BAREKENG: Journal of Mathematics and Its Applications
June 2025 Volume 19 Issue 2 Page 1393-1408
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol19iss2pp1393-1408

PERFORMANCE ANALYSIS OF GRADIENT BOOSTING MODELS VARIANTS IN PREDICTING THE DIRECTION OF STOCK CLOSING PRICES ON THE INDONESIA STOCK EXCHANGE

Delvian Christoper Kho [1*], Hindriyanto Dwi Purnomo[2], Hendry[3]

[1,2,3]*Information System Study Program, Fakulty of Information Technology, Universitas Kristen Satya Wacana Jln. Diponegoro, Salatiga, 50711, indonesia*

*Corresponding author's e-mail: * **972022018@student.uksw.edu***

Article History:

Received: 18th November 2024
Revised: 27th January 2025
Accepted: 28th February 2025
Published: 1st April 2025

Keywords:

Catboost;
Gradient Boosting;
Performance Analysis;
Random Forest;
Stock Prediction;
Xgboost.

ABSTRACT

Accurately predicting stock market trends remains a significant challenge for investors due to its dynamic nature. This study explores the performance of Gradient Boosting models, including XGBoost, XGBoost Random Forest, CatBoost, and Gradient Boosting Scikit-Learn, in predicting stock market trends such as sideways movement, uptrends, downtrends, and volatility. Using four datasets from the Indonesia Stock Exchange, the research integrates technical, fundamental, and sentiment data, encompassing 37 features. Modeling and testing are conducted using Orange tools and Python, with performance evaluated through metrics such as Mean Absolute Percentage Error (MAPE), R-squared (R^2), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). Results indicate that XGBoost and XGBoost Random Forest consistently outperform other models in predicting stock price movements. These findings highlight the potential of Gradient Boosting models in providing accurate and reliable predictions, offering valuable insights for investors, financial analysts, and researchers to enhance investment strategies and adapt to market fluctuations effectively.

How to cite this article:

D. C. Kho, H. D. Purnomo and Hendry., "PERFORMANCE ANALYSIS OF GRADIENT BOOSTING MODELS VARIANTS IN PREDICTING THE DIRECTION OF STOCK CLOSING PRICES ON THE INDONESIA STOCK EXCHANGE," *BAREKENG: J. Math. & App.*, vol. 19, iss. 2, pp. 1393-1408, June, 2025.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*
Research Article · Open Access

1. INTRODUCTION

When facing the complexities of the stock market, investors often encounter challenges in making accurate predictions [1]. One primary reason is that price trends are not always linear [2][3]. In certain conditions, the market may move laterally (sideways), making predictions more difficult as stock prices fluctuate within a specific range without a clear direction [4]. Additionally, in both uptrend and downtrend conditions, a deep understanding is required to make optimal investment decisions, as in an uptrend, buying too late or selling too early can be detrimental. In contrast, in a downtrend, decisions to sell or cancel investments must be made judiciously [5][6]. High levels of volatility or instability also affect prediction accuracy and investment decisions, as high volatility generates uncertainty regarding stock price movements [7][8]. These complex conditions highlight the need to understand and overcome the challenges of predicting stock closing prices in various market situations. Therefore, improving the accuracy of stock price predictions through advanced tools and models is essential, especially in volatile market conditions [9].

This research aims to address a significant gap in stock price prediction by analyzing the performance of various Gradient Boosting models, including XGBoost, XGBoost Random Forest, CatBoost, and Scikit-Learn Gradient Boosting, on various stock data characteristics such as sideways, uptrend, downtrend, and volatility. While Gradient Boosting models have shown potential in stock market predictions, the novelty of this study lies in exploring different variants of these models and their ability to adapt to the dynamic and diverse characteristics of stock market data. Specifically, this study evaluates how these models identify and predict stock trends under various market conditions, providing new insights into the effectiveness of Gradient Boosting models in this domain. Gradient Boosting is a model in which decision trees are built sequentially, with each tree attempting to correct the prediction errors of the previous one. This model can adjust to emerging patterns adaptively, making it potentially effective in handling the dynamic nature of the stock market [10][11]. The effectiveness of Gradient boosting has also been proven in other domains, such as education, demonstrating high accuracy and robust performance in handling classification problems with diverse datasets [12].

Technical factors alone are insufficient in predicting the direction of stock closing prices [13]. A combination of technical, fundamental, and sentiment analysis can provide a more comprehensive understanding of market conditions. Technical analysis aids in understanding stock price patterns and trends [14], fundamental analysis provides insights into the overall performance of a company [15], and sentiment analysis helps gauge investor attitudes and perceptions toward the stock [16]. This research aims to analyze the capabilities of various Gradient Boosting models in handling specific stock data characteristics such as sideways, uptrend, downtrend, and volatile, with the specific goal of assessing model accuracy in forecasting stock closing price directions, evaluating model success in recognizing and predicting price direction and examining the models' responses to diverse stock data characteristics. This study contributes to understanding the effectiveness of Gradient Boosting model variants in addressing the unique challenges associated with stock data characteristics. The research findings can provide insights for investors, financial analysts, and researchers in optimizing predictive models by considering a combination of technical, fundamental, and sentiment factors to make more intelligent and more responsive investment decisions in response to changing market conditions.

Previous studies have successfully implemented Gradient Boosting models, particularly Extreme Gradient Boosting (XGBoost), to predict stock performance with good model efficacy across various markets. Study [17] developed ensemble machine learning models, including XGBoost, Gradient Boosting, and AdaBoost, to predict stock returns of Indian banks using technical indicators such as price, volume, and turnover. This study found that XGBoost outperformed other models, with MAE and RMSE values ranging between 3-5%. Study [18] porposed a hybrid ensemble model that combines Extreme Gradient Boosting and Long Short-Term Memory with CNN-based feature selection. By optimizing input features, the study achieved higher accuracy in predicting stock closing prices, surpassing traditional models such as SVM and Kernel Extreme Learning Machine. Study [19] examined using of XGBoost, Random Forest, and Logistic Regression in predicting stock movements using technical indicators and Google Trends data in the Thai Stock market. XGBoost demonstrated the best performance during periods of extreme market volatility, while Randome Forest performed better in stable market conditions. Study [20] focused on applying ensemble models, including XGBoost, to predict stock price movements in European markets. By integrating macroeconomic indicators and historical price trends, the study showed that XGBoost achieved high prediction accuracy in volaitle market conditions. Study [21] explored the integration of XGBoost with advanced sentiment analysis techniques to predict stock prices in the US market. This study analyzed a large

dataset from social media, financial news, and market indicators, demonstrating XGBoost's superiority in capturing price movements influenced by sentiment.

These previous studies produced multiple evaluation metrics demonstrating model accuracy and performance, such as RMSE, MAPE, MAE, and R^2. This research has limitations worth noting, including a narrow research focus on certain stocks within specific indices or categories, often overlooking the unique characteristics of each dataset, such as the trends of individual stocks. This limitation can hinder the generalizability of research findings across different market conditions. Conducting tests that incorporate various asset classes and market situations could provide deeper insights into the distinctiveness and constraints of the model. Additionally, there has been insufficient exploration of different variants within Gradient Boosting models, such as XGBoost in Scikit-Learn, XGBoost, XGBoost Random Forest, and CatBoost. Comparing these model variations and assessing their suitability for specific stock data characteristics could yield a more comprehensive understanding. Another limitation of prior studies is the lack of integration of technical, fundamental, and sentiment analysis in stock price prediction. While technical indicators like moving averages or RSI are commonly used, fundamental metrics such as P/E ratios, earnings growth, and macroeconomic indicators (e.g., GDP or interest rates) remain underrepresented. Sentiment data, including news sentiment or social media trends, is rarely incorporated. This gap may lead to an incomplete understanding of stock market dynamics, as combining these three aspects offers a more holistic view of market conditions and potential future price movements.

This study addresses these gaps by leveraging diverse stock datasets that represent a variety of market conditions, including uptrends, downtrends, sideways trends, and periods of high volatility, sourced from the Indonesia Stock Exchange. Furthermore, it includes an in-depth comparison of multiple Gradient Boosting variants, such as XGBoost, XGBoost Random Forest, CatBoost, and Gradient Boosting Scikit-Learn, under different stock data scenarios. This study offers a novel and robust framework for stock price prediction by integrating technical, fundamental, and sentiment data. These contributions aim to provide a deeper understanding of the strengths and weaknesses of Gradient Boosting models and their variants while offering actionable insights for researchers and practitioners alike.

