iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VII **Month of publication:** July 2024

DOI: https://doi.org/10.22214/ijraset.2024.63695

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VII July 2024- Available at www.ijraset.com

Prediction of Stock Price Using XG Boost: A Machine Learning Technique

Vismayaa Yadav BKV[1], Shailaja K.P [2]
[1]*Dept. of Computer Applications, B.M.S. College Of Engineering, Bangalore, India*
[2]*Dept. of Computer Applications, B.M.S. College Of Engineering, Bangalore, India*

Abstract: ***Forecasting stock prices is a very difficult task due to the sudden and volatile nature of financial markets. This paper reviews recent developments in the use of the XGBoost algorithm for stock price forecasting. XGBoost, a robust and efficient gradient enhancement implementation, has demonstrated excellent performance in a variety of predictive modeling environments. The analysis uses various experimental methods, including data generation, feature engineering, model training, and validation procedures. It also compares the performance of XGBoost with other machine learning algorithms. The findings show that XGBoost is able to capture complex non-linear relationships in stock market data, resulting in improved forecasting accuracy. However, challenges such as excessive packaging and reliance on quality remain. The paper presents possible future research directions, including the integration of mixed models, the use of new data sources, and the enhancement of model interpretation and real-time predictive capabilities.***

Keywords: XGBoost-eXtreme Gradient Boosting, ARIMA- Autoregressive Integrated Moving Average.

I. INTRODUCTION

Stock price forecasting has always been at the core of financial research because of the importance of its implications to investors, financial organizations, and overall economic health. The primary difficulty when it comes to stock price prediction is the stochastic and non-linear nature of financial markets, which are influenced by a vast array of factors, ranging from economic indicators to market sentiment and geopolitical events, as well as investors' behavior.

The linear regression and autoregressive integrated moving average (ARIMA) models used in the conventional approaches for predicting the stock prices fail to capture the non-linear structure and the complex relationships present in the financial time series data. The traditional statistical models that were used in the past to solve such problems do not have the capability of handling such complexities, but the new machine learning approaches can. Of these techniques, the ensemble learning methods especially gradient boosting have become popular due to high accuracy and less overfitting.

XGBoost is an advanced gradient boosting algorithm that has received much attention and appreciation from the machine learning community due to its effectiveness. XGBoost is a more advanced version of the Gradient Boosting algorithm which was developed by Tianqi Chen and Carlos Guestrin; the advancements include regularization, parallel processing, and tree pruning which enhance both computational speed and model performance. These features make the XGBoost particularly suitable for high-dimensional and large-scale data sets, typical for the analysis of the stock market.

This paper aims to provide an overview of the literature on stock price prediction using XGBoost, combining methodologies, findings, and discussions from the current literature. Here, we will discuss the application of XGBoost in predicting the stock prices and identify the advantages, shortcomings, and future directions of this approach. The proposed review covers data pre-processing, feature engineering, model training, and model evaluation in the context of the XGBoost algorithm.

II. RELATED WORK

In the realm of stock price prediction using machine learning (ML) techniques, recent literature reflects a broad spectrum of methodologies and innovations aimed at improving forecasting accuracy and reliability. Researchers from various institutions globally have contributed to this field:

Sumeet Sarode, Harsha G. Tolani, Prateek Kak, and Lifna C S from Vivekanand Education Society's Institute of Technology, Mumbai, India, explore the application of ML algorithms for stock price prediction, focusing on regression and classification methods. Their work emphasizes leveraging historical data patterns to forecast future price movements.

Gourav Bathla, at the University of Petroleum & Energy Studies, Dehradun, India, investigates LSTM (Long Short-Term Memory) and SVR (Support Vector Regression) models for predicting stock prices. Bathla's research highlights LSTM's capability to capture long-term dependencies and SVR's robustness in handling noisy data, aiming to enhance prediction accuracy.

YaoHu Lin, Shancun Liu, Haijun Yang, and Harris Wu from Beihang University, China, propose combining candlestick charting with ensemble ML techniques. Their approach includes innovative feature engineering to improve the predictive power of models, integrating technical indicators with machine learning methodologies.

Sondo Kim, Seungmo Ku, Woojin Chang, and Jae Wook Song from Seoul National University, South Korea, utilize transfer entropy alongside ML techniques to forecast the direction of US stock prices. Their study focuses on identifying causal relationships and dependencies within stock market data, enhancing predictive capabilities.

