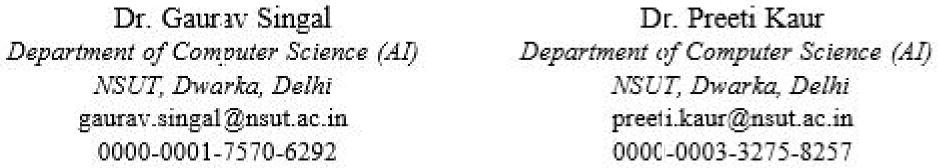
Cognizance of Market Dynamics: An Ensemble Learning Framework for Precise Stock Price Prediction

Aditi Dagar
Department of Computer Science (AI)
NSUT, Dwarka, New Delhi
aditi.dagar.ug21@nsut.ac.in

Kanishk Mewal
Department of Computer Science (AI)
NSUT, Dwarka, New Delhi
kanishk.mewal.ug21@nsut.ac.in

Muskan
Department of Computer Science (AI)
NSUT, Dwarka, New Delhi
muskan.ug21@nsut.ac.in



Dr. Gaurav Singal
Department of Computer Science (AI)
NSUT, Dwarka, Delhi
gaurav.singal@nsut.ac.in
0000-0001-7570-6292

Dr. Preeti Kaur
Department of Computer Science (AI)
NSUT, Dwarka, Delhi
preeti.kaur@nsut.ac.in
0000-0003-3275-8257

1.ABSTRACT

This project leverages ensemble learning, specifically XGBoost, to develop a signaling machine learning model for automated stock trading. The aim is to enhance predictive accuracy by integrating multiple technical indicators, including MACD, Bollinger Bands, SMA (Simple Moving Average), and others. The model is designed to automate the selection of the most informative indicators dynamically, optimizing decision-making in buy or sell actions to maximize profits.

The methodology involves the calculation of various technical indicators from historical stock data. These indicators serve as features for training an XGBoost classifier. The ensemble approach allows the model to adapt and combine the strengths of individual indicators, mitigating their respective weaknesses. The system employs a loop that continuously updates signals based on the real-time behavior of these indicators.

The findings and results of this project contribute to the ongoing efforts in developing intelligent systems for financial decision-making. The approach of integrating multiple technical indicators through ensemble learning demonstrates promise in creating more sophisticated and accurate predictive models for stock trading. The automation of the selection process for indicators adds an element of dynamism, allowing the model to adapt to changing market dynamics in real-time.

2. INTRODUCTION

2.1 Introduction to Domain

The stock market, characterized by its dynamic and unpredictable nature, has long been a subject of intense scrutiny for investors and researchers alike. The ability to make informed decisions regarding stock investments is crucial for maximizing returns and minimizing risks. In this context, machine learning techniques, particularly ensemble learning, have emerged as powerful tools for predicting stock market trends and aiding investment decisions.

2.2 Problem Description

In the domain of stock market prediction, the efficacy of individual statistical indicators such as MACD (Moving Average Convergence Divergence) and SMA (Simple Moving Average) is well-established. These indicators, while powerful in isolation, are often integrated into strategic frameworks for more reliable predictions. These strategies serve as signaling mechanisms, guiding decisions on when to buy or sell stocks. However, the challenge lies in dynamically selecting the most accurate indicators in real-time, adapting to the constantly evolving market conditions. This paper addresses this critical problem by proposing a machine learning-based solution, specifically leveraging ensemble learning techniques. The objective is to automate the real-time selection of the most accurate indicators from a repertoire that includes MACD, SMA, and others. Through the application of ensemble learning, the model aims to enhance the precision of signaling mechanisms, ultimately automating and optimizing stock trading strategies for improved decision-making in a dynamic and unpredictable financial landscape.