2. RESEARCH METHODS

A quantitative comparative approach is employed in this study, focusing on four stocks listed on the Indonesia Stock Exchange as of December 2023. These stocks represent various stock data trends being analyzed: BBRI (PT Bank Rakyat Indonesia Tbk) represents uptrend stocks, ASII (PT Astra International Tbk) represents sideways stocks, UNVR (PT Unilever Indonesia Tbk) represents downtrend stocks, and PTBA (PT Bukit Asam Tbk) represents volatile stocks.ent). We chose the selected stocks based on their distinct price movement patterns, which indicate various market trends. The datasets were sourced from Kaggle, provided in CSV format, and processed using Python and the Pandas library. The closing price data for each stock was used as the primary feature to analyze current stock trends. These stocks were specifically chosen to represent a broad spectrum of stock behaviors in the Indonesian stock market. By examining these diverse stocks, the study aims to capture a variety of market conditions. However, we acknowledge that the sample size of only four stocks may not fully represent the market as a whole, as it is limited to the Indonesian stock exchange and may not account for various sectors or external economic factors that could influence stock movements. Researchers conducted preliminary checks on the data to address potential biases, ensure consistency, and minimize data quality issues. The stock selection reflects diverse trends, but further research could explore broader datasets or additional sectors to enhance the results' generalizability.

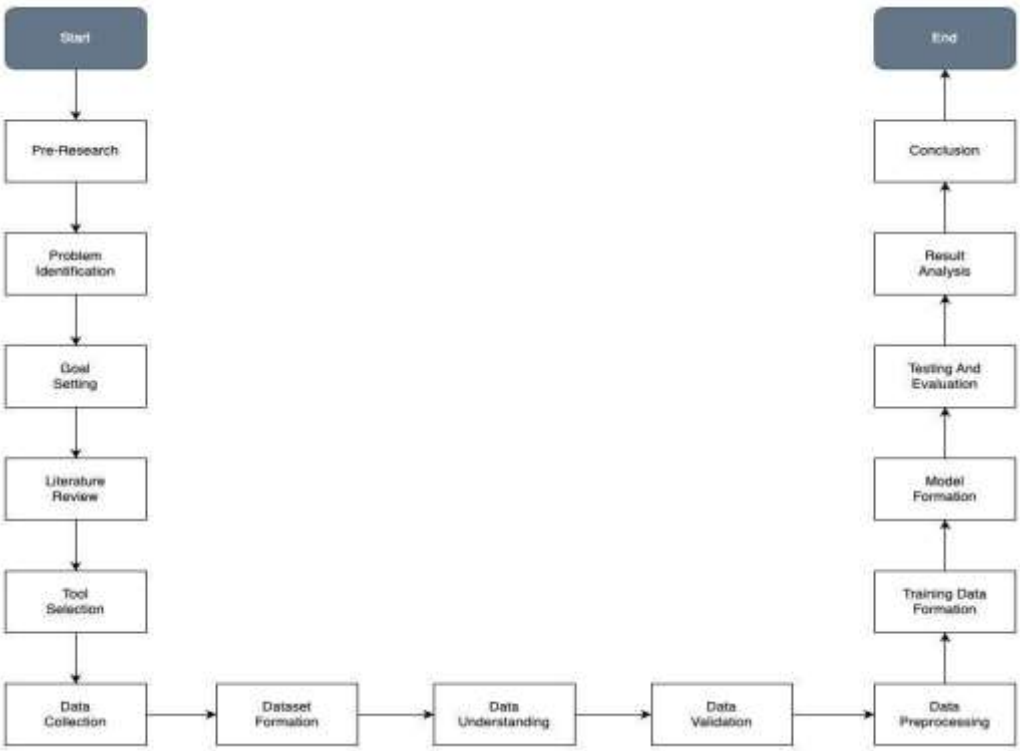

Figure 1. Research Stages

Figure 1 shows 15 stages, starting from Pre-Research, Problem Identification, Goal Setting, Literature Review, Tool Selection, Data Collection, Dataset Formation, Data Understanding, Data Validation, Data Preprocessing, Training Data Formation, Model Formation, Testing and Evaluation, Result Analysis, to Conclusion.

In the Pre-Research stage, a preliminary study is conducted to understand the background and context of the research. Information is gathered regarding the prediction of stock price closing trends and methods that have been previously used from both journal books and the internet. The Goal of this stage is to develop a deep understanding of the research topic. The outcome is a profound understanding of the gradient boosting method as a predictive algorithm and how to process data in the form of technical, fundamental, and sentiment aspects as datasets for prediction. In the Problem Identification stage, problems that need to be solved or areas for improvement in predicting stock price closing trends on the Indonesia Stock Exchange are identified. The identified issues focus on the performance of the Gradient Boosting model variants in identifying and predicting various characteristics of diverse stock data. This stage aims to determine the problem or area that will be the focus of the research. In the Goal Setting stage, the objectives of the study are established, namely to evaluate the ability of Gradient Boosting model variants to address specific characteristics of stock data, including sideways, uptrend, downtrend, and volatility, with particular goals of analyzing the accuracy of the model in forecasting stock price closing directions, measuring the model's success in recognizing and predicting stock movements and evaluating the model's response to diverse stock data characteristics. This stage aims to establish the parameters and focus of the research being conducted.

In the Literature Review stage, a deeper literature study is conducted on Gradient Boosting models, their usage in predicting stock price closing trends, and related research. This stage aims to provide a better knowledge base for designing predictive models for stock price closing trends. This stage plays a role in providing the knowledge and understanding necessary to design and implement quality research that contributes to the field of predicting stock price closing trends. In the Tool Selection stage, tools are chosen for model formation, testing, and visual analysis of stock data. The tools used in this research include Orange 3.36.1, Python 3.11.5, Jupyter Notebook 6.5.4, Node.js 20.11.0, Tweet Harvest 2.2.8, and NLTK (Natural Language Toolkit) 3.8.1.

Orange is used for model formation and testing because it has features that facilitate the testing variants of the Gradient Boosting model, including XgBoost, XgBoost Random Forest, Catboost, and Gradient Boosting Scikit-Learn. The data used in Orange consists of a CSV dataset containing fundamental, technical, and sentiment data, which will then be processed into stock price closing predictions from each model variant. These models were selected due to their established performance in handling diverse and complex datasets,

making them well-suited for stock price prediction. Python scripts with the Jupyter Notebook platform are used for stock data analysis through comparison graphs. Jupyter was selected for its flexibility in data processing and ability to easily compare actual data with predictions from different Gradient Boosting variants [22].

The graphs compare actual data with datasets and comparisons between each Gradient Boosting variant. The data used in the Jupyter Notebook platform consists of technical CSV datasets and prediction results from each Gradient Boosting variant. Node.js with the Tweet Harvest library is used to collect sentiment data from Twitter. Tweet Harvest is a crawler that automatically visits Twitter web pages, follows links, and retrieves relevant data as instructed. The collected data is then downloaded in CSV format. The CSV data is processed using the NLTK library with the SentimentIntensityAnalyzer function in Python. This tool is used to perform sentiment analysis on the obtained Twitter data. The SentimentIntensityAnalyzer is a natural language processing tool used to analyze text sentiment, provide sentiment scores, and determine whether the text is positive, negative, or neutral [23]. Current tools are also considered to take advantage of the latest features that can enhance the accuracy and precision of the formed stock analysis models [24]. The tools selected ensure the incorporation of the latest features, enhancing the accuracy and precision of the models. However, it is important to acknowledge the limitations of the chosen methods. While Gradient Boosting models are powerful, they may not perform well on certain datasets with high noise levels or outliers. Additionally, the sentiment analysis tool, though effective, may not always capture the full nuance of sentiment expressed in social media text. These limitations should be considered when interpreting the results of the analysis.

In the Data Collection stage, stock data that can be used for the research is searched. The purpose of this stage is to obtain data that is relevant and aligned with the research focus. Technical data is obtained from the Kaggle platform in a zip file, which is then extracted into a collection of datasets per stock in CSV format. The data used consists of stocks listed on the Indonesia Stock Exchange from July 29, 2019, to December 7, 2023. The dataset includes information such as date, previous, open_price, first_trade, high, low, close, change, volume, value, frequency, index_individual, offer, offer_volume, bid, bid_volume, listed_shares, tradable_shares, weight_for_index, foreign_sell, foreign_buy, delisting_date, non_regular_volume, non_regular_value, and non_regular_frequency. An example of technical stock data can be seen in Table 1.

