ARTIFICIAL INTELLIGENCE

Adaptive Machine Learning Framework for Stock Market Prediction under Non-Stationary Conditions

Victor Gbenga Lijoka[1], B. C. Iloka[2], Hope Anyaragbu[2], Emmanuel Omeye[2], Emeka Okorie[2]

[1]ICT Unit, Admiralty University of Nigeria, Delta State, Nigeria
[2]Computer Science Department, Tansian University, Umuya, Anambra State, Nigeria

Published November 2026

Abstract

This research seeks to contribute to the evolving landscape of financial forecasting by proposing an innovative and comprehensive machine learning framework for modeling and predicting stock market dynamics, while considering the impact of beta on model performance. With the unprecedented growth of financial data and the increasing complexity of global markets, traditional models are often limited in their ability to capture the intricate patterns and volatilities inherent in stock price movements. The primary objective of this research is to develop a robust and adaptive machine learning model that leverages advanced techniques in data analysis, feature engineering, and algorithmic optimization, while accounting for the effect of beta, to enhance the accuracy and reliability of stock market predictions. The study will explore a diverse range of machine learning methodologies, including but not limited to deep learning, ensemble methods, and reinforcement learning, to extract meaningful insights from historical market data and adapt to changing market conditions, with particular attention to how beta influences the performance of these methodologies.

Keywords: financial forecasting; stock market; machine learning; data analysis; algorithmic optimization

1 Introduction

The stock market is critical to economic growth. This is because stock markets help companies to raise capital, it helps generate personal wealth for individual and corporate investors, it also serves as an indicator of the state of the economy (Fengrong et al., 2023; Peidong et al., 2023). The implication is that strong economies will have a vibrant stock market, while weak economies will have a weak stock market. It also implies that a healthy company will have a strong stock market indicator (Liu and Chen, 2023).

Investors in the stock market seek to grow their stock portfolio over time, they also seek to maximize their profit while minimizing their loss. To achieve these goals, the investors need to read the market and predict the future movement (Leippold et al., 2022).

The study of financial markets and asset pricing has a rich historical backdrop, tracing back centuries with early forms of financial analysis focused on fundamental factors like earnings, dividends, and economic indicators (MacKenzie, 2008). However, it was not until the late 19th century that statistical methods began to be applied to financial data, setting the stage for quantitative analysis in finance.

The landscape of financial modeling experienced a transformative shift in the 1950s with the development of Modern Portfolio Theory (MPT) by Harry Markowitz. MPT introduced the concept of

diversification to optimize risk-adjusted returns, revolutionizing investment management practices (Mahadevan, 2023). Building upon this framework, the Capital Asset Pricing Model (CAPM) emerged in the 1960s, spearheaded by William Sharpe and his colleagues. CAPM linked expected returns to systematic risk, as measured by beta relative to the market, providing a foundation for understanding asset pricing and risk management (Dempsey, 2013).

Advancements in computing technology in the 1980s and 1990s paved the way for the rise of computational finance. With the proliferation of computers and increased computational power, complex mathematical models could be applied to financial markets more effectively. This era witnessed the development of option pricing models such as the Black-Scholes model in 1973, which revolutionized derivatives pricing and quantitative finance (Schaefer, 1998).

The turn of the millennium brought forth a new era characterized by the integration of machine learning techniques into financial modeling. In recent decades, machine learning has gained prominence across various financial domains, including credit scoring, fraud detection, and algorithmic trading. The emergence of high-frequency trading (HFT) further emphasized the need for advanced data analytics and predictive modeling in real-time market environments (Gomber and Haferkorn, 2015).

With the advent of big data and predictive analytics, financial markets have become increasingly complex and interconnected. This complexity has spurred interdisciplinary research efforts, combining finance, statistics, and computer science to develop innovative solutions for financial forecasting and risk assessment (Pappas, 2018). The current research focus on integrating beta (systematic risk) into machine learning frameworks for stock price prediction represents a continuation of this trajectory, aiming to enhance the accuracy and reliability of predictions amidst dynamic and interconnected global markets.

