed via standard classification metrics including accuracy, precision, recall, and F1-score. A time-aware cross-validation strategy was applied to evaluate generalization using TimeSeriesSplit, which retains the temporal order of events and avoids the problem of lookahead bias. Model performance was averaged for each fold to capture stability across different time windows. Also, checks for overfitting and underfitting were carried out by comparing the metrics of training and testing.

At last, hyperparameter tuning for XGBoost was performed using GridSearchCV with Time Series Split as the model's learning rate, tree depth, and regularization parameters were further refined without compromising temporal consistency. The whole methodology was supported by the

 doi:10.20944/preprints202511.2007.v1

fact that all preprocessing, feature engineering, and validation techniques followed the best practices in financial machine learning, thus ensuring that the models were realistic, robust, and safe from future information leakage.

4.1. Data Mining Techniques & Justification

4.1.1. Logistic Regression (Baseline Model)

Logistic Regression was chosen as the basic model for the research because of its straightforwardness, ease of understanding, and high effectiveness. It works through a linear combination of features and a sigmoid activation function estimating the likelihood of the next day stock price going up or down. Logistic Regression is helpful for recognizing the linear connections between financial indicators and price movements. The model has a fast training time, gives probability outputs that can be tuned for trading signals, and gives a crucial basis for the comparison of more sophisticated models' performances. Its benchmark position guarantees that the following models make substantial progress over just simple linear separability.

4.1.2. Decision Tree Classifier

The Decision Tree Classifier got picked because it can detect non-linear relationships that exist in financial datasets. The model can then pull off complex patterns where stock price changes are one of the influencing factors by classifying the data through hierarchical branches with different characteristic thresholds. Moreover, Decision Trees grant high interpretability, thus enabling users to see the path of decisions being made and also know how much each feature contributes to the overall decision. This level of clarity is especially necessary in financial modelling where sometimes the stakeholders want too much to be told about how the model works. In addition to that, Decision Trees can work on both types of variables—continuous and categorical—making them very adaptable for structured financial data.

4.1.3. Random Forest Classifier

Another reason for including random forest in the list was its strength and ability to generalize pretty well. It is an ensemble learning method that is very powerful and robust. By building several decision trees based on random data subsets and features, the model is less likely to overfit—the major issue one faces while working with financial time-series data. Moreover, Random Forests combine forecasts for every tree through voting that takes place over the majority, thus improving stability and accuracy as compared to a single tree. They're also great at providing the so-called reliable feature importance scores, which are based on Gini impurity or information gain, thus allowing us to see which technical indicators are powerful in affecting the stock price movements. This model works excellently even in the noisy environments of the financial markets where it can capture the "noise" by applying the ensemble averaging technique.

4.1.4. XGBoost Classifier

XGBoost (Extreme Gradient Boosting) was the advanced ensemble method chosen in this research primarily because of its excellence in the case of structured datasets. This method constructs trees one after another, wherein the new tree rectifies the mistakes made by the old ones, thus, highly optimizing the prediction accuracy. Moreover, XGBoost applies regularization techniques that limit overfitting, efficiently takes care of missing data points, and deals with the problem of multicollinearity among the financial indicators. The model's versatility and the wide range of tuning options for hyperparameters contribute to its classification as a top-notch model for highly competitive predictive tasks. Due to its documented success in data science competitions and practical applications, XGBoost can be considered a viable option for capturing non-linear and dense financial patterns.

 doi:10.20944/preprints202511.2007.v1

4.1.5. Model Comparison and Evaluation Criteria

Four important measures were used in this study to assess the performance of the model: Accuracy, Precision, Recall, and F1-Score. A general sense of prediction reliability is provided by accuracy, which gauges the model's overall correctness. While recall is the number of real positive cases that were correctly detected, precision measures the percentage of anticipated positive cases that were actually correct. When dealing with imbalanced classes where one kind of outcome may predominate, the F1-Score—which is the harmonic means of precision, and recall—is very helpful.

To balance interpretability, scalability, and prediction performance, a mix of models was chosen. A clear grasp of feature influence was made possible by the quick and comprehensible baseline provided by logistic regression. Because decision trees provide an understandable and visual depiction of feature splits, they were incorporated. In order to balance resilience and model complexity, Random Forest was used to produce forecasts that were more reliable in the face of noisy financial data. Lastly, XGBoost was used as a potent ensemble model that followed ethical modeling guidelines and could capture intricate patterns. The final suggestions are more credible and in line with the course's learning objectives when these models are compared and their capacity to generalize to new data is validated.