Table 1. Example of Technical Stock Data for BBRI.

date	open_price	high	low	close
2019-07-29	4480	4480	4440	4460
2020-07-29	3140	3159	3100	3120
2021-07-29	3730	3769	3769	3840
2022-07-29	4360	4430	4340	4360
2023-07-31	5700	5700	5650	5650

Table 1, Shows sample technical data for BBRI stock. The data includes various stock price information on specific dates, such as the opening price (open_price), highest price (high), lowest price (low), and closing price (close).

The fundamental data was obtained from PT Ajaib Sekuritas from Q1 2019 to Q4 2023 in JSON format, which was then converted to CSV format. The dataset includes information such as ROE (Return on Equity), ROA (Return on Asset), GPM_PCT (Gross Profit Margin), OPM_PCT (Operating Profit Margin), NPM_PCT (Net Profit Margin), earnings_before_tax, EPS (Earnings Per Share), PBV_ratio, PE_ratio, current_ratio, and debt_equity_ratio. Sample fundamental stock data can be seen in Table 2.

Table 2. Example of Fundamental Stock Data for BBRI.

year	qrt	roe	roa	gpm
2019	Q1	0.0424	0.006375	0.2815
2020	Q2	0.00885	0.00135	0.2122
2021	Q4	0.03815	0.006575	0.2146
2022	Q3	0.0443	0.00825	0.3164
2023	Q3	0.043225	0.007475	0.3297

Table 2 shows sample fundamental data for BBRI stock. This data includes several fundamental indicators such as Return on Equity (ROE), Return on Assets (ROA), and Gross Profit Margin (GPM) for certain years and quarters.

Sentiment data was obtained from Twitter from 2019 to 2023 using the Tweet Harvest library, focusing on tweets from prominent media accounts that provide business, economic, and capital market news, such as KontanNews, CNBC Indonesia, and Bisnis.com. The data was collected in CSV format and then processed using the SentimentIntensityAnalyzer library. SentimentIntensityAnalyzer is part of VADER (Valence Aware Dictionary and Sentiment Reasoner), a sentiment analysis model designed to analyze social media text. First, the text undergoes preprocessing through tokenization, normalization, and identification of informal language elements such as emoticons and acronyms. Each word in the text is then compared with the VADER sentiment lexicon, which maps words to positive or negative sentiment values. VADER then generates four principal scores: positive, representing the proportion of positive text; negative, the proportion of negative text; neutral, the proportion of neutral text; and compound, an overall normalized score ranging from -1 to +1. These scores provide an overview of the overall sentiment of the analyzed text. SentimentIntensityAnalyzer is used to obtain sentiment and sentiment_score based on user tweets. The data can then be used for further analysis. Sample sentiment data can be seen in **Table 3**.

Table 3. Example of Sentiment Stock Data for BBRI.

Created_at	Full_text	Username	Sentiment_score	Sentiment
2019-01-02T06:23:06	BBRI & BMRI Stocks Become the Main Pressure on IHSG in Session I	Bisniscom	0	Neutral
2020-02-18T01:50:03	Foreign investors buy BBRI stocks; buyers from a month ago have a potential profit of nearly 4%.	KontanNews	0.4404	Positive
2021-02-10T02:43:19	Becoming the Ultra Micro Holding, BBRI Stock Remains Steady in the Green Zone	cnbcindonesia	0	Neutral
2022-05-09T10:14:32	IHSG Drops After Eid Holiday, BMRI and BBRI Stocks Among Today's Top Losers	Bisniscom	-0.3818	Negative
2023-12-20T03:50:34	Set to Distribute Interim Dividend, BBRI Stock Reaches Rp 5,700	KontanNews	0	0

Table 3 shows sample sentiment data for BBRI stock. This data includes information on messages or text related to BBRI stock, the date the message was created, the username of the message creator, the sentiment score, and the sentiment associated with the message.

We collected the technical, fundamental, and sentiment data from various sources with 37 features. The feature selection process was guided by domain-specific knowledge, statistical analysis, and the relevance of each feature to stock price prediction. Features were chosen based on their direct or indirect influence on stock price movements. For technical data, selected features for their ability to capture historical price patterns, trends, volatility, and momentum. Fundamental metrics were included because they reflect a company's financial health, profitability, and valuation, which are critical for assessing intrinsic value. Sentiment data, derived from social media posts, was incorporated to reflect market sentiment and investor behavior, which can significantly influence short-term price fluctuations.

Statistical analysis was conducted to evaluate the correlation of features with stock price movements, further validating their inclusion. For example, technical indicators like volume and price change strongly correlate with price volatility and momentum. At the same time, sentiment scores were found to signal positive or negative trends during periods of high market activity. Domain-specific justifications further supported the inclusion of certain features. For instance, we selected metrics such as foreign_sell and foreign_buy to capture the impact of foreign investor activity, which is particularly significant in the Indonesian stock market. Ratios like current_ratio and debt_equity_ratio were chosen to assess liquidity and financial stability, both crucial for predicting long-term price performance. The inclusion of sentiment analysis aligns with behavioral finance principles, highlighting the influence of public opinion and market reactions on short-term price dynamics. The selected features provide a comprehensive representation of stock price influences by combining technical trends, financial fundamentals, and real-time sentiment data.

Technical data supports predicting short-term trends and volatility, fundamental data offers insights into long-term price direction, and sentiment data captures immediate market reactions to news and events. This holistic approach ensures that the model can adapt to various market conditions, enhancing its accuracy in predictions.

The data is combined into a single CSV file using Python's Pandas library. First, the daily technical, quarterly fundamental, and sentiment data were imported from their respective sources using the Pandas library. The frequency of the fundamental data was converted to a daily format using a Python script. Afterward, the three data types were merged using the date as the primary key and extracted into a new CSV file. An example of the combined stock data is shown in **Table 4**.

Table 4. Example of merged BBRI stock data.

Date	Open_price	Close	Roe	Sentiment
2019-07-29	4480	4460	0.03995	Neutral
2020-06-02	3290	3180	0.00885	Negative
2021-01-04	4150	4310	0.03562	Positive
2022-07-29	4360	4430	0.0443	Neutral
2023-02-09	4790	4810	0.05545	Positive

Table 4 shows a sample of merged data for BBRI stock. This data includes information on the opening price (open_price), closing price (close), return on equity (ROE), and related sentiment for specific dates. In the Data Validation stage, the tested stock dataset is compared with the actual available stock data. This stock data validation is essential to ensure the accuracy and reliability of the data used in the analysis.

In the Data Pre-Processing stage, preparation steps are carried out to ensure data reliability and compatibility before it is used for model building or further analysis. This stage begins with handling missing attribute values and filling any missing values with a value of 0. The decision to use this method was based on the nature of the dataset, where missing values often indicated the absence of a particular event or transaction rather than an error in data collection. Alternative imputation methods, such as mean, median, or mode substitution, were considered; however, they were deemed less suitable in this case, as they could introduce bias or inaccurately represent the original data's characteristics. Using 0 ensured a neutral, non-influential value that maintained the dataset's integrity while avoiding the loss of valuable information. Next, data normalization is performed to ensure that all attributes are uniform, minimizing the potential bias in the analysis due to scale differences across features. This step ensured that no single attribute disproportionately influenced the model due to its range of values. Finally, feature selection is conducted to choose only the features that significantly impact the target. This process involved evaluating each feature's relevance and relationship with the target using statistical analyses and domain-specific knowledge. Focusing on impactful features reduced the model's complexity, and its performance and interpretability were enhanced.

In the Training Data Formation stage, the dataset is divided into training and test data. The training data covers the period from July 2019 to September 2023, while the test data spans from October 2023 to December 2023, covering a total period of the next three months. Splitting the data with the appropriate proportion can yield good accuracy levels **[19]**. The goal of this stage is to ensure that the developed model not only performs well on the training data but also provides accurate and reliable predictions on new or test data. In the Modeling stage, various Gradient Boosting models are built, namely XgBoost, XgBoost Random Forest, CatBoost, and Scikit-Learn Gradient Boosting. The modeling process involves parameter tuning to ensure that each algorithm can optimize its performance in processing stock data. The aim of this stage is to ensure that each model can be adjusted to meet the specific needs of the stock dataset being use This stage aims to ensure that each model can be adjusted to meet the specific needs of the stock dataset being used.

