Vietnamese Stock Price Prediction Using Statistical Model And Machine Learning Algorithm

1st Nguyen Thi My Tran
Faculty of Information Systems
University of Information Technology
Ho Chi Minh City, Vietnam
20520322@gm.uit.edu.vn

2nd Ton Nu Tu Quyen
Faculty of Information Systems
University of Information Technology
Ho Chi Minh City, Vietnam
20520296@gm.uit.edu.vn

3rd Pham Thanh Nhut
Faculty of Information Systems
University of Information Technology
Ho Chi Minh City, Vietnam
20521728@gm.uit.edu.vn

Abstract – The application of machine learning for stock prediction is attracting a lot of attention these years. A large amount of research has been conducted in this area and multiple existing results have shown that machine learning methods could be successfully used toward stock predicting using stocks historical data. In this paper, we used Vietnam's stocks dataset to perform some basic time series machine learning models (Linear regression, Arima, GRU, LSTM) and main machine learning model of this project (VAR, SSA, MCMC, Seq2Seq, FCN).

Keywords – Stocks, Predict, Linear regression, Arima, GRU, LSTM, VAR, SSA, MCMC, Seq2Seq, FCN

I. INTRODUCTION

Stocks play a significant role in the development and promotion of Vietnam's economy. The stock market provides an important financial mechanism for companies to attract capital and expand their businesses. It also creates investment opportunities and generates profits for investors. Stocks facilitate the buying and selling of shares and financial assets. Furthermore, the development of the stock market demonstrates the maturity and reliability of Vietnam's financial system in the eyes of domestic and foreign investors. In summary, stocks are crucial in driving economic growth and fostering financial development in Vietnam.

As a result, in this project, our group has chosen to predict stocks price using machine learning and deep learning models in the hope of assisting Vietnam's economy and investment in stocks.

There are numerous amounts of time-series models, this project uses nine time-series models to predict stock price in Vietnam: Linear regression, Arima, GRU, LSTM, VAR, SSA, MCMC, Seq2Seq, and FCN.

In this project, predictive models are evaluated according to four criteria: MAPE, RMSE, Vendi score and result of data division methods. We used these criteria to determine which one is best for estimating the price.

II. RELATED WORKS

In the field of finance, stock price forecasting is an important and highly regarded issue. In recent years, forecasting algorithms have been widely applied to solve this problem. In this study, we will utilize the following 9 algorithms: LSTM, ARIMA, GRU, Linear Regression, VAR, SSA, MCMC, Seq2Seq, and FCN, to forecast the stock prices in Vietnam. David Spade (2020) [1] provides an introduction to MCMC methods and their applications in statistical inference, emphasizing the importance of assessing convergence and mixing and providing practical guidance for implementing MCMC in data analysis, highlighting its usefulness for sampling from complex probability distributions. Sima Siami-Namini and Akbar Siami Namin (2018) [2] discuss the use of ARIMA and LSTM, two popular time series forecasting techniques, and highlights the advantages of each method, with ARIMA being useful for short-term forecasting of stationary data, and LSTM being particularly effective for capturing long-term dependencies and non-linear patterns in the data. Aytan Osmanzade (2017) [3] has performed SSA to the dataset of "AS Tallink Grupp" stock. This paper has used the main algorithm of SSA: decomposition and reconstruction for forecasting. Ge Zhang et al. (2021) [4] propose a CNN-Seq2Seq model with an attention mechanism based on a multi-task learning method for short-time multi-energy load forecasting. Junyoung Chung et al. (2014) [5] have empirically evaluated Gated Recurrent Neural Networks on Sequence Modeling, including gated recurrent unit (GRU) model. Bayraci et al. (2011) [6] develop a vector autoregressive (VAR) model of the Turkish financial markets for the period of June 15 2006 – June 15 2010 and forecasts ISE100 index, TRY/USD exchange rate, and short-term interest rates. Hassan Ismail Fawaz et al. (2019) [7] studied the current state-of-the-art performance of deep learning algorithms for Time Series Classification (TSC) by presenting an empirical study of the most recent DNN architectures for TSC, including Fully Convolutional Neural Networks (FCN) model.

MATERIALS

A. Petrovietnam Technical Services Corp (PVS)

Petrovietnam Technical Services Corp (PVS) is a publicly traded company on the Ho Chi Minh City Stock Exchange, specializing in providing technical services and solutions for the oil and gas industry in Vietnam. The dataset was gathered from Investing website, containing 7 columns and 1371 rows of data from December 2017 to the present.

	Date	Price	Open	High	Low	Vol.	Change %
2	12/15/2017	17,656.00	17,292.00	17,929.00	17,292.00	6.37M	1.04%
3	12/18/2017	19,021.00	17,747.00	19,385.00	17,747.00	15.17M	7.73%
4	12/19/2017	19,385.00	19,112.00	19,658.00	18,839.00	9.39M	1.91%
5	12/20/2017	20,022.00	19,385.00	21,206.00	19,385.00	12.97M	3.29%
6	12/21/2017	20,114.00	20,296.00	20,660.00	19,750.00	10.97M	0.46%
7	12/22/2017	20,022.00	20,114.00	20,296.00	19,658.00	7.38M	-0.46%
8	12/25/2017	20,660.00	20,204.00	21,206.00	20,022.00	7.68M	3.19%
9	12/26/2017	21,570.00	20,933.00	21,570.00	20,660.00	8.53M	4.40%
10	12/27/2017	21,479.00	21,843.00	22,298.00	21,388.00	9.22M	-0.42%

