# FORECASTING VIETNAM BANKING'S STOCK PRICES USING TIME SERIES APPROACH. (DECEMBER 2023)

**NGUYEN DAI ANH TUAN[1], NGUYEN HUU THIEN[2], and NGUYEN TUAN TU[3]**

[1] *University of Information Technology (e-mail: 21522753@gm.uit.edu.vn)*
[2] *University of Information Technology (e-mail: 21522625@gm.uit.edu.vn)*
[3] *University of Information Technology (e-mail: 21522744@gm.uit.edu.vn)*

**ABSTRACT** In the current scenario of increasing investor interest in Vietnam, this study guides the context of increasing investment activities in Vietnam's banking sector. With a focus on BIDV, Sacombank (STB), and MB Bank (MBB), we employ accessible models Linear Regression, ARIMA, SVR, RNN, VAR, XGBoost, CNN-LSTM, and AdaBoost to predict stock prices.

**INDEX TERMS** ***Keywords - Time series ,statistical method, forecasting bank stock prices, machine learning, deep learning***

## I. INTRODUCTION

This report focus on the task of forecasting stock prices in the Vietnamese stock market through a time series approach, concentrating on three major banks: BIDV, Sacombank (STB), and MB Bank (MBB). By employing advanced time series analysis techniques on historical stock price data, the study aims to uncover patterns and trends that can increase our understanding of future stock price movements.

The purpose of this research lies in its ability to assist investors, financial analysts and policymakers in making informed decisions regarding risk management, investment strategies and financial planning.

In the course of this research, we use Linear Regression models, ARIMA, SVR, RNN, VAR, XGBoost, CNN-LSTM, and adaBoost to analyze and forecast stock prices of Vietnamese banks. Each model brings distinct strengths to the analysis, facilitating a thorough examination of the intricate dynamics shaping stock price movements.

## II. RELATED RESEARCH

Linear regression has proven its flexibility in different domains, showcasing its versatility in a range of applications within the field of predictive modeling. In 2023, Smith and Jones conducted a study that centers on foreseeing sales, providing valuable insights that can be utilized for business strategies [1].

Time series forecasting demands precision, and traditional methods like ARIMA, while adept at simple patterns, often falter on complex data. This report explores a groundbreaking hybrid approach by Wu et al. (2020) that merges ARIMA's linear prowess with the non-linear learning power of neural networks. This potent fusion promises not only to surpass individual model accuracy but also unveil hidden data dynamics and adapt to diverse datasets. Buckle up as we dissect this model's architecture, witness its real-world performance, and unlock the transformative potential of hybrid forecasting! [2]

Bui Thanh Khoa and colleagues (2022) conducted a study wherein they employed Support Vector Regression (SVR) in combination with the Capital Asset Pricing Model (CAPM) to forecast the yield rates of individual stocks. Experiments conducted on a dataset of companies listed from December 2012 to September 2020 on the Ho Chi Minh City Stock Exchange demonstrated that the SVR model outperforms CAPM in terms of accuracy in stock return prediction. [3]

Amalou et al. [4] conducted a comparative assessment of deep learning models, comprising the basic RNN and its variants, namely LSTM and GRU, using the energy datasets of the Smart Grid Smart City (SGSC) project spanning from 2010 to 2014. The study reveals that, in addressing the challenges of this time series problem, GRU exhibited superior performance when compared to the three models

In the year 2022, Zhang and his team conducted a research project where they employed Vector Autoregression (VAR) to predict the stock prices of ten companies listed on the Chinese stock market. By utilizing experimental techniques, the study proved that VAR has the ability to accurately forecast stock prices, which empowers investors to make better-informed investment choices [5]

This paper introduces XGBoost, a scalable tree boosting system. XGBoost is a machine learning algorithm based on decision trees and gradient boosting. It has been shown to

be effective in a variety of machine learning tasks, including classification, regression, and clustering. The paper describes the architecture of XGBoost and its core algorithms. It also presents a number of improvements to these algorithms to improve performance and accuracy. [6]

The paper describes the architecture of XGBoost and its core algorithms. It also presents a number of improvements to these algorithms to improve performance and accuracy.

Bharadi et al.[7] conducted research on implementing CNN-LSTMs and other deep learning models to forecast COVID-19 cases. The findings indicated that CNN-LSTM models demonstrated superior performance compared to other models in predicting COVID-19 cases, as assessed by the MSE metric.

The paper "An Improved AdaBoost Algorithm for Face Detection" by Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja, published in the IEEE Transactions on Pattern Analysis and Machine Intelligence in 2002, focuses on enhancing AdaBoost for the specific task of face detection. Recognizing the critical role of accurate face detection in computer vision applications, the authors introduce modifications to the AdaBoost algorithm, aiming to improve its performance in varied conditions.

## III. MATERIALS

### A. DATASET SOURCE

The dataset utilized in this study is sourced from https://www.investing.com/, a reputable financial platform known for its comprehensive and up-to-date market information. The dataset encompasses stock price data for three prominent Vietnamese banks: Sai Gon Thuong Tin Commercial Joint Stock Bank (STB), Joint Stock Commercial Bank for Investment and Development of Vietnam (BID), and Military Commercial Joint Stock Bank (MBB).

By leveraging the data available on this platform, we ensure a reliable foundation for our analysis.

The dataset spans from January 2, 2018, to December 21, 2023, providing a robust temporal scope for our analysis.

Each entry in the dataset encompasses key financial indicators, including:

Price: The closing stock price on a given day.

Open: The opening stock price at the beginning of the trading day.

High: The highest recorded stock price during the trading day.

Low: The lowest recorded stock price during the trading day.

Vol.: Trading volume, indicating the total number of shares traded on a particular day.

Change %: The percentage change in stock price from the previous day's closing price.

### B. DESCRIPTIVE STATISTICS

| | STB | BIDV | MBB |
|---|---|---|---|
| **Count** | 1492 | 1492 | 1492 |
| **Mean** | 16,250 | 32,741 | 14,503 |
| **Std** | 7879.081109 | 7164.07205 | 5559.773523 |
| **Min** | 7,300 | 16,531 | 7,207 |
| **Q1** | 11650 | 27705.7 | 10846.125 |
| **Q2** | 16250 | 32741.15 | 14503.3 |
| **Q3** | 26712.5 | 38785 | 19700 |
| **Max** | 35,850 | 49,100 | 28,667 |
| **Mode** | 11400 | 24873.7 | 18250 |
| **Median** | 16250 | 32741.15 | 14503.3 |
| **Var** | 62079919.12 | 51323928.33 | 30911081.63 |
| **Kurtosis** | -1.389548718 | -0.728150526 | -1.075570258 |
| **Skewness** | 0.371104931 | 0.19043032 | 0.455508884 |
| **CV** | 0.413504039 | 0.213528704 | 0.351912689 |

