



FORECASTING OF BANK'S STOCK PRICE USING USING STATISTIC MODEL, MACHINE LEARNING AND DEEP LEARNING

DANG QUANG NHAT¹, TRAN PHUC THINH², AND NGUYEN DAI DUONG³

¹21522413, Faculty of Information Systems, University of Information Technology, (e-mail: 21522413@gm.uit.edu.vn)

²21521475, Faculty of Information Systems, University of Information Technology, (e-mail: 21521475@gm.uit.edu.vn)

³21520756, Faculty of Information Systems, University of Information Technology, (e-mail: 21520756@gm.uit.edu.vn)

ABSTRACT

Forecasting stock market trends is crucial for investment decision-making, especially in the Vietnamese banking sector. This scientific report examines the application of statistical models and machine learning techniques to predict the future movements of three major Vietnamese bank stocks. Eight forecasting models, including ARIMA, GRU, linear regression, GARCH, MICN, RNN, LSTM, and GradientBoosting, provide valuable insights into the dynamics and trends within this crucial sector of the Vietnamese economy. The findings are expected to contribute to the existing knowledge on stock market forecasting and inform the development of investment strategies and risk management practices in the Vietnamese banking stock market.

INDEX TERMS: bank stocks in Vietnam, stock market forecasting, statistical models, machine learning, neural network, ARIMA, GRU, Linear Regression, GARCH, RNN, MICN, GradientBoosting, LSTM.

I. INTRODUCTION

The stock market is characterized as dynamic, unpredictable, and non-linear in nature. Predicting stock prices is a challenging task as it depends on various factors including, but not limited to, political conditions, the global economy, and company financial reports and performance. Thus, to maximize profit and minimize losses, techniques to predict stock values in advance by analyzing trends over the last few years could prove to be highly useful for making stock market movements. [1] [2]

While Vietnam's booming stock market entices investors, accurately predicting prices within it proves difficult due to the general complexities of financial markets. This study tackles this challenge by applying machine learning and deep learning algorithms to forecast the value of stocks from three major Vietnamese banks: Vietcombank, ACB, and VietinBank.

This research explores how various machine and deep learning algorithms can predict stock prices in Vietnam. The study will test popular algorithms, including Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), Recurrent Neural Networks (RNNs), Linear Regression, and Autoregressive Integrated Moving Average (ARIMA) models, to determine their effectiveness in capturing stock price trends. Each algorithm's unique strengths in modeling stock prices will be evaluated using performance metrics such as

Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE).

To further improve accuracy the study will experiment with advanced techniques such as MICN (Multi-scale Isometric Convolution Network), Gradient Boosting Regressors, and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models.

Overall, this research aims to improve the accuracy of stock price predictions in Vietnam's stock market using machine and deep learning models. By making these predictions more reliable, investors and other stakeholders can make smarter investment decisions.

II. RELATED WORKS

The application of machine learning and deep learning algorithms to stock price prediction has garnered significant attention in recent years. Various studies have explored different models and techniques to enhance prediction accuracy and reliability. Most previous work in this area utilizes classical algorithms like linear regression [3] and linear models such as Autoregressive Integrated Moving Average (ARIMA) [4] for predicting stock prices. Some techniques based on neural networks, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and deep neural networks like Long Short-Term Memory (LSTM), have also shown promising results [5] [6]. Recent advance-

ments in stock forecasting have seen a rise in the use of Gated Recurrent Units (GRUs). Studies have compared GRU models to other methods like Support Vector Machines (SVM) for predicting trading signals based on stock indicators [7]. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have proven effective in time series forecasting. LSTMs excel at capturing long-term dependencies within data, making them suitable for analyzing historical trends alongside current data for stock price prediction [8] [9]. In recent years, researchers have applied LSTMs to various stock markets globally. For example, Chen et al. [10] utilized LSTMs to forecast China's Shanghai and Shenzhen stock markets. In addition to traditional neural networks, advanced techniques like Gradient Boosting Regressors and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models have been explored. Gradient Boosting is an ensemble technique that builds multiple weak learners, usually decision trees, to improve predictive accuracy. GARCH models are used to predict volatility in financial markets and have been integrated with other predictive models to enhance performance. Kim, Jong-Min et al [11]. Comparing the Performances of GARCH-type Models in Capturing the Stock Market Volatility in Malaysia Gradient Boosting has been extensively applied in financial time-series modeling due to its ability to handle complex, non-linear relationships within data. Regularised Gradient Boosting, in particular, has shown promise in improving predictive accuracy and robustness. Agapitos, Brabazon, and O'Neill (2017) [12] explored the application of Regularised Gradient Boosting for financial time-series modeling. Their study highlighted the advantages of using gradient boosting algorithms in capturing the intricate patterns present in financial data. The Multi-scale Isometric Convolution Network (MICN) model by Wang et al. (2022) [13] represents a significant advancement in long-term series forecasting. MICN integrates multi-scale local and global context modeling to effectively capture complex patterns and dependencies within time-series data.

Prior to MICN, Convolutional neural networks (CNN) are widely used in computer vision, natural language processing and speech recognition. Meanwhile, Transformer architectures (Vaswani et al., 2017) [14] introduced attention mechanisms for learning intricate temporal relationships across different scales.

MICN distinguishes itself by leveraging multi-scale isometric convolutions, which enhance its ability to handle both short-term fluctuations and long-term trends in time-series forecasting. This approach improves prediction accuracy and interpretability, making MICN suitable for applications in finance, climate science, and healthcare.

III. MATERIALS

A. DATASET

Dataset We will be analyzing historical stock data for Joint Stock Commercial Bank For Foreign Trade Of Vietnam (VCB), Asia Commercial Joint Stock Bank (ACB), and Viet-

nam Joint Stock Commercial Bank for Industry and Trade (CTG), covering the period from March 1st, 2019 to June 1st, 2024. The dataset includes columns such as Date, Open, High, Low, Close, Adj Close, and Volume. Our focus will be on processing the "Close" price for forecasting purposes.

B. DESCRIPTIVE STATISTICS

TABLE 1. VCB, ACB, CTG's Descriptive Statistics

	VCB	ACB	CTG
Count	1311	1313	1317
Mean	66758.246	15745.649	23713.87
Std	13660.147	5246.805	6526.769
Min	37957.312	6626.145	11925.876
25%	56604.175	9157.481	16710.095
50%	65198.984	18071.833	25192.072
75%	75867.910	19608.695	28635.972
Max	97400	25782.609	37719.05

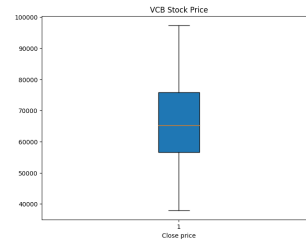


FIGURE 1. VCB stock price's boxplot

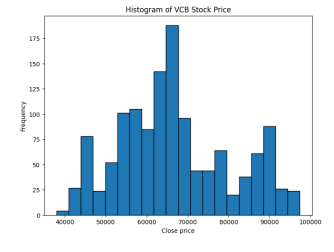


FIGURE 2. VCB stock price's histogram

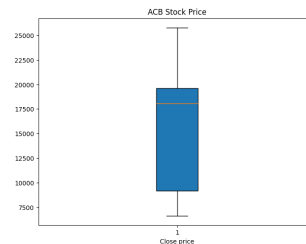


FIGURE 3. ACB stock price's boxplot

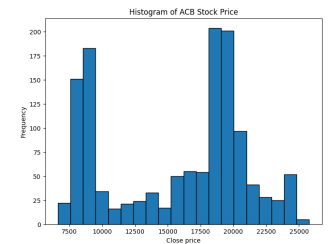


FIGURE 4. ACB stock price's histogram

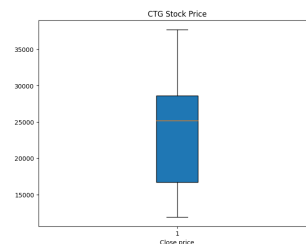


FIGURE 5. CTG stock price's boxplot

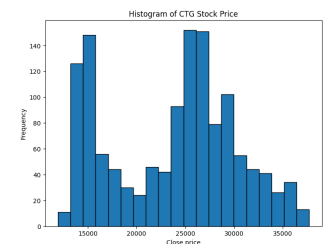


FIGURE 6. CTG stock price's histogram

IV. METHODOLOGY

A. LINEAR REGRESSION

Linear regression is a statistical procedure for calculating the value of a dependent variable from an independent variable.

Linear regression measures the association between two variables. It is a modeling technique where a dependent variable is predicted based on one or more independent variables. Linear regression analysis is the most widely used of all statistical techniques. [15] A multiple linear regression model has the form: [16]

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon$$

Where:

- Y is the dependent variable (Target Variable).
- X_1, X_2, \dots, X_k are the independent (explanatory) variables.
- β_0 is the intercept term.
- β_1, \dots, β_k are the regression coefficients for the independent variables.
- ε is the error term.

B. RNN

A recurrent neural network (RNN) is a type of artificial neural network designed to handle sequential or time series data. RNNs are a form of deep learning architecture that is particularly adept at predicting subsequent steps in a sequence. Their distinguishing feature is the ability to remember past inputs through internal memory, making them well-suited for tasks such as stock price prediction, text generation, transcription, and machine translation.

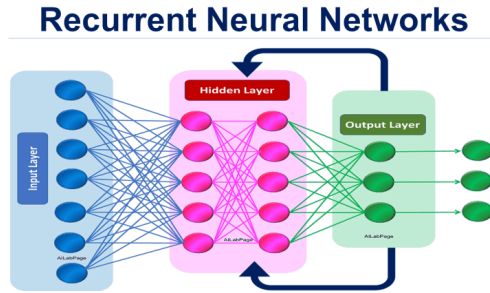


FIGURE 7. Model of RNN

In RNN architectures, different time steps share the same weight, allowing the network to form cyclical connections over time. This weight sharing significantly reduces the temporal parameters in the RNN network system.

However, one limitation of RNNs is their capacity to retain only a limited number of steps from previous sequences, which makes them less effective for longer sequences. To address this limitation, Long Short-Term Memory (LSTM) networks were introduced, offering improved capabilities for handling longer sequences.

C. ARIMA

ARIMA models (Autoregressive Integrated Moving Average) were first introduced by George Box and Gwilym Jenkins in the early 1970s. They have since become a fundamental tool in time series analysis and forecasting. ARIMA model is commonly denoted as (p, d, q) :

- AR (Autoregressive): Auto-Regressive (AR): using a linear combination of past values of the variable. An autoregressive model of order p can be written:

$$Y_t = \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \dots + \Phi_p Y_{t-p} + \varepsilon_t$$

Where:

- Y_t is the current value.
- $\Phi_1, \Phi_2, \dots, \Phi_p$ are model parameters.
- ε_t is the random error.
- I (Integrated): refers to the differentiation of the time series data.
- Moving Average (MA): uses past forecast errors in a regression-like model. “ q ” is the number of previous error values to consider for the forecast.

$$Y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}$$

Where:

- Y_t is the current value.
- ε_t is the random error.
- $\theta_1, \theta_2, \dots, \theta_q$ are coefficients.
- c is a constant.

D. LSTM

The Long Short-Term Memory (LSTM) model represents a widely used variant of recurrent neural networks (RNNs). Specifically designed to address the challenge of long-term dependencies, LSTM is well-suited for processing and predicting time series data. This model employs a sophisticated gating mechanism to regulate the flow of information within memory cells. The gate structure includes input gates, forget gates, and output gates, each managed by sigmoid and tanh layers[8].

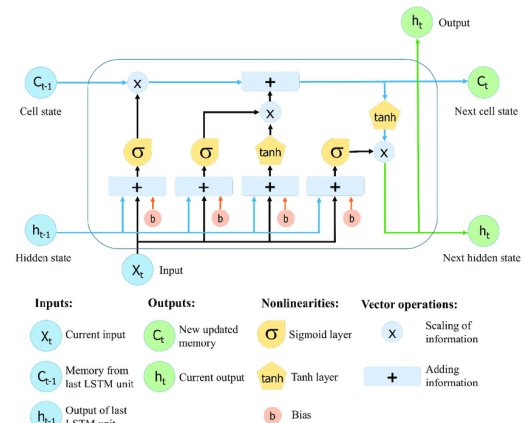


FIGURE 8. Model of LSTM

Schematic diagram of the LSTM neural network structure:

$$\text{Sigmoid function: } \sigma(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

$$\text{Tanh activation function: } \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\text{Forget gate: } f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (4)$$

$$\text{Input gate: } C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t \quad (5)$$

(Where the calculation methods of i_t and \hat{C}_t are shown in Eqs. 4 and 5)

$$\text{Output gate: } i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (6)$$

$$\hat{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (7)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (9)$$

Bi-LSTM: A bidirectional LSTM consists of two LSTM layers: a forward LSTM that processes the input sequence from start to end, and a backward LSTM that processes it from end to start. The outputs from both LSTMs are concatenated to produce the final output. By leveraging information from both past and future contexts, bidirectional LSTMs excel in tasks where understanding the entire context of the input sequence is crucial, such as natural language processing.

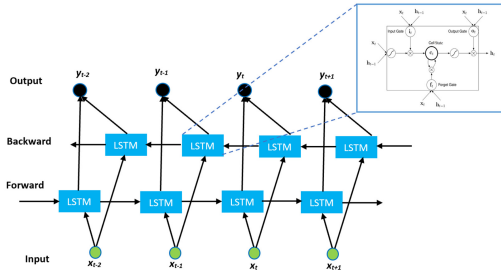


FIGURE 9. Model of LSTM

E. GRU

Gated Recurrent Unit (GRU) is a variant of the recurrent neural network (RNN) architecture, designed to improve the ability to store and process information in long-term data sequences without losing information. GRU uses gate mechanisms to selectively adjust the flow of information into and out of the hidden states of the network, allowing for efficient and selective updates of the state. The two main gates in GRU are the reset gate and the update gate.

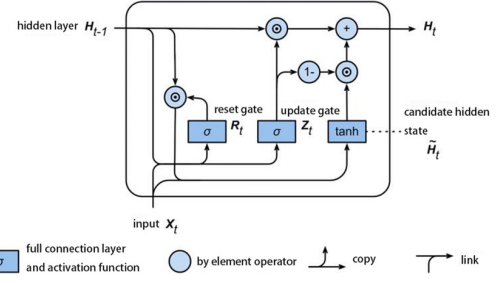


FIGURE 10. Model of GRU

- Reset gate: Adjusts the amount of information from the previous hidden state that needs to be forgotten.
- Update gate: Decides the amount of new information from the input that will be used to update the hidden state.

GRU computes the output based on the updated hidden state through these gates, helping to maintain information effectively throughout the sequence processing. Below are the equations describing the GRU computation:

- Reset gate:

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (1)$$

- Update gate:

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (2)$$

- Candidate hidden state:

$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t * h_{t-1}) + b_h) \quad (3)$$

- Final hidden state:

$$h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t \quad (4)$$

Where:

- W_{xr}, W_{xz}, W_{xh} are the learnable weight matrices.
- x_t is the input at time step t .
- h_{t-1} is the previous hidden state.
- σ is the sigmoid activation function and \tanh is the hyperbolic tangent activation function.
- b_r, b_z, b_h are the bias terms.

By selectively adjusting the information flow through the gates, GRU can effectively handle long-term data sequences, especially useful in applications such as cryptocurrency price forecasting. GRU is simpler than LSTM but still maintains the powerful ability to process complex data sequences.

F. GARCH

The GARCH (Generalized AutoRegressive Conditional Heteroskedasticity) model is a statistical model that describes how the variance of error in a time series depends on past errors and variances. Unlike simple models that assume constant variance (homoskedasticity), GARCH allows for volatility, which means that periods of high and low volatility can change based on historical data.

The general form of a GARCH(p, q) model is:

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-2}^2 + \dots + \alpha_p \varepsilon_{t-p}^2$$

$$+ \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2 + \dots + \beta_q \sigma_{t-q}^2$$

Where:

- σ_t^2 represents the conditional variance at time t .
- ω is the constant term or intercept.
- $\alpha_1, \alpha_2, \dots, \alpha_p$ are numeric coefficient for the squared residual for the past period.
- $\beta_1, \beta_2, \dots, \beta_q$ are numeric coefficient for the conditional variance from last period.
- $\varepsilon_{t-1}^2, \varepsilon_{t-2}^2, \dots, \varepsilon_{t-p}^2$ are squared residual for the past period.
- $\sigma_{t-1}^2, \sigma_{t-2}^2, \dots, \sigma_{t-q}^2$ are the past conditional variances.

In case GARCH(1:1), the formula will be:

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

G. GRADIENT BOOSTING REGRESSOR

Gradient Boosting Regressor (GBR) is an ensemble machine learning technique that sequentially builds a series of weak learners, typically decision trees, to minimize a loss function and improve predictive accuracy.

Let $\{(x_i, y_i)\}_{i=1}^n$ be a training dataset where x_i represents the input features and y_i denotes the corresponding target values. GBR aims to approximate the target function $F(x)$ by iteratively fitting additive models:

$$F(x) = \sum_{m=1}^M \gamma_m h_m(x)$$

where $h_m(x)$ is the m -th weak learner (often a decision tree), and γ_m is the learning rate for the m -th iteration.

The GBR algorithm can be summarized as follows:

1. Initialize the model with a constant value, typically the mean of the target values $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$.
2. For $m = 1$ to M :
 - Compute the negative gradient of the loss function with respect to the current model predictions $-\frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}$.
 - Fit a weak learner $h_m(x)$ to predict the negative gradient residuals.
 - Determine the optimal step size γ_m using line search or another optimization method.
 - Update the model: $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$.

3. Continue iterating until a predefined number of iterations M is reached or until further iterations do not significantly improve performance.

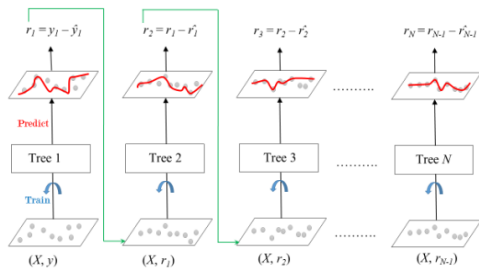


FIGURE 11. Model of GBR

GBR combines the strengths of gradient descent optimization with boosting, effectively minimizing the error through

an ensemble of weak learners. This method is particularly effective in scenarios where complex relationships exist between input features and target variables, making it widely used in predictive modeling across various domains.

H. MULTI-SCALE ISOMETRIC CONVOLUTION NETWORK

1) MICN Model

The Multi-scale Isometric Convolution Network (MICN), presented by Wang et al. in 2022, represents a significant advancement in long-term series forecasting. This model integrates multi-scale local and global context modeling techniques to effectively capture complex patterns and dependencies within time-series data.

2) MICN Framework

MICN Framework Overview: The MICN framework aims to predict a future time series of length O based on a historical series of length I . It incorporates multi-scale hybrid decomposition (MHDecomp) blocks to separate complex patterns in the input series. It uses Seasonal Prediction Blocks to forecast seasonal information and Trend-cyclical Prediction Blocks to predict trend-cyclical patterns. The final prediction Y_{pred} is obtained by aggregating these prediction results. Here, d represents the number of variables in the multivariate time series, and D denotes the hidden state of the series.

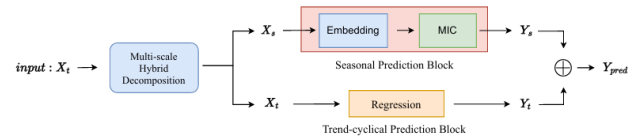


FIGURE 12. Model of MICN

For the input series $X \in \mathbb{R}^{I \times d}$:

Trend-Cyclical Prediction Block: MICN employs a linear regression strategy within the Trend-Cyclical Prediction Block to forecast the trend component of the time series. This method contrasts with simpler averaging techniques used in previous models, demonstrating improved forecasting accuracy for non-stationary series.

$$Y_t^{regre} = \text{regression}(X_t)$$

where $Y_t^{regre} \in \mathbb{R}^{O \times d}$ denotes the prediction of the trend part using the linear regression strategy.

Seasonal Prediction Block: The Seasonal Prediction Block focuses on modeling complex seasonal variations using multi-scale isometric convolution. It embeds input sequences to capture local features and global correlations across different scales, enhancing the model's capability to handle seasonal patterns robustly.

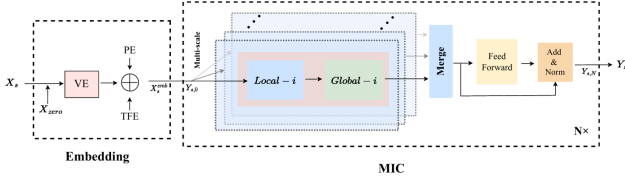


FIGURE 13. Model of MICN

3) Methodologies

Multi-scale Isometric Convolution: MICN utilizes multi-scale isometric convolution to capture both local and global dependencies within time-series data. This technique allows the model to extract comprehensive information from input sequences and improve forecasting accuracy.

Embedding Layer: The embedding layer in MICN integrates time features encoding, positional encoding, and value embedding to represent input sequences effectively. This approach minimizes redundant calculations and adapts well to varying prediction lengths O .

V. RESULT

A. EVALUATION METHODS

Mean Percentage Absolute Error (MAPE): is a metric for evaluating the accuracy of a forecasting model. It measures the average magnitude of errors between predicted and actual values, expressed as a percentage. MAPE is defined as:

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100$$

where:

- n is the number of observations.
- y_t is the actual value at time t .
- \hat{y}_t is the predicted value at time t .

Root Mean Squared Error (RMSE): measures the square root of the average of the squared differences between the predicted and actual values. It is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

where:

- n is the number of observations.
- y_t is the actual value at time t .
- \hat{y}_t is the predicted value at time t .

Mean Absolute Error (MAE): measures the average magnitude of the errors between predicted and actual values without considering their direction. MAE is defined as:

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t|$$

where:

- n is the number of observations.
- y_t is the actual value at time t .
- \hat{y}_t is the predicted value at time t .

B. VCB DATASET

VCB Dataset's Evaluation				
Model	Training:Testing	MAPE (%)	RMSE	MAE
LN	7:3	9.8	9308.642	8307.594
	8:2	11.6	10929.857	10378.652
	9:1	6.9	7141.888	6311.754
ARIMA	7:3	1.04	1175.745	848.698
	8:2	0.92	1128.364	813.264
	9:1	0.807	1053.277	728.787
RNN	7:3	2.756	84912.220	84743.142
	8:2	2.768	87906.756	87839.897
	9:1	1.676	92428.848	92400.334
LSTM	7:3	1.83	2064.552	1601.095
	8:2	2.13	2328.359	1922.049
	9:1	1.54	2122.498	1455.057
GRU	7:3	1.04	1240.390	911.386
	8:2	0.99	1231.421	889.385
	9:1	0.6	798.720	614.843
GARCH	7:3	nan	2.097	1.741
	8:2	nan	1.950	1.661
	9:1	nan	1.865	1.617
GBR	7:3	8.6	9507.772	7663.113
	8:2	9.3	9202.547	8350.244
	9:1	1.9	2230.027	1748.122
MICN	7:3	19.0	18350.763	16502.455
	8:2	22.2	20136.471	19749.414
	9:1	16.6	15341.417	15027.681

TABLE 2. VCB Dataset's Evaluation



FIGURE 14. LN model's result with 9:1 splitting proportion

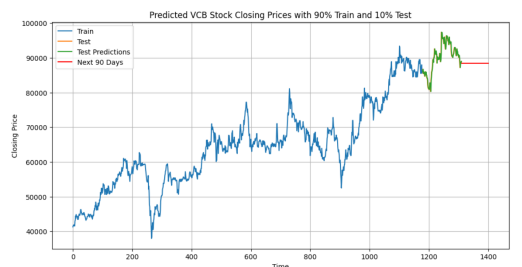


FIGURE 15. ARIMA model's result with 9:1 splitting proportion

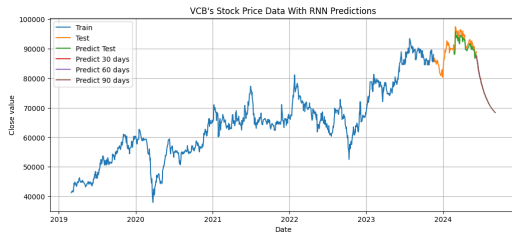


FIGURE 16. RNN model's result with 9:1 splitting proportion

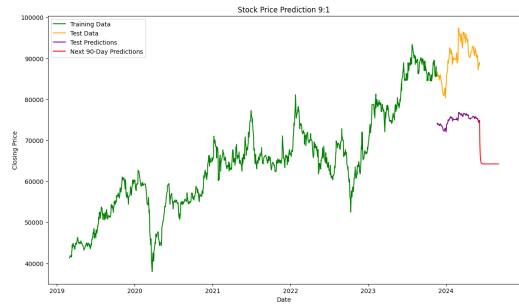


FIGURE 21. MICN model's result with 9:1 splitting proportion

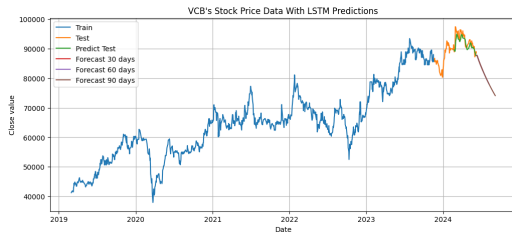


FIGURE 17. LSTM model's result with 9:1 splitting proportion

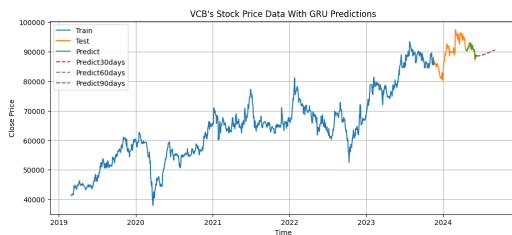


FIGURE 18. GRU model's result with 9:1 splitting proportion

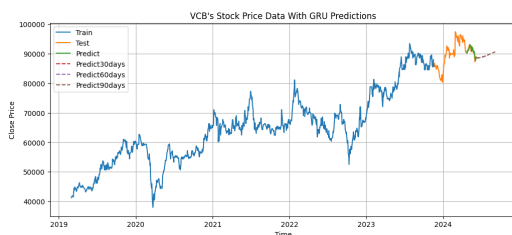


FIGURE 19. GARCH model's result with 9:1 splitting proportion



FIGURE 20. GBR model's result with 9:1 splitting proportion

C. ACB DATASET

ACB Dataset's Evaluation				
Model	Training:Testing	MAPE (%)	RMSE	MAE
LN	7:3	24.1	4812.948	4679.671
	8:2	11.7	2682.869	2320.709
	9:1	5.3	1454.617	1158.013
ARIMA	7:3	1.06	294.170	208.677
	8:2	0.90	274.825	192.154
	9:1	0.95	308.451	218.907
RNN	7:3	2.01	20638.496	20543.834
	8:2	2.37	21701.767	21614.888
	9:1	3.03	23590.512	23585.406
LSTM	7:3	3.23	917.183	705.928
	8:2	2.39	666.151	531.216
	9:1	3.45	963.564	840.762
GRU	7:3	1.02	307.166	218.966
	8:2	1.37	409.880	310.602
	9:1	1.23	388.034	301.499
GARCH	7:3	nan	2.171	1.718
	8:2	nan	1.832	1.514
	9:1	nan	1.848	1.518
GBR	7:3	3.6	1418.069	819.109
	8:2	4.6	1677.833	1086.011
	9:1	7.8	2260.802	1865.096
MICN	7:3	6.6	2186.285	1438.314
	8:2	15.1	3922.045	3363.858
	9:1	19.3	4786.054	4513.602

TABLE 3. ACB Dataset's Evaluation



FIGURE 22. LN model's result with 9:1 splitting proportion

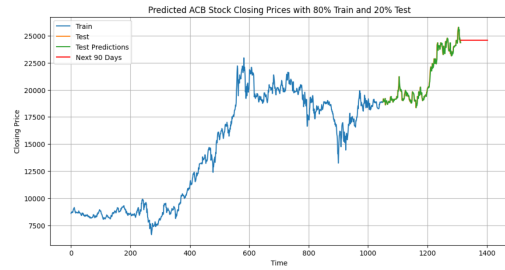


FIGURE 23. ARIMA model's result with 8:2 splitting proportion

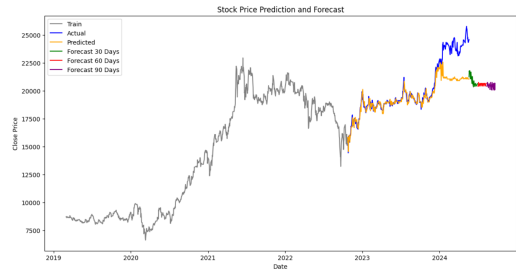


FIGURE 28. GBR model's result with 7:3 splitting proportion

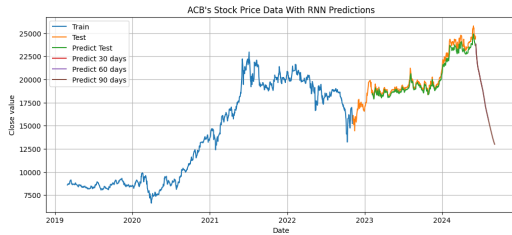


FIGURE 24. RNN model's result with 7:3 splitting proportion

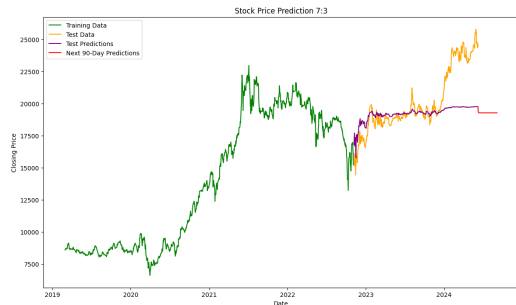


FIGURE 29. MICN model's result with 7:3 splitting proportion



FIGURE 25. LSTM model's result with 9:1 splitting proportion

D. CTG DATASET

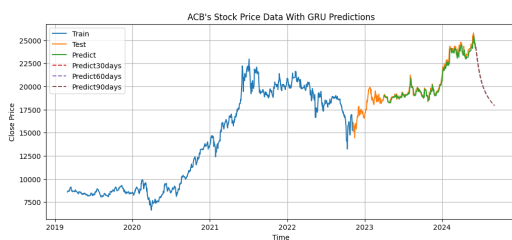


FIGURE 26. GRU model's result with 7:3 splitting proportion

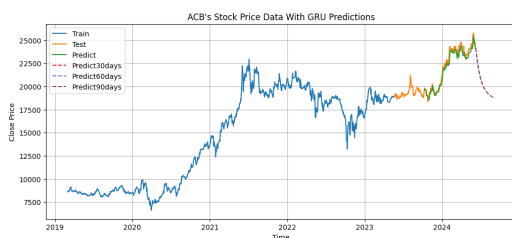


FIGURE 27. GARCH model's result with 8:2 splitting proportion

CTG Dataset's Evaluation				
Model	Training:Testing	MAPE (%)	RMSE	MAE
LN	7:3	23.5	6628.914	6240.569
	8:2	11.8	3731.196	3233.027
	9:1	8.4	3032.375	2643.229
ARIMA	7:3	1.336	523.846	371.441
	8:2	1.187	521.978	356.312
	9:1	1.294	610.492	422.452
RNN	7:3	2.465	28772.902	28581.666
	8:2	2.353	30566.344	30365.371
	9:1	2.568	33546.902	33520.913
LSTM	7:3	3.127	1141.069	924.030
	8:2	2.304	936.059	713.297
	9:1	3.542	1411.940	1207.337
GRU	7:3	1.236	508.833	363.608
	8:2	1.360	595.657	427.719
	9:1	1.344	574.250	440.853
GARCH	7:3	nan	2.641	2.088
	8:2	nan	2.425	1.945
	9:1	nan	2.594	2.059
GBR	7:3	1.6	615.558	441.555
	8:2	1.4	626.922	432.788
	9:1	1.7	781.218	568.495
MICN	7:3	9	3456.450	2711.102
	8:2	5.7	2695.131	1861.530
	9:1	12.5	4704.741	4176.037

TABLE 4. CTG Dataset's Evaluation

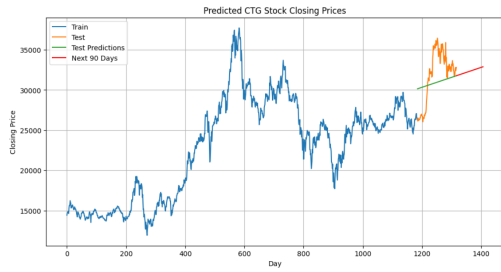


FIGURE 30. CTG model's result with 9:1 splitting proportion

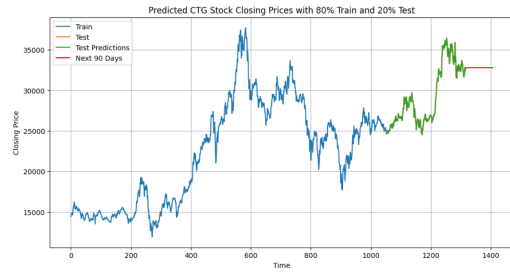


FIGURE 31. ARIMA model's result with 8:2 splitting proportion

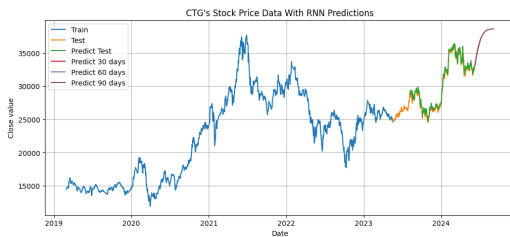


FIGURE 32. RNN model's result with 8:2 splitting proportion

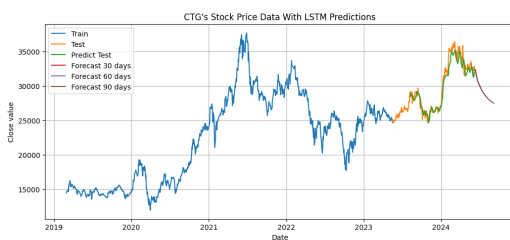


FIGURE 33. LSTM model's result with 8:2 splitting proportion

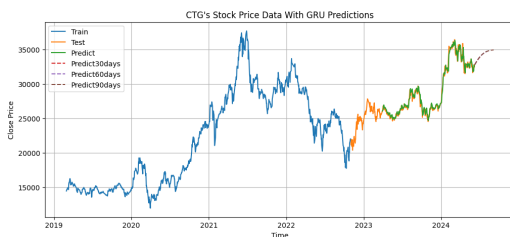


FIGURE 34. GRU model's result with 7:3 splitting proportion

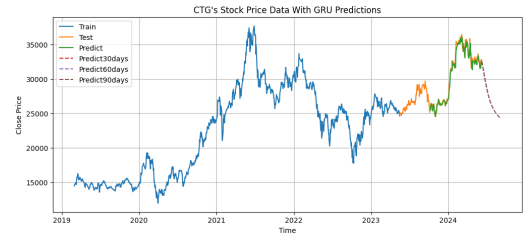


FIGURE 35. GARCH model's result with 8:2 splitting proportion



FIGURE 36. GBR model's result with 8:2 splitting proportion

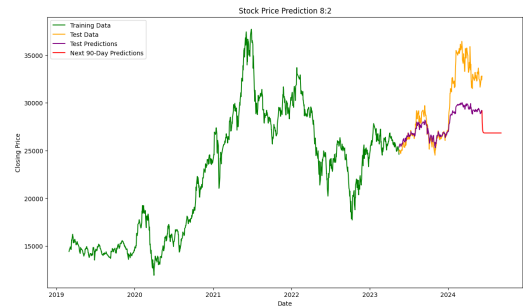


FIGURE 37. MICN model's result with 8:2 splitting proportion

VI. CONCLUSION

A. SUMMARY

The results of this study highlight that out of the eight models tested (Linear Regression, ARIMA, RNN, LSTM, GRU, GARCH, GBR, and MICN), the GRU model emerged as the most suitable for predicting future values of VCB, ACB, and CTG bank stocks within the respective time series. This underscores the significant impact of algorithm selection on forecasting accuracy. Further development of deep learning algorithms in this area requires fine-tuning input parameters and expanding sample sizes. However, these efforts are challenged by the complex interplay of social variables that can drastically influence forecasting outcomes. Therefore, the findings presented here are for reference purposes and should be carefully considered in relation to specific research inquiries proposed by the investigator.

ACKNOWLEDGMENT

First and foremost, we extend our heartfelt gratitude to **Assoc. Prof. Dr. Nguyen Dinh Thuan** for his invaluable

guidance and enthusiastic support throughout this project. His lectures have provided our team with the knowledge and inspiration needed to successfully accomplish our goals. We would also like to express our sincere thanks to **Ms. Trinh Thi Thanh Truc** for her unwavering support during the course of this endeavor. We appreciate **Assoc. Prof. Dr. Nguyen Dinh Thuan** for granting us permission to undertake this project. Lastly, we deeply appreciate **Assoc. Prof. Dr. Nguyen Dinh Thuan, Ms. Trinh Thi Thanh Truc** and **Mr. Nguyen Minh Nhut** for their tireless efforts in guiding and encouraging our team every step of the way.

REFERENCES

- [1] Masoud, N. M. H. (2017). The impact of stock market performance upon economic growth. *International Journal of Economics and Financial Issues*, 3(4), 788-798.
- [2] Murkute, A., and Sarode, T. (2015). Forecasting market price of stock using artificial neural network. *International Journal of Computer Applications*, 124(12), 11-15.
- [3] Seber, G. A. F., and Lee, A. J. (2012). *Linear regression analysis*. John Wiley and Sons.
- [4] Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159-175.
- [5] Li, L., Wu, Y., Ou, Y., Li, Q., Zhou, Y., and Chen, D. (2017). Research on machine learning algorithms and feature extraction for time series. In *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* (pp. 1-5). IEEE.
- [6] Oyeeyemi, E. O., McKinnell, L.-A., and Poole, A. W. V. (2007). Neural network-based prediction techniques for global modeling of M(3000)F2 ionospheric parameter. *Advances in Space Research*, 39(5), 643-650.
- [7] Shen, G., Tan, Q., Zhang, H., Zeng, P., and Xu, J. (2018). Deep learning with gated recurrent unit networks for financial sequence predictions. *Procedia Computer Science*, 131, 895-903. <https://doi.org/10.1016/j.procs.2018.04.298>
- [8] Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [9] Hewamalage, H., Bergmeir, C., and Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1), 388-427. <https://doi.org/10.1016/j.ijforecast.2020.06.008>
- [10] Chen, K., Zhou, Y., and Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. In *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data)*, October 2015, Santa Clara, CA, USA (pp. 2823-2824). IEEE. <https://doi.org/10.1109/BigData.2015.7364089>
- [11] Kim, J.-M., Kim, D. H., and Jung, H. (2021). Estimating yield spreads volatility using GARCH-type models. *The North American Journal of Economics and Finance*, 57, Article 100880. <https://doi.org/10.1016/j.najef.2021.100880>
- [12] Agapitos, A., Brabazon, A., and O'Neill, M. (2017). Regularised gradient boosting for financial time-series modelling. *Computational Management Science*, 14, 367-391. <https://doi.org/10.1007/s10287-017-0280-y>
- [13] Wang, H., Peng, J., Huang, F., Wang, J., Chen, J., and Xiao, Y. (2022, September). Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*.
- [14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *NeurIPS*.
- [15] Kumari, K., and Yadav, S. (2018). Linear Regression Analysis Study. *Journal of the Practice of Cardiovascular Sciences*, 4(1), 33. <https://doi.org/10.4103/jpcs.jpcs-8-18>
- [16] Hyndman, R. J., and Athanasopoulos, G. (2021). *Forecasting: Principles and practice*. Melbourne: OTexts. Available at: <https://otexts.com/fpp2/AR.html>