Figure 1. PVS Dataset

Calculate the values of Count, Min, Max, Mean, Median, Mode, Quantile, Range, Variance, Standard Deviation, Coefficient of Deviation, Skewness, and Kurtosis for the stock prices (Price) in the PVS Dataset

	Price
count	1.372000e+03
mean	2.104488e+04
std	5.254251e+03
min	9.000000e+03
25%	1.770000e+04
50%	2.126200e+04
75%	2.457700e+04
max	3.812400e+04
mode	1.240000e+04
range	2.912400e+04
variance	2.760716e+07
coefficient of deviation	2.496689e-01
skewness	1.244311e-01
kurtosis	-1.992745e-01

Figure 2. Descriptive Statistics for the stock prices in PVS Dataset

B. Asia Commercial Joint Stock Bank (ACB)

Asia Commercial Joint Stock Bank (ACB) is a Vietnam-based financial institution that offers commercial banking services. ACB stock price dataset was gathered from the website investing.com. The dataset consists of 7 columns and 1378 rows, gathered from December 2017 to the present.

	Date	Price	Open	High	Low	Vol.	Change %
2	12/1/2017	10,537.70	10,477.80	10,567.60	10,417.90	7.36M	0.86%
3	12/4/2017	10,926.90	10,537.70	10,956.80	10,537.70	9.45M	3.69%
4	12/5/2017	10,477.80	10,956.80	10,956.80	10,477.80	8.19M	-4.11%
5	12/6/2017	10,447.90	10,447.90	10,627.50	10,328.10	9.81M	-0.29%

Figure 3. ACB Dataset preview

Calculate the values of Count, Min, Max, Mean, Median, Mode, Quantile, Range, Variance, Standard Deviation, Coefficient of Deviation, Skewness, and Kurtosis for the ACB Dataset.

	Price
count	1.378000e+03
mean	1.761803e+04
std	6.401105e+03
min	8.763100e+03
25%	1.152000e+04
50%	1.512595e+04
75%	2.447250e+04
max	3.036000e+04
mode	2.500000e+04
range	2.159690e+04
variance	4.097414e+07
coefficient of deviation	3.633269e-01
skewness	3.115853e-01
kurtosis	-1.580813e+00

Figure 4. Descriptive Statistics for the stock prices in ACB Dataset

C. Petrolimex Petrochemical JSC (VNM)

Vietnam Dairy Products JSC (VNM) is a Vietnam-based food manufacturer. The Company is mainly engaged in manufacturing, marketing, and distribution of dairy products, especially milk of various forms and other derived products, as well as nutritious food and non-alcoholic beverages. Through subsidiaries, the Company is also involved in real estate investment activities.. VNM stock price dataset was gathered in Investing website since December 2017 to the present. The dataset consists of 7 columns and 1381 rows.

	Date	Price	Open	High	Low	Trading Volume	% Change
2	1/12/2017	159,039.00	151,736.00	159,039.00	151,736.00	1.50M	4.98%
3	4/12/2017	164,719.00	159,039.00	166,423.00	159,039.00	1.70M	3.57%
4	5/12/2017	161,473.00	164,719.00	166,342.00	160,662.00	852.08K	-1.97%
5	6/12/2017	158,227.00	158,227.00	159,850.00	151,898.00	1.14M	-2.01%

Figure 5. VNM Dataset preview

	Date	Price	Open	High	Low	Trading V	% Change
2	1/12/2017	159,039.00	151,736.00	159,039.00	151,736.00	1.50M	4.98%
3	4/12/2017	164,719.00	159,039.00	166,423.00	159,039.00	1.70M	3.57%
4	5/12/2017	161,473.00	164,719.00	166,342.00	160,662.00	852.08K	-1.97%
5	6/12/2017	158,227.00	158,227.00	159,850.00	151,898.00	1.14M	-2.01%

Figure 5. VNM Dataset preview

Calculate the values of Count, Min, Max, Mean, Median, Mode, Quantile, Range, Variance, Standard Deviation, Coefficient of Deviation, Skewness, and Kurtosis for the VNM Dataset

	Price
count	1.381000e+03
mean	9.643309e+04
std	2.546024e+04
min	6.126010e+04
25%	7.892580e+04
50%	9.287700e+04
75%	1.025160e+05
max	1.755780e+05
mode	7.440000e+04
range	1.143179e+05
variance	6.482240e+08
coefficient of deviation	2.640198e-01
skewness	1.459260e+00
kurtosis	1.628681e+00

Figure 6. Descriptive Statistics for the stock prices in VNM Dataset

III. METHOD

A. Model

1) Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable.