**TABLE 1.** Descriptive Statistics

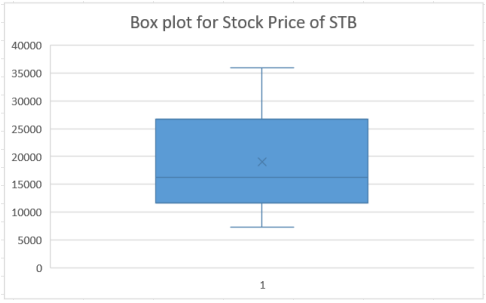


**FIGURE 1.** Box plot for stock price of STB

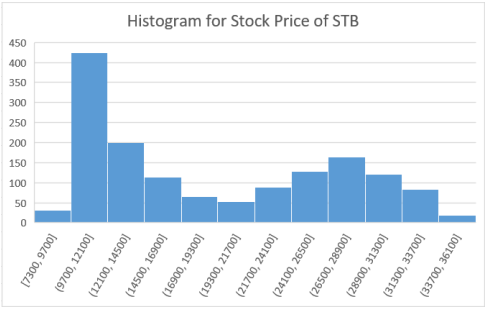


**FIGURE 2.** Histogram for Stock Price of STB

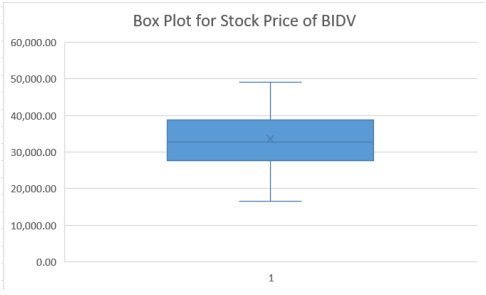


**FIGURE 3.** Box plot for stock price of BIDV

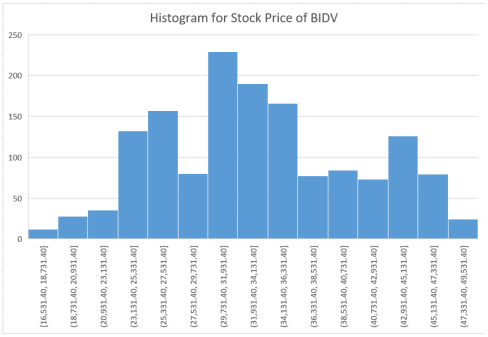


**FIGURE 4.** Histogram for Stock Price of BIDV

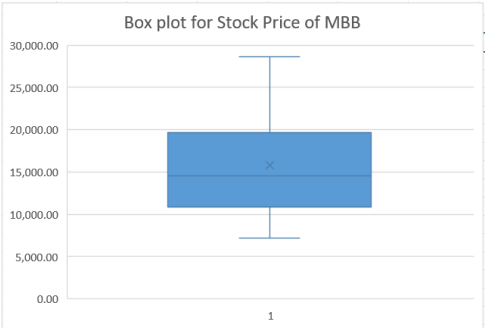


**FIGURE 5.** Box plot for stock price of MBB

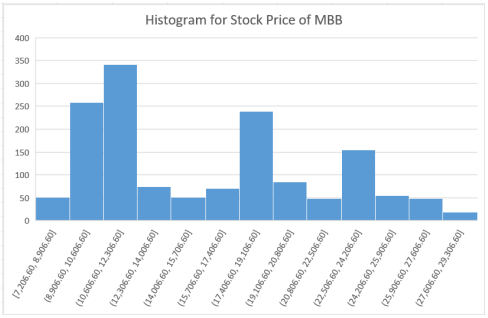


**FIGURE 6.** Histogram for Stock Price of MBB

### C. TOOL

In the course of our research and data analysis, we utilized a set of statistical analysis tools in Python to gain a deeper understanding of the data patterns and draw meaningful conclusions. The key tools include: numpy, pandas, sklearn, matplotlib.pyplot,...The statistical analysis tools in Python have facilitated a deeper understanding of the data, leading to significant findings. Detailed results can be found in the accompanying descriptive table and charts.

### D. DATA SPLIT RATIO

In our analysis of time series data, we divided the dataset into training and testing sets using various proportions: 70% for training and 30% for testing, 80% for training and 20% for testing, and 90% for training and 10% for testing. These ratios allow us to assess their impact on the model's performance by examining the distribution of data in each set. The commonly used 7:3 ratio allocates 70% for training and 30% for testing, striking a balance between providing sufficient training data and ensuring distinct sets for fine-tuning and evaluation. An alternative is the 8:2 ratio, favoring an 80% training set, beneficial for more complex models that require a larger training dataset. In certain scenarios, a cautious approach like the 9:1 ratio may be preferred, especially when dealing with a large dataset and a simpler model. This ratio ensures enough training data while allowing for a substantial testing set for performance evaluation.

### E. MODEL EVALUATION

RMSE is the square root of the average squared error in the predicted $yi$ values. In simpler terms, it measures how far off your predictions are from the actual values. The lower the RMSE, the better the prediction model.

Mean Absolute Error (MAE) is a metric used to measure the average magnitude of errors between predicted and actual values in a forecasting or regression model.

Mean Average Percentage Error (MAPE) is a metric used to evaluate the accuracy of a forecasting or prediction model.

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

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{2}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

Where:
$y_i$ is the observer value,
$\hat{y}_i$ is the predicted value,
$n$ is the number of observers

## IV. METHODOLOGY

### A. LINEAR REGRESSION

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data. A multiple linear regression model has the form: [8]

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \epsilon \tag{4}$$

Where:
$Y$ is the dependent variable,
$X_1, \ldots, X_k$ are the independent variables,
$\beta_0$ is the intercept term,
$\beta_1, \ldots, \beta_k$ are the regression coefficients for the independent variables,
$\epsilon$ is the error term

### B. ARIMA

The general form of ARIMA is ARMA(p, q) which is the combination of the Autoregressive model and the Moving Average model.

The Autoregressive (AR) component represents the relationship between the current observation and the past observations at several previous time steps. The current observation within the AR models depends on the value of the previous p time steps.

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \epsilon_t \quad (5)$$

Where:
$X_t$ is the current observation,
$X_{t-1}, \ldots, X_{t-p}$ the values of time series at previous time step,
$\phi_1, \ldots, \phi_p$ the autoregressive coefficients
$c$ is the intercept
$\epsilon_t$ is the error term

The Moving Average (MA) component represents the relationship between the current observation and the past error term at several previous time steps. The current observation within the AR models depends on the value of q past error terms

$$X_t = c + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t \quad (6)$$

Where:
$X_t$ is the current observation,
$c$ is the intercept
$\epsilon_{t-1}, \ldots, \epsilon_{t-q}$ are the error terms at previous q time steps
$\theta_1, \ldots, \theta_q$ are the moving average coefficients

Integrated I(d) performs differencing to the d observations between the present values and past values. The differencing transformation transforms a non stationary time series to a stationary time series, which could be represented as

$$Y_t = X_t - X_{t-d} \quad (7)$$

Where: $Y_t$ is the differenced time series
In general, the ARIMA(p, d, q) can be represented in the following form:

$$Y_t = c + \phi_1 Y_{t-1} + \ldots + \phi_p Y_{t-p} + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t \quad (8)$$

Where:
$Y_t$ the observation at time t
$\phi_j$ the parameter of the Autoregressive model
$\theta_k$ the parameter of the Moving Average model
$\epsilon_t$ the error term
$c$ constant value

### C. SVR

Support Vector Regression (SVR) is a regression analysis machine learning algorithm. Its main goal is to find a function that approximates the relationship between input variables and a continuous target variable, minimizing prediction errors. Unlike Support Vector Machines (SVMs) for classification, SVR seeks a hyperplane in a continuous space, achieved by mapping inputs to a high-dimensional feature space. It identifies the hyperplane maximizing margin from the nearest data points while minimizing prediction errors, handling non-linear relationships through kernel functions. SVR is a powerful tool for regression tasks with complex input-target variable relationships.