2 Methodology

2.1 Long short-term memory (LSTM) network

Let $x_t \in R^n$Input vector at time step t, $h_t \in R^m$ be the Hidden state vector at time step t. $c_t \in R^m$is the Cell state vector at time step t. W, U, and b: Weight matrices and bias vectors for different gates. The LSTM unit has the following components: input gate (i_t), forget gate (f_t), output gate (o_t), and the cell state (c_t).

The forget gate (f_t) controls how much of the previous cell state c_{t-1} is retained. It outputs a value between 0 and 1, where 0 means forget everything, and 1 means keep everything.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{1}$$

Where:

$W_f \in R^{m \times n}$ and $b_f \in R^m$ are the weight matrices and bias for the forget gate.

σ is the element-wise sigmoid function.

The input gate (i_t) controls how much of the new candidate cell state to add to the cell state.

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{2}$$

Where:

$W_i \in R^{m\times n}$, $U_i \in R^{m\times m}$, and $b_i \in R^m$ are the weight matrices and bias for the input gate.

This is the new candidate cell state (c_t), scaled by the input gate before being added to the current cell state.

$$c_t = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{3}$$

Where:

$W_c \in R^{m\times n}$, $U_c \in R^{m\times m}$, and $b_c \in R^m$ are the weight matrices and bias for the candidate cell state.

tanh is the hyperbolic tangent activation function.

The cell state is updated based on the forget gate and the input gate. The forget gate determines what part of the previous cell state to forget, and the input gate determines how much of the new candidate cell state to add.

$$c_{t+1} = f_t \odot c_{t-1} + i_t \odot c_t \tag{4}$$

Where:

$\odot$ represents element-wise multiplication.

The output gate (o_t) controls how much of the cell state to pass to the hidden state.

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{5}$$

Where:

$W_o \in R^{m\times n}$, $U_o \in R^{m\times m}$, and $b_o \in R^m$ are the weight matrices and bias for the output gate.

The hidden state is computed as the output gate scaled by the $\tanh$ of the current cell state:

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{6}$$

2.1.1 *Loss function and optimization*

LSTMs are typically trained using backpropagation through time (BPTT), where the loss function (e.g., mean squared error or cross-entropy) is computed over the entire sequence, and gradients are computed to update the weight matrices W, U, and biases b. Optimization is often performed using gradient descent or a variant like Adam.

2.2 Random forest

A random forest is an ensemble method that builds multiple decision trees and combines their outputs to make more accurate predictions. It reduces overfitting and increases accuracy compared to a single decision tree.

2.2.1 *Steps to build a random forest*

Bootstrapping: Random Forest builds B decision trees using different bootstrap samples $D^{(b)}$ drawn from the original training set D, where $b = 1, \ldots, B$.

$$D^{(b)} = \left\{\left(x_i, y_i\right)\right\} \text{ sampled with replacement from } D \tag{7}$$

Random Feature Selection: At each node of each tree, instead of evaluating all possible features for splitting, a random subset of M features is selected. The best feature from this subset is chosen for the split.

- If there are d features in total, $M \approx \frac{d}{3}$.

Tree Growth: Each decision tree is grown independently by following the standard decision tree construction process, but using the bootstrap sample and random feature selection.

Prediction Aggregation: Random Forest takes the average of the predictions:

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B}\hat{y}^{(b)} \tag{8}$$

2.3 Ensemble learning

Ensemble learning is a machine learning paradigm that combines multiple models to improve performance compared to individual models. The primary idea is to leverage the strengths of various models to achieve better predictive accuracy and robustness.

Let $D = \left\{(x_i,\ y_i)\right\}_{i=1}^{N}$ be a dataset consisting of N instances, where $x_i \in \mathrm{R}^d$ is the feature vector and $y_i \in \mathrm{R}$.

An ensemble model f_E is defined as a weighted combination of M base models f_m:

$$f_E(x) = \sum_{m=1}^{M} w_m f_m(x) \tag{9}$$

where w_m are the weights assigned to each base model f_m.

2.4 Bagging (bootstrap aggregating)

Sampling: From the original dataset D, create M bootstrap samples D_m (samples drawn with replacement) for $m = 1,\ \ldots,\ M$.