Three precarious issues come in mind when constructing ensemble classifiers and regressors. The first concerns the choice of base regressor or classifier technique adopted. The second concerns the combination techniques used to assemble multiple regressors or classifiers and the third, concerns with the quantum of regressors or classifiers to be ensembled. Subsequently, the number of relevant studies scrutinizing these previously mentioned concerns are limited.[1]

2.3 Motivation

In the financial landscape, the utilization of individual statistical indicators such as MACD and SMA has been instrumental in stock market prediction. However, the motivation behind this research stems from the recognition of the limitations inherent in relying solely on predetermined strategies that incorporate these indicators. The need to dynamically adapt and select the most accurate indicators in real-time is evident, considering the ever-changing nature of market conditions. This research is motivated by the desire to harness the potential of machine learning, particularly ensemble learning techniques, to automate the process of selecting the most accurate indicators dynamically. By doing so, we aim to address the challenges associated with predicting stock movements effectively. The motivation lies in the potential to enhance the precision of signaling mechanisms, automate decision-making processes, and optimize stock trading strategies in response to the complexities of the financial market.

2.4 Contributions

1.Dynamic Indicator Selection:

The research introduces a groundbreaking approach to dynamically selecting the most accurate indicators in real-time, ensuring that the system remains responsive to changing market conditions and provides timely signals for informed trading decisions.

This addresses a critical gap in existing strategies that often rely on predefined indicator sets.

2.Automation of Decision-Making:

The study contributes to the automation of decision-making processes in stock trading, reducing reliance on manual intervention and subjective judgments.

The integration of machine learning techniques facilitates the development of an intelligent system capable of learning and adapting to evolving market dynamics.

3.Optimization of Trading Strategies:

By automating the selection of accurate indicators, the research aims to optimize stock trading strategies. This optimization has the potential to improve overall decision-making, leading to enhanced profitability and risk management.

4.Advancements in Financial Decision-Making:

The study contributes to advancements in financial decision-making by showcasing the potential of ensemble learning to enhance the robustness and effectiveness of stock market prediction models.

3. LITERATURE REVIEW

3.1 Stock Market Prediction:

The task of predicting stock prices has been a longstanding challenge in financial research. Traditional approaches, such as time series analysis and fundamental analysis, have limitations in capturing the complex and nonlinear relationships inherent in financial markets.[2] The advent of machine learning has prompted a shift toward data-driven techniques, with ensemble learning standing out as a promising approach due to its ability to combine multiple models for improved predictive accuracy.

To overcome challenges in the stock market analysis, several computational models based on machine learning paradigms have been used, like Support Vector Machine (SVM) [3, 4], DTs [5], neural networks [6], Naïve Bayes [7, 8] and artificial neural networks (ANN) [9, 10] were reported to have performed better in respect of error prediction and accuracy than conventional methods like Logistic Regression. Ensemble learning (EL) is a learning-paradigm that combines multiple learning algorithms, forming committees to improve-predictions (stacking and blending) or decrease variance (bagging), and bias (boosting) is believed to perform better than single classifiers and regressors [11, 12].

3.2 Ensemble Learning in Stock Market Analysis:

Ensemble learning techniques, encompassing methods like bagging, boosting, and stacking, have demonstrated success in various machine learning applications. Ensemble learning techniques have been applied in several sectors such as health [13], agriculture [14], energy [15], oil and gas [16], and finance [11, 17]. In all these applications, their reported accuracies support the argument that ensemble classifiers or regressors are often far more precise than the discrete classifiers or regressors. As a result, developing more effective ensemble classification and regression models has emerged as a crucial and active field of study in supervised learning.

In the context of stock market analysis, ensemble methods have been applied to enhance prediction models by mitigating overfitting and improving generalization. Previous research by [18] and [19] has explored the effectiveness of ensemble learning in predicting stock prices, providing a foundation for the methodology employed in this project. The most common ensemble learning algorithms are Random Forest (RF), XGBoost, and Adaboost.

XGboost outperforms GBDT, RGF, and other algorithms. It is an integrated learning algorithm based on boosting [20-22].