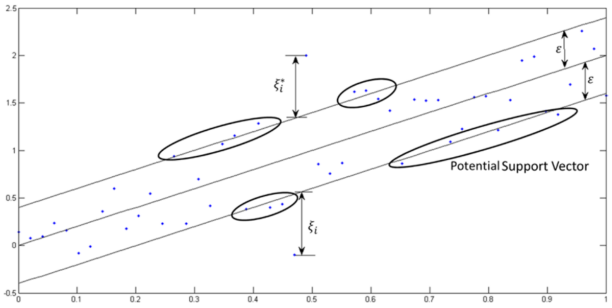


**FIGURE 7.** One-dimensional linear SVR

The continuous-valued function being approximated can be written: [9]

$$y = f(x) = < w, x > + b = \sum_{j=1}^{M} w_j x_i + b, y, b \in \mathbb{R}, w \in \mathbb{R}^M \quad (9)$$

For multidimensional data, you augment x by one and include b in the w vector to simply the mathematical notation, and obtain the multivariate regression:

$$f(x) = \begin{bmatrix} w \\ b \end{bmatrix}^T \begin{bmatrix} x \\ 1 \end{bmatrix} = w^t x + b \qquad x, w \in \mathbb{R}^M + 1 \quad (10)$$

The objective function in Support Vector Regression (SVR) is given by:

$$min \frac{1}{2} ||w||^2 + C \sum_{i=1}^{N} \xi_i + \xi_i^* \quad (11)$$

Where:
$w$ is the weight vector of the model.
$C$ is the regularization parameter, adjusting the trade-off between optimizing the complexity of the model and minimizing prediction errors.
$\xi_i, \xi_i^*$ are slack variables introduced to handle noisy or outlying data points
Constraints
$y_i - w^t x_i \leq \varepsilon + \xi_i^* \qquad i = 1...N$
$w^t x_i - y_i \leq \varepsilon + \xi_i \qquad i = 1...N$
$\xi_i, \xi_i^* \geq 0 \qquad i = 1...N$

Above, we asume $f(x)$ is linear. For non linear functions, the data can be mapped into a higher dimensional space, called kernel space, to achieve a higher accuracy.

**TABLE 2.** Kernel Functions in Support Vector Regression

| Kernel Name | Function |
|---|---|
| Linear | $k(x,u) = x^T * u$ |
| Polynomial | $k(x,u) = (ax^T u + c)^q, q > 0$ |
| Gaussian Radial Basis Function (RBF) | $k(x,u) = exp(-\frac{\|\|x-u\|\|^2}{\sigma^2})$ |
| Sigmoid | $k(x,u) = tanh(\beta x^T u + \gamma)$ |

### D. RNN

Recurrent Neural Network (RNN) [4] was designed for processing sequential and time-series data. RNNs have connections that form directed cycles, allowing them to maintain memory and capture dependencies in sequences. The below equation represent a RNN cell:

$$h^{(t)} = \sigma(Wx^{(t)} + Uh^{(t-1)} + b) \tag{12}$$

Here, $h^{(t)}$ and $x^{(t)}$ represent the hidden state and input state at time t, respectively. $W$ and $U$ are the weight matrices, $h^{(t-1)}$ denotes the hidden state at time $t-1$,and $b$ is the bias term associated with that cell.

### E. VAR

Vector Autoregression (VAR) is a statistical model used in econometrics and time series analysis to capture the dynamic relationships among multiple time series variables. It is a natural extension of the univariate autoregressive model to dynamic multivariate time series. Forecasts from VAR models are quite flexible because they can be made conditional on the potential future paths of specified variables in the model. [10]

Let $Y_t = (y_{1t}, y_{2t}, ..., y_{nt})'$ denote an $(n \times 1)$ vector of time series variables. The basic p-lag vector autoregressive (VAR(p)) model has the form

$$Y_t = c + \Pi_1 Y_{t-1} + \Pi_2 Y_{t-2} + ... + \Pi_p Y_{t-p} + \varepsilon_t, t = 1, ..., T \tag{13}$$

Where:
$Y_t$ is a vector of endogenous variables at time t,
$\Pi_i$ are $(n \times n)$ coefficient matrices,
$Y_{t-1}, Y_{t-2}, ..., Y_{t-p}$ are lagged values of the endogenous variables,
$\varepsilon_t$ is an $(n \times 1)$ is a vector of error terms assumed to be white noise.

### F. XGBOOST

XGBoost, a leading machine learning algorithm, excels in predictive modeling. Leveraging ensemble learning, it sequentially constructs decision trees to correct errors, showcasing adaptability and robustness. With regularization and adept handling of missing data, XGBoost proves to be a powerful solution for unraveling intricate data relationships. Understanding its streamlined architecture is key to exploring its wide-ranging applications in machine learning.

n examples and m features

$$\mathcal{D} = \{(x_i, y_i)\}(|\mathcal{D}| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}) \tag{14}$$

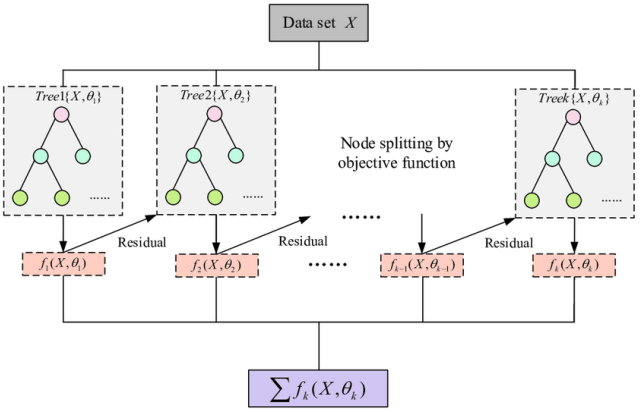


**FIGURE 8.** Flow chart of XGBoost

$$\hat{y}_i = \varphi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{15}$$

where $\mathcal{F} = \{f(x) = w \cdot q(x) \mid q : \mathbb{R}^m \to \mathbb{T}, w \in \mathbb{R}^T\}$
$f_k$ is an independent tree structure of model, and $y_i^t$ be the prediction of the i-th instance at the t-th iteration

The core formula for XGBoost is derived from the objective function, which is optimized during the training process. The objective function for the regression task in XGBoost is:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{16}$$

$$\text{where } \Omega(f_k) = \gamma T + \frac{1}{2}\lambda||w||^2 \tag{17}$$

$l$ is a differentiable convex loss function, $\omega$ penalizes the complexity of the model, $T$ is the number of leaves in the tree, $w$ is leaf weight, $\gamma$ is a regularization parameter that penalizes the number of leaves, and $\lambda$ is another regularization parameter that penalizes the L2 norm of the leaf scores.

The process repeats until the final predicted value is reached.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_k) \tag{18}$$

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[ l(y_i, \hat{y_i}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_k) \tag{19}$$

### G. CNN-LSTM

The CNN-LSTM model [7, 11], a powerful fusion of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), represents an advanced architecture for handling diverse data types. CNNs, with their specialized convolutional and pooling layers, excel at pattern recognition, particularly in image data, and have demonstrated adaptability for time series problems. Meanwhile, Recurrent Neural Networks (RNNs) are known for their effectiveness in managing sequential data but grapple with the vanishing gradient problem. To overcome the limitations of RNNs, the