4.1.6. Evaluation Results of Data Mining Techniques

We trained and tested four classifiers: Logistic Regression, Decision Tree, Random Forest, and XGBoost, to assess their capabilities in predicting whether the stock price would rise on the following day. The target variable was binary (up or down), and the test set was comprised of 246 samples with a near-equal distribution of the two classes, with 121 for class 0 and 125 for class 1. This allowed for a fair comparison of the models with respect to the class distributions. The following evaluation makes a definite and clear statement. From the comparison, we see that Logistic Regression has a perfect recall of 1.0, identifying all positive price movements, but moderate precision, and fair detecting of the negative movements, but it had the highest F1-score of 0.67. The Random Forest and XGBoost have the highest precision (0.5248), and the highest accuracy (0.5163), have better distribution of incoming predictions; however, they failed to identify the true positive labels compared to Logistic Regression. The Decision Tree model managed to attain a modest precision of 0.5167 but produced the lowest recall (0.25) and had the resulting weakest F1-score of 0.3370, suggesting a lack of sensitivity to positive price movements. All the models had relatively low accuracy in predicting the future outcomes of financial markets, which demonstrates the unpredictable nature of financial markets, but the F1-scores yielded more useful results. The Logistic Regression model proved capable of capturing all of the upward movements; the Random Forest and XGBoost were better able to balance the trade-off of precision and recall, which is particularly valuable in a financial forecasting context where missed opportunities to long (or short) and false signals have expensive risks.

5. Model Validation

The model validation procedure and performance assessment using the cleaned Yahoo Finance time-series dataset are presented in this section. Predicting whether the stock will move upward or downward in the future was the main goal. Finding the most dependable model required a thorough validation technique because financial markets are nonlinear and extremely volatile. By maintaining the chronological order of observations, the validation method sought to guarantee that models avoided overfitting or underfitting, extended well to new data, and were free from lookahead bias. TimeSeriesSplit and other time-aware validation approaches were used to get accurate and trustworthy performance estimations.

5.1. Cross-Validation Strategy with TimeSeriesSplit

 doi:10.20944/preprints202511.2007.v1

Because future information may unintentionally affect model training, traditional cross-validation techniques that shuffle data might induce lookahead bias in time-series forecasting. The TimeSeriesSplit method was employed to avoid this. By gradually extending the training window and assessing performance on forward-looking test windows, this technique preserves the temporal sequence of the dataset. TimeSeriesSplit enables a realistic evaluation of how well each model might generalize to novel, untested market situations by honoring the chronological order.

5.2. Evaluating Models Using TimeSeriesSplit

Accuracy, precision, recall, and F1-score were the main metrics used to assess each model's performance throughout five time-based folds. These measures revealed information about stability over time as well as predictive power. We were able to evaluate robustness and find models that held steady even as market conditions changed by looking at differences across the folds.

5.3. Model Training and Comparison

The TimeSeriesSplit configuration was used to train and assess models for Random Forest, Decision Tree, Logistic Regression, and XGBoost. A thorough comparison was made possible by the cross-validation results, which offered variance and mean performance scores across the folds. This process not only identified the top-performing models but also demonstrated the consistency of their forecasts throughout the dataset's various temporal segments.

5.4. Overfitting vs. Underfitting Check

Training and testing accuracies were examined to make sure models did not perform too poorly or memorize historical data. Overfitting, where the model performed well on training data but badly on unseen data, was indicated by the wide difference between the two. On the other hand, consistently low scores on both sets indicated underfitting, indicating that the model was unable to extract meaningful patterns from the data. Model selection and improvement were informed by this analysis.

5.5. Preventing Data Leakage and Lookahead Bias

To prevent future information from influencing model training, strict precautions were made to prevent data leaking. Every feature engineering process, including volatility calculations and moving averages, was only calculated using historical data. In order to represent future price movement, the target variable was created using.shift(-1). Furthermore, no data shuffling was done at any point, maintaining the observations' organic temporal flow.
Important Steps Taken:

- Rolling() was only used for previous observations.
- The next-day target variable is created using shift (-1).
- All dataset splits have shuffle=False.
- Future value dependency was avoided in all engineered features.

5.6. Optional Hyperparameter Tuning

GridSearchCV and Timeseries Split were used to undertake optional hyperparameter adjustment for the XGBoost classifier in order to further improve model performance. This method preserved the integrity of the time series while enabling systematic optimization. The model's generalization and predictive accuracy were enhanced by adjusting variables like learning rate, tree depth, and regularization.

5.7. Limitations of the Validation Strategy

 doi:10.20944/preprints202511.2007.v1

Timeseries Split offers a robust validation framework for financial time-series modeling, although there are still a number of drawbacks:

- No online simulation: Unlike in actual trading settings, models are not updated on a regular basis.
- Stationarity assumption: Because market circumstances change quickly, historical data might not accurately predict future distributions.
- Metric sensitivity: For datasets that are unbalanced, metrics like accuracy might not be as meaningful.
- Feature instability: During significant market occurrences, rolling or volatility-based features may exhibit erratic behavior.

Notwithstanding these drawbacks, this study's validation strategy is reliable and appropriate for supervised learning in financial forecasting. All things considered, Random Forest and XGBoost performed better, especially in terms of precision and recall, making them the best models for forecasting the direction of stock prices in this dataset.