Gradient Boosting is an ensemble learning method that produces a predictive model by sequentially combining several weak models. This model corrects the prediction errors of the previous model in a series of iterations to improve overall performance **[25]**.

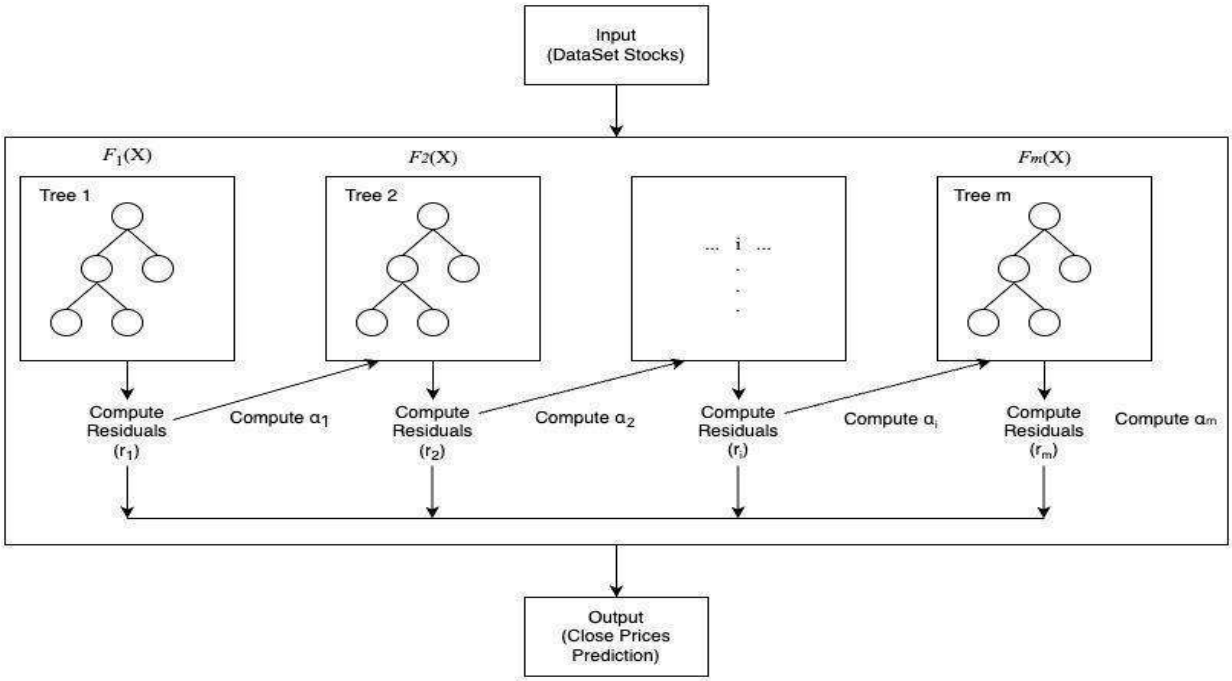

Figure 2. Illustration of Gradient Boosting

Figure 2 shows an illustration of Gradient Boosting. Gradient Boosting calculations can be performed with a scenario where there is a dataset consisting of n samples, represented by **Equation (1)**, where x_n is the feature vector for the n sample, and y_n is its label. The goal is to build a predictive model that maps x to y.

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad (1)$$

The steps in Gradient Boosting begin with initializing the model F_0(x) with a constant value, such as the average of all y_n. For each iteration m = 1 to M, where M is the total number of trees or desired iterations, the next step is to calculate the gradient of the loss function with respect to the previous prediction using **Equation (2)**.

$$r_i = -\frac{\partial L(y, f(x))}{\partial f(x)} \quad (2)$$

Equation (2) shows the required change in the model's prediction with respect to the change in the loss function value, which is then used to determine the residuals that the model will focus on in the next iteration. Here, L is the loss function, y is the actual value, and $f(x)$ is the previous model prediction.

Once the residual values are obtained, the next step is to build a new tree. The new tree (h_i) is built to predict the residuals (r_i). This tree is trained using all the available features. The goal of this tree is to predict the errors from the previous model so that the model can be improved by continuously correcting the mistakes made. Next, the new prediction for iteration i $(f_i(x))$ is updated by adding the prediction of the new tree, scaled by the learning rate (a). This can be mathematically expressed in **Equation (3)**.

$$f_i(x) = f_{i-1}(x) + a \, . \, h_i(x) \quad (3)$$

Equation (3), Shows the new prediction equation, where (a) is the learning rate, typically a small value between 0.01 and 1. These steps are repeated until the maximum number of iterations (M) is reached. At each iteration, the model becomes more complex by adding a new tree that predicts the residuals of the previous prediction, gradually reducing errors and increasing accuracy. After the final iteration, the final model $(f_M(x))$ is the combination of all the previous models that have been added to the ensemble. The final prediction is the result of the sum of all predictions from the individual models in the ensemble.

XGBoost (Extreme Gradient Boosting) is a highly popular and effective machine learning algorithm for regression and classification problems **[26]**. Based on the ensemble learning technique known as gradient

boosting, XGBoost strengthens the model gradually by adding weak models sequentially, focusing on reducing the residual errors from the previous model [27]. XGBoost Random Forest (XGBoost RF) is a combination of the XGBoost algorithm and the Random Forest concept. Random Forest is an ensemble learning method consisting of many decision trees built randomly and independently from each other XGBoost Random Forest (XGBoost RF) combines the XGBoost algorithm and the Random Forest concept. Random Forest is an ensemble learning method consisting of many decision trees built randomly and independently from each other [28][29][30]. CatBoost is a machine learning algorithm developed by Yandex that is specifically designed to handle classification and regression problems with categorical data, but it can also handle numerical variables. CatBoost uses a gradient boosting approach similar to XGBoost but has several unique features and optimizations that set it apart [31][32][33]. Gradient Boosting Scikit-Learn is an implementation of the gradient boosting algorithm available in the Scikit-Learn library, one of the popular libraries in the Python ecosystem for machine learning [34][35][36]. XGBoost, XGBoost Random Forest (XGBoost RF), CatBoost, and Gradient Boosting from Scikit-Learn are popular machine learning algorithms for regression and classification tasks. XGBoost is a boosting method that gradually strengthens the model and is known for its speed and efficiency in processing data. XGBoost RF combines the XGBoost boosting technique with the Random Forest method, using many random trees to improve performance. CatBoost, developed by Yandex, is explicitly designed to handle categorical data quickly and efficiently.

Meanwhile, Gradient Boosting from Scikit-Learn is an easy-to-use implementation that integrates with other Scikit-Learn tools, making it ideal for beginners and researchers. Each algorithm has its advantages. XGBoost is very fast and provides many tuning options for optimal results. It also reduces the risk of overfitting with special techniques. XGBoost RF performs better with variable data and reduces the risk of overfitting by combining boosting and Random Forest techniques. CatBoost excels at handling categorical data without requiring much preprocessing and supports automatic tuning for ease of use. Gradient Boosting from Scikit-Learn is accessible, highly flexible, and promotes various regression and classification applications, making it ideal for basic learning. There are also several drawbacks to consider. XGBoost requires extensive tuning for the best results and can take a long time on large datasets. XGBoost RF requires more memory and processing time and more complex parameter tuning. CatBoost is less user-friendly for new users and has more limited support as Yandex develops it. Gradient Boosting from Scikit-Learn, although easy to use, performs slower than XGBoost and CatBoost and is less efficient on large datasets and when handling missing values.

In the Testing and Evaluation Stage, the built model is tested and evaluated using data separate from the training data. Evaluation metrics such as Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared (R2), Train Time, and Test Time are used to measure the model's performance. The evaluation metrics used in this study are carefully selected to provide a comprehensive understanding of the model's performance in stock price prediction tasks. Mean Absolute Percentage Error (MAPE) offers a percentage-based perspective on prediction errors, making it particularly relevant for financial decision-making, where understanding relative errors is crucial. Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are chosen for their ability to emphasize more significant errors, which is essential in stock prediction scenarios where outliers or significant deviations can heavily impact the model's reliability. Mean Absolute Error (MAE) complements these metrics by providing a straightforward average of absolute errors, ensuring an easy-to-interpret measure of the model's overall accuracy.