Audeliano Wolian Li and Guilherme Sousa Bastos, affiliated with the Federal University of Itajubá, Brazil, conduct a systematic review on deep learning and technical analysis integration for stock market forecasting. Their review highlights deep learning's ability to extract intricate patterns from financial data, complementing traditional technical analysis methods.

Donghwan Song, Adrian Matias Chung Baek, and Namhun Kim from Ulsan National Institute of Science and Technology, South Korea, propose a novel approach using padding-based Fourier transform denoising and deep learning models. Their method focuses on improving data quality before applying deep learning techniques to forecast stock market indices.

Empirical studies by Vaibhav Gaur, Shubham Sood, Lisha Uppal, and Manpreet Kaur from Manav Rachna University, Haryana, India, and Sahil Vazirani, Abhishek Sharma, and Pavika Sharma from Amity University Uttar Pradesh, India, demonstrate practical applications of ML algorithms in stock market prediction. Their research spans from comparative studies of ML models to the development of hybrid approaches, aiming to optimize prediction accuracy and adaptability to market dynamics.

Additionally, Huei Wen Teng, Yu-Hsien Li, and Shang-Wen Chang from National Chiao Tung University, Taiwan, and Kartika Maulida Hindrayani, Prismahardi Aji R., Tresna Maulana Fahrudin, and Eristya Maya Safitri from UPN "Veteran" Jawa Timur, Indonesia, contribute insights into various ML algorithms and their application in empirical asset pricing and during the COVID-19 era, respectively.

Collectively, these studies underscore ongoing advancements in ML techniques, feature engineering, and data preprocessing strategies to enhance the efficacy of stock price prediction systems. Future research directions may focus on further integrating AI advancements, enhancing model interpretability, and addressing real-time prediction challenges in dynamic financial markets.

III. TECHNIQUES FOR STOCK MARKET PREDICTION

1) *Importing Libraries:* To begin the stock market prediction task, essential libraries are imported. NumPy and Pandas are utilized for numerical operations and data manipulation, enabling efficient handling of large datasets and providing functionalities for data cleaning and preprocessing. Matplotlib is used to create basic static visualizations, allowing for exploratory data analysis and preliminary visual inspection of trends. XGBoost, a robust and efficient implementation of gradient boosting algorithms, is employed for predictive modeling, known for its high performance in regression tasks and handling of missing data. Scikit-learn offers various utilities for data preprocessing, model evaluation, and hyperparameter tuning through grid search, facilitating the development of robust and accurate predictive models.
2) *Chart Drawing with Plotly:* Plotly libraries are imported to create interactive and visually appealing charts. Plotly offers a variety of interactive visualization tools, allowing for dynamic data exploration and presentation. Integrating Plotly with Jupyter notebooks enables seamless embedding of interactive plots, which enhances data analysis interpretability and presentation. Plotly's flexibility in customizing plots ensures that complex data relationships can be clearly communicated, making it a powerful tool for both analysis and reporting in research.
3) *Suppressing Warnings:* To maintain a clean and readable output, warnings related to future and deprecation issues from the Scikit-learn library are suppressed. This ensures that the output remains focused on relevant results and analysis, free from non-critical warning messages. By suppressing these warnings, we can avoid distractions and maintain the flow of analysis, which is particularly important when presenting findings in a professional or academic setting.
4) *Initializing Plotly Notebook Mode:* Plotly's notebook mode is initialized to ensure that charts are correctly displayed within Jupyter notebooks. This enhances the user experience by allowing interactive visualizations to be rendered directly in the notebook environment. By enabling this mode, researchers can interact with the data visualizations, zooming in on specific areas of interest, and gaining deeper insights from the presented data.

5) *Customizing Plotly Layout:* The default background color for all Plotly visualizations is customized, setting a transparent paper background and a lightly shaded plot background to improve visual appeal. A custom template is created to ensure consistent styling across all visualizations, contributing to a cohesive and professional presentation. This step is crucial for creating high-quality, publication-ready figures that adhere to the aesthetic standards of academic and professional publications.
6) *Installing yfinance:* The yfinance library is installed to facilitate the fetching of historical stock data from Yahoo Finance. This library provides a user-friendly interface for accessing stock data, which is crucial for building and testing predictive models in financial analysis. By leveraging yfinance, researchers can easily obtain up-to-date and comprehensive financial data, ensuring the accuracy and relevance of their predictive models.
7) *Fetching Stock Data:* Historical stock data for Apple Inc. (AAPL) is retrieved from Yahoo Finance for a specified date range. This data includes attributes such as open, high, low, close prices, and volume, which are essential for comprehensive stock market analysis and prediction. Access to accurate and detailed historical data allows researchers to perform thorough analyses, identify patterns, and build robust predictive models.

```
data.head(10)
```