6. Conclusions and Recommendation

6.1. Conclusions

Investigating the predictive ability of data mining algorithms in financial forecasting—more especially, predicting whether a stock price would rise or fall on the next trading day—was the main goal of this research work. To improve model performance, pertinent technical indicators, like moving averages and volatility measures, were dynamically produced using historical stock data that was obtained from Yahoo Finance via Kaggle. We developed and assessed four supervised learning models: Random Forest, Decision Tree, Logistic Regression, and XGBoost.

The challenge was presented as a binary classification exercise, with 1 denoting an increase in price and 0 denoting a fall for the following trading day. To prevent lookahead bias, models were trained and evaluated using a time-aware chronological split. Accuracy, Precision, Recall, and F1-score measures were used to evaluate the models.

All positive or price-increasing situations were effectively identified by Logistic Regression, which had the highest recall (1.0) and F1-score (0.67) of all the models. High recall is especially important in financial contexts because it might be more expensive to miss advantageous chances than to act on misleading signals. However, Logistic Regression's poor precision suggests that it had trouble identifying price declines.

Random Forest and XGBoost had more balanced performance between precision and recall, with the best accuracy (~0.5163) and precision of 0.5248, even though they did not get the highest F1-scores. For tactics where both long and short signals are equally essential, these models are therefore appropriate.

The constraints of shallow decision boundaries in noisy and unpredictable financial markets were highlighted by the Decision Tree model's good interpretability but poor generalization. Overall, the study showed that technical indicators combined with supervised learning can generate important trading signals, even though none of the models reached high accuracy—likely due to market randomness, noise, and limited features. Furthermore, it emphasizes how crucial it is to use evaluation metrics other than accuracy, especially in finance, where asymmetric risk is substantial.

6.2. Improvements and Future Work

1. Market Regime Adaptation: Bull and bear markets are examples of how market habits change over time. To capture evolving trends, models should be retrained on new data on a regular basis. To account for structural changes in market behavior, methods such as clustering models or regime-switching segmentation might be used.

2. Back testing and Deployment: Future research should focus on practical application, turning the

 doi:10.20944/preprints202511.2007.v1

models into useful trading instruments. This includes creating real-time data pipelines from APIs like Yahoo Finance, integrating risk-adjusted performance metrics like the Sharpe ratio or maximum drawdown, and back testing logic to mimic trades based on predictions.

3. Ethical Considerations: Investment choices can be greatly influenced by financial forecasting models. Models must be impartial, make it obvious that they are advisory tools rather than profit-guaranteed methods, and guard against abuse in high-risk trading situations without careful accuracy testing.

4. Extending the Scope of the Dataset: The current study limited generalization by concentrating on a particular stock. To evaluate model resilience across asset classes and different market regimes, future research should include several equities from a variety of industries and markets, such as technology, energy, and developing economies.

References

1. Amoanu, S. (2025). Comparative Forecasting of Financial Time Series Using ARIMA, GARCH, Random Forest, and XGBoost Models.
2. Bendale, M. (2024). *Comparative study among ARIMA, SARIMA & XGBoost for prediction of NIFTY IT index* (Doctoral dissertation, Dublin Business School).
3. Gifty, A., & Li, Y. (2024). A comparative analysis of LSTM, ARIMA, XGBoost algorithms in predicting stock price direction. *Engineering and Technology Journal*, *9*(8), 4978-4986.
4. Esmailizade, S., Ebrahimi, A., Soltani, H., Sam, A., & Rahimi, M. (2024). Machine Learning Approaches for Retail Forecasting: A Study on XGBoost and Time-Series Models. *Authorea Preprints*.
5. Hossain, S., & Kaur, G. (2024, May). Stock market prediction: XGBoost and LSTM comparative analysis. In *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)* (pp. 1-6). IEEE.
6. Nhat, N. M. (2024). Applied Random Forest Algorithm for News and Article Features on The Stock Price Movement: An Empirical Study of The Banking Sector in Vietnam. *Journal of Applied Data Sciences*, *5*(3), 1311-1324.
7. Amruth, S. J., Nigelesh, T. M., Shruthik, V. S., Reddy, V. S., & Venugopalan, M. (2024, June). Time-Series-Based Stock Market Analysis using Machine Learning. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-7). IEEE.
8. Ampomah, E. K., Qin, Z., & Nyame, G. (2020). Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. *Information*, *11*(6), 332.
9. Goldani, M. (2023). Comparative analysis on forecasting methods and how to choose a suitable one: case study in financial time series. *Journal of Mathematics and Modeling in Finance*, *3*(2), 37-61.
10. Sayın, M. G. (2021). *Performance Of Hybrid Machine Learning Algorithms on Financial Time Series Data* (master's thesis, Middle East Technical University (Turkey)).
11. Elena, P. (2021). Predicting the movement direction of omxs30 stock index using xgboost and sentiment analysis.
12. Uhunmwangho, O. P. (2024). Comparing XGBoost and LSTM Models for Prediction of Microsoft Corp's Stock Price Direction. *Mountain Top University Journal of Applied Science and Technology (MUJAST)*, *4*(2), 64-88.