The formula of Linear Regression can present below [8]:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

- y: dependent variable (Target Variable)
- x: Independent variable
- β_0= intercept of the line
- β_1: linear regression coefficient
- ε: random error

Applying Linear Regression algorithm to the VNM dataset:

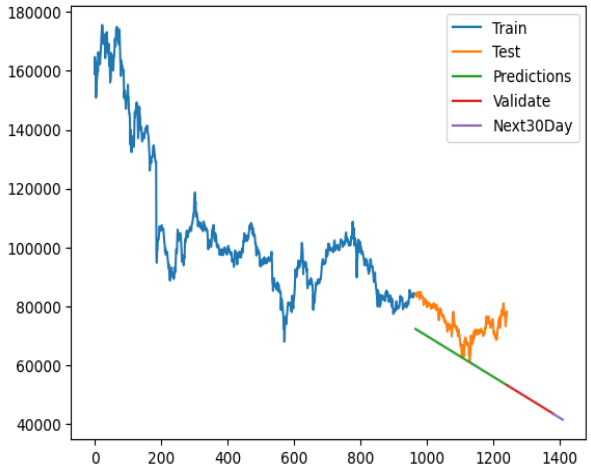


Figure 6. Result of Linear Regression (7-2-1)

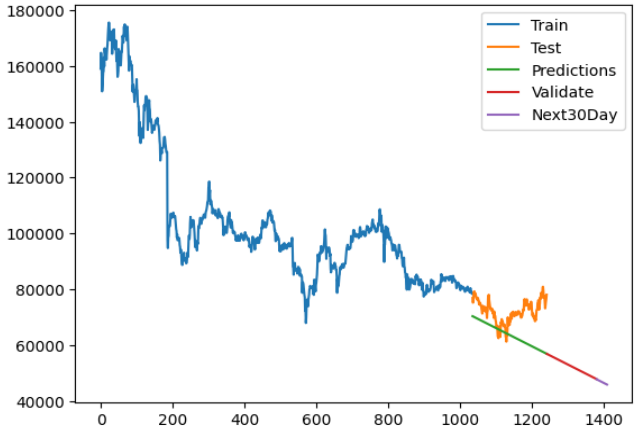


Figure 7. Result of Linear Regression (7.5-1.5-1)

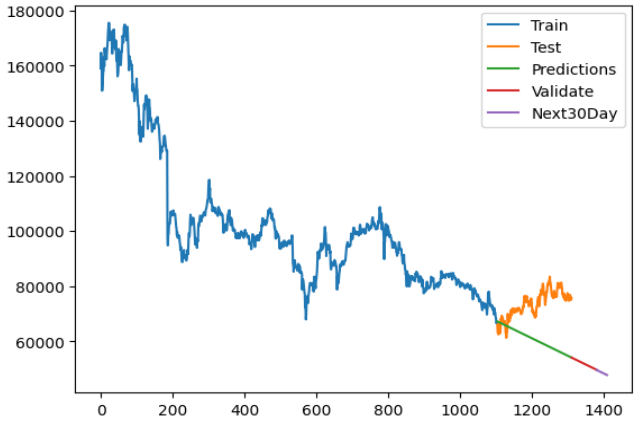


Figure 8. Result of Linear Regression (8-1.5-0.5)

2) Autoregressive Integrated Moving Average (ARIMA)

ARIMA model is a generalized model of Autoregressive Moving Average (ARMA) that combines Autoregressive (AR) process and Moving Average (MA) processes and builds a composite model of the time series. [9]

The ARIMA model has three parameters: p, d, q

As acronym indicates, ARIMA(p, d, q) captures the key elements of the model [2]:

- AR: Autoregression. A regression model that uses the dependencies between an observation and a number of lagged observations (p).
- I: Integrated. To make the time series stationary by measuring the differences of observations at different time (d).
- MA: Moving Average. An approach that takes into accounts the dependency between observations and the residual error terms when a moving average model is used to the lagged observations (q).

A simple form of an AR model of order p, i.e., AR (p), can be written as a linear process given by:

$$x_t = c + \sum_{i=1}^{p} \emptyset_i x_{t-i} + \varepsilon_t$$

Where χ_t is the stationary variable, c is constant, the terms in $\emptyset_i$ are autocorrelation coefficients at lags *1, 2, … , p* and ε_t the residuals, are the Gaussian white noise series with mean zero and variance $б_\varepsilon^2$

An MA model of order q, i.e., MA(q), can be written in the form:

$$x_t = \mu + \sum_{i=0}^{q} \theta_i \varepsilon_{t-i}$$

Where μ is the expectation of χ_t (usually assumed equal to zero), the $\emptyset_i$ terms are the weights applied to the current and prior values of a stochastic term in the time series, and $\theta_0 =$ 1. We assume that χ_t is a Gaussian white noise series with mean zero and variance $б_\varepsilon^2$.

We can combine these two models by adding them together and form an ARMA model of order (p, q):

$$x_t = c + \sum_{i=1}^{p} \emptyset_i x_{t-i} + \varepsilon_t + \sum_{i=0}^{q} \theta_i \varepsilon_{t-i}$$

Where $\emptyset_i \neq 0$, $\theta_i \neq 0$, and $б_\varepsilon^2 > 0$. The parameters p and q are called the AR and MA orders, respectively. ARIMA forecasting, also known as Box and Jenkins forecasting, is capable of dealing with non-stationary time series data because of its "integrate" step.

In fact, the "integrate" component involves differencing the time series to convert a non-stationary time series into a stationary. The general form of a ARIMA model is denoted as ARIMA(p, d, q). [2]

Applying the ARIMA algorithm to the PVS dataset:

ARIMA (7-2-1)

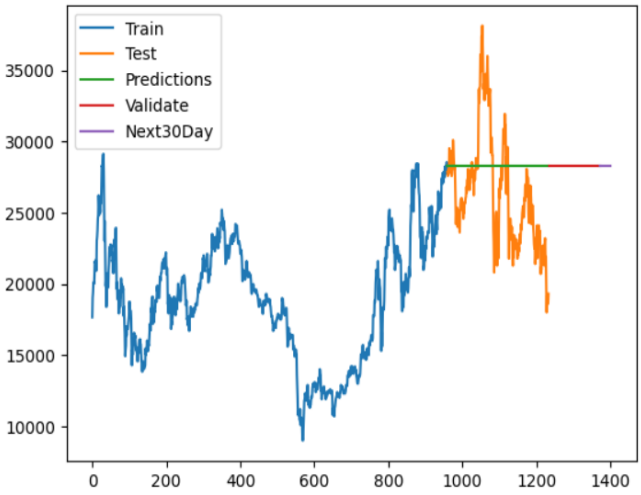


Figure 9. Result of ARIMA (7-2-1)