CNN-LSTM model integrates the strengths of both architectures. The CNN component efficiently extracts hierarchical features from the input data, while the LSTM component, with its memory cells and gating mechanisms, addresses the challenge of learning long-term dependencies. The LSTM memory cells incorporate three gates: the input gate $i^{(t)}$, deciding on new information addition or updates; the forget gate $f^{(t)}$, responsible for discarding irrelevant information; and the output gate $o^{(t)}$, controlling the information output from the memory. This amalgamation enhances the model's ability to capture intricate patterns and relationships in diverse datasets, making the CNN-LSTM model a versatile and robust choice for various machine learning tasks. The following equations describes the operation of LSTM:

$$i^{(t)} = \sigma(U^{(i)}x^{(t)} + W^{(i)}h^{(t-1)} + b^{(i)}), \quad (20)$$

$$f^{(t)} = \sigma(U^{(g)}x^{(t)} + W^{(g)}h^{(t-1)} + b^{(g)}), \quad (21)$$

$$\bar{c^{(t)}} = tanh(U^{(c)}x^{(t)} + W^{(c)}h^{(t-1)} + b_c), \quad (22)$$

$$c^{(t)} = g^{(t)} * c^{(t-1)} + i^{(t)} * \bar{c^{(t)}}, \quad (23)$$

$$o^{(t)} = \sigma(U^{(0)}x^{(t)} + W^{(0)}h^{(t-1)} + b^{(0)}), \quad (24)$$

$$h^{(t)} = o^{(t)} * tanh(c^{(t)}) \quad (25)$$

Where $x^{(t)}$ is the input, $U$ and $W$ are weight matrix, $b$ is the vectors of bias, $\sigma$ is the sigmoid function, (*) denotes component-wise multiplication, $h^{(t)}$ is the hidden state.
In our specific implementation, the CNN-LSTM model comprises 64 filters of size (3,) in the first convolutional layer, followed by a pooling layer, a flatten layer, an LSTM layer, and a fully connected layer.

### H. ADABOOST

AdaBoost, or Adaptive Boosting, is an ensemble learning algorithm designed to enhance the performance of weak learners. It operates iteratively by assigning weights to training instances, focusing more on misclassified ones in each iteration. The algorithm combines the outputs of weak learners, assigning weights based on their accuracy. The final model is a weighted sum of these learners, providing a robust predictive model. AdaBoost is versatile, working well with various weak learners, and it excels at reducing bias and variance for improved generalization.

AdaBoost typically uses decision stumps(classifier) as its weak learners. Decision stumps are simple decision trees with only one internal node and two terminal leaves.At first, the weight wi of i'th instance is 1/the number of trees. Each loop chooses a feature to make a stump which is determined by their lowest Gini Index. The weight error is the sum of the weight of the observations that are wrongly forecasted. [12]

$$\epsilon_t = \sum_{t}^{n} \omega_i \quad (26)$$

And the weight of weaker learner is calculate by the following formula

$$\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t}) \quad (27)$$

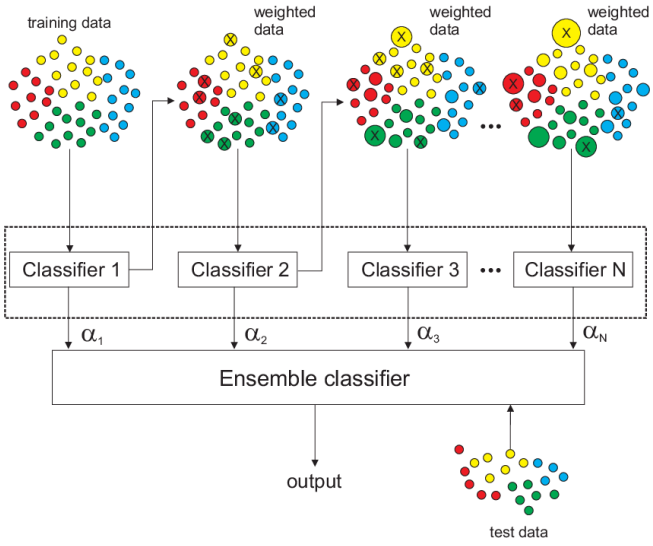


**FIGURE 9.** Flow chart of AdaBoost

The weight of each instance is then updated

$$\omega_i = \omega_i * e^{\pm\alpha_t} \quad (28)$$

When the sample is successfully identified, the amount of, say, (alpha) will be negative.
When the sample is misclassified, the amount of (alpha) will be positive.
Finally, the final prediction $H(x)$ of the strong classifier based on T weaker learn and is calculated by this following equation.