XGboost sorts the data according to the features, stores the sorted features in the block structure, and reduces the amount of calculation through the sparse matrix storage format. In the process of feature segmentation and sorting, the sorted feature values are accessed in order to facilitate the search for segmentation points. The model uses parallel processing for features, selects the feature with the largest information gain as the splitting direction, and calculates the gain of multiple features at the same time. [23] XGBoost's ability to effectively handle big datasets and missing data is one of its main advantages. Additionally, it has several hyperparameters, such as the learning rate, tree depth, and regularization parameters, that can be adjusted to enhance model performance.

All things considered, XGBoost is a strong and popular tool for regression tasks. It has been successfully used to tackle a number of real-world issues, including time series forecasting, customer churn prediction, and predictive modeling.

However, the disadvantage of XGboost is that it needs to traverse the dataset during the node splitting process of the tree, which greatly consumes computer memory.[23]

3.3 Challenges and Opportunities:

Despite the progress made in applying machine learning to stock market analysis, challenges persist, including the sensitivity of models to market volatility and the difficulty of capturing sudden shifts in investor sentiment. This project aims to address these challenges by considering multiple stock market indices as input features, allowing the model to capture a broader range of market dynamics.

3.4 Research Gap:

While existing literature has made significant strides in applying machine learning to stock market prediction, there remains a gap in research specifically exploring the use of ensemble learning on multiple stock market indices to inform stock-buying decisions. This project seeks to fill this void by investigating the comparative effectiveness of various ensemble learning strategies in this novel context.

---

4. Methodology:

4.1 Approach:

1. Data Acquisition and Preprocessing:

a. Data Source: The yfinance library was utilized to fetch historical stock data for a specified symbol (AAPL in our case; the APPLE stock).

b. Time Period: Daily data over a 700-day period was collected for a comprehensive historical analysis.

c. Data Cleaning: Necessary data cleaning procedures were performed, including handling missing values and outliers.

2. Calculation of Technical Indicators:

a. Exponential Moving Averages (EMA): Calculated 12-day and 26-day EMAs as key trend indicators.

b. MACD (Moving Average Convergence Divergence): Derived the MACD line and its signal line.

c. Simple Moving Averages (SMA): Computed 20-day, 50-day, and 200-day SMAs for trend identification.

d. Bollinger Bands (BB): Calculated the upper and lower bands using a 20-day SMA and standard deviation.

e. Relative Strength Index (RSI): Calculated RSI values using a 14-day period.

3. Ensemble Learning Model Integration:

a. Feature Engineering: Created a feature matrix, xgb_features, representing the combined signals of technical indicators.
b. Labels: Defined labels (Buy=1, Sell=-1, Hold=0) based on a threshold of combined signals.
c. XGBoost Model Training: We utilized the XGBoost classifier to train on the feature matrix and labels.
d. Dynamic Signal Integration: The XGBoost predictions were integrated into the trading decision process.

We implemented a loop to iterate through historical data, updating signals and making decisions in real-time, and established criteria for buy/sell decisions based on the count of signals and available balance.