Date	Open	High	Low	Close	Adj Close	Volume
2020-01-02	74.059998	75.150002	73.797501	75.087502	72.960464	135480400
2020-01-03	74.287498	75.144997	74.125000	74.357498	72.251144	146322800
2020-01-06	73.447502	74.989998	73.187500	74.949997	72.826843	118387200
2020-01-07	74.959999	75.224998	74.370003	74.597504	72.484344	108872000
2020-01-08	74.290001	76.110001	74.290001	75.797501	73.650352	132079200
2020-01-09	76.809998	77.607498	76.550003	77.407501	75.214745	170108400
2020-01-10	77.650002	78.167503	77.062500	77.582497	75.384789	140644800
2020-01-13	77.910004	79.267502	77.787498	79.239998	76.995316	121532000
2020-01-14	79.175003	79.392502	78.042503	78.169998	75.955635	161954400
2020-01-15	77.962502	78.875000	77.387497	77.834999	75.630135	121923600

Fig 1. Dataset Used

8) *Changing Date Index Format:* A function is defined to change the date index of the dataset to DateTime format, ensuring that the data is correctly formatted for subsequent analysis. This step is crucial for handling time series data accurately and efficiently. Properly formatted date indices allow for effective time-based data manipulation, such as resampling, time-based feature extraction, and alignment of multiple time series datasets.

```
data.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2020-01-02	74.059998	75.150002	73.797501	75.087502	72.960464	135480400
1	2020-01-03	74.287498	75.144997	74.125000	74.357498	72.251144	146322800
2	2020-01-06	73.447502	74.989998	73.187500	74.949997	72.826843	118387200
3	2020-01-07	74.959999	75.224998	74.370003	74.597504	72.484344	108872000
4	2020-01-08	74.290001	76.110001	74.290001	75.797501	73.650352	132079200

Fig 2. Date Index Format

9) *Installing Altair:* Altair, a declarative statistical visualization library, is installed to create interactive and meaningful visualizations. By leveraging Altair, a line chart of the closing prices can be constructed, providing a clear visual representation of stock price trends over time. Altair's intuitive syntax and powerful capabilities enable researchers to create complex visualizations with minimal code, facilitating exploratory data analysis and communication of results.

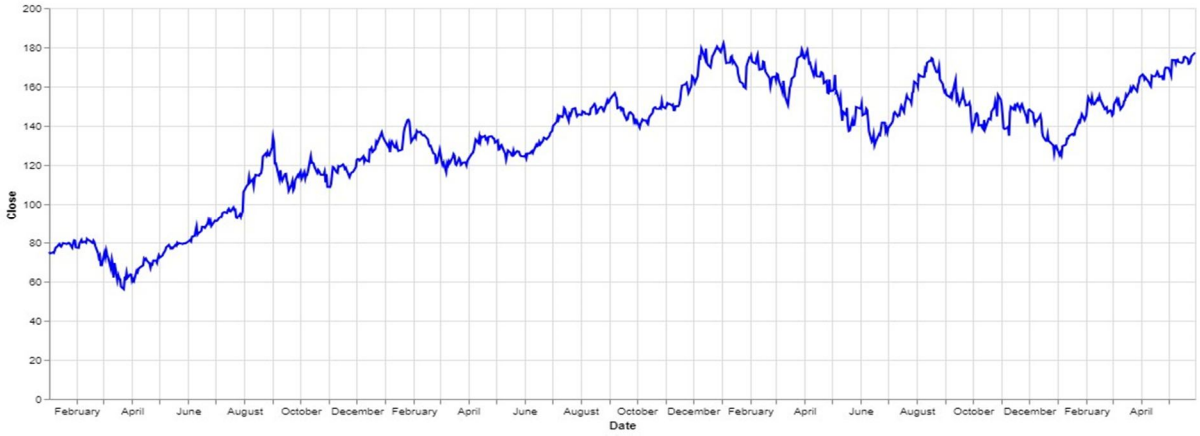


Fig 3. Graph using Altair

10) *Data Preparation:* The dataset is filtered to include data from the year 2020 onwards, ensuring that the analysis focuses on recent trends and patterns. The index is reset to maintain a sequential order, ensuring that the dataset is prepared correctly for analysis and modeling. This step is essential for aligning the dataset with the time period of interest and for ensuring the integrity of subsequent analyses and model training processes.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2020-01-02	74.059998	75.150002	73.797501	75.087502	72.960464	135480400