Training: Train a base model f_m on each bootstrap sample D_m.

Prediction: For a new instance x, the ensemble prediction is obtained by averaging the predictions

$$f_E(x) = \frac{1}{M}\sum_{m=1}^{M} f_m(x) \tag{10}$$

2.5 Support vector machines

Support vector machines (SVM) are a supervised machine learning model primarily used for regression tasks. The goal of SVM is to find the optimal hyperplane that separates the classes of data in a feature space with the maximum margin.

2.5.1 Formulation of support vector machines

For linearly separable data, the goal of SVM is to find a hyperplane that maximally separates two classes. Let the training dataset be:

$$D = \left\{(x_i,\ y_i)\right\}_{i=1}^{N},\ x_i \in \mathrm{R}^d,\ y_i \in \left\{-1,\ 1\right\} \tag{11}$$

where x_i is the feature vector of the i-th sample, and y_i is the class label, which is either $+1$ or -1.

2.5.2 Equation of the hyperplane

A hyperplane in R^d is defined by:

$$w^{\top}x + b = 0 \tag{12}$$

where $w \in \mathrm{R}^d$ is the normal vector to the hyperplane, and $b \in \mathrm{R}$ is the bias term (or offset from the origin). The vector w determines the orientation of the hyperplane, and b determines its position.

2.6 Regression metrics

2.6.1 Mean squared error (MSE)

MSE is defined as:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{13}$$

2.6.2 Root mean squared error (RMSE)

RMSE is the square root of MSE:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{14}$$

2.6.3 Coefficient of determination

The R^2 score is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{15}$$

3 Results and discussion

3.1 Descriptive statistics

The summary statistics of the stock data are given in Table 1.

Table 1. Summary statistics of the stock data.

Variable	Count	Mean	SD	Min	25%	50%	75%	Max
Close	807	276.179	43.429	195.456	240.813	273.449	310.928	380.620
Volume	807	28,296,090	10,313,580	9,200,800	21,680,250	25,978,600	32,430,900	90,428,900
SMA_200	807	257.670	35.713	180.170	233.672	260.770	288.200	327.222
RSI	807	54.029	16.078	6.806	42.142	53.458	67.198	90.537
%K	807	33.619	34.104	−78.185	4.721	36.064	64.192	95.003
%D	807	33.595	31.812	−62.196	4.613	35.425	61.162	90.971
Volatility	807	7.526	2.899	1.773	5.428	7.077	9.601	16.614

Variable	Count	Mean	SD	Min	25%	50%	75%	Max
ATR	807	9.131	1.724	5.292	7.839	9.202	10.184	14.919
Daily return	807	0.001	0.017	−0.077	−0.009	0.001	0.011	0.082
Cumulative return	807	0.794	0.282	0.270	0.564	0.776	1.020	1.473
Rolling beta	807	1.215	0.273	0.561	1.035	1.224	1.384	2.090

SD, standard deviation; SMA_200, 200-day simple moving average; RSI, relative strength index; %K and %D, stochastic oscillator; ATR, average true range.

This stock summary statistics table provides insights into the stock's historical price behavior, volatility, and momentum indicators.