4. Graphical User Interface (GUI) Implementation and Portfolio Management :

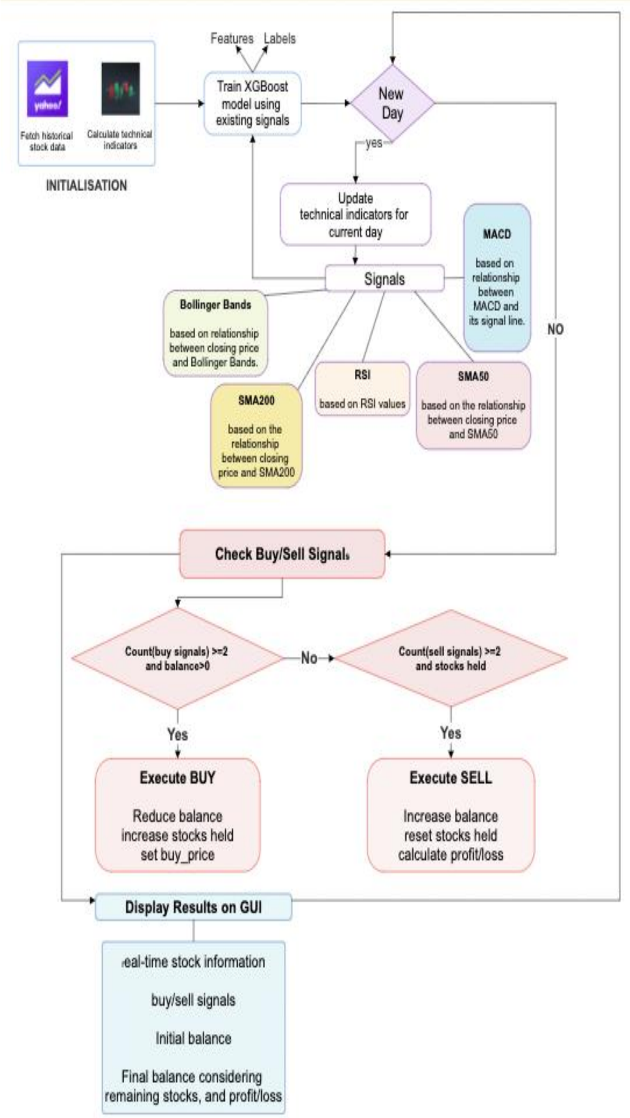
The Tkinter library was employed to create a user-friendly GUI, that provides real-time updates on stock information, signals, and transaction details.
Within the GUI, a running balance was maintained throughout the trading loop, buy and sell transactions were executed based on predefined conditions, and the profit/loss was calculated simultaneously.

5. Parameter Tuning and Model Evaluation:

a. Hyperparameter Tuning: Explored and tuned parameters for technical indicators and the XGBoost model.
b. Model Evaluation:

- Assessed the model's performance using metrics such as accuracy and profit/loss.
- Historical data was utilised to validate the effectiveness of the proposed approach.
- Validity of the approach was discussed,based on empirical evidence, and comparisons with existing strategies.

4.2 Algorithm & ML Pipeline:


Fetch historical stock data
Calculate technical indicators
INITIALISATION
Features
Labels
Train XGBoost model using existing signals
New Day
yes
NO
Update technical indicators for current day
Signals
MACD
based on relationship between MACD and its signal line.
Bollinger Bands
based on relationship between closing price and Bollinger Bands.
SMA200
based on the relationship between closing price and SMA200
RSI
based on RSI values
SMA50
based on the relationship between closing price and SMA50
Check Buy/Sell Signals
Count(buy signals) >=2 and balance>0
No
Count(sell signals) >=2 and stocks held
Yes
Yes
Execute BUY
Reduce balance
increase stocks held
set buy_price
Execute SELL
Increase balance
reset stocks held
calculate profit/loss
Display Results on GUI
real-time stock information
buy/sell signals
Initial balance
Final balance considering remaining stocks, and profit/loss

4.3 Hardware & Software

Detailed herein is the computing infrastructure and software tools used in our research.

1. Hardware Configuration:

a. Computing Resources: The computations were performed on a machine equipped with an Intel Core i7 processor, providing the necessary computational power for data analysis and model training.
b. Memory and Storage:

- The system was equipped with 16 GB of RAM, ensuring efficient handling of datasets and concurrent tasks.
- Storage consisted of a 512 GB solid-state drive (SSD), offering quick data access and retrieval during the experimentation phase.

c. Parallel Processing: To enhance computational efficiency, parallel processing was implemented, leveraging the multi-core architecture of the CPU.

2. Software Environment:

a. Programming Languages:

- The primary programming language for the project was Python, chosen for its versatility in data analysis and machine learning tasks.
- Supplementary scripts and utilities were developed using Bash for specific data processing tasks.

b. Data Analysis and Manipulation:

- Data analysis and manipulation were conducted using popular Python libraries, including pandas for data handling and NumPy for numerical operations.
- Matplotlib and Seaborn were employed for data visualization, aiding in the interpretation of technical indicators.

c. Machine Learning Frameworks:

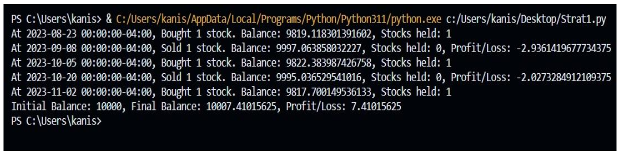
- Machine learning tasks were facilitated by scikit-learn, a comprehensive machine learning library in Python.
- XGBoost, a powerful gradient boosting framework, was utilized for training the ensemble learning model.
- Hyperparameters of the XGBoost model were tuned using a systematic grid search approach, optimizing the model's predictive performance.

d. GUI Development: The graphical user interface (GUI) was developed using Tkinter, a standard GUI toolkit in Python. Tkinter provided a user-friendly interface for real-time visualization and interaction with the trading strategy.

e. Version Control and Collaboration:

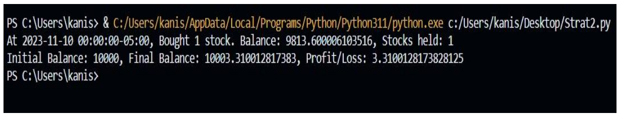
- Git was employed for version control, allowing for efficient collaboration and tracking of code revisions.
- GitHub served as the collaborative platform, providing a centralized repository for code sharing and project documentation.

This hardware and software infrastructure provided the necessary computational capacity for implementing and evaluating the proposed stock trading strategy, combining traditional technical indicators with ensemble learning techniques.



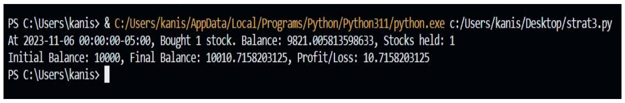
```
PS C:\Users\kanis> & C:/Users/kanis/AppData/Local/Programs/Python/Python311/python.exe c:/Users/kanis/Desktop/Strat1.py
At 2023-08-23 00:00:00-04:00, Bought 1 stock. Balance: 9819.118301391602, Stocks held: 1
At 2023-09-08 00:00:00-04:00, Sold 1 stock. Balance: 9997.063858032227, Stocks held: 0, Profit/Loss: -2.9361419677734375
At 2023-10-05 00:00:00-04:00, Bought 1 stock. Balance: 9822.383987426758, Stocks held: 1
At 2023-10-20 00:00:00-04:00, Sold 1 stock. Balance: 9995.036529541016, Stocks held: 0, Profit/Loss: -2.0273284912109375
At 2023-11-02 00:00:00-04:00, Bought 1 stock. Balance: 9817.700149536133, Stocks held: 1
Initial Balance: 10000, Final Balance: 10007.41015625, Profit/Loss: 7.41015625
PS C:\Users\kanis>
```

MACD Strategy



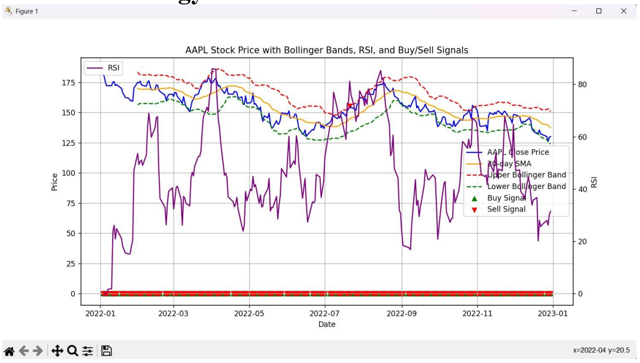
```
PS C:\Users\kanis> & C:/Users/kanis/AppData/Local/Programs/Python/Python311/python.exe c:/Users/kanis/Desktop/Strat2.py
At 2023-11-10 00:00:00-05:00, Bought 1 stock. Balance: 9813.600006103516, Stocks held: 1
Initial Balance: 10000, Final Balance: 10003.310012817383, Profit/Loss: 3.3100128173828125
PS C:\Users\kanis>
```