1	2020-01-03	74.287498	75.144997	74.125000	74.357498	72.251144	146322800
2	2020-01-06	73.447502	74.989998	73.187500	74.949997	72.826843	118387200
3	2020-01-07	74.959999	75.224998	74.370003	74.597504	72.484344	108872000
4	2020-01-08	74.290001	76.110001	74.290001	75.797501	73.650352	132079200

Fig 4. Data Preparation

11) *Plotting OHLC and Volume:* An interactive subplot is created to visualize the Open, High, Low, Close (OHLC) values along with trading volume. This provides a comprehensive view of stock price movements and trading activity over time, which is essential for technical analysis. By examining OHLC charts, researchers can identify key price levels, volatility patterns, and potential trading signals, enhancing their understanding of market dynamics.

Fig 5. Plotting OHLC and Volume

12) *Seasonal Decomposition:* Seasonal decomposition is performed on the closing price data to identify underlying trends and seasonal patterns. This analysis helps in understanding the cyclical behavior of stock prices and is useful for making informed predictions. By decomposing the time series into trend, seasonal, and residual components, researchers can isolate and analyze these effects separately, leading to more accurate and interpretable models.

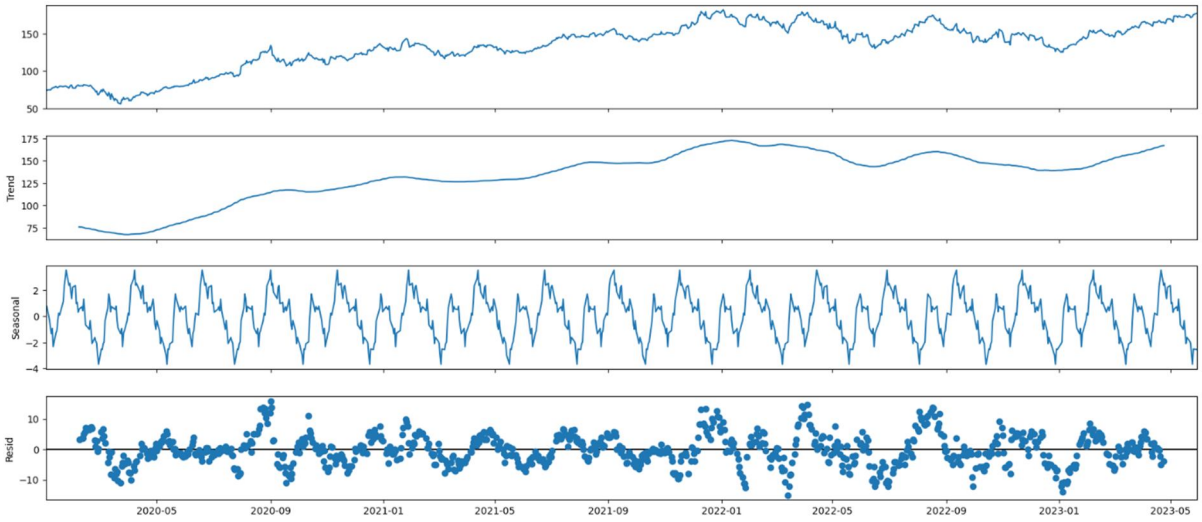


Fig 6. Seasonal Decomposition

13) *Calculating Moving Averages:* Exponential Moving Average (EMA) and Simple Moving Averages (SMA) for various periods are calculated. These moving averages are commonly used in technical analysis to smooth out price data and identify trends. Moving averages help in filtering out noise from the price data, providing clearer signals for trend direction and potential reversals, which are crucial for making trading decisions.
14) *Plotting Moving Averages:* The calculated moving averages are plotted along with the closing price to visually inspect the trends and signals. This helps in understanding the market momentum and potential reversal points. By overlaying moving averages on the price chart, researchers can visually identify crossovers and divergences, which are key indicators of trend strength and potential changes in market direction.

Fig 7. Plotting Moving Averages

15) *Calculating Relative Strength Index (RSI):* A function is defined to calculate the Relative Strength Index (RSI), a momentum oscillator that measures the speed and change of price movements. RSI is used to identify overbought or oversold conditions in the stock market. By quantifying the magnitude of recent price changes, RSI provides insights into the strength of price trends and potential reversal points, aiding in the development of trading strategies.