Additionally, R-squared (R^2) is included to evaluate how well the model explains the variability in stock prices, offering insights into its predictive power. Finally, Train Time and Test Time are considered critical metrics to assess the computational efficiency of the model, which is particularly important in time-sensitive domains like stock market analysis. By integrating these metrics, the evaluation process provides a detailed assessment of the model's accuracy, precision, and practical applicability in predicting stock price closing directions. This stage aims to provide a comprehensive understanding of the model's accuracy and precision in predicting stock price closing directions.

In the Result Analysis Stage, the prediction results obtained from the model testing are analyzed to evaluate the Gradient Boosting variants' ability to handle the specific characteristics of stock data, such as sideways, uptrend, downtrend, and volatile conditions. This stage aims to gain a deep understanding of the model's response to different market conditions, such as its ability to identify stock movement patterns.

In the final stage, the Conclusion Stage, conclusions are drawn from the analysis and research findings. These conclusions reflect whether the research objectives have been achieved, provide insights into the

Gradient Boosting model variants' ability to handle specific stock data characteristics, such as sideways, uptrend, downtrend, and volatile conditions, and offer recommendations for future research. The goal of this stage is to provide a summary of the research findings and suggestions for future studies.

3. RESULTS AND DISCUSSION

The data testing is conducted using the Orange Tools. The testing flow using Orange Tools can be seen in **Figure 3**.

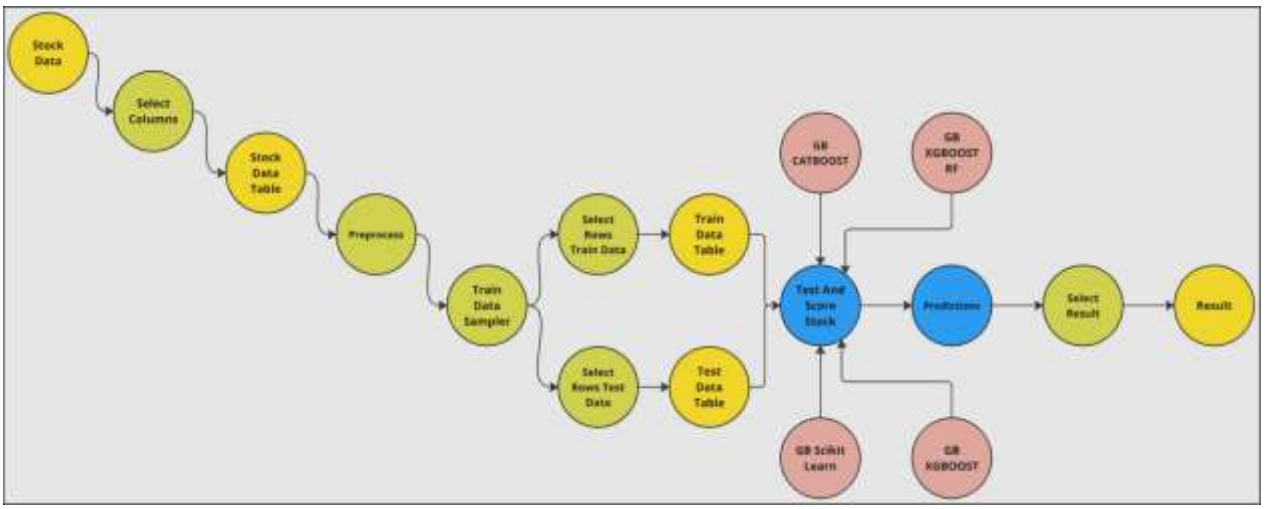

Figure 3. Stock Data Testing Flow Using Orange Tools

Figure 3 shows the stock data testing flow. The process begins by importing stock data using the CSV File Import feature. Next, feature selection is performed using the Select Column feature to determine the relevant variables. These variables include technical aspects such as previous, open_price, first_trade, high, low, volume, value, frequency, foreign_sell, foreign_buy, change, index_individual, offer, offer_volume, bid, bid_volume, listed_shares, tradeable_shares, weight_for_index, non_regular_volume, non_regular_value, and non_regular_frequency. From the fundamental side, the variables considered include roe, roa, gpm, opm, npm, earn_before_tax, eps, pbv, pe, current_ratio, and der. From sentiment analysis, the variables considered are sentiment_score and sentiment, with the target being the closing price, labeled as close. The data preprocessing involves handling missing attribute values by replacing rows with missing values with 0. Next, normalization is performed with a data interval from 0 to 1 using the "Normalize to interval [0,1]" feature in the Preprocess tool.

Table 5. Example of Normalization Stock Data for BBRI.

Date	Low	High	Roe	Close
2019-08-05	0.60028	0.61781	0.6856	4270
2020-06-02	0.22475	0.29310	0.0548	3180
2021-01-04	0.56614	0.58908	0.739	4310
2022-07-29	0.62019	0.62069	0.7738	4360
2023-02-10	0.74253	0.75	1	4860

Table 5 shows an example of the data normalization results for BBRI stock within the range of 0 to 1, aimed at simplifying comparison or analysis. In this table, data such as the lowest price (low), highest price (high), and Return on Equity (ROE) have been normalized. The Data Sampler feature is used to generate the training and test datasets with the help of the Select Rows feature to define the data range based on dates. The training data is taken from July 29, 2019, to October 6, 2023, while the testing data is taken from October 9, 2023, to December 7, 2023.

Modeling is performed using the Gradient Boosting feature, which includes four model variants: XgBoost, XgBoost Random Forest, CatBoost, and Gradient Boosting Scikit-Learn. For each model, hyperparameter tuning is carried out using 1000 decision trees, a learning rate of 1, and a maximum depth of

10 for each individual tree. With a large number of trees, the model can capture complex relationships in the data, while the high learning rate allows the model to quickly adapt to existing patterns [37]. However, to prevent overfitting, the depth of each individual tree is limited to 10, ensuring that each tree has good generalization capability on the test data [38]. This combination of hyperparameter settings is designed to create a predictive model that maintains strong performance without compromising generalization on unseen data [39]. To optimize these parameters and improve model performance, a grid search approach with cross-validation was implemented. The grid search explored hyperparameter variations, such as the number of estimators, learning rates, and tree depths, ensuring that the final settings balanced accuracy and generalization. Limiting the depth of each tree to 10 helped ensure that individual trees generalized well to unseen data, preventing overfitting despite the large number of trees. After modeling, the Test and Score feature is used to evaluate the model's performance on stock data based on the test dataset, which was not used during the modeling phase.

Table 6. Composition of the Training and Test Data.

Stock Code	Characteristics	Train Data	Testing Data	Total
BBRI	Uptrend	23258	1158	24416
ASII	Sideways	11710	657	12367
UNVR	Downtrend	8910	474	9348
PTBA	Volatile	6414	474	6888

Table 6 shows the training and test data composition for four stocks: BBRI, ASII, UNVR, and PTBA. The data composition is divided based on the characteristics of stock price movements: uptrend, sideways, downtrend, and volatile. This table provides information on the total training data, total test data, and total data for each stock and its respective price movement characteristics. For example, for the BBRI stock with an uptrend characteristic, the total data is 244,216, which is divided into 23,258 training data and 1,158 test data.

Table 7. Testing results based on Execution Time, MAPE, and R^2.