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \quad (29)$$

## V. EXPERIMENCE

### A. MODEL SETTING

#### 1) Linear Regression

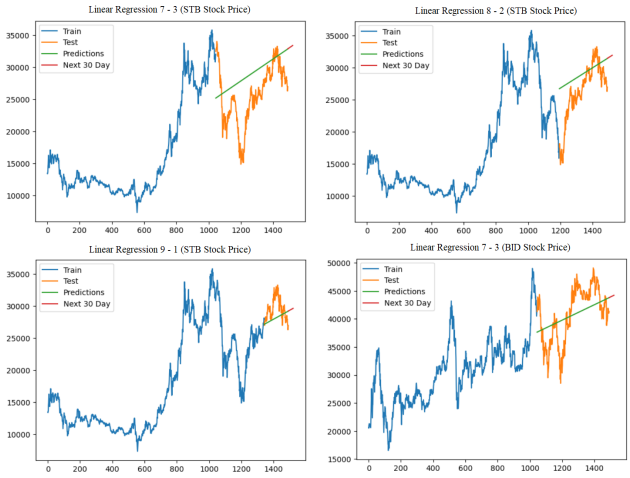


**FIGURE 10.** The result of LN model

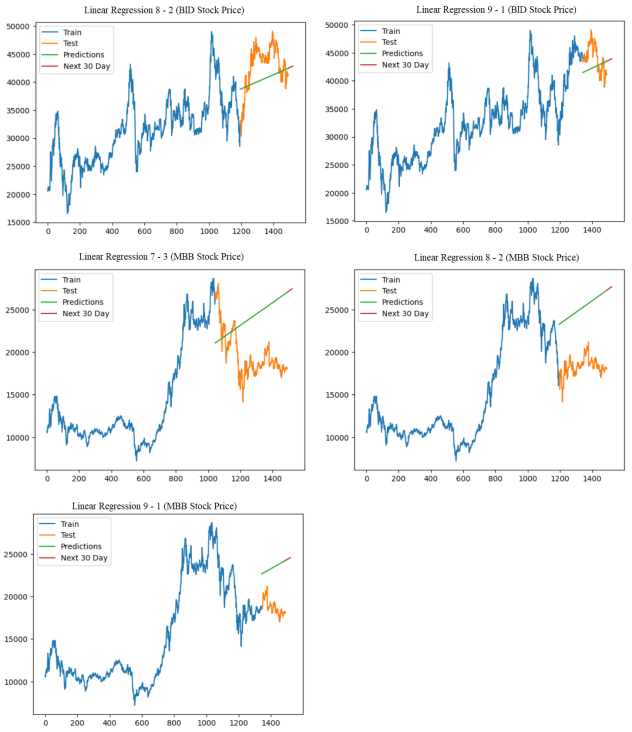


**FIGURE 11.** The result of LN model

## 2) ARIMA

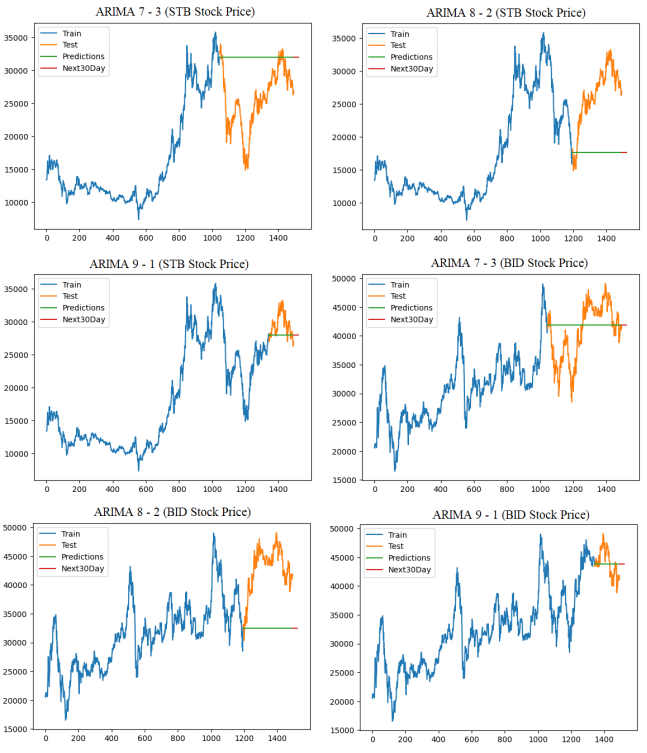


**FIGURE 12.** The result of ARIMA model

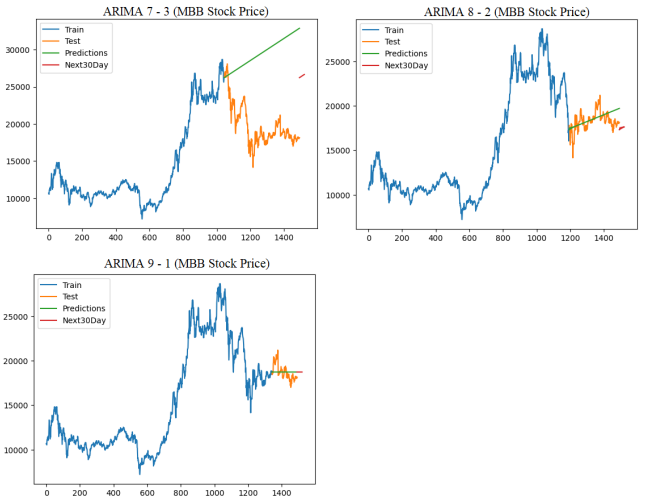


**FIGURE 13.** The result of ARIMA model

## 3) SVR

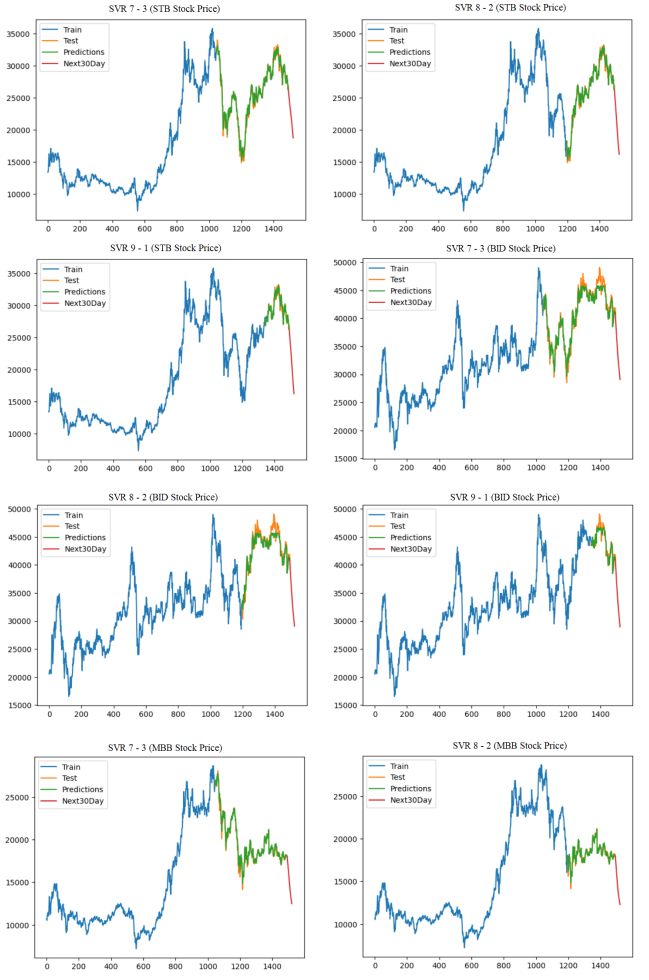


**FIGURE 14.** The result of SVR model

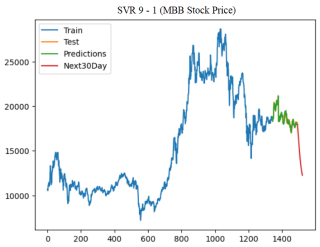


**FIGURE 15.** The result of SVR model

## 4) RNN

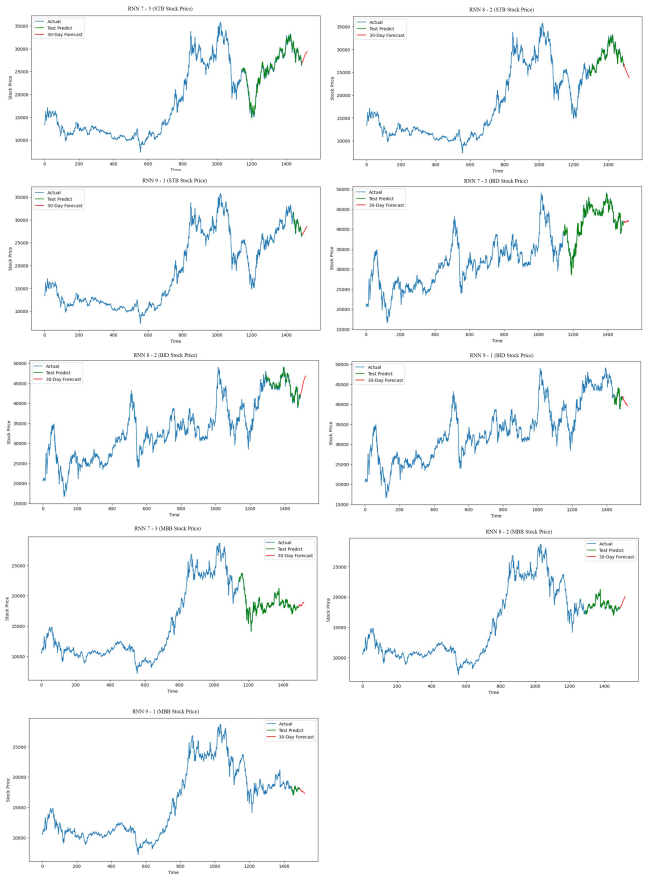


**FIGURE 16.** The result of RNN model

## 5) VAR

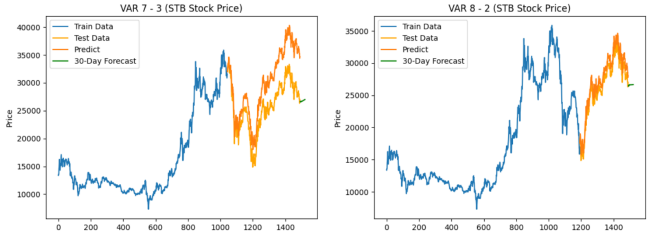


**FIGURE 17.** The result of VAR model

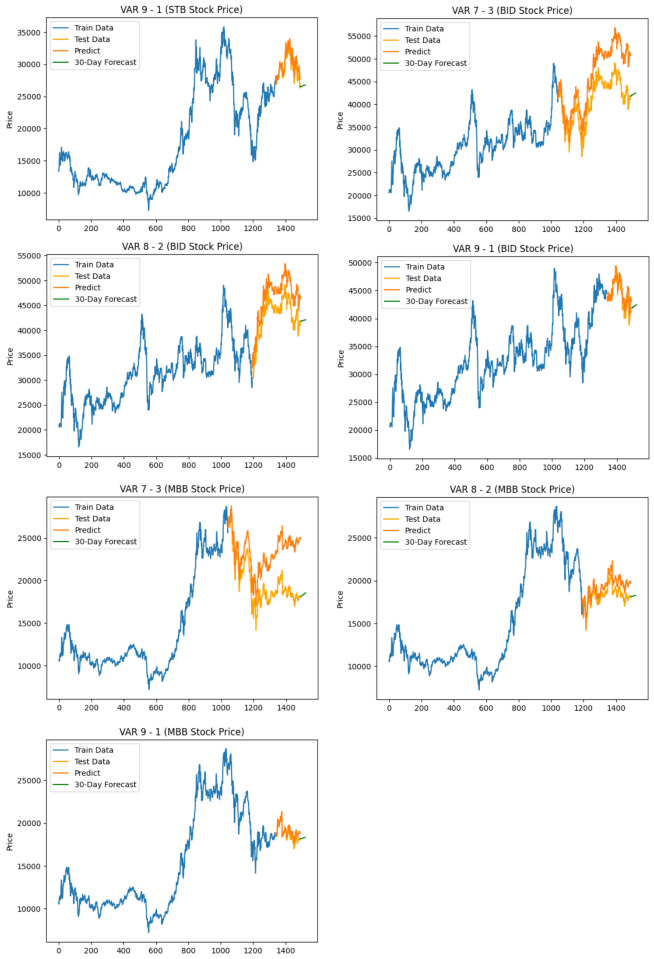


**FIGURE 18.** The result of VAR model

## 6) XGBoost

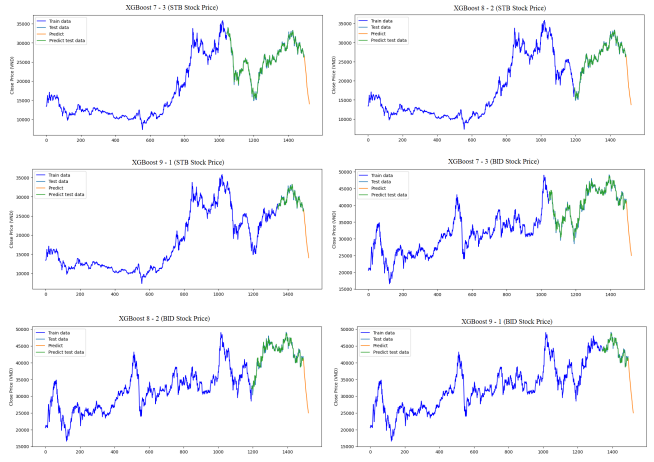


**FIGURE 19.** The result of XGBoost model

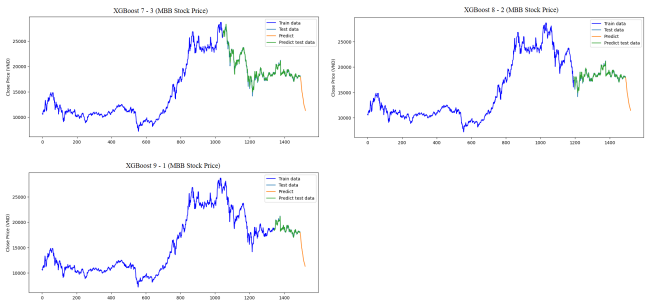


**FIGURE 20.** The result of XGBoost model

### 7) CNN-LSTM

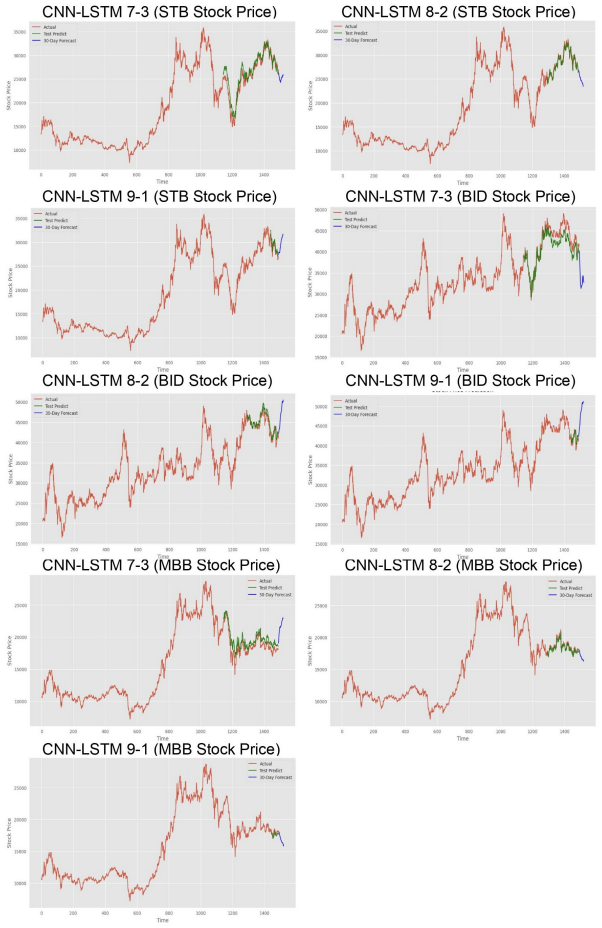


**FIGURE 21.** The result of CNN-LSTM model

### 8) AdaBoost

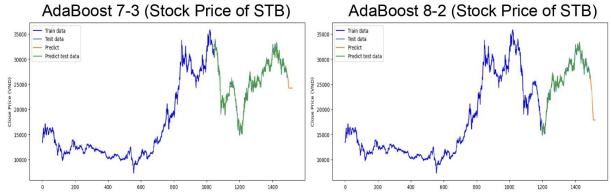


**FIGURE 22.** The result of AdaBoost model

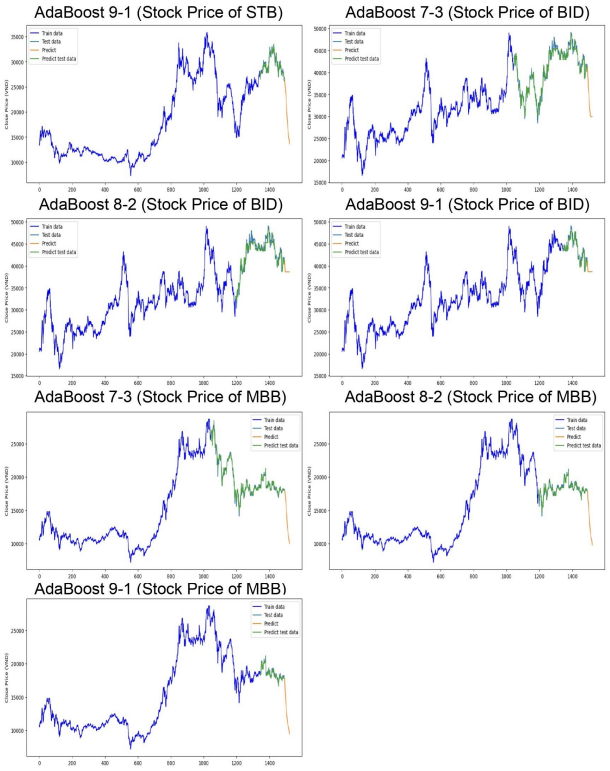


**FIGURE 23.** The result of AdaBoost model

## *B. EVALUATION MODELS AND DISCUSSION*

### 1) Evaluation models with STB dataset

**TABLE 3.** Performance Metrics for STB dataset

| Model | Ratio | RMSE | MAE | MAPE (%) |
|---|---|---|---|---|
| *Linear Regression* | 7-3 | 5167.63 | 4286.47 | 19.12 |
| | 8-2 | 4334.29 | 3243.14 | 15.06 |
| | 9-1 | 2315.66 | 1891.83 | 6.20 |
| *ARIMA* | 7-3 | 7725.97 | 6441.62 | 29.00 |
| | 8-2 | 9724.32 | 8924.92 | 32.12 |
| | 9-1 | 1862.66 | 1862.67 | 6.04 |
| *SVR* | 7-3 | 443.11 | 331.78 | 1.37 |
| | 8-2 | 401.96 | 316.23 | 1.28 |