16) *Plotting RSI:* The RSI is plotted to visualize periods where the stock might be overbought or oversold. This helps in making trading decisions based on the momentum and relative strength of the stock price movements. Visualizing RSI alongside price data allows researchers to identify potential entry and exit points, enhancing the effectiveness of their trading strategies.

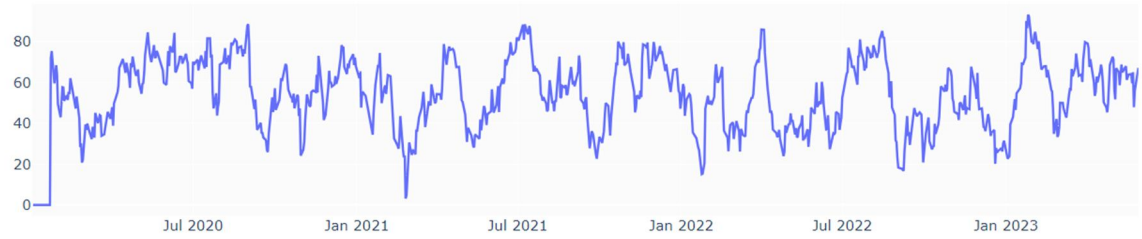


Fig 8. Plotting RSI

17) *Calculating MACD and Signal Line:* The Moving Average Convergence Divergence (MACD) and its signal line are calculated. MACD is a trend-following momentum indicator that shows the relationship between two moving averages of a stock's price. By calculating the difference between the short-term and long-term moving averages, MACD helps in identifying the direction and strength of the trend, as well as potential turning points in the market.
18) *Plotting MACD and Signal Line:* The MACD and its signal line are plotted along with the stock's closing price. This helps in identifying potential buy or sell signals based on the convergence or divergence of these lines. By visualizing MACD and the signal line, researchers can detect bullish or bearish momentum shifts, enhancing their ability to predict future price movements and make informed trading decisions.

Fig 9. Plotting MACD and Signal Line

19) *Splitting Data into Train, Validation, and Test Sets:* The dataset is split into training, validation, and test sets to ensure that the model can be trained, tuned, and evaluated effectively. This step is crucial for preventing overfitting and ensuring the model's generalizability to unseen data. By creating separate datasets for training, validation, and testing, researchers can objectively assess the model's performance and fine-tune it to achieve optimal predictive accuracy.

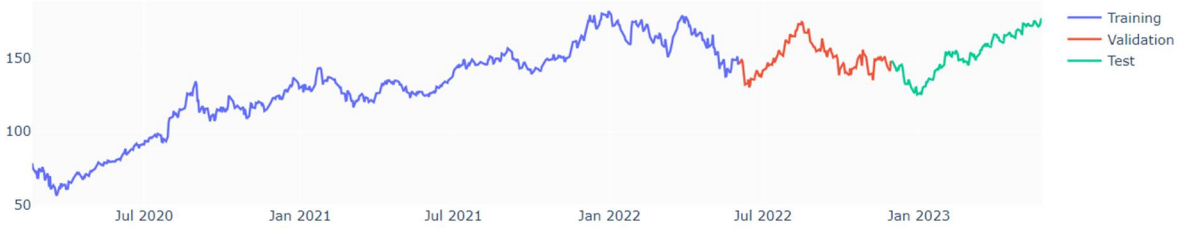


Fig 10. Splitting Data into Train, Validation, and Test Sets

20) *Dropping Unwanted Columns:* Unnecessary columns are dropped from the datasets to focus on the relevant features for modeling. This step simplifies the data and enhances the model's performance by eliminating noise. By removing irrelevant or redundant features, researchers can improve the computational efficiency of the model and reduce the risk of overfitting, leading to more robust and interpretable predictions.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 577 entries, 0 to 576
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Adj Close    577 non-null    float64
 1   EMA_9        577 non-null    float64
 2   SMA_5        577 non-null    float64
 3   SMA_10       577 non-null    float64
 4   SMA_15       577 non-null    float64
 5   SMA_30       577 non-null    float64
 6   RSI          577 non-null    float64
 7   MACD         577 non-null    float64
 8   MACD_signal  577 non-null    float64
dtypes: float64(9)
memory usage: 40.7 KB
```