ARIMA(7.5-1.5-1)

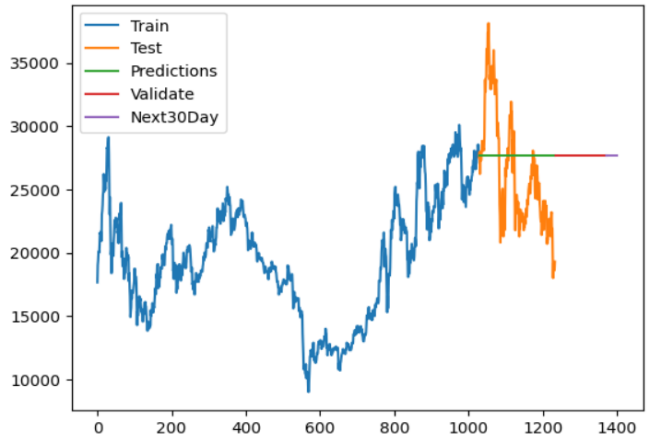


Figure 10. Result of ARIMA (7.5-1.5-1)

ARIMA(8-1.5-0.5)

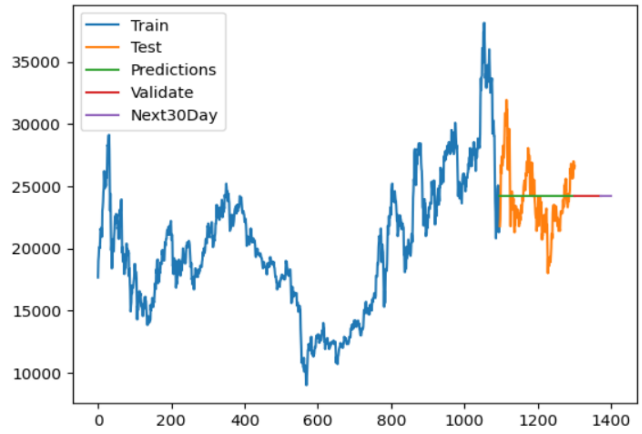


Figure 11. Result of ARIMA (8-1.5-0.5)

3) Gated Recurrent Unit (GRU)

A gated recurrent unit (GRU) was proposed by Cho et al. [2014] to make each recurrent unit to adaptively capture dependencies of different time scales. Similarly to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having a separate memory cells.

The activation h_t^j of the GRU at time t is a linear interpolation between the previous activation h_{t-1}^j and the candidate activation $\tilde{h}_t^j$:

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j,$$

where an update gate z_t^j decides how much the unit updates its activation, or content. The update gate is computed by

$$z_t^j = \sigma\left(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1}\right)^j$$

This procedure of taking a linear sum between the existing state and the newly computed state is similar to the LSTM unit. The GRU, however, does not have any mechanism to control the degree to which its state is exposed, but exposes the whole state each time.

The candidate activation $\tilde{h}_t^j$ is computed similarly to that of the traditional recurrent unit and as in [Bahdanau et al., 2014],

$$\tilde{h}_t^j = \tanh\left(W \mathbf{x}_t + U\left(\mathbf{r}_t \odot \mathbf{h}_{t-1}\right)\right)^j$$

where $\mathbf{r}_t$ is a set of reset gates and $\odot$ is an element-wise multiplication. When off (r_t^j close to 0), the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state.

The reset gate r_t^j is computed similarly to the update gate:

$$r_t^j = \sigma\left(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1}\right)^j \quad [5]$$

Result of GRU model with:

- Train/Test/Validate ratio as 7/2/1

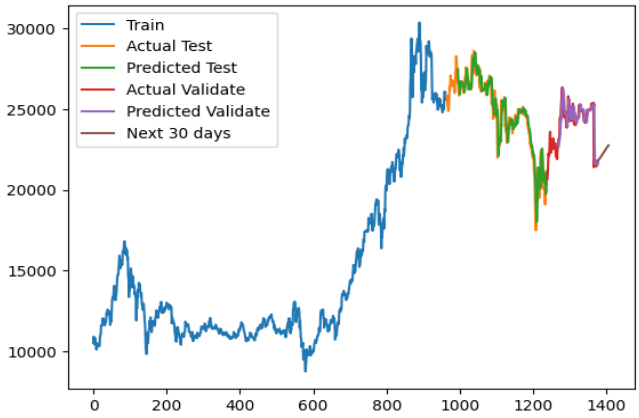


Figure 12. The result of GRU model with data split into the ratio of 7/2/1

- Train/Test/Validate ratio as 7.5/1.5/1

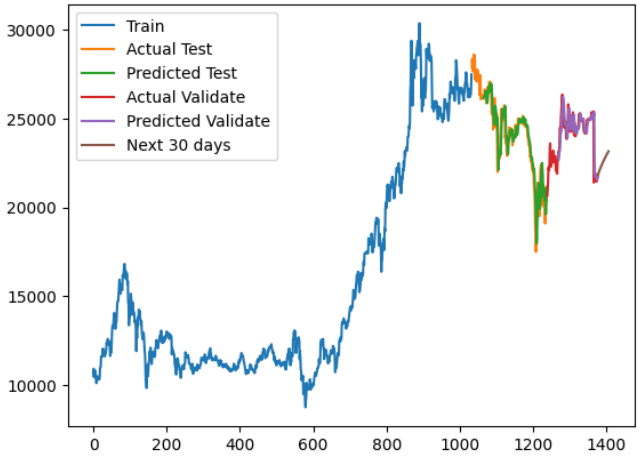


Figure 13. The result of GRU model with data split into the ratio of 7.5/1.5/1

- Train/Test/Validate ratio as 8/1.5/0.5

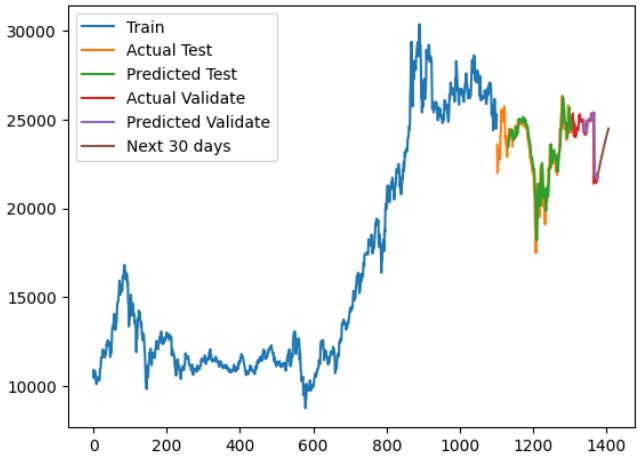
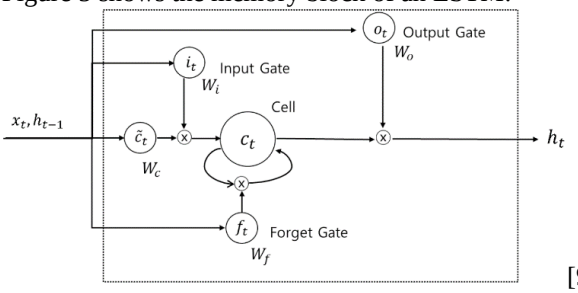


Figure 14.The result of GRU model with data split into the ratio of 8/1.5/0.5

4) Long short-term memory (LSTM)

LSTM is a special kind of RNNs with additional features to memorize the sequence of data. The memorization of the earlier trend of the data is possible through some gates along with a memory line incorporated in a typical LSTM. [2]

Figure 5 shows the memory block of an LSTM.



[9]

Figure 15 Memory block of LSTM

Three types of gates are involved in each LSTM with the goal of controlling the state of each cell:

- Forget Gate outputs a number between 0 and 1, where 1 shows "completely keep this"; whereas, 0 implies "completely ignore this."
- Memory Gate chooses which new data need to be stored in the cell. First, a sigmoid layer, called the "input door layer" chooses which values will be modified. Next, a tanh layer makes a vector of new candidate values that could be added to the state.
- Output Gate decides what will be yield out of each cell. The yielded value will be based on the cell state along with the filtered and newly added data. [2]

Specifically, the mathematical expression of LSTM in Figure.5 can be written as follows:

$$f_t = \sigma\left(W_{fh}h_{t-1} + W_{fx}x_t + b_f\right) \quad (1)$$
$$i_t = \sigma\left(W_{ih}h_{t-1} + W_{ix}x_t + b_i\right) \quad (2)$$
$$\bar{c}_t = \tanh\left(W_{\mathrm{ch}}h_{t-1} + W_{\bar{r}r}x_t + b_{\bar{r}}\right) \quad (3)$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot \bar{c}_t \quad (4)$$
$$o_t = \sigma\left(W_{ch}h_{t-1} + W_{ox}x_t + b_o\right) \quad (5)$$
$$h_t = o_t \cdot \tanh\left(c_t\right) \quad (6)$$

Where W represents weight matrices, b is a bias term, $\sigma(\cdot)$ is a sigmoid function, and $tanh(\cdot)$ is a hyperbolic tangent function. In Equ (1) to Equ (6), the input variables x_t, and h_{t-1} go into the four gates labelled as f_t, i_t, o_t, $\tilde{c}_t$. For the input and output gates, the weights corresponding to each gate are calculated, and the sigmoid function is taken as the activation function. The sigmoid function takes a value between zero and one. If the output value is one, the corresponding value should be kept, but if it zero, the corresponding value should be completely discarded. For the remaining gate, the input modulate gate, *tanh* is used to determine how much new information should be reflected in the cell state. Finally, the information to be reflected in c_t is calculated by adding the point-wise multiplication of the previously calculated i_t, $\tilde{c}_t$ values and the values calculated from the forget gate, the previous cell state value, and the point-wise multiplication of c_t. Finally, in order to calculate the output value h_t, point-wise multiplication is performed on the value calculated from the output gate and the value that is obtained by adding $tanh(\cdot)$ to the calculated cell state value. [9]

Applying the LSTM algorithm to the PVS dataset

LSTM(7-2-1)