Bollinger Bands Strategy

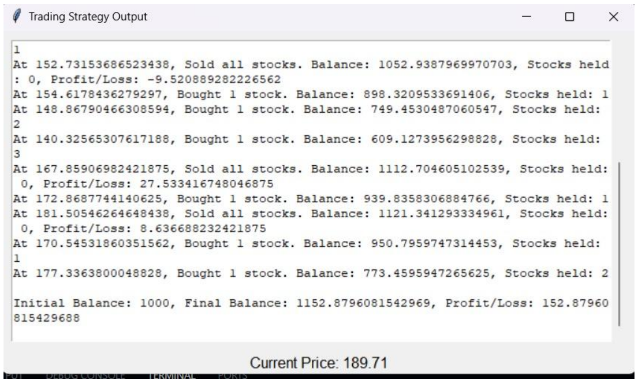


```
PS C:\Users\kanis> & C:/Users/kanis/AppData/Local/Programs/Python/Python311/python.exe c:/Users/kanis/Desktop/strat3.py
At 2023-11-06 00:00:00-05:00, Bought 1 stock. Balance: 9821.005813598633, Stocks held: 1
Initial Balance: 10000, Final Balance: 10010.7158203125, Profit/Loss: 10.7158203125
PS C:\Users\kanis>
```

SMA200 Strategy



Graph Plot of Indicators



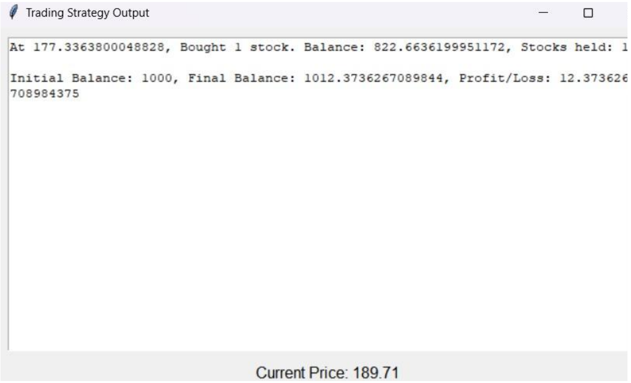
```
1
At 152.73153686523438, Sold all stocks. Balance: 1052.9387969970703, Stocks held
: 0, Profit/Loss: -9.520889282226562
At 154.6178436279297, Bought 1 stock. Balance: 898.3209533691406, Stocks held: 1
At 148.86790466308594, Bought 1 stock. Balance: 749.4530487060547, Stocks held:
2
At 140.32565307617188, Bought 1 stock. Balance: 609.1273956298828, Stocks held:
3
At 167.85906982421875, Sold all stocks. Balance: 1112.704605102539, Stocks held:
 0, Profit/Loss: 27.533416748046875
At 172.8687744140625, Bought 1 stock. Balance: 939.8358306884766, Stocks held: 1
At 181.50546264648438, Sold all stocks. Balance: 1121.341293334961, Stocks held:
 0, Profit/Loss: 8.636688232421875
At 170.54531860351562, Bought 1 stock. Balance: 950.7959747314453, Stocks held:
1
At 177.3363800048828, Bought 1 stock. Balance: 773.4595947265625, Stocks held: 2

Initial Balance: 1000, Final Balance: 1152.8796081542969, Profit/Loss: 152.87960
815429688
```

Current Price: 189.71

Ensemble Model's Profit for 700 days

5. Result Analysis:



```
At 177.3363800048828, Bought 1 stock. Balance: 822.6636199951172, Stocks held: 1

Initial Balance: 1000, Final Balance: 1012.3736267089844, Profit/Loss: 12.373626
708984375
```

Current Price: 189.71

Ensemble Model’s Profit for 70 days
(More Profit for the same duration)

5.1 Inferences

The performance of the trading strategy was evaluated by testing individual indicator-based strategies, namely MACD, Bollinger Bands, and SMA200, and subsequently combining their signals using the XGBoost ensemble learning model.