Fig 11. Dropping Unwanted Columns

21) *Training the XGBRF Regressor Model:* An XGBRF Regressor model is trained using the specified hyperparameters. Grid search and randomized search are employed to find the best parameters, optimizing the model's performance on the training and validation data. This process involves systematically testing different combinations of hyperparameters to identify the optimal configuration that maximizes the model's predictive accuracy and minimizes errors.

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
Best params: {'random_state': 42, 'n_estimators': 400, 'max_depth': 8, 'learning_rate': 0.05, 'gamma': 0.005}
Best validation score = -294.28530289815524
CPU times: total: 4min 59s
Wall time: 48.4 s
```

```
CPU times: total: 7.34 s
Wall time: 1.1 s
```

```
XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=0.005, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.05, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=8, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=400, n_jobs=None,
             num_parallel_tree=None, random_state=42, ...)
```

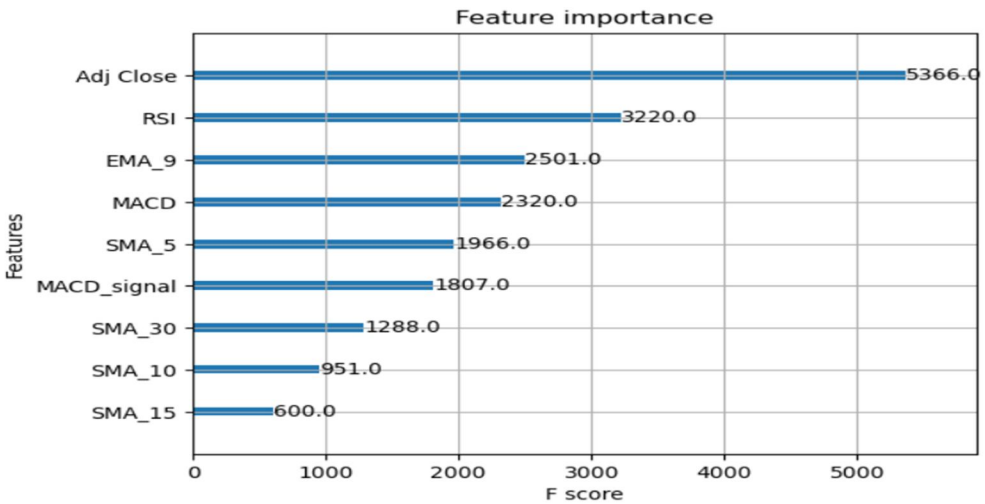


Fig 12. Training the XGBRF Regressor Model

22) *Evaluating the Model:* The model's performance is evaluated by predicting the stock prices on the test set and comparing them with the actual values. Mean squared error is calculated to quantify the prediction accuracy. The predicted and actual prices are plotted to visually inspect the model's performance. This comprehensive evaluation provides insights into the model's strengths and weaknesses, guiding further refinement and improvement.

```
y_true = [148.30999756 147.80999756 146.63000488 142.91000366 140.94000244]
y_pred = [148.96973 149.33739 149.15268 147.57622 145.24706]