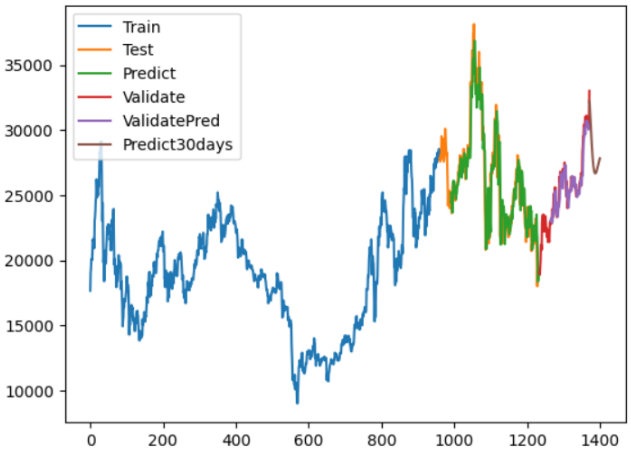


Figure 16. Result of LSTM(7-2-1)

LSTM(7.5-1.5-1)

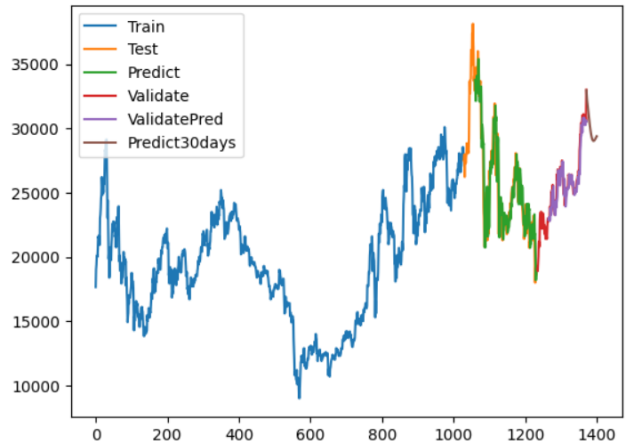


Figure 17. Result of LSTM(7.5-1.5-1)

LSTM(8-1.5-0.5)

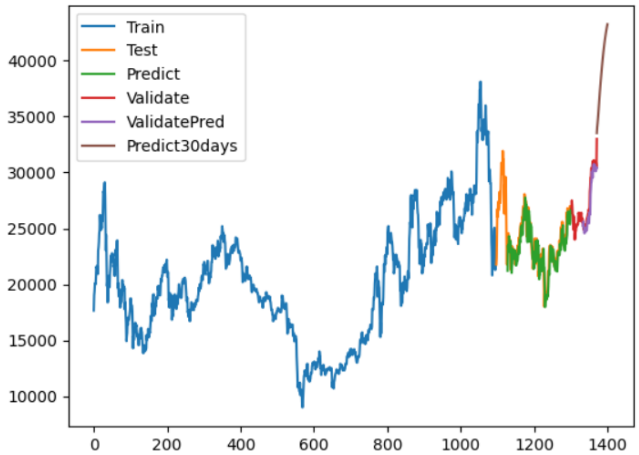


Figure 18. Result of LSTM(8-1.5-0.5)

5) Vector Autoregression (VAR)

The vector autoregression (VAR) model is one of the most successful, flexible,and easy to use models for the analysis of multivariate time series.

The time series Y_t follows a VAR(p) model if it satisfies

$$Y_t = \phi_0 + \Phi_1 Y_{t-1} + \ldots + \Phi_p Y_{t-p} + a_t \quad , p > 0,$$

where ϕ_0 is a k-dimensional vector, and a_t is a sequence of serially uncorrelated random vectors with mean zero and covariance matrix Σ. [6]
Result of VAR model with:
- Train/Test/Validate ratio as 7/2/1