1. **Close Price:** Mean: 276.179, with a standard deviation (Std) of 43.429, indicates that the stock's price has historically centered around $276 but fluctuates quite significantly (±$43). Min/Max: The price ranged between 195.456 and 380.620, suggesting periods of both strong declines and substantial rallies. This is a stock with moderate price volatility, and investors may expect significant price swings over time.
2. **Volume:** The Mean volume is 28.3 million shares with a Std of 10.3 million, suggesting a high average trading volume but with large variability. The Min/Max shows that the stock had as few as 9.2 million shares traded and as many as 90.4 million, indicating periods of both low and high liquidity. The stock sees robust trading volumes, making it fairly liquid. Large volume spikes could indicate periods of heightened investor interest or news-driven events.
3. **200-day Simple Moving Average (SMA_200):** Mean SMA_200 is 257.670, which means the stock has, on average, traded higher than its long-term price trend, as the mean close price (276.179) is above the SMA_200. The Std of 35.713 suggests the 200-day moving average has also seen significant changes over time.
4. **Relative Strength Index (RSI):** Mean RSI is 54.029, indicating a neutral momentum, as RSI values range from 0 to 100, with 50 being neutral. Min/Max RSI: 6.806 to 90.537 shows periods of oversold conditions (below 30) and overbought conditions (above 70). Market Implication: The stock occasionally enters both oversold and overbought territories, which could signal short-term trading opportunities.
5. **Stochastic Oscillator (%K, %D):** Mean %K: 33.619 and Mean %D: 33.595 are both well below 50, suggesting the stock tends to be in the lower range of its price over time. Min/Max: %K and %D have significant ranges, with %K ranging from -78.185 to 95.003, and %D ranging from -62.196 to 90.971. Market Implication: These extreme values suggest volatility and significant swings in momentum, offering potential opportunities for traders looking to capitalize on short-term price movements.
6. **Volatility:** Mean Volatility: 7.526, with a Std of 2.899, indicates moderate historical price fluctuations. Max Volatility: 16.614 shows some periods of extremely high volatility, likely during earnings reports or macroeconomic events. This stock generally experiences moderate volatility, with occasional spikes. Traders should be cautious of sudden price swings.

7. **Average True Range (ATR):** Mean ATR of 9.131, with a Std of 1.724, reflects a moderate level of price fluctuation in absolute terms. Investors can expect the stock to move by around $9 on average per trading day, with some variations during highly volatile periods.
8. **Daily Return and Cumulative Return:** Mean Daily Return: 0.001, indicating the stock has a slight upward bias on a daily basis. Cum Return: The mean of 0.794 suggests that cumulatively, the stock has returned about 79% over the observed period. The Min/Max Cum Return of 0.270 and 1.473 suggests periods of both drawdowns and substantial growth. The stock generally trends upwards in the long term, although there are occasional periods of significant short-term losses.
9. **Rolling Beta:** Mean Beta: 1.215, indicating that the stock is more volatile than the market (beta > 1). Std: 0.273, showing variability in its correlation with the market. Min/Max: Beta ranges from 0.561 to 2.090, suggesting periods of lower and higher market sensitivity. This stock is typically more volatile than the overall market, which implies higher risk but also higher potential reward for investors during bullish periods. However, in market downturns, it may experience sharper declines.

The summary statistics of MSFT stock displays significant volatility and variability in its returns, making it suitable for active traders and risk-tolerant investors. Long-term investors may benefit from the stock's upward trend (as indicated by the cumulative return), but they must be prepared for potential drawdowns due to its high beta and occasional overbought/oversold conditions (as seen from the RSI and stochastic indicators).

The correlations among the variables are shown in Figure 1.