mean_squared_error = 9.361054727379365
```

23) *Comparing Predicted Data with Actual Data:* The predicted stock prices are compared with the actual prices to assess the model's accuracy. An interactive plot is created to visualize the comparison, highlighting areas where the model performs well and where it may need improvement. This step is crucial for validating the model's effectiveness and reliability, ensuring that the predictions are accurate and actionable. By thoroughly comparing predicted and actual data, researchers can identify any discrepancies and make necessary adjustments to enhance the model's predictive capability.

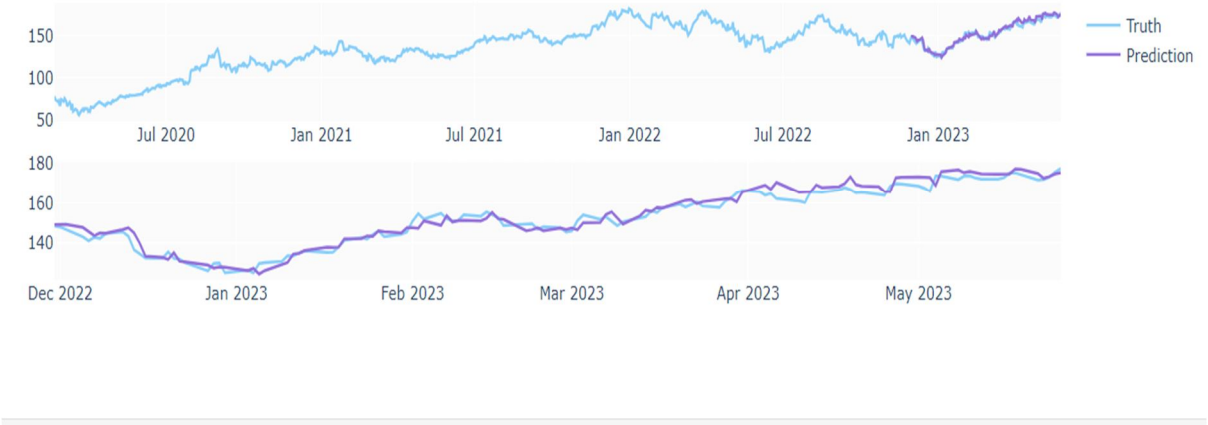


Fig 13. Comparing Predicted Data with Actual Data

IV. COMPARISON WITH OTHER MODELS

When compared to traditional statistical models such as ARIMA or simple linear regression, the XGBoost model offers several advantages:

1) *Handling Non-Linearity:* Unlike linear models, XGBoost can capture non-linear relationships within the data, making it more suitable for the inherently volatile and complex nature of stock prices.
2) *Feature Importance:* XGBoost provides insights into feature importance, helping in understanding which indicators most significantly impact the stock price predictions.
3) *Speed and Performance:* XGBoost is optimized for speed and performance, especially with large datasets, making it an efficient choice for real-time stock market prediction.
4) *Regularization:* XGBoost includes built-in regularization techniques that prevent overfitting, enhancing the model's generalizability to unseen data.

However, it is essential to note that while XGBoost performs exceptionally well in many scenarios, it also requires careful tuning of hyperparameters, which can be computationally intensive. Other machine learning models like LSTM (Long Short-Term Memory) networks could potentially offer better performance for sequential data due to their capability to retain long-term dependencies, which is a significant factor in time series analysis.

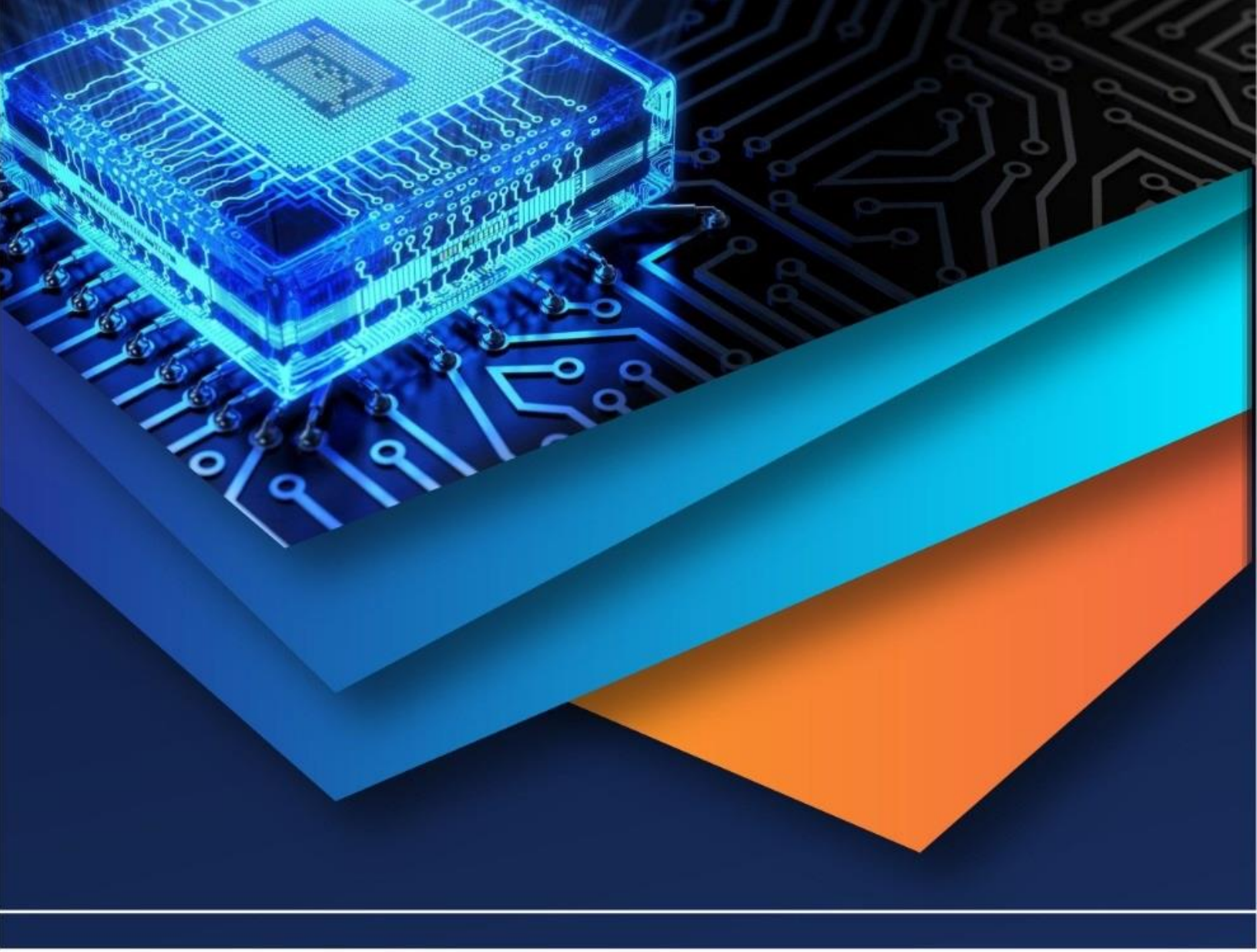
V. CONCLUSION

The stock market prediction model developed in this research leverages advanced machine learning techniques, particularly the XGBoost regressor, to predict future stock prices. XGBoost has been chosen for its robustness, efficiency, and superior performance in handling complex datasets with multiple features and time dependencies. The model incorporates various technical indicators such as moving averages, RSI, and MACD, providing a comprehensive analysis of stock price movements.