| | 9-1 | 371.61 | 300.74 | 1.02 |
| *RNN* | 7-3 | 651.29 | 504.68 | 2.11 |
| | 8-2 | 576.10 | 448.36 | 1.56 |
| | 9-1 | 650.16 | 484.03 | 1.70 |
| *VAR* | 7-3 | 4898.41 | 4335.46 | 16.75 |
| | 8-2 | 1381.55 | 1170.21 | 4.56 |
| | 9-1 | 1095.14 | 858.52 | 2.94 |
| *XGBoost* | 7-3 | 380.75 | 299.25 | 1.22 |
| | 8-2 | 364.04 | 295.35 | 1.18 |
| | 9-1 | 359.59 | 290.60 | 0.98 |
| *CNN-LSTM* | 7-3 | 13394.63 | 11356.47 | 29.56 |
| | 8-2 | 7529.91 | 6837.71 | 20.68 |
| | 9-1 | 8681.70 | 8277.78 | 21.70 |
| *adaBoost* | 7-3 | 572.50 | 444.54 | 1.78 |
| | 8-2 | 483.70 | 403.58 | 1.58 |
| | 9-1 | 539.09 | 447.25 | 1.50 |

### 2) Evaluation models with BID dataset

**TABLE 4.** Performance Metrics for BID dataset

| Model | Ratio | RMSE | MAE | MAPE (%) |
|---|---|---|---|---|
| *Linear Regression* | 7-3 | 3928.70 | 3333.52 | 8.51 |
| | 8-2 | 4150.78 | 3593.61 | 8.34 |
| | 9-1 | 3235.16 | 2808.89 | 6.27 |
| *ARIMA* | 7-3 | 4874.41 | 4006.81 | 10.57 |
| | 8-2 | 11082.72 | 10438.79 | 23.70 |
| | 9-1 | 2456.93 | 2025.33 | 4.60 |
| *SVR* | 7-3 | 816.41 | 588.25 | 1.42 |
| | 8-2 | 916.76 | 657.92 | 1.50 |
| | 9-1 | 671.43 | 466.17 | 1.04 |
| *RNN* | 7-3 | 887.98 | 636.60 | 1.57 |
| | 8-2 | 780.73 | 534.92 | 1.22 |
| | 9-1 | 885.58 | 603.70 | 1.46 |
| *VAR* | 7-3 | 5922.78 | 5224.80 | 12.56 |
| | 8-2 | 3998.91 | 3719.29 | 8.67 |
| | 9-1 | 1524.42 | 1131.65 | 2.63 |
| *XGBoost* | 7-3 | 505.60 | 370.94 | 0.94 |
| | 8-2 | 496.47 | 360.50 | 0.86 |
| | 9-1 | 434.94 | 332.48 | 0.76 |
| *CNN-LSTM* | 7-3 | 16551.92 | 14273.81 | 24.66 |
| | 8-2 | 13411.61 | 10916.23 | 18.43 |
| | 9-1 | 22072.01 | 21637.34 | 33.59 |
| *adaBoost* | 7-3 | 768.57 | 594.65 | 1.45 |
| | 8-2 | 843.40 | 667.33 | 1.54 |
| | 9-1 | 647.07 | 534.11 | 1.21 |

### 3) Evaluation models with MBB dataset

**TABLE 5.** Performance Metrics for MBB dataset

| Model | Ratio | RMSE | MAE | MAPE (%) |
|---|---|---|---|---|
| *Linear Regression* | 7-3 | 5889.29 | 5273.74 | 28.22 |