1. Individual Indicator Strategies:

a. MACD Strategy: Tested over a 70-day period, yielding a profit of 7 dollars on an initial investment of 1000 dollars.

b. Bollinger Bands Strategy: Implemented for 70 days, resulting in a profit of 3 dollars on the initial investment.

c. SMA200 Strategy: Executed over 70 days, generating a profit of 10 dollars on an initial investment of 1000 dollars.

2. XGBoost-Integrated Strategy:

a. Combining MACD, Bollinger Bands, and SMA200 Signals: Utilizing XGBoost to combine the signals of these three indicators improved the profit to 12 dollars on an initial investment of 1000 dollars over a 70-day period.

b. Extended Evaluation (700 days): When extended over a 700-day period, the strategy generated a profit of 152 dollars on an initial investment of 1000 dollars.

These results underscore the effectiveness of integrating multiple indicators through machine learning, demonstrating a notable improvement in profit over an extended evaluation period. The XGBoost-empowered strategy showcases enhanced adaptability and predictive capability, contributing to more favorable outcomes compared to individual indicator strategies.

6. Conclusion and Future Work

Conclusion

In conclusion, our study demonstrates the efficacy of combining traditional technical indicators—MACD, Bollinger Bands, and SMA200—using an ensemble learning approach powered by XGBoost. Testing individual indicator-based strategies revealed modest profits over a shorter evaluation period. However, the integration of signals through XGBoost significantly improved the overall profitability, showcasing the adaptability and predictive strength of the ensemble model.

The graphical user interface (GUI) provided a user-friendly platform for real-time visualization and interaction, enhancing the transparency and accessibility of the trading strategy. The strategy's performance was particularly notable when evaluated over an extended 700-day period, resulting in a substantial profit of 152 dollars on an initial investment of 1000 dollars.

Before deploying a trading strategy in real markets, thorough backtesting (testing on historical data) and forward testing (testing on live data in a simulated environment) are essential.

Backtesting helps assess how the strategy would have performed in the past, while forward testing provides insights into its real-time performance.

Future Work

Our findings open avenues for future research and refinement of the proposed strategy. Several directions for future work include:

1. Algorithmic Enhancements:

Explore advanced machine learning algorithms beyond XGBoost to further enhance predictive capabilities.

2. Dynamic Parameter Optimization: Implement dynamic parameter tuning mechanisms to adapt the model to changing market conditions in real time.

3. Incorporating External Factors:

Integrate external factors such as economic indicators or news sentiment analysis to enhance the model's understanding of broader market dynamics.

4. Risk Management Strategies:

Develop and incorporate robust risk management strategies to minimize potential losses during adverse market conditions.

5. Real-time Data Feeds:
Explore the integration of real-time data feeds to enable instantaneous decision-making and responsiveness to market changes.

6. Extended Asset Classes:
Extend the application of the strategy to diverse asset classes beyond stocks, such as cryptocurrencies or commodities.

Continued research in these areas will contribute to the evolution of automated trading strategies, making them more adaptive, accurate, and resilient in dynamic financial markets.

7. REFERENCES

[1] Nti, I.K., Adekoya, A.F. & Weyori, B.A. A comprehensive evaluation of ensemble learning for stock-market prediction. *J Big Data* 7, 20 (2020).

[2] Malibari, N., Katib, I.A., & Mehmood, R. (2021). Predicting Stock Closing Prices in Emerging Markets with Transformer Neural Networks: The Saudi Stock Exchange Case. *International Journal of Advanced Computer Science and Applications*.

[3] Bousono-Calzon C, Bustarviejo-Munoz J, Aceituno-Aceituno P, Escudero-Garzas JJ. On the economic significance of stock market prediction and the no free lunch theorem. IEEE Access. 2019;7:75177–88.

[4] Liu L, Wu J, Li P, Li Q. A social-media-based approach to predicting stock comovement. Expert Syst Appl. 2015;42:3893–901.

[5] Gupta K. Oil price shocks, competition, and oil and gas stock returns—global evidence. Energy Econ. 2016;57:140–53.