Stock Code	Model	Execution Time (s)		MAPE	R2
		Train	Test		
BBRI	Scikit Learn	4.163	0.004	0.002	0.982
	XGBoost	0.751	0.001	0.005	0.907
	XGBoost RF	6.915	0.007	0.005	0.907
	CatBoost	14.75	0.026	0.015	0.502
ASII	Scikit Learn	1.808	0.004	0.010	0.632
	XGBoost	0.239	0.001	0.009	0.713
	XGBoost RF	4.870	0.009	0.009	0.726
	CatBoost	7.714	0.009	0.025	-0.64
UNVR	Scikit Learn	1.403	0.002	0.011	0.887
	XGBoost	0.243	0.001	0.013	0.862
	XGBoost RF	4.681	0.006	0.013	0.868
	CatBoost	7.430	0.007	0.032	-1.88
PTBA	Scikit Learn	1.147	0.002	0.009	0.962
	XGBoost	0.290	0.001	0.010	0.951
	XGBoost RF	5.129	0.009	0.009	0.953
	CatBoost	7.405	0.003	0.099	-2.77

Table 7 shows the results of several Gradient Boosting model variants, namely Scikit Learn, XGBoost, XGBoost RF, and CatBoost used to predict stock performance for the stock codes BBRI, ASII, UNVR, and PTBA. Each model is evaluated based on execution time (in seconds) for the training (Train) and test (Test)

data, Mean Absolute Percentage Error (MAPE), and R-squared (R^2) values. The testing results indicate that the Scikit Learn model performs better in predicting the direction of stock price closing in some cases, such as for the BBRI stock code. This is supported by a Mean Absolute Percentage Error (MAPE) of 0.002 and an R-squared value of 0.982. However, in general, the XGBoost and XGBoost RF models provide better performance than Scikit Learn and CatBoost. For example, for the ASII stock code, XGBoost and XGBoost RF have the same MAPE of 0.009 and R-squared values that are relatively close, at 0.713 and 0.726, respectively. Additionally, the execution time of XGBoost is faster than the other models, as shown for the UNVR stock code, with a training time of 0.243 seconds and a test time of 0.001 seconds.est data.

Table 8. Testing results based on MSE, RMSE and MAE.

Stock Code	Model	MSE	RMSE	MAE
BBRI	Scikit Learn	4.163	0.002	0.982
	XGBoost	0.751	0.005	0.907
	XGBoost RF	6.915	0.005	0.907
	CatBoost	14.75	0.015	0.502
ASII	Scikit Learn	1.808	0.010	0.632
	XGBoost	0.239	0.009	0.713
	XGBoost RF	4.870	0.009	0.726
	CatBoost	7.714	0.025	-0.64
UNVR	Scikit Learn	1.403	0.011	0.887
	XGBoost	0.243	0.013	0.862
	XGBoost RF	4.681	0.013	0.868
	CatBoost	7.430	0.032	-1.88
PTBA	Scikit Learn	1.147	0.009	0.962
	XGBoost	0.290	0.010	0.951
	XGBoost RF	5.129	0.009	0.953
	CatBoost	7.405	0.099	-2.77

Table 8 shows the advantages of the XGBoost and XGBoost RF models in several cases. For example, for the ASII stock code, the XGBoost model has an MSE of 4670.79, an RMSE of 68.343, and an MAE of 53.957, which are lower compared to the Scikit Learn and CatBoost models.

The analysis results highlight differences in model performance based on various stock characteristics: uptrend, sideways, downtrend, and volatile conditions, providing insights into the robustness and adaptability of each model. For uptrend stocks (such as BBRI), the Scikit Learn model demonstrated exceptional performance with a MAPE of 0.002 and an R^2 of 0.982, reflecting its effectiveness in predicting precise, linear price movements. More complex models like XGBoost and CatBoost performed relatively worse on BBRI. XGBoost showed a MAPE of 0.005 and an R^2 of 0.907, suggesting these models may struggle with overfitting or capturing the simplicity of clear upward trends. In contrast, for sideways stocks (such as ASII), which exhibit more minor and less predictable fluctuations, XGBoost and XGBoost RF performed better with a MAPE of 0.009 and R^2 values of 0.713 and 0.726, respectively, due to their ability to capture non-linear relationships and feature interactions. Scikit Learn showed higher error metrics, while CatBoost struggled with a negative R^2 value (-0.64), indicating its inability to capture less defined trends. For downtrend stocks (such as UNVR), XGBoost and XGBoost RF again excelled with low MAPE and high R^2 values (0.862 and 0.868), as these models effectively handled downward price movements. Scikit Learn performed slightly worse (MAPE 0.011, R^2 0.887), while CatBoost once again showed negative R^2 values, indicating its poor performance on downward trends. For volatile stocks (such as PTBA), which exhibit erratic price behavior, XGBoost, and XGBoost RF provided the most accurate predictions with MAPE close to 0.009 and R^2 of 0.953 due to their ability to capture unpredictable and high-variance patterns. Scikit Learn also performed well with an R^2 of 0.962, but CatBoost again underperformed with negative R^2 and high MAPE (0.099), reflecting its difficulty in handling highly volatile data. Overall, the consistent outperformance of XGBoost

and XGBoost RF across various market conditions indicates that these models are well-suited for handling non-linear and complex relationships in stock data.

In contrast, Scikit Learn is effective for more apparent linear trends, while CatBoost requires further adjustment to handle erratic or volatile patterns. However, limitations persist for all models, such as the risk of overfitting, even after hyperparameter tuning. While setting a maximum tree depth of 10 helps maintain generalization, it may not always capture the complexities of highly dynamic datasets. The Prediction feature is used to view the model's predictions on the direction of stock price closing. The prediction results are then extracted into a CSV file and used to create a graph using Jupyter Notebook. The comparison graph of the Gradient Boosting model variants for BBRI stock can be seen in **Figure 4**.

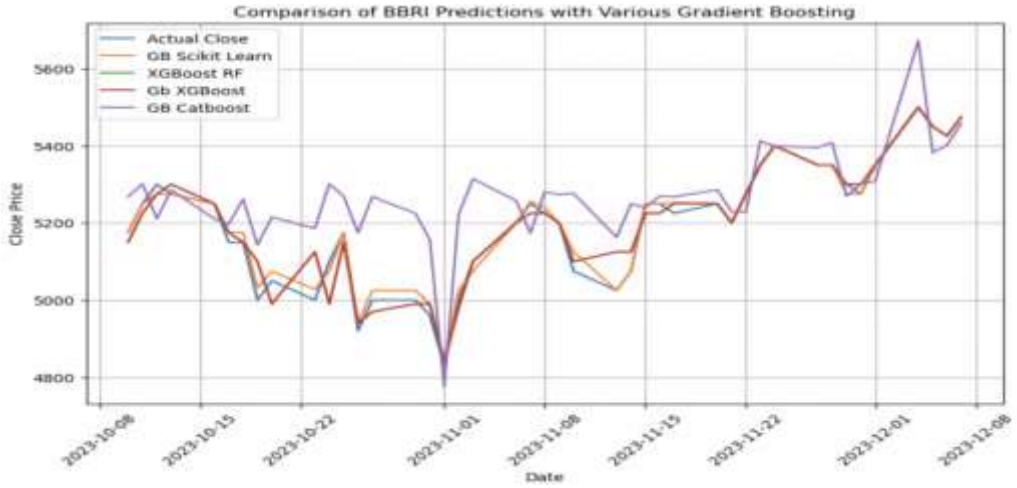

Figure 4. The Comparison Graph of the Gradient Boosting Model Variants for BBRI stock

Figure 4 shows the comparison graph of Gradient Boosting model variants for BBRI stock. The blue graph represents the actual data, the orange graph shows predictions from Scikit-Learn, the green graph shows predictions from XGBoost RF, the red graph shows predictions from XGBoost, and the purple graph shows predictions from CatBoost. From the graph, it is clear that CatBoost provides the least accurate predictions compared to the other models. The XGBoost, XGBoost RF, and Scikit-Learn models show predictions closer to the actual data, with XGBoost and XGBoost RF having very similar prediction results.

These findings have critical real-world applications. For investors, models like XGBoost or XGBoost RF can predict stock price movements more accurately, allowing them to gauge market trends and make better-informed investment decisions. These models excel at identifying underlying patterns in stock price movements, which is crucial for effective portfolio management and risk mitigation. Stock analysts can also leverage these models to forecast stock prices more confidently, especially in volatile market conditions. More accurate predictions enable analysts to advise clients on the optimal times to buy or sell stocks, providing insights into both short-term opportunities and long-term trends. Different market conditions, such as a rising bull market, XGBoost, and XGBoost RF can help pinpoint ideal entry points. Conversely, in a declining bear market, these models can assist in identifying stock bottoms or support levels to minimize losses. With its enhanced generalization capabilities, the Random Forest variant of XGBoost may perform even better in highly volatile conditions than simpler models like CatBoost, which struggled in this context. This reinforces the importance of choosing the right model based on performance. Investors and analysts can select models that align with current market conditions by considering these differences, improving forecasting accuracy.

The results emphasize the importance of selecting models suited explicitly to stock data's unique characteristics. While XGBoost's speed and consistency make it well-suited for real-time applications, the unexpectedly poor performance of CatBoost on certain datasets highlights the need for further optimization before deployment. Issues like negative R^2 values and high MAPE suggest that CatBoost may not be ideal for all types of data or might require additional tuning. In conclusion, adopting a flexible, data-driven approach that integrates multiple models and includes thorough evaluation is crucial for enhancing the accuracy of stock market predictions.