| | 8-2 | 7094.71 | 6983.85 | 38.51 |
| | 9-1 | 4881.25 | 4721.33 | 25.56 |
| *ARIMA* | 7-3 | 10670.39 | 9892.94 | 53.21 |
| | 8-2 | 1040.33 | 802.08 | 4.45 |
| | 9-1 | 863.24 | 692.01 | 3.66 |
| *SVR* | 7-3 | 285.24 | 184.14 | 0.95 |
| | 8-2 | 219.60 | 158.86 | 0.89 |
| | 9-1 | 154.54 | 129.35 | 0.69 |
| *RNN* | 7-3 | 408.18 | 274.62 | 1.51 |
| | 8-2 | 316.27 | 200.51 | 1.08 |
| | 9-1 | 238.25 | 182.64 | 1.02 |
| *VAR* | 7-3 | 4148.14 | 3682.86 | 19.70 |
| | 8-2 | 1155.51 | 1009.18 | 5.54 |
| | 9-1 | 644.17 | 494.61 | 2.68 |
| *XGBoost* | 7-3 | 248.00 | 174.65 | 0.89 |
| | 8-2 | 200.01 | 147.07 | 0.82 |
| | 9-1 | 155.29 | 123.81 | 0.66 |
| *CNN-LSTM* | 7-3 | 11409.18 | 10173.17 | 32.04 |
| | 8-2 | 8610.85 | 7820.98 | 28.54 |
| | 9-1 | 6101.55 | 5991.01 | 25.15 |
| *adaBoost* | 7-3 | 314.47 | 223.89 | 1.14 |
| | 8-2 | 267.84 | 206.22 | 1.14 |
| | 9-1 | 280.84 | 243.25 | 1.30 |

### 4) Discussion

**TABLE 6.** Performance Metrics for Different Datasets and Train-Test Ratio 7-3

| Dataset | Train-Test Ratio 7-3 | | |
|---|---|---|---|
| | RMSE | MAE | MAPE |
| **STB** | 380.75 (XGBoost) | 299.25 (XGBoost) | 1.22 (XGBoost) |
| | 443.11 (SVR) | 444.54 (AdaBoost) | 1.36 (SVR) |
| **BIDV** | 768.57 (AdaBoost) | 370.94 (XGBoost) | 0.94 (XGBoost) |
| | 505.59 (XGBoost) | 588.24 (SVR) | 1.41 (SVR) |
| **MBB** | 248.00 (XGBoost) | 174.65 (XGBoost) | 0.894 (XGBoost) |
| | 314.47 (AdaBoost) | 223.88 (AdaBoost) | 0.95 (SVR) |

**TABLE 7.** Performance Metrics for Different Datasets and Train-Test Ratio 8-2

| Dataset | Train-Test Ratio 8-2 | | |
|---|---|---|---|
| | RMSE | MAE | MAPE |
| **STB** | 364.04 (XGBoost) | 316.23 (SVR) | 1.17 (XGBoost) |
| | 401.95 (SVR) | 403.57 (AdaBoost) | 1.28 (SVR) |
| **BIDV** | 496.47 (XGBoost) | 360.49 (XGBoost) | 0.86 (XGBoost) |
| | 843.39 (AdaBoost) | 534.91 (RNN) | 1.22 (RNN) |
| **MBB** | 200.00 (XGBoost) | 148.07 (XGBoost) | 0.81 (XGBoost) |
| | 219.60 (SVR) | 158.86 (SVR) | 0.88 (SVR) |

**TABLE 8.** Performance Metrics for Different Datasets and Train-Test Ratio 9-1

| Dataset | Train-Test Ratio 9-1 | | |
|---|---|---|---|
| | RMSE | MAE | MAPE |
| **STB** | 359.59 (XGBoost) | 290.60 (XGBoost) | 0.98 (XGBoost) |
| | 539.09 (AdaBoost) | 300.74 (SVR) | 1.02 (SVR) |
| **BIDV** | 434.94 (XGBoost) | 332.48 (XGBoost) | 0.76 (XGBoost) |
| | 539.09 (AdaBoost) | 466.17 (SVR) | 1.04 (SVR) |
| **MBB** | 154.54 (SVR) | 123.81 (XGBoost) | 0.66 (XGBoost) |
| | 155.29 (XGBoost) | 129.35 (SVR) | 0.69 (SVR) |

### C. STOCK PRICE PREDICTIONS FOR THE NEXT 30 DAYS

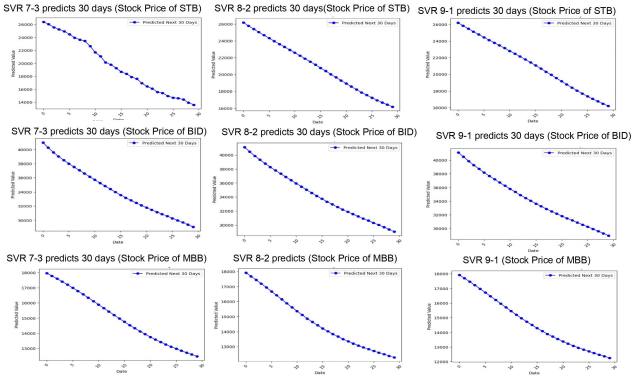


**FIGURE 24.** 30-Days Prices Forecast Result from SVR Model

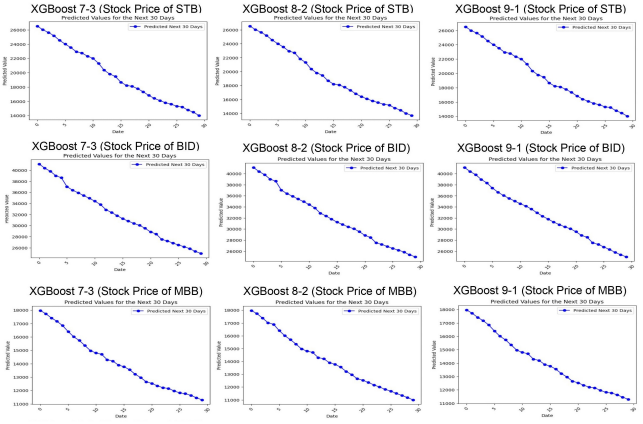


**FIGURE 25.** 30-Days Prices Forecast Result from XGBoost Model

## VI. CONCLUSION

### A. OVERALL CONCLUSION

In conclusion, predicting stock prices is complex, influenced by factors like market sentiment and economic indicators. The intricate and dynamic nature of stock price data poses unique challenges for models. In our project, eight models

were implemented with three data split ratios: 7-3, 8-2, and 9-1. Evaluation metrics (RMSE, MAPE, MAE) showed that SVR, XGBoost, and AdaBoost effectively identify patterns, while RNN excels in capturing long-term dependencies. Integrating both machine learning and deep learning provides a comprehensive solution, leveraging their strengths for accurate stock price predictions. Future work involves exploring advanced deep learning architectures, integrating diverse data sources, refining feature engineering, exploring ensembling techniques, and evolving evaluation metrics.

### B. CHALLENGES ENCOUNTERED

During the implementation of the research project " FORECASTING VIETNAM BANKING'S STOCK PRICES USING TIME SERIES APPROACH." we encountered several challenges, including:

The complexity of processing stock data requires the application of precise and scientific methods to ensure accuracy. Our goal was to guarantee the viability and precision of our prediction models.

Developing sophisticated prediction models for stocks involves leveraging profound domain knowledge. Critical decisions, including the selection of suitable algorithms, defining data processing techniques, and identifying key variables to incorporate into the models, had to be made.

Creating sophisticated prediction models for stocks necessitates a profound understanding of the domain. Critical decisions, including the selection of suitable algorithms, determination of data processing techniques, and identification of significant variables for inclusion in the models, were imperative.

Assessing the effectiveness of our models posed a challenge. We utilized a range of algorithmic and statistical indicators for evaluation. Despite these efforts, the outcomes indicated that the accuracy of the models remained unsatisfactory.

### C. FUTURE INTENTIONS

To address the challenges outlined, we've devised strategies for improving stock price modeling accuracy:

Strengthen foundational knowledge: Prioritizing mathematical concepts like calculus, linear algebra, and statistics alongside research methodologies to enhance understanding.

Improve data searching and preprocessing: Continuously exploring advanced techniques to augment model robustness.

Embrace advanced models: Deploying diverse models on datasets to adapt to varied time series data and identify the most effective models.

Team-based projects: Dedicate more time to collaborative projects, fostering a mutually supportive environment for collective success.

## ACKNOWLEDGMENT

We extend sincere appreciation to Prof. Assoc. Prof. Dr. Nguyen Dinh Thuan and TA. Nguyen Minh Nhut for their steadfast dedication, guiding us through this project and imparting invaluable lessons that will shape our future endeavors. The completion of our project would not have been possible without their supervision. Throughout this endeavor, we encountered challenges, but together, we successfully surmounted them and completed our work. This project provided us with a hands-on opportunity to participate in real-world exercises. Additionally, we express gratitude to our fellow teammates for their supportive contributions to our project, from whom we gained valuable insights.

## REFERENCES

[1] A Linear Regression Model for Predicting Sales" by A. Smith and B. Jones (2023)

[2] Zhang, G.P.: Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. Neurocomputing 50, 159-175

[3] Bui, T. K., Tran, T. H., Thai, D. T., Nguyen, N. D., and Nguyen, V. D. (2022) [Factor affecting the error in individual stock's return forecasting: Appling machine learning with Spark MLlib]

[4] I. Amalou, N. Mouhni, and A. Abdali, "Multivariate time series prediction by RNN architectures for energy consumption forecasting," Energy Reports, vol. 8, no. 9, pp. 1084-1091, 2022. DOI: 10.1016/j.egyr.2022.07.139.

[5] Liu, Y., Zhang, Y., and Wang, L. (2021). Forecasting customer demand with vector autoregressive models: A case study of a smartphone retailer. Journal of Retailing and Consumer Services, 60, 102376.

[6] Tianqi Chen-University of Washington, and Carlos Guestrin-University of Washington: XGBoost: A Scalable Tree Boosting System

[7] V. A. Bharadi, S. Alegavi and B. Nemade, "Using CNN-LSTMs and Transformer RNNs for COVID19 Impact Prediction," 2023 International Conference on Network, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2023, pp. 1-10, doi: 10.1109/NMITCON58196.2023.10275886.

[8] Maulud, D., and Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning. Journal of Applied Science and Technology Trends, 1(4), 140-147.

[9] Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, April 2015, Rahul Khanna, Mariette Awad. Chapter 4 Support Vectoer Regression pages 68.

[10] Modelling Financial Time Series with S-PLUS, Second Edition - Eric Zivot and Jiahui Wang, March 30, 2006. Vector Autoregression pages 405-406.

[11] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," Neural Computing and Applications, vol. 32, pp. 17351–17360, 2020, doi: 10.1007/s00521-020-04867-x.

[12] Improved Boosting Algorithms - Using Confidence-rated Predictions, Machine Learning, 37, 297–336 (1999), 1999 Kluwer Academic Publishers. Manufactured in The Netherlands, pages 300-301.