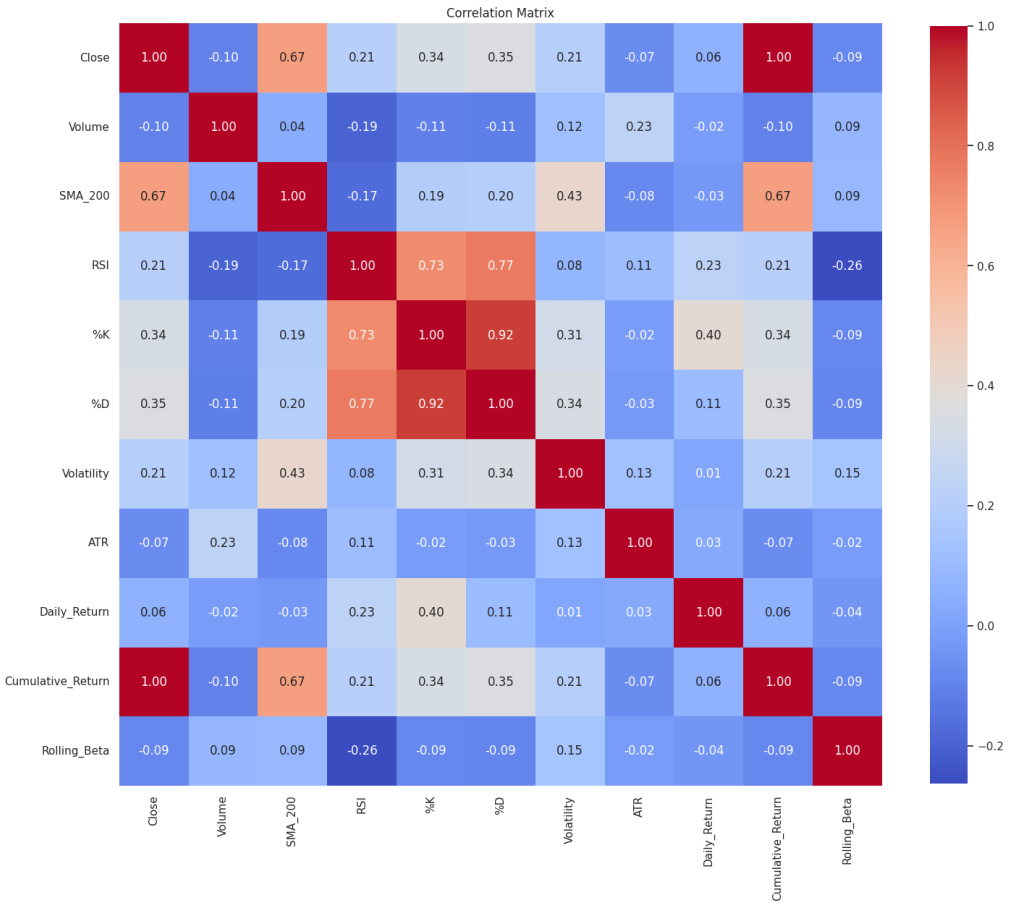


Figure 1. Correlation matrix of the stock variables.

3.2 Model performance

The performance of the regression models on the test data is summarised in Table 2, and the actual and predicted closing prices are compared in Figures 2-4.

Table 2. Performance of the regression models.

Model	MSE	RMSE	R^2
Random Forest Regression	327.8448	18.1065	0.2130
Gradient Boosting Regression	311.5370	17.6504	0.2521
Support Vector Regression	3315.1990	57.5778	−6.9582
K-Nearest Neighbors Regression	469.0018	21.6565	−0.1259
Multi-layer Perceptron Regression	1801.3100	42.4418	−3.3241
Ensemble Model Regression	142.0018	11.9165	0.6591

MSE, mean squared error; RMSE, root mean squared error.