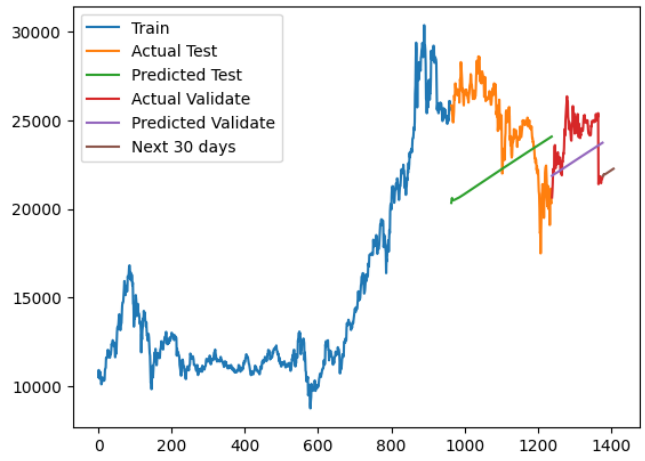


Figure 19.The result of VAR model with data split into the ratio of 7/2/1

- Train/Test/Validate ratio as 7.5/1.5/1

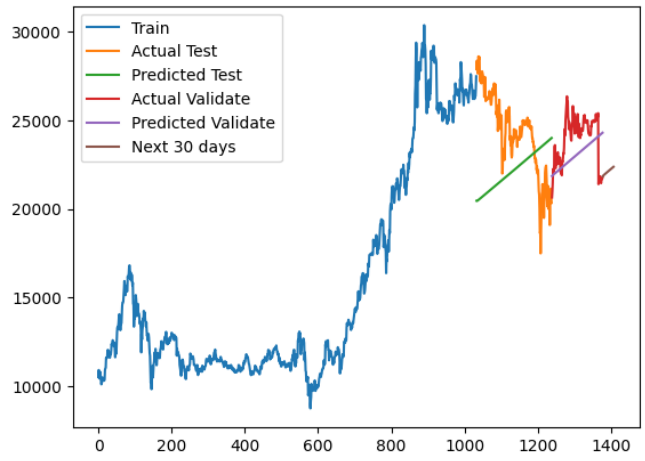


Figure 20. The result of VAR model with data split into the ratio of 7.5/1.5/1

- Train/Test/Validate ratio as 8/1.5/0.5

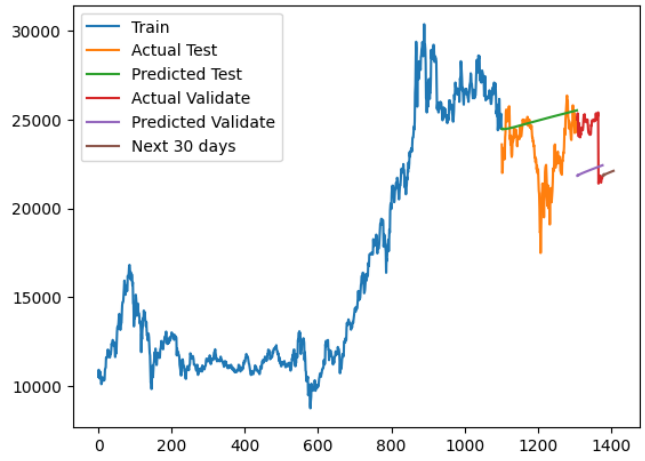


Figure 21. The result of VAR model with data split into the ratio of 8/1.5/0.5

6) Singular spectrum analysis (SSA)

Singular spectrum analysis (SSA) is a technique of time series analysis and forecasting combining elements of classical time series analysis, multivariate statistics, multivariate geometry, dynamical systems and signal processing. SSA seeks to decompose the original series into a sum of a small number of interpretable components such as trend, oscillatory components and noise. It is based on the singular value decomposition of a specific matrix constructed upon the time series.

The basic SSA primarily involves two stages: decomposition and reconstruction. The decomposition consists of embedding and singular value decomposition (SVD). The reconstruction stage consists of eigentriple grouping and diagonal average. These two stages make up the basic SSA algorithm.

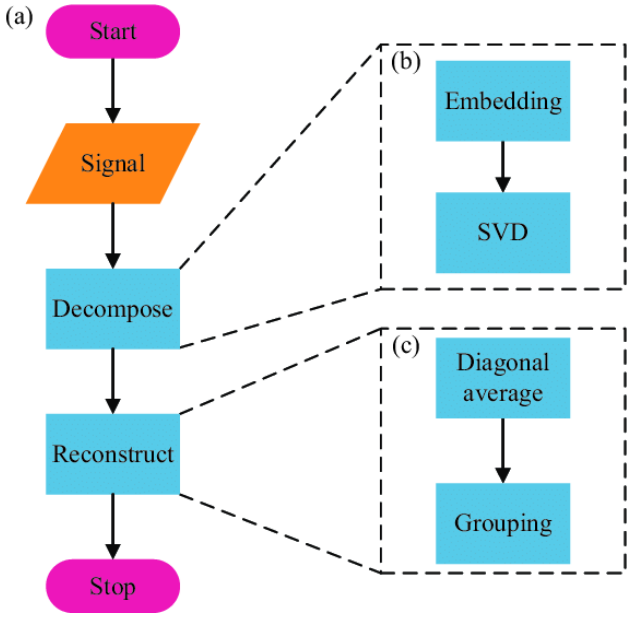


Figure 22: Flowchart of basic singular spectrum analysis (SSA). (a) Procedure of basic SSA; (b) sub-procedure: decomposition; (c) sub-procedure: reconstruction.

Applying Singular spectrum analysis algorithm to the VNM dataset:

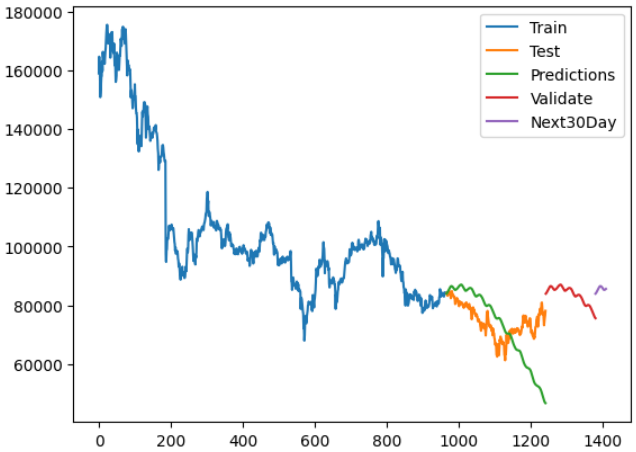


Figure 23. Result of SSA (7-2-1)

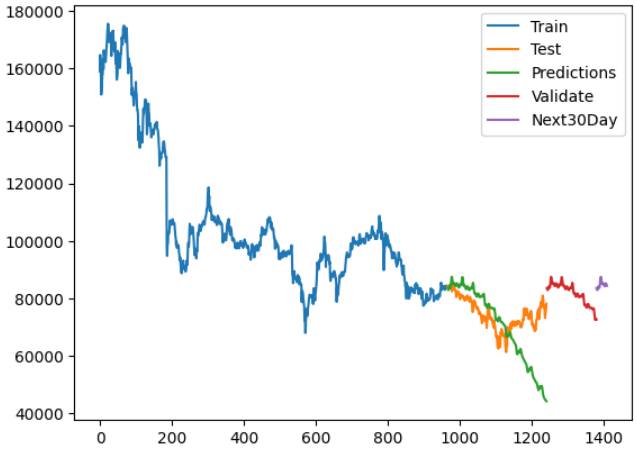


Figure 24. Result of SSA (7.5-1.5-1)