In conclusion, the use of XGBoost for stock market prediction demonstrates significant potential due to its advanced capabilities in handling complex and non-linear relationships in the data. While this model shows promising results, continuous refinement and comparison with other sophisticated models like LSTM or hybrid approaches can further enhance predictive accuracy and reliability. The choice of model should always be aligned with the specific requirements and constraints of the prediction task at hand.

REFERENCES

[1] Gourav Bathla, Stock Price prediction using LSTM and SVR, January 2021, DOI: 10.1109/PDGC50313.2020.9315800.
[2] Srinath Ravikumar, Prediction of Stock Prices using Machine Learning (Regression, Classification) Algorithms, June 2020.
[3] Yaohu lin, Stock Trend Prediction Using Candlestick Charting and Ensemble Machine Learning Techniques With a Novelty Feature Engineering Scheme, May 2021, DOI: 10.1109/ACCESS.2021.3096825
[4] Sondo Kim, Predicting the Direction of US Stock Prices Using Effective Transfer Entropy and Machine Learning Techniques, June 2020, DOI: 10.1109/ACCESS.2020.3002174
[5] Stock Market Forecasting Using Deep Learning and Technical Analysis: A Systematic Review, AUDELIANO WOLIAN LI, October 2020, DOI: 10.1109/ACCESS.2020.3030226
[6] DONGHWAN SONG1, Forecasting Stock Market Indices Using Padding-Based Fourier Transform Denoising and Time Series Deep Learning Models, June 2021, DOI: 10.1109/ACCESS.2021.3086537
[7] Parag P. Kadu, Comparative Study of Stock Price Prediction using Machine Learning, April 2020, DOI: 10.1109/EICT48899.2019.9068850
[8] Shoban Dinesh, Prediction of Trends in Stock Market using Moving Averages and Machine Learning, April 2021, DOI: 10.1109/I2CT51068.2021.9418097
[9] S. Nithya Tanvi Nishitha, Stock Price Prognosticator using Machine Learning Techniques, December 2020, DOI: 10.1109/ICECA49313.2020.929764
[10] Vaibhav Gaur, Revitalizing Stock Predictions with Machine Learning Algorithms – An Empirical Study, February 2021, DOI: 10.1109/INDICON49873.2020.9342571
[11] Huei Wen Teng, Machine Learning in Empirical Asset Pricing Models, December 2020, DOI: 10.1109/ICPAI51961.2020.00030
[12] Kartika Maulida Hindrayani, Indonesian Stock Price Prediction including Covid19 Era Using Decision Tree Regression, January 2021, DOI: 10.1109/ISRITI51436.2020.9315484
[13] Sahil Vazirani, Analysis of various machine learning algorithm and hybrid model for stock market prediction using Python, December 2020, DOI: 10.1109/ICSTCEE49637.2020.9276859



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY