Predicting Vietnamese airlines' stock prices utilizing statistical, machine learning, and deep learning algorithms (2019 - 2024)

TRAN HOANG NHAT[1], THAI NGOC DUNG[2], AND NGUYEN HOANG VU[3]

[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21522420@gm.uit.edu.vn
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21521982@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 21522799@gm.uit.edu.vn)

ABSTRACT

Stock prediction plays a crucial role in economic planning and investment decision-making. This study focuses on leveraging advanced statistical, machine learning and deep learning algorithms to predict stock prices in the Vietnamese market, with a particular emphasis on HVN, SCS, VJC. Eight time-series models, including Linear Regression, ARIMA, RNN, GRU, LSTM, SEMOS, Stacking and FCN, are utilized for prediction. Evaluation of these models is conducted using metrics such as MAPE, RMSE, and MSLE. Furthermore, related research on stock price prediction utilizing 8 algorithms has been examined and analyzed to demonstrate their effectivenes. The dataset spans from 01/03/2019 to 01/06/2024, and only the "Close" prices (VND) are considered for analysis.After testing on HVN, SCS and VJC datasets for comparison, the result showed that the Stacking (with ARIMAX and RNN as the base models, Linear Regression as the meta model) gave the best performance in predicting stock prices of these 3 datasets.

INDEX TERMS Vietnam , Stock prices , LR , ARIMA , RNN , GRU , LSTM , SEMOS , Stacking , FCN

I. INTRODUCTION

Stocks represent a vital component of Vietnam's economic landscape, serving as a cornerstone for economic development and fostering investor participation. The stock market not only facilitates capital infusion into enterprises, enabling their expansion and innovation but also serves as a lucrative avenue for investors to seek returns on their investments.

Recognizing the pivotal role of stocks in driving economic progress, our endeavor focuses on leveraging advanced statistical, machine learning, and deep learning algorithms to predict stock prices in the Vietnamese market, with a particular emphasis on Vietnam Airlines JSC (HVN), SCSC Cargo Service Corporation (SCS), and Vietjet Aviation Joint Stock Company (VJC).

There are numerous amounts of time-series models, this project uses eight time-series models to predict stock price in Vietnam: Linear regression, ARIMA, RNN, GRU, LSTM, SEMOS, Stacking, FCN.

We evaluate predictive models through a comprehensive analysis based on multiple criteria such as MAPE, RMSE, MSLE, and the results of data division methods. Through this process, we determine whether this model is good or not, which model should be used, which model should not be used to estimate stock prices, providing stakeholders with clear insights and decision-making support tools.

II. RELATED WORKS

Due to the profitability and importance of stocks, many methods have been developed to predict stock prices. Here, we will utilize 8 algorithms: Stacking, FCN, SEMOS, Linear regression, ARIMA, RNN, GRU and LSTM. Below are some related research papers on stock price prediction using these 8 algorithms.

In the study by Philip Ngare, Dennis Ikpe, and Samuel Asante Gyamerah, three models were employed: AdaBoost, KNN, and Stacking (AdaBoost and KNN serve as base-level classifiers, and GBM acts as the meta-level classifier). They utilized a dataset obtained from the Nairobi Stock Exchange, and the results showed that the Stacking model outperformed the two individual models, AdaBoost and KNN. This demonstrates that combining different models using the stacking ensemble learning method can lead to better results[1].

In the research by Shima Nabiee and Nader Bagherzadeh, they compared the performance of six different models, including FCN, SegNet, U-Net, DeepLab V3+, and two proposed models with 20-day and 40-day input frames, respectively. The results showed that when using price frames as input, FCN performed extremely well and ranked second among the six models. Notably, when the input was changed to trends, the FCN model achieved the best results among all six models, demonstrating that FCN is very suitable for stock prediction. Additionally, it was observed that when the input

frame is 1 (40 days), it provides better short-term prediction results, and when the number of input frames increases, the long-term results are improved [2].

The research of David Jobst, Annette Möller, and Jürgen Groß has shown that SEMOS is used to enhance the forecasting performance of numerical weather prediction (NWP) models by employing finite Fourier series, making it a clearly seasonal and trend-aware time series model. The results show that SEMOS provides more accurate weather forecasts based on evaluation metrics such as CRPS, LogS, and RMSE. Additionally, SEMOS maintains good forecasting performance across different time horizons. These advantages indicate that SEMOS could be a good choice for applying to stock market forecasting[3].

In the study by Xiwen Jin and Chaoran Yi, they conducted a comparison of the effectiveness among 6 models: LSTM, GRU, RandomForest, XGBoost, LightGBM, and Linear Regression, and found that GRU yielded the best results, followed by LSTM and Linear Regression. The R2 scores were as follows: LSTM 0.84, GRU 0.86, Linear Regression 0.73, and the MSE scores were: LSTM 7.06, GRU 6.26, Linear Regression 6.64 [4]. In another research by Dias Satria, four models were employed: ARIMA, RNN, LSTM, and GRU. The results showed that GRU exhibited the best performance in predicting stock prices, followed by LSTM and RNN, while ARIMA was deemed unsuitable due to the nonlinear characteristics of the data, violating the assumption of white noise in the estimation of ARIMA Box-Jenkins parameters [5].

III. MATERIALS

A. DATASET

The historical stock price of Vietnam Airlines JSC (HVN), SCSC Cargo Service Corporation (SCS) and Vietjet Aviation Joint Stock Company (VJC) from 01/03/2019 to 01/06/2024 will be applied. The data contains column such as Date, Open, High, Low, Close, Volume. As the goal is to forecast close prices, only data relating to column “close" (VND) will be processed.

B. DESCRIPTIVE STATISTICS

TABLE 1. HVN, SCS, VJC's Descriptive Statistics

	HVN	SCS	VJC
Count	1,307	1,315	1,314
Mean	20,134	64,511	117,339
Std	6,404	9,116	14,156
Min	8,610	37,320	93,800
25%	13,500	59,605	105,000
50%	20,853	65,030	116,200
75%	24,900	68,180	128,575
Max	34,753	90,900	149,000

FIGURE 1. HVN stock price's boxplot

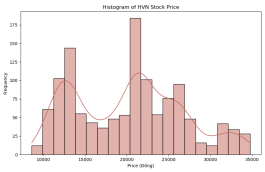

FIGURE 2. HVN stock price's histogram

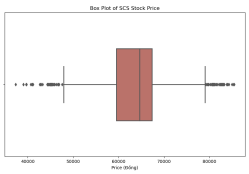

FIGURE 3. SCS stock price's boxplot

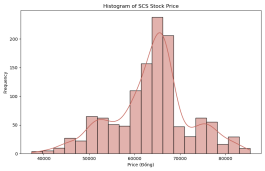

FIGURE 4. SCS stock price's histogram

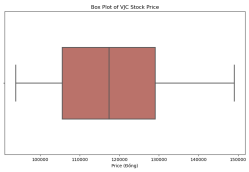

FIGURE 5. VJC stock price's boxplot

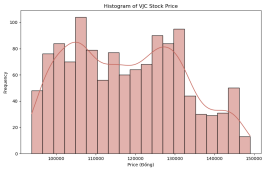

FIGURE 6. VJC stock price's histogram

IV. METHODOLOGY

A. LINEAR REGRESSION

Linear Regression is a method of statistical analysis used to determine the relationship between a dependent variable and one or more independent variables. The objective is to find the best-fitting linear equation that describes how the dependent variable changes as the independent variables change. A simple linear regression model has the form:

$$Y = \beta_0 + \beta_1 X + \varepsilon \quad [6]$$

Where:

- Y is the dependent variable.
- X is the independent variable.
- β_0 is the intercept of the regression line.
- β_1 is the regression coefficient for the independent variable.
- ε is the error term.

B. ARIMA

ARIMA(Autoregressive Integrated Moving Average) is a statistical forecasting method widely used in time series analysis. This model incorporates autoregressive (AR), moving average (MA) and Integrated (I) components to capture the relationship between current and past values of a time series. Auto Regression (AR):

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \epsilon_t \quad [7]$$

Where :
- y_t is the current value
- μ is the constant term
- p is the number of autoregressive terms
- γ_i is the autocorrelation coefficient
- ε_t is error

Moving Average (MA):

$$y_t = \mu + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \quad [7]$$

Where:
- y_t is the current value
- μ is the constant term
- q is the number of terms in the moving average
- θ_i is the moving average coefficient
- ε_t is error

Integrated (I):

$$\mathrm{I}(d=0) : \Delta y_t = y_t$$

$$\mathrm{I}(d=1) : \Delta y_t = y_t - y_{t-1}$$

$$\mathrm{I}(d=2) : \Delta\left(\Delta y_t\right) = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$$

Where: d is the number of differences required to make it a stationary sequence.

After combining them, we will have the ARIMA (p, d, q) express as follow:

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \quad [7]$$

C. SEMOS

SEMOS (Smooth Ensemble Model Output Statistics) is a statistical approach used for post-processing ensemble forecasts, particularly to handle seasonal variations in the predictive distribution parameters. SEMOS integrates seasonal effects directly into the estimation of location and scale parameters, enhancing the accuracy of ensemble forecasts, especially for quantities exhibiting seasonal variability.

The first parameter is location, in general, represents the central tendency or the "location" of the predictive distribution. In SEMOS, the location parameter is modeled as a function of the ensemble mean forecast and seasonal effects in order to capture systematic biases or trends in the ensemble forecasts:

$$\mu_S(t) = a_0 + f_0(t) + (a_1 + f_1(t)).x(t) \quad [3]$$

Where:
- $\mu_S(t)$ is the location parameter at time t.
- a_0, a_1 are coefficients representing the baseline intercept and slope.
- $f_0(t), f_1(t)$ are seasonal effects modeled using cyclic regression splines.
- $x(t)$ is the ensemble mean forecast at time t.

The scale parameter represents the spread or "scale" of the predictive distribution.It accounts for variations in forecast uncertainty over time, including factors such as model errors and ensemble spread, modeled as a function of the empirical ensemble standard deviation and seasonal effects.

$$log(\sigma_S(t)) = b_0 + g_0(t) + (b_1 + g_1(t)).s(t) \quad [3]$$

Where:
- $\sigma_S(t)$ is the scale parameter at time t.
- b_0, b_1 are coefficients representing the baseline intercept and slope.
- $g_0(t), g_1(t)$ are seasonal effects modeled using cyclic regression splines.
- $s(t)$ is the ensemble standard deviation at time step t.

To apply the smooth into the model, $f_i(t)$ and $g_i(t)$ can be defined below in order to reduce the coefficients to smoothly evolve over the year:

$$f_i(t) = \alpha_{i1} \sin\left(\frac{2\pi t}{365.25}\right) + \alpha_{i2} \cos\left(\frac{2\pi t}{365.2}\right) + \beta_{i3} \sin\left(\frac{4\pi t}{365.2}\right) + \beta_{i4} \cos\left(\frac{4\pi t}{365.2}\right) \quad [3]$$

$$g_i(t) = \beta_{i1} \sin\left(\frac{2\pi t}{365.25}\right) + \beta_{i2} \cos\left(\frac{2\pi t}{365.25}\right) + \beta_{i3} \sin\left(\frac{4\pi t}{365.25}\right) + \beta_{i4} \cos\left(\frac{4\pi t}{365.25}\right) \quad [3]$$

D. STACKING

Stacking, short for Stacked Generalization, is a machine learning algorithm belonging to the Ensemble Learning category. A basic Stacking model is usually divided into two levels: level-0 models and the level-1 model. Level-0 models (base-models) are the foundational models that learn directly from the dataset and produce predictions for the level-1 model (meta-model). The meta-model is trained based on the predicted outputs of the base models. These outputs, combined with the labels, form the input-output pairs during the training process of the meta-model. In this work, ARIMAX and RNN are used as base-models and Linear Regression is selected as meta-model.

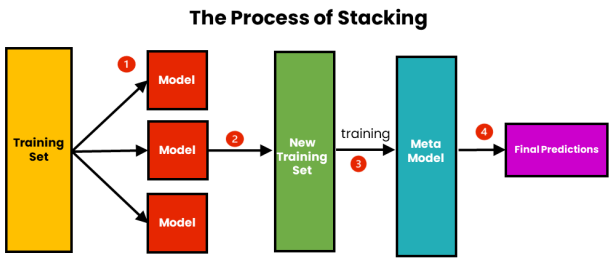

FIGURE 7. Stacking Architecture [8]

E. RNN

RNN (Recurrent Neural Network) is a type of artificial neural network with feedback connections closed by loop. The looping structure allows the network to store past information in the hidden state and use that past information to improve the performance of the network. In figure 8, the inputs x_t will be combined with the hidden layer h_{t-1} using an activation function to compute the current hidden layer h_t.

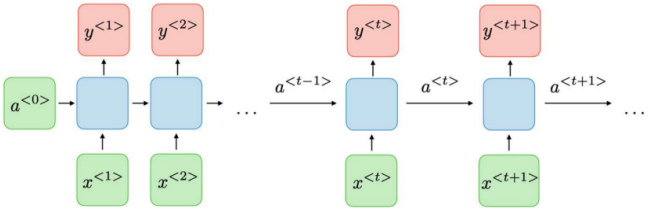

FIGURE 8. RNN Architecture [9]

Forward propagation :

$$a^{<t>} = g_1\left(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a\right) \quad [10]$$

$$y^{<t>} = g_2\left(W_{ya}a^{<t>} + b_y\right) \quad [10]$$

Where:

- $x^{<t>}$ is the input value at time step t
- $a^{<t>}$ is the state at time step t
- $y^{<t>}$ is the output value at time step t
- $\mathrm{W_{aa}}, \mathrm{W_{ax}}, \mathrm{W_{ya}}$ are weights
- b_a và b_y are bias
- $\mathrm{g_1}$ is activation function(e.g., tanh, ReLU)
- $\mathrm{g_2}$ is activation function(e.g., softmax)

Backward propagation :

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad [10]$$

$$W_{ay}^{(t+1)} = W_{ay}^{(t)} - \eta\frac{\partial\mathcal{L}}{\partial W_{ay}} \quad [10]$$

$$W_{aa}^{(t+1)} = W_{aa}^{(t)} - \eta\frac{\partial\mathcal{L}}{\partial W_{aa}} \quad [10]$$

$$W_{xa}^{(t+1)} = W_{xa}^{(t)} - \eta\frac{\partial\mathcal{L}}{\partial W_{xa}} \quad [10]$$

$$b_y^{(t+1)} = b_y^{(t)} - \eta\frac{\partial\mathcal{L}}{\partial b_y} \quad [10]$$

$$b_h^{(t+1)} = b_h^{(t)} - \eta\frac{\partial\mathcal{L}}{\partial b_h} \quad [10]$$

Where:

- y_i is actual value of the i-th element
- $\hat{y}_i$ is predicted value of the i-th element
- η is learning rate

F. LSTM

Long Short-Term Memory (LSTM) is an advanced variant of recurrent neural network (RNN) architecture used in the field of deep learning. Its goal is to give RNN a "long short-term memory" — a short-term memory that can endure thousands of timesteps.

A vanilla LSTM unit is composed of a cell, an input gate, an output gate and a forget gate.

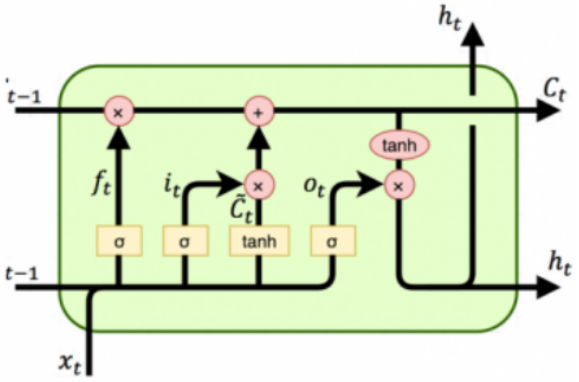

FIGURE 9. LSTM Model At Time Step t [11]

Where:

- Input gate: conditionally decides which values from the input to update the memory state.
- Output Gate: conditionally decides what to output based on input and the memory of the block.
- Forget Gate: conditionally decides what information to throw away from the block.

The update equations for the LSTM unit are expressed by equation below:

$$h^{(t)} = g_0^{(t)} f_h(s^{(t)}) \quad [11]$$

$$s^{(t-1)} = g_f^{(t)} s^{(t-1)} + g_i^{(t)} f_s(wh^{(t-1)}) + uX^{(t)} + b \quad [11]$$

$$g_i^{(t)} = sigmold(w_i h^{(t-1)} + u_i X^{(t)} + b_i \quad [11]$$

$$g_f^{(t)} = sigmold(w_f h^{(t-1)} + u_f X^{(t)} + b_f \quad [11]$$

$$g_f^{(t)} = sigmold(w_o h^{(t-1)} + u_o X^{(t)} + b_o \quad [11]$$

Where: f_h and f_s represent the activation functions of the system state and internal state, typically utilizing the hyperbolic tangent function.

G. GRU

GRU stands for Gated Recurrent Unit, which is a type of recurrent neural network (RNN) architecture that is similar to LSTM (Long Short-Term Memory). This means that GRU also have the input gate, output gate and forget gate. The differences are that GRU combines the input and forget gate into a single update gate, resulting in a more streamlined design and separate cell state is not included in GRU.

A GRU unit consists of three main components: an update gate, a reset gate and the current memory content.

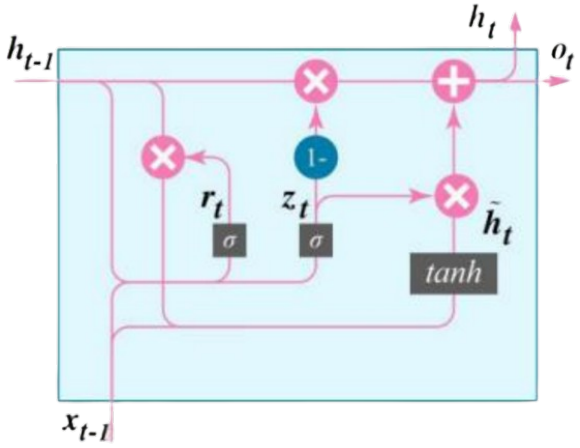

FIGURE 10. GRU Model At Time Step t [11]

The update gate determines how much of the past information should be retained and combined with the current input at a specific time step.

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \quad [11]$$

The reset gate decides how much of the past information should be forgotten.

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \quad [11]$$

The current memory content is computed based on the reset gate and the concatenation of the transformed previous hidden state and the current input.

$$\tilde{h}_t = \tanh(W_h[r_t h_t, x_t]) \quad [11]$$

The final memory state h_t is determined by a combination of the previous hidden state and the candidate activation.

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \quad [11]$$

Finally, the output gate is computed using the current memory state h_t and is typically followed by an activation function, such as the sigmold function.

$$o_t = \sigma_o(W_o h_t + b_o) \quad [11]$$

H. FCN

Fully Convolutional Neural Networks (FCNs) were first proposed in Wang et al. (2017b) for classifying univariate time series and validated on 44 datasets from the UCR/UEA archive. FCNs are mainly convolutional networks that do not contain any local pooling layers which means that the length of a time series is kept unchanged throughout the convolutions. In addition, one of the main characteristics of this architecture is the replacement of the traditional final FC layer with a Global Average Pooling (GAP) layer which reduces drastically the number of parameters in a neural network while enabling the use of the CAM (Zhou et al., 2016) that highlights which parts of the input time series contributed the most to a certain classification. [12]

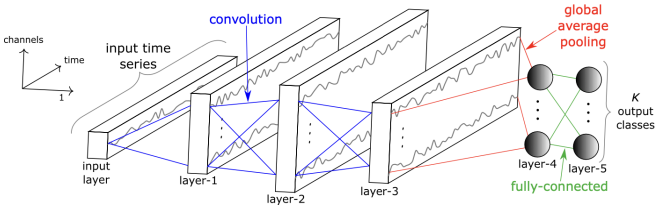

FIGURE 11. Fully Convolutional Neural Network architecture [12]

The FCN is designed for end-to-end classification tasks on univariate time series data. It leverages the power of Convolutional Neural Networks (CNNs) to capture hierarchical features directly from the input time series.

The fundamental building block of FCN is the convolutional layer. The formula for calculating the output of a convolutional layer is as follows:

$$\text{conv}(i,j) = R\left(\sum_{u=0}^{M-1}\sum_{v=0}^{M-1} w_{u,v}x_{i+u,j+v} + b\right) \quad [13]$$

Where:

- $\text{conv}(i,j)$ is the convolution result, also known as the feature map.
- M indicates the size of the convolution kernel ($M \times M$).
- $w_{u,v}$ is the weight of the convolution kernel in line u and column v.
- $x_{i+u,j+v}$ is the input time series.
- b is the bias term.
- R is the activation function, which brings a nonlinear factor that allows FCN to approximate any nonlinear function.

V. RESULT

A. EVALUATION METHODS

Root Mean Squared Error (RMSE): is the square root of average value of squared error in a set of predicted values.

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

Mean Absolute Percentage Error (MAPE): is the average percentage error in a set of predicted values.

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

Mean Squared Logarithmic Error (MSLE):is the relative difference between the log-transformed actual and predicted values.

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(log(1+\hat{y_i}) - log(1+y_i))^2$$

Where:

- n is the number of observations in the dataset.
- y_i is the true value.
- $\hat{y_i}$ is the predicted value.

B. HVN DATASET

HVN Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MSLE
LR	7:3	3993.14	24.3741	0.07304
	8:2	4163.78	13.5631	0.07678
	9:1	5940.92	22.6138	0.15792
ARIMA	7:3	3581.65	16.3043	0.06087
	8:2	3341.08	10.9181	0.04085
	9:1	5375.16	19.9011	0.11742
SEMOS	7:3	3962.49	18.2408	0.08073
	8:2	3947.04	13.3932	0.06523
	9:1	6018.41	25.2501	0.16638
Stacking	**7:3**	**197.562**	**0.92907**	**0.00015**
	8:2	**240.401**	**1.01158**	**0.00021**
	9:1	**434.257**	**1.59473**	**0.00038**
RNN	7:3	779.731	4.25189	0.00269
	8:2	813.972	3.60593	0.00227
	9:1	1337.81	5.16975	0.00387
FCN	7:3	5147.77	32.5057	0.17553
	8:2	3881.08	16.3598	0.05691
	9:1	8015.63	35.2079	0.20307
GRU	7:3	424.191	1.74601	0.00069
	8:2	440.368	2.00419	0.00072
	9:1	880.814	3.36061	0.00171
LSTM	7:3	450.836	1.89826	0.00078
	8:2	600.371	2.27099	0.00116
	9:1	917.374	3.41183	0.00184

TABLE 2. HVN Dataset's Evaluation

FIGURE 12. Stacking model's result with 7:3 splitting proportion

FIGURE 13. Stacking model's result with 8:2 splitting proportion

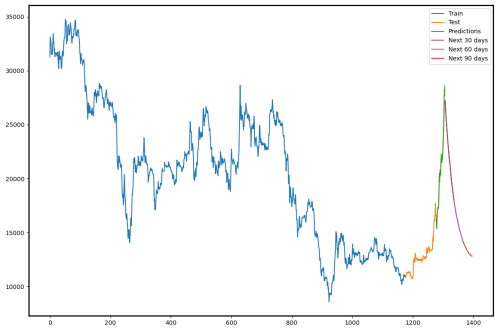

FIGURE 14. Stacking model's result with 9:1 splitting proportion

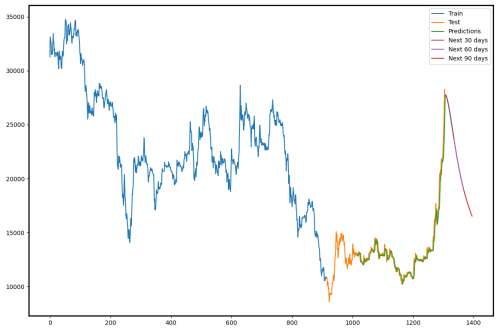

FIGURE 15. GRU model's result with 7:3 splitting proportion

FIGURE 16. GRU model's result with 8:2 splitting proportion

FIGURE 17. LSTM model's result with 7:3 splitting proportion

C. SCS DATASET

SCS Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MSLE
LR	7:3	9905.85	13.9523	0.02007
	8:2	7788.89	10.3625	0.01199
	9:1	8985.64	9.69887	0.01395
ARIMA	7:3	7633.15	6.23802	0.01103
	8:2	11243.2	10.7119	0.02545
	9:1	13349.1	12.7524	0.03356
SEMOS	7:3	6451.95	7.50616	0.00812
	8:2	8462.03	6.90038	0.01311
	9:1	11244.7	10.5766	0.02261
Stacking	**7:3**	**457.389**	**0.48271**	**0.00004**
	8:2	**532.678**	**0.53753**	**0.00005**
	9:1	**678.649**	**0.64756**	**0.00006**
RNN	7:3	1383.81	1.35395	0.00036
	8:2	1828.88	1.83307	0.00061
	9:1	2145.41	2.02933	0.00064
FCN	7:3	3276.81	3.68393	0.00226
	8:2	3450.99	3.83999	0.00208
	9:1	7511.55	7.62490	0.00803
GRU	7:3	946.301	0.88267	0.00017
	8:2	1123.88	0.97304	0.00022
	9:1	1475.51	1.31043	0.00031
LSTM	7:3	1183.07	1.12567	0.00027
	8:2	1242.92	1.18776	0.00029
	9:1	2017.59	1.84989	0.00057

TABLE 3. SCS Dataset's Evaluation

FIGURE 18. Stacking model's result with 7:3 splitting proportion

FIGURE 19. Stacking model's result with 8:2 splitting proportion

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

FIGURE 21. GRU model's result with 7:3 splitting proportion

FIGURE 22. GRU model's result with 8:2 splitting proportion

FIGURE 23. LSTM model's result with 7:3 splitting proportion

D. VJC DATASET

VJC Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MSLE
LR	7:3	21720.7	20.7604	0.03762
	8:2	12454.9	11.3989	0.01373
	9:1	4220.91	2.76031	0.00149
ARIMA	7:3	7056.66	5.94446	0.00465
	8:2	7807.39	5.87333	0.00579
	9:1	4857.61	4.07812	0.00203
SEMOS	7:3	7241.33	6.12661	0.00489
	8:2	10233.5	8.45117	0.01041
	9:1	4496.74	2.22865	0.00169
Stacking	**7:3**	**798.243**	**0.57291**	**0.00006**
	8:2	**862.898**	**0.57633**	**0.00007**
	9:1	**984.336**	**0.65617**	**0.00008**
RNN	7:3	2051.11	1.44206	0.00038
	8:2	2312.08	1.49321	0.00046
	9:1	3826.83	2.24303	0.00121
FCN	7:3	8798.61	7.04139	0.00681
	8:2	8950.51	7.41324	0.00712
	9:1	5649.02	4.77011	0.00264
GRU	7:3	1502.23	0.95296	0.00021
	8:2	1697.52	0.99987	0.00025
	9:1	2567.96	1.47789	0.00054
LSTM	7:3	1860.28	1.32234	0.00032
	8:2	1780.11	1.06262	0.00027
	9:1	2791.86	1.69092	0.00063

TABLE 4. VJC Dataset's Evaluation

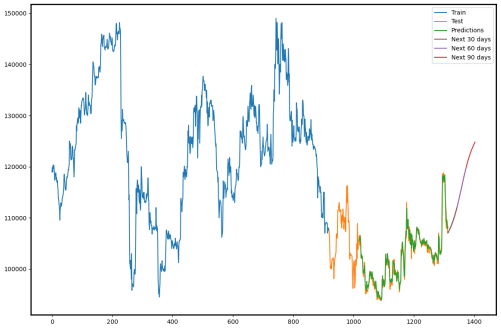

FIGURE 24. Stacking model's result with 7:3 splitting proportion

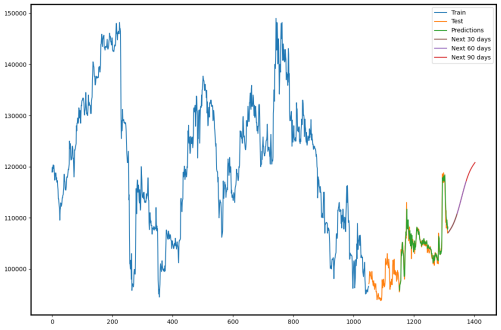

FIGURE 25. Stacking model's result with 8:2 splitting proportion

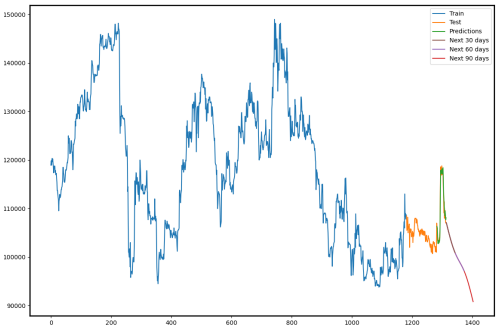

FIGURE 26. Stacking model's result with 9:1 splitting proportion

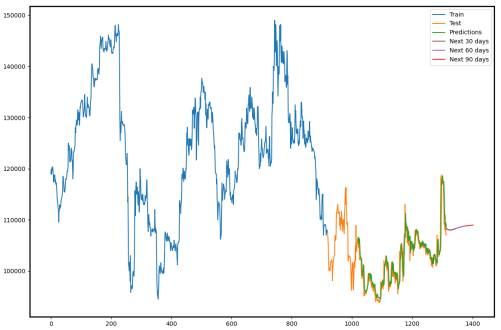

FIGURE 27. GRU model's result with 7:3 splitting proportion

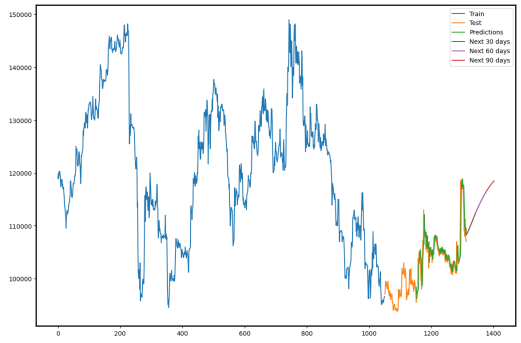

FIGURE 28. GRU model's result with 8:2 splitting proportion

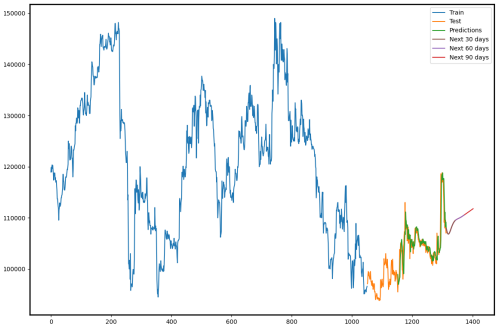

FIGURE 29. LSTM model's result with 8:2 splitting proportion

VI. CONCLUSION

A. SUMMARY

This study has explored the application of advanced statistical, machine learning, and deep learning algorithms to predict stock prices in the Vietnamese market, focusing on three major companies: Vietnam Airlines JSC (HVN), SCSC Cargo Service Corporation (SCS), and Vietjet Aviation Joint Stock Company (VJC). The time-series models utilized in this study include Linear Regression, ARIMA, RNN, GRU, LSTM, SEMOS, Stacking, and FCN.

The results demonstrate that each model has its strengths and weaknesses in predicting stock prices. Specifically, after comparing models with different splitting ratios on 3 datasets spanning from 01/03/2019 to 01/06/2024, we have obtained the model that give the best forecast result for each dataset as follows:

• For the HVN dataset, the Stacking model outperformed others with a RMSE of 197.562, MAPE of 0.92907%, and MSLE of 0.00015 for a 7:3 train:test ratio. Additionally, it obtained the highest accuracy for both the 8:2 and 9:1 ratios.

• For the SCS dataset, the Stacking model again showed superior performance with a RMSE of 457.389, MAPE of 0.48271%, and MSLE of 0.00004 for a 7:3 ratio. It also had the best results for the 8:2 and 9:1 ratios.

• For the VJC dataset, the Stacking model demonstrated the best performance with a RMSE of 798.243, MAPE of 0.57291%, and MSLE of 0.00006 for a 7:3 ratio. The model also achieved the best results for the 8:2 and 9:1 ratios.

Through rigorous evaluation, the study confirms that the Stacking algorithm, despite demonstrating good performance in predicting stock prices in Vietnam, utilizes exogenous variables such as High and Low in both the base model and meta model. In reality, data for these attributes on any given day are only available once the stock market trading session for that day has concluded. This limitation implies that the model's effectiveness may not fully reflect its predictive capabilities under real-world conditions when actual data on the highest and lowest prices of the day are lacking.

These studies provide valuable insights for investors and fund managers in Vietnam, enabling them to make informed decisions and enhance investment efficiency. This contributes to the sustainable development and stability of the Vietnamese stock market in general, including the specific airlines Vietnam Airlines JSC, SCSC Cargo Service Corporation and Vietjet Aviation Joint Stock Company. Moreover, it fosters investor confidence in market analysis and forecasting capabilities.

B. FUTURE CONSIDERATIONS

While the current study presents significant advancements in stock price prediction, several avenues for future research remain. These include:

• Expansion of dataset and feature inclusion: future research should consider expanding the dataset to include a broader range of features beyond the "Close" prices, such as trading volume, market sentiment indicators, and macroeconomic variables. This could enhance the predictive power of the models by providing a more comprehensive view of the factors influencing stock prices.

• Real-time prediction and algorithm optimization: implementing real-time prediction systems that continuously update model parameters as new data becomes available can significantly improve the applicability of the models in dynamic market conditions. Additionally, optimizing algorithm parameters through techniques such as grid search or Bayesian optimization can further refine model performance.

• Incorporation of advanced techniques: future studies could explore the incorporation of more sophisticated techniques such as transfer learning, which can leverage knowledge from related tasks to improve prediction accuracy. Additionally, the use of explainable AI methods could provide deeper insights into the decision-making processes of complex models, enhancing trust and interpretability for stakeholders.

ACKNOWLEDGMENT

First and foremost, we would like to express our sincere gratitude to **Assoc. Prof. Dr. Nguyen Dinh Thuan** and **Mr. Nguyen Minh Nhut** for their exceptional guidance, expertise, and invaluable feedback throughout the research process. Their mentorship and unwavering support have been instrumental in shaping the direction and quality of this study. Their profound knowledge, critical insights, and attention to detail have significantly contributed to the success of this research.

This research would not have been possible without the support and contributions of our mentors. We would like to extend our heartfelt thanks to everyone involved for their invaluable assistance, encouragement, and belief in our research. Thank you all for your invaluable assistance and encouragement.

REFERENCES

[1] S. A. Gyamerah, P. Ngare and D. Ikpe, "On Stock Market Movement Prediction Via Stacking Ensemble Learning Method," 2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr), Shenzhen, China, 2019, pp. 1-8, doi: 10.1109/CIFEr.2019.8759062.

[2] S. Nabiee và N. Bagherzadeh, "Stock Trend Prediction: A Semantic Segmentation Approach," Mar 2023, doi: 10.48550/arXiv.2303.09323.

[3] D. Jobst, A. Möller, and J. Groß, "Time Series based Ensemble Model Output Statistics for Temperature Forecasts Postprocessing", Feb 2024, doi: 10.48550/arXiv.2402.00555.

[4] Xiwen Jin and Chaoran Yi, "The Comparison of Stock Price Prediction Based on Linear Regression Model and Machine Learning Scenarios," in Proceedings of the 2022 International Conference on Bigdata Blockchain and Economy Management (ICBBEM 2022), December 2022, pp. 837-842, doi: 10.2991/978-94-6463-030-5_82.

[5] D. Satria, "Predicting Banking Stock Prices Using RNN, LSTM, and GRU Approach," Applied Computer Science, vol. 19, no. 1, pp. 82-94, March 2023, doi: 10.35784/acs-2023-06

[6] D. C. Montgomery, E. A. Peck, and G. G. Vining, "Introduction to Linear Regression Analysis," 5th ed., Hoboken, NJ, USA: Wiley, 2012.

[7] Y. Zhao, "Comparison of Stock Price Prediction in Context of ARIMA and Random Forest Models," BCP Business & Management, vol. 38, pp. 1880-1885, Mar. 2023, doi: 10.54691/bcpbm.v38i.3996.

[8] B. Soni, "Stacking to Improve Model Performance: A Comprehensive Guide on Ensemble Learning in Python", Medium. Accessed: May 12,

2024. [Online]. Available: https://medium.com/@brijesh_soni/stacking-to-improve-model-performance-a-comprehensive-guide-on-ensemble-learning-in-python-9ed53c93ce28

[9] A. Amidi and S. Amidi, "Recurrent Neural Networks cheatsheet" Stanford University. [Accessed: May 25, 2024]. [Online]. Available: https://stanford.edu/ shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[10] Nguyen Minh Nhut, “Mo Hinh Mang Hoi Quy (RNN) trong chuoi thoi gian”

[11] Farhad Mortezapour Shiri, Thinagaran Perumal, Norwati Mustapha and Raihani Mohamed, "A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU", May 2023, doi: 10.48550/arXiv.2305.17473

[12] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," Data Mining and Knowledge Discovery, vol. 33, no. 4, pp. 917-963, Jul. 2019, doi: 10.1007/s10618-019-00619-1.

[13] J. Wu, B. Liu, H. Zhang, S. He, and Q. Yang, "Fault Detection Based on Fully Convolutional Networks (FCN)," J. Mar. Sci. Eng., vol. 9, no. 3, p. 259, Mar. 2021. doi: 10.3390/jmse9030259.