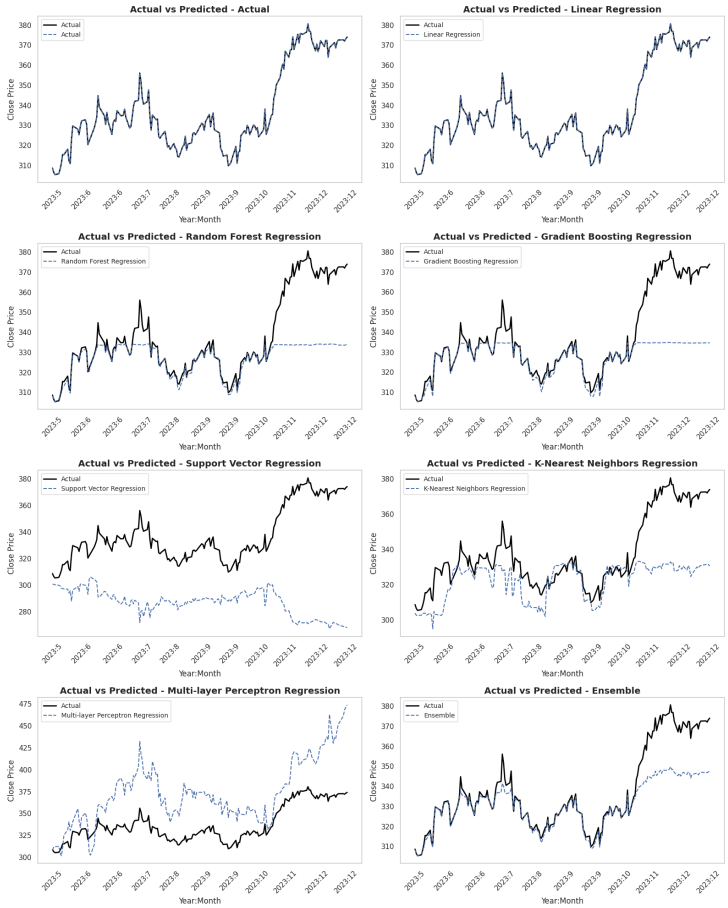


Figure 2. Actual and predicted closing prices for each model (test period, May–December 2023).

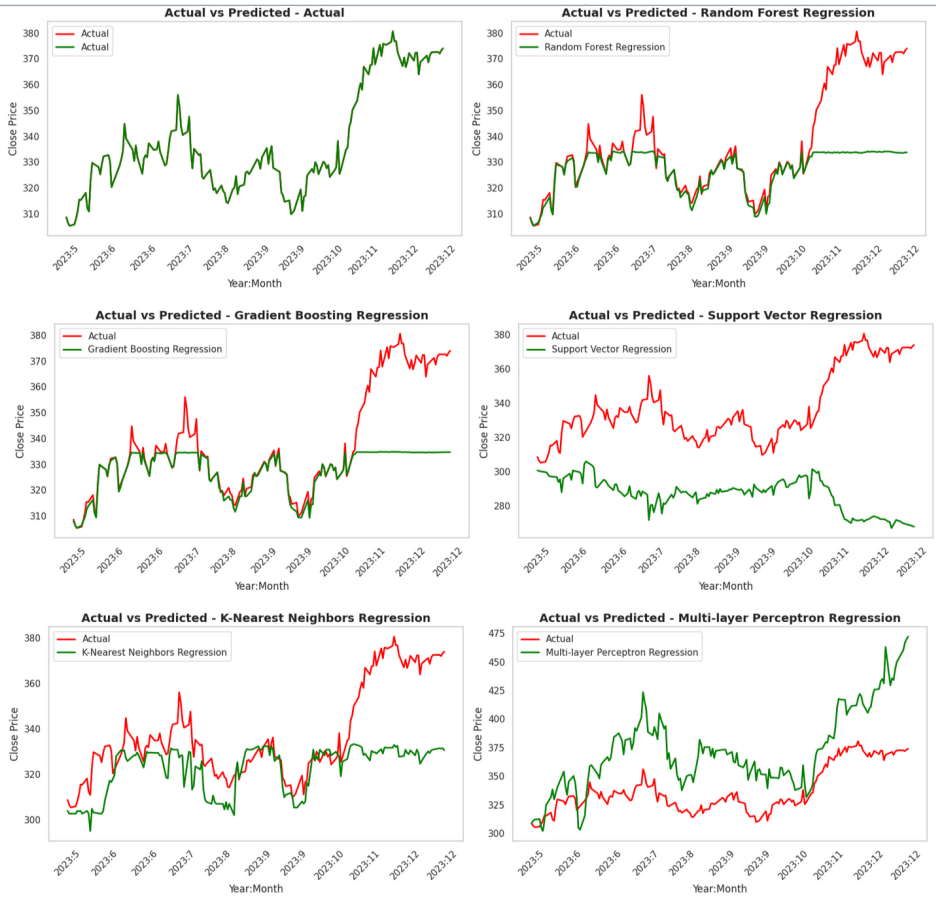


Figure 3. Actual and predicted closing prices: (a) actual, random forest; (b) gradient boosting, support vector regression; (c) k-nearest neighbours, multi-layer perceptron.