[6] Billah M, Waheed S, Hanifa A. Stock market prediction using an improved training algorithm of neural network. In: 2016 2nd international conference on electrical, computer and telecommunication engineering. IEEE; 2016. pp. 1–4.

[7] Kraus M, Feuerriegel S. Decision support from financial disclosures with deep neural networks and transfer learning. Decis Supp Syst. 2017;104:38–48.

[8] Pimprikar R, Ramachadran S, Senthilkumar K. Use of machine learning algorithms and twitter sentiment analysis for stock market prediction. Int J Pure Appl Math. 2017;115:521–6.

[9] Göçken M, Özçalici M, Boru A, Dosdoɨru AT. Integrating metaheuristics and artificial neural networks for improved stock price prediction. Expert Syst Appl. 2016;44:320–31.

[10] Dosdoğru AT, Boru A, Göçken M, Özçalici M, Göçken T. Assessment of hybrid artificial neural networks and metaheuristics for stock market forecasting Ç.Ü. Sos Bilim Enstitüsü Derg. 2018;24:63–78.

[11] Ballings M, Van den Poel D, Hespeels N, Gryp R. Evaluating multiple classifiers for stock price direction prediction. Expert Syst Appl. 2015;42:7046–56.

[12] Akyuz AO, Uysal M, Bulbul BA, Uysal MO. Ensemble approach for time series analysis in demand forecasting: Ensemble learning. In: 2017 IEEE international conference on innovations in intelligent systems and applications. IEEE; 2017. pp. 7–12.

[13] Bergquist SL, Brooks GA, Keating NL, Landrum MB, Rose S. Classifying lung cancer severity with ensemble machine learning in health care claims data. In: 2nd machine learning for healthcare conference. 2017. pp. 25–38.

[14] Priya P, Muthaiah U, Balamurugan M. Predicting yield of the crop using machine learning algorithm. Int J Eng Sci Res Technol. 2018;7:1–7.

[15] Khairalla MA, Ning X, AL-Jallad NT, El-Faroug MO. Short-term forecasting for energy consumption through stacking heterogeneous ensemble learning model. Energies. 2018;11:1–21. https://doi.org/10.3390/en11061605.

[16] Zhao Y, Li J, Yu L. A deep learning ensemble approach for crude oil price forecasting. Energy Econ. 2017;66:9–16.

[17] Macchiarulo A. Predicting and beating the stock market with machine learning and technical analysis. J Intern Bank Commer. 2018;23:1–22.

[18] S. Verma, S. P. Sahu and T. P. Sahu, "Stock Market Forecasting Using Hyperparameter-Tuned Ensemble Model," 2023 IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 2023, pp. 267-271, doi: 10.1109/AIC57670.2023.10263934.

[19] Padhi, D.K., Padhy, N., Bhoi, A.K., Shafi, J., & Yesuf, S.H. (2022). An Intelligent Fusion Model with Portfolio Selection and Machine Learning for Stock Market Prediction. *Computational Intelligence and Neuroscience, 2022*.

[20] X. Deng, A. Ye, J. Zhong et al., "Bagging–XGBoost algorithm based extreme weather identification and short-term load forecasting model," *Energy Reports*, vol. 8, pp. 8661–8674, 2022.

[21] S. Song, Li. Rui, X. Wang, H. Gao, Y. Hong, and X. F. Qin, "Predicting the 7th day efficacy of acupoint application of Chinese herbs (Xiao Zhong Zhi Tong Tie) in patients with diarrhea – a machine-learning model based on XGBoost algorithm," *World Journal of Traditional Chinese Medicine*, vol. 8, no. 2, 2022.

[22] H. Su, K. Qi, L. Fu, T. Guo, and L. Zhang, "Anomaly detection method of cloud platform timing monitoring data based on wavelet packet and xgboost," *Digital technology and application*, vol. 40, no. 07, pp. 6–8, 2022.

[23] Hong Chen, Wei Wan, "Analysis of E-Commerce Marketing Strategy Based on Xgboost Algorithm", *Advances in Multimedia*, vol. 2023, Article ID 1247890, 9 pages, 2023.