4. CONCLUSIONS

a. The objective of this research was to evaluate the effectiveness of various Gradient Boosting models (XGBoost, XGBoost Random Forest, CatBoost, and Scikit-Learn) in predicting stock closing prices using a combination of technical, fundamental, and sentiment data. The results demonstrate that integrating these diverse data sources into a structured dataset can improve stock price prediction accuracy. XGBoost and XGBoost Random Forest performed particularly well across a range of stocks (BBRI, ASII, UNVR, and PTBA), outperforming other models like CatBoost regarding accuracy and execution time.

b. Based on evaluation metrics such as MAPE, MSE, RMSE, MAE, and R-squared, XGBoost and XGBoost Random Forest generally delivered superior performance in predicting stock closing price direction, with lower errors and higher R-squared values compared to other models. Specifically, XGBoost outperformed CatBoost in all stocks tested, showing significantly lower error values and faster execution times. For example, on the BBRI stock, XGBoost required only 0.751 seconds for training and 0.005 seconds for testing, whereas CatBoost took 14.75 seconds for training and 0.015 seconds for testing. These findings underscore the importance of selecting the appropriate model for stock price prediction, with XGBoost and XGBoost Random Forest being the preferred choices due to their accuracy and efficiency.

c. The results highlight several limitations of the models. XGBoost and XGBoost Random Forest generally perform well, but CatBoost struggles, especially in volatile and downtrend conditions, showing poor performance with negative R^2 values and high MAPE. Overfitting remains a concern despite hyperparameter tuning, and the models may not capture the complexity of dynamic datasets. Additionally, there's a tradeoff between execution time and accuracy—while XGBoost is fast, CatBoost's longer training time doesn't improve accuracy for volatile stocks. Scikit Learn models are better for linear movements but less effective in volatile markets. Finally, model performance varies depending on stock characteristics, meaning no single model is universally best, and selecting the right model for each scenario is crucial.

d. The findings emphasize the importance of model selection, as it significantly impacts prediction accuracy. XGBoost and XGBoost Random Forest are recommended for stock price prediction modeling because they deliver more accurate and efficient results. Future research could explore incorporating additional data sources, such as macroeconomic factors, political events, or global financial indicators, to provide a more comprehensive understanding of the forces affecting stock prices. Furthermore, testing these models under different market conditions, such as during periods of high volatility or economic uncertainty, could reveal how well they adapt to changing financial environments. Integrating ensemble models that combine various prediction techniques could also lead to more robust and consistent results.

REFERENCES

[1] Melina, Sukono, H. Napitupulu, and N. Mohamed, "A CONCEPTUAL MODEL OF INVESTMENT-RISK PREDICTION IN THE STOCK MARKET USING EXTREME VALUE THEORY WITH MACHINE LEARNING: A SEMISYSTEMATIC LITERATURE REVIEW," *Multidisciplinary Digital Publishing Institute (MDPI)*, Mar. 01, 2023, doi: 10.3390/risks11030060.

[2] M. Ali, D. M. Khan, H. M. Alshanbari, and A. A. A. H. El-Bagoury, "PREDICTION OF COMPLEX STOCK MARKET DATA USING AN IMPROVED HYBRID EMD-LSTM MODEL," *Applied Sciences (Switzerland)*, vol. 13, no. 3, Feb. 2023, doi: 10.3390/app13031429.

[3] M. Wątorek, J. Kwapień, and S. Drożdż, "CRYPTOCURRENCIES ARE BECOMING PART OF THE WORLD GLOBAL FINANCIAL MARKET," *Entropy*, vol. 25, no. 2, Feb. 2023, doi: 10.3390/e25020377.

[4] N. U. Devi and R. Mohan, "A BLENDED SOFT-COMPUTING MODEL FOR STOCK-VALUE PREDICTION," *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 09, no. 3, Sep. 2023, doi: 10.5455/jjcit.71-1683995072.

[5] R. Abraham *et al.*, "FORECASTING A STOCK TREND USING GENETIC ALGORITHM AND RANDOM FOREST," *Journal of Risk and Financial Management*, vol. 15, no. 5, May 2022, doi: 10.3390/jrfm15050188.

[6] J. M. T. Wu, Z. Li, N. Herencsar, B. Vo, and J. C. W. Lin, "A GRAPH-BASED CNN-LSTM STOCK PRICE PREDICTION ALGORITHM WITH LEADING INDICATORS," *Multimedia Systems*, Jun. 2023, pp. 1751–1770. doi: 10.1007/s00530-021-00758-w.

[7] C. Yuan, X. Ma, H. Wang, C. Zhang, and X. Li, "COVID19-MLSF: A MULTI-TASK LEARNING-BASED STOCK MARKET FORECASTING FRAMEWORK DURING THE COVID-19 PANDEMIC," *Expert System Application*, vol. 217, May 2023, doi: 10.1016/j.eswa.2023.119549.

[8] Y. F. Zhang and M. Umair, "EXAMINING THE INTERCONNECTEDNESS OF GREEN FINANCE: AN ANALYSIS OF DYNAMIC SPILLOVER EFFECTS AMONG GREEN BONDS, RENEWABLE ENERGY, AND CARBON MARKETS," *Environmental Science and Pollution Research*, vol. 30, no. 31, pp. 77605–77621, Jul. 2023, doi: 10.1007/s11356-023-27870-w.

[9] D. Maulana, A. Sofro, D. Ariyanto, R. Romadhonia, A. Oktaviarina, and D. Purnama, "STOCK PRICE PREDICTION AND SIMULATION USING GEOMETRIC BROWNIAN MOTION-KALMAN FILTER- A COMPARISON BETWEEN KALMAN FILTER ALGORITHMS," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 19, no. 1, pp. 0097–0106, Mar. 2025, doi: 10.30598/barekengvol19iss1pp0097-0106.

[10] S. Alves Basilio and L. Goliatt, "GRADIENT BOOSTING HYBRIDIZED WITH EXPONENTIAL NATURAL EVOLUTION STRATEGIES FOR ESTIMATING THE STRENGTH OF GEOPOLYMER SELF-COMPACTING CONCRETE," *Knowledge-Based Engineering and Sciences*, vol. 3, no. 1, pp. 1–16, Apr. 2022, doi: 10.51526/kbes.2022.3.1.1-16.

[11] M. Liang, Z. Chang, Z. Wan, Y. Gan, E. Schlangen, and B. Šavija, "INTERPRETABLE ENSEMBLE-MACHINE-LEARNING MODELS FOR PREDICTING CREEP BEHAVIOR OF CONCRETE," *Cement and Concrete Composites*, vol. 125, Jan. 2022, doi: 10.1016/j.cemconcomp.2021.104295.

[12] M. W. Dwinanda, N. Satyahadewi, and W. Andani, "CLASSIFICATION OF STUDENT GRADUATION STATUS USING XGBOOST ALGORITHM," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 17, no. 3, pp. 1785–1794, Sep. 2023, doi: 10.30598/barekengvol17iss3pp1785-1794.

[13] G. Sonkavde, D. S. Dharrao, A. M. Bongale, S. T. Deokate, D. Doreswamy, and S. K. Bhat, "FORECASTING STOCK MARKET PRICES USING MACHINE LEARNING AND DEEP LEARNING MODELS: A SYSTEMATIC REVIEW, PERFORMANCE ANALYSIS AND DISCUSSION OF IMPLICATIONS," *Multidisciplinary Digital Publishing Institute (MDPI)*, Sep. 01, 2023, doi: 10.3390/ijfs11030094.

[14] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "MACHINE LEARNING TECHNIQUES AND DATA FOR STOCK MARKET FORECASTING: A LITERATURE REVIEW," *Elsevier Limited*, Jul. 01, 2022, doi: 10.1016/j.eswa.2022.116659.

[15] V. Drakopoulou, "A REVIEW OF FUNDAMENTAL AND TECHNICAL STOCK ANALYSIS TECHNIQUES," *Journal of Stock and Forex Trading*, vol. 05, no. 01, 2016, doi: 10.4172/2168-9458.1000163.

[16] Y. Pei *et al.*, "TWEETFINSENT: A DATASET OF STOCK SENTIMENTS ON TWITTER," *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*, pp. 37–47, Dec. 2022, doi: 10.18563/v1/2022.finnlp-1.5.