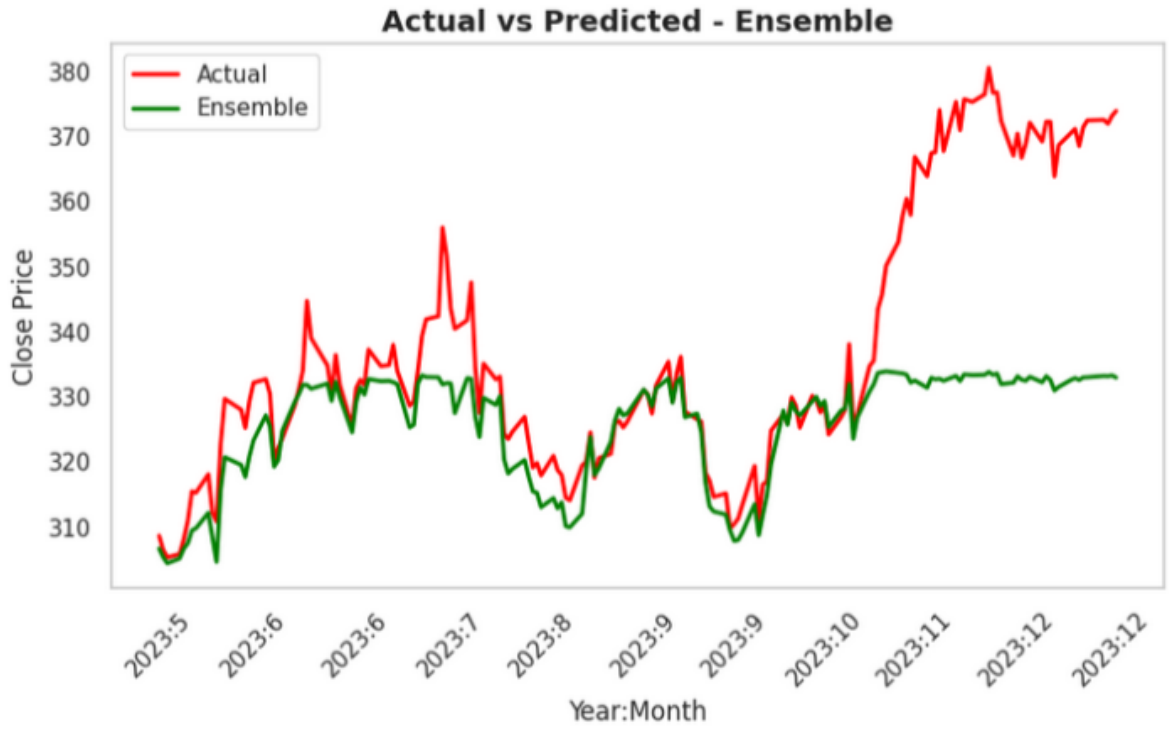


Figure 4. Actual and predicted closing prices of the ensemble model.

3.2.1 Analysis of the performance metrics

The performance metrics for each regression model, along with the ensemble model, offer a detailed view of how well each model predicts the target variable, (the closing price).

Linear regression. The linear regression model shows nearly perfect predictions, indicated by an R^2 of 1. This suggests that it explains all the variance in the closing prices. However, the reliability of this model can be questioned due to its simplistic assumptions and the possibility of overfitting, especially in more complex financial data.

Random forest regression. The random forest model shows considerable error with a much lower R^2 value. This indicates it explains only about 21.3% of the variance in the closing price. This indicate it may not be reliable for investors as it suggests high prediction uncertainty.

Gradient boosting regression. Similar to random forest, the gradient boosting regression also displays significant prediction errors, with less than 26% variance explained. This also implies a lack of reliability for investors relying on this model for forecasts.

Support vector regression. This model performs poorly, as indicated by a negative R^2 indicating that the model performs worse than a horizontal line (mean prediction). Investors should be wary of using this model as it could lead to misguided investment decisions.

K-nearest neighbours regression. This model also shows poor performance, with an R^2 less than zero. Its predictive capability is questionable, making it unsuitable for investment strategies.

Multi-layer perceptron regression. Like support vector regression and K-nearest neighbors, this model also performs poorly, resulting in negative R^2 values. It is a further indication that the model does not capture the data trends effectively.

Ensemble model. The ensemble model performs significantly better than individual models like random forest, gradient boosting, and others, with an R^2 of approximately 66%. This suggests that the ensemble model effectively combines the strengths of various models, leading to improved predictive performance. This is crucial for investors, as it can offer a more reliable basis for decision-making.

3.3 Implications for investors

Model Selection: Investors should focus on models that demonstrate strong performance metrics. In this analysis, the ensemble model emerges as a suitable candidate for making predictions about future closing prices. Its ability to balance performance across various modeling approaches is beneficial.

Risk Management: The high variance and errors associated with several models indicate the presence of unpredictability in the market. Investors must remain cautious when relying solely on models that have poor performance metrics. Decision-making based on unreliable models can lead to significant financial losses.

Diversification of Approaches: The diverse results from different models reinforce the idea that relying on a single predictive model could expose investors to greater risks. Using an ensemble approach or a combination of models may yield more robust predictions and a better understanding of market movements.

4 Conclusion

While the results show potential in using machine learning models for predicting closing prices, investors should approach these predictions with caution. The ensemble model shows promise, but continuous refinement, monitoring, and a diversified approach to model selection are essential strategies to mitigate risk and enhance decision-making in investment strategies.

References

[1] Alzaman C (2024). Deep learning in stock portfolio selection and predictions. *Expert Systems with Applications*, 237, 121404. https://doi.org/10.1016/j.eswa.2023.121404

[2] Ayyildiz N, Iskenderoglu O (2024). How effective is machine learning in stock market predictions? *Heliyon*, e24123. https://doi.org/10.1016/j.heliyon.2024.e24123

[3] Beniwal M, Singh A, Kumar N (2024). Forecasting multistep daily stock prices for long-term investment decisions: A study of deep learning models on global indices. *Engineering Applications of Artificial Intelligence*, 129, 107617. https://doi.org/10.1016/j.engappai.2023.107617

[4] Bonga-Bonga L, Mwamba MJ (2021). Multivariate models for the prediction of stock returns in an emerging market economy: comparison of parametric and non-parametric models. *Macroeconomics and Finance in Emerging Market Economies*,1-17. https://doi.org/10.1080/17520843.2021.1997289

[5] Bui DG, Kong DR, Lin CY, Lin TC (2023). Momentum in machine learning: Evidence from the Taiwan stock market. *Pacific-Basin Finance Journal*, 82, 102178. https://doi.org/10.1016/j.pacfin.2023.102178

[6] Byun SJ, Cho S, Kim DH (2024). Can a machine learn from behavioral biases? Evidence from stock return predictability of deep learning models. *Journal of Behavioral and Experimental Finance*, 41, 100881. https://doi.org/10.1016/j.jbef.2023.100881

[7] Cagliero L, Fior J, Garza P (2023). Shortlisting machine learning-based stock trading recommendations using candlestick pattern recognition. *Expert Systems with Applications*, 216, 119493. https://doi.org/10.1016/j.eswa.2022.119493

[8] Cakici N, Fieberg C, Metko D, Zaremba A (2023). Machine learning goes global: Cross-sectional return predictability in international stock markets. *Journal of Economic Dynamics and Control*, 155, 104725. https://doi.org/10.1016/j.jedc.2023.104725

[9] Campisi G, Muzzioli S, Baets BD (2023). A comparison of machine learning methods for predicting the direction of the US stock market on the basis of volatility indices. *International Journal of Forecasting*. https://doi.org/10.1016/j.ijforecast.2023.07.002

[10] Chaudhari K, Thakkar A (2023). Neural network systems with an integrated coefficient of variation-based feature selection for stock price and trend prediction. *Expert Systems with Applications*, 219, 119527. https://doi.org/10.1016/j.eswa.2023.119527

[11] Chen J, Wen Y, Nanehkaran YA, Suzauddola MD, Chen W, Zhang D (2023). Machine learning techniques for stock price prediction and graphic signal recognition. *Engineering Applications of Artificial Intelligence*, 121, 106038. https://doi.org/10.1016/j.engappai.2023.106038

[12] Chen Y, Wu J, Wu Z (2022). China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach. *Expert Systems with Applications*, 202, 117370. https://doi.org/10.1016/j.eswa.2022.117370