[17] S. Mohapatra, R. Mukherjee, A. Roy, A. Sengupta, and A. Puniyani, "CAN ENSEMBLE MACHINE LEARNING METHODS PREDICT STOCK RETURNS FOR INDIAN BANKS USING TECHNICAL INDICATORS?," *Journal of Risk and Financial Management*, vol. 15, no. 8, Aug. 2022, doi: 10.3390/jrfm15080350.

[18] P. R. Mohapatra, A. K. Parida, S. K. Swain, and S. S. Basa, "GRADIENT BOOSTING AND LSTM BASED HYBRID ENSEMBLE LEARNING FOR TWO STEP PREDICTION OF STOCK MARKET," *Journal of Advances in Information Technology*, vol. 14, no. 6, pp. 1254–1260, 2023, doi: 10.12720/jait.14.6.1254-1260.

[19] K. Saetia and J. Yokrattanasak, "STOCK MOVEMENT PREDICTION USING MACHINE LEARNING BASED ON TECHNICAL INDICATORS AND GOOGLE TREND SEARCHES IN THAILAND," *International Journal of Financial Studies*, vol. 11, no. 1, Mar. 2023, doi: 10.3390/ijfs11010005.

[20] Y. Sheng and D. Ma, "STOCK INDEX SPOT–FUTURES ARBITRAGE PREDICTION USING MACHINE LEARNING MODELS," *Entropy*, vol. 24, no. 10, Oct. 2022, doi: 10.3390/e24101462.

[21] J. Xu *et al.*, "FINANCIAL TIME SERIES PREDICTION BASED ON XGBOOST AND GENERATIVE ADVERSARIAL NETWORKS," *International Journal of Circuits, Systems and Signal Processing*, vol. 16, pp. 637–645, 2022, doi: 10.46300/9106.2022.16.79.

[22] M. Botlagunta *et al.*, "CLASSIFICATION AND DIAGNOSTIC PREDICTION OF BREAST CANCER METASTASIS ON CLINICAL DATA USING MACHINE LEARNING ALGORITHMS," *Scientific Reports*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-27548-w.

[23] J. Yao, "AUTOMATED SENTIMENT ANALYSIS OF TEXT DATA WITH NLTK," *Journal of Physics: Conference Series*, May 2019. doi: 10.1088/1742-6596/1187/5/052020.

[24] D. Dwivedi, S. Batra, and Y. K. Pathak, "A MACHINE LEARNING BASED APPROACH TO IDENTIFY KEY DRIVERS FOR IMPROVING CORPORATE'S ESG RATINGS," *Journal of Law and Sustainable Development*, vol. 11, no. 1, Jan. 2023, doi: 10.37497/sdgs.v11i1.242.

[25] W. Corey, "HANDS-ON GRADIENT BOOSTING WITH XGBOOST AND SCIKIT-LEARN : PERFORM ACCESSIBLE MACHINE LEARNING AND EXTREME GRADIENT BOOSTING WITH PYTHON," *Birmingham: Packt Publishing*, 1st ed., vol. 1, 2020. Accessed: Oct. 01, 2024. [Online]. Available: https://studylib.net/doc/25882439/hands-on-gradient-boosting-with-xgboost-and-scikit-learn

[26] S. Elsa Suryana and B. Warsito, "PENERAPAN GRADIENT BOOSTING DENGAN HYPEROPT UNTUK MEMPREDIKSI KEBERHASILAN TELEMARKETING BANK," *Jurnal Gaussian*, vol. 10, no. 4, pp. 617–623, 2021, [Online]. Available: https://ejournal3.undip.ac.id/index.php/gaussian/

[27] I. D. Mienye and Y. Sun, "A SURVEY OF ENSEMBLE LEARNING: CONCEPTS, ALGORITHMS, APPLICATIONS, AND PROSPECTS," *Institute of Electrical and Electronics Engineers Inc, 2022*, doi: 10.1109/ACCESS.2022.3207287.

[28] M. Lotfirad, H. Esmaeili-Gisavandani, and A. Adib, "DROUGHT MONITORING AND PREDICTION USING SPI, SPEI, AND RANDOM FOREST MODEL IN VARIOUS CLIMATES OF IRAN," *Journal of Water and Climate Change*, vol. 13, no. 2, Feb. 2022, doi: 10.2166/wcc.2021.287.

[29] M. G. El-Shafiey, A. Hagag, E. S. A. El-Dahshan, and M. A. Ismail, "A HYBRID GA AND PSO OPTIMIZED APPROACH FOR HEART-DISEASE PREDICTION BASED ON RANDOM FOREST," *Multimedia Tools Application*, vol. 81, no. 13, pp. 18155–18179, May 2022, doi: 10.1007/s11042-022-12425-x.

[30] L. J. Wu, X. Li, J. D. Yuan, and S. J. Wang, "REAL-TIME PREDICTION OF TUNNEL FACE CONDITIONS USING XGBOOST RANDOM FOREST ALGORITHM," *Frontiers of Structural and Civil Engineering*, vol. 17, no. 12, pp. 1777–1795, Dec. 2023, doi: 10.1007/s11709-023-0044-4.

[31] S. Chehreh Chelgani, H. Nasiri, A. Tohry, and H. R. Heidari, "MODELING INDUSTRIAL HYDROCYCLONE OPERATIONAL VARIABLES BY SHAP-CATBOOST - A 'CONSCIOUS LAB' APPROACH," *Powder Technology*, vol. 420, Apr. 2023, doi: 10.1016/j.powtec.2023.118416.
[32] J. Wu *et al.*, "FAULT DIAGNOSIS OF THE HVDC SYSTEM BASED ON THE CATBOOST ALGORITHM USING KNOWLEDGE GRAPHS," *Front Energy Research*, vol. 11, 2023, doi: 10.3389/fenrg.2023.1144785.
[33] W. Chang, X. Wang, J. Yang, and T. Qin, "AN IMPROVED CATBOOST-BASED CLASSIFICATION MODEL FOR ECOLOGICAL SUITABILITY OF BLUEBERRIES," *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23041811.
[34] D. Zhou *et al.*, "ESTABLISHMENT OF A DIFFERENTIAL DIAGNOSIS METHOD AND AN ONLINE PREDICTION PLATFORM FOR AOSD AND SEPSIS BASED ON GRADIENT BOOSTING DECISION TREES ALGORITHM," *Arthritis Research And Therapy*, vol. 25, no. 1, Dec. 2023, doi: 10.1186/s13075-023-03207-3.
[35] R. Alkentar and T. Mankovits, "OPTIMIZATION OF ADDITIVELY MANUFACTURED AND LATTICE-STRUCTURED HIP IMPLANTS USING THE LINEAR REGRESSION ALGORITHM FROM THE SCIKIT-LEARN LIBRARY," *Crystals*, vol. 13, no. 10, Oct. 2023, doi: 10.3390/cryst13101513.
[36] M. F. El-Amin, B. Alwated, and H. A. Hoteit, "MACHINE LEARNING PREDICTION OF NANOPARTICLE TRANSPORT WITH TWO-PHASE FLOW IN POROUS MEDIA," *Energies*, vol. 16, no. 2, Jan. 2023, doi: 10.3390/en16020678.
[37] K. Xu, Z. Han, H. Xu, and L. Bin, "RAPID PREDICTION MODEL FOR URBAN FLOODS BASED ON A LIGHT GRADIENT BOOSTING MACHINE APPROACH AND HYDROLOGICAL–HYDRAULIC MODEL," *International Journal of Disaster Risk Science*, vol. 14, no. 1, pp. 79–97, Feb. 2023, doi: 10.1007/s13753-023-00465-2.
[38] V. Raj, S. Q. Dotse, M. Sathyajith, M. I. Petra, and H. Yassin, "ENSEMBLE MACHINE LEARNING FOR PREDICTING THE POWER OUTPUT FROM DIFFERENT SOLAR PHOTOVOLTAIC SYSTEMS," *Energies*, vol. 16, no. 2, Jan. 2023, doi: 10.3390/en16020671.
[39] D. N. Gono, H. Napitupulu, and Firdaniza, "SILVER PRICE FORECASTING USING EXTREME GRADIENT BOOSTING (XGBOOST) METHOD," *Mathematics*, vol. 11, no. 18, Sep. 2023, doi: 10.3390/math11183813.