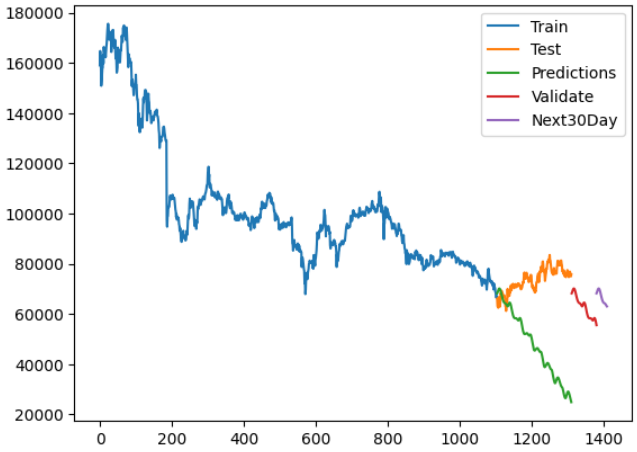


Figure 25. Result of SSA (8-1.5-0.5)

7) Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo (MCMC) is a class of algorithms used to generate samples from complex probability distributions. MCMC methods are particularly useful for Bayesian inference, which is a statistical framework for updating beliefs about uncertain quantities based on observed data.

The basic idea behind MCMC methods is to construct a Markov chain whose stationary distribution is the target probability distribution of interest.

Define a Markov chain :

A Markov chain $(X_t)_{t\geq 0}$ is a discrete-time stochastic process $\{X_0, X_1, \ldots\}$ with the property that, given $X_0, X_1, \ldots, X_{t-1}$, the distribution of X_t depends only on X_{t-1}. Formally, $(X_t)_{t\geq 0}$ is a Markov chain if for all $A \subset S$, where S is the state space,

$$P(X_t \in A | X_0, \ldots, X_{t-1}) = P(X_t \in A | X_{t-1}). \quad [1]$$

Although the MCMC algorithm has several popular versions, in this paper we will use the Metropolis–Hastings algorithm to implement our project.

Let p(.) denote the posterior distribution of a random vector X. Then for a Metropolis–Hastings chain $(X_t)_{t\geq 0}$, p(.) is the target distribution. To begin, choose an initial value x_0 from some distribution. The prior distribution is a common choice of distribution from which to draw the initial state. At iteration t = 1,2,…, propose a new value, say x*, from a density of $q(.|x_t)$ that can be used as a candidate for the next value of x. Let

$$\alpha(\mathbf{x}^*, \mathbf{x}_t) = \min\left\{\frac{p(\mathbf{x}^*)q(\mathbf{x}_t|\mathbf{x}^*)}{p(\mathbf{x}_t)q(\mathbf{x}^*|\mathbf{x}_t)}, 1\right\}. \quad [1]$$

Then x* is selected as the value of x_{t+1} with probability $\alpha(x^*, x_t)$, and $x_{t+1} = x_t$ with probability $1 - \alpha(x^*, x_t)$.

Applying the MCMC algorithm to the PVS dataset

MCMC(7-2-1)

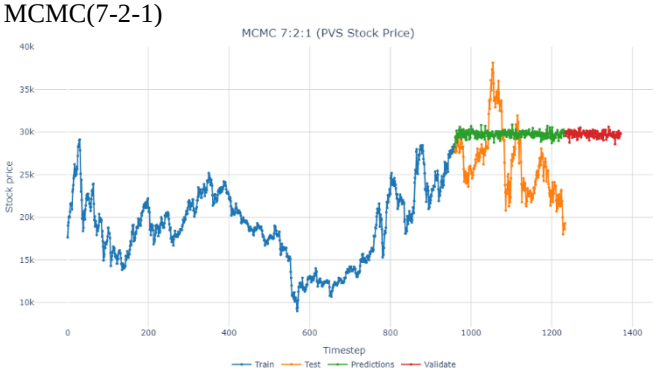


Figure 26. Result of MCMC (7-2-1)

MCMC(7.5-1.5-1)

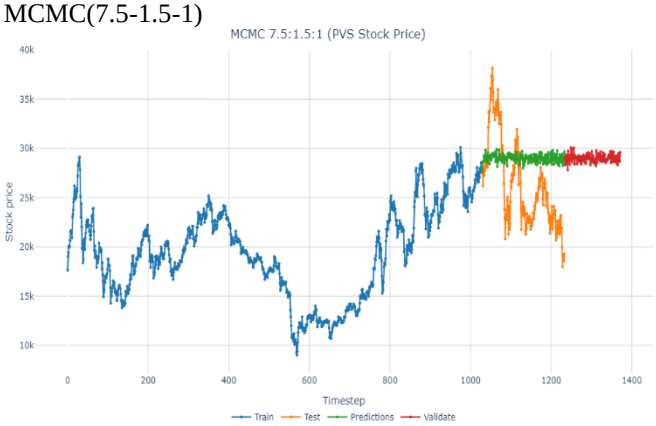


Figure 27. Result of MCMC (7.5-1.5-1)

MCMC(8-1.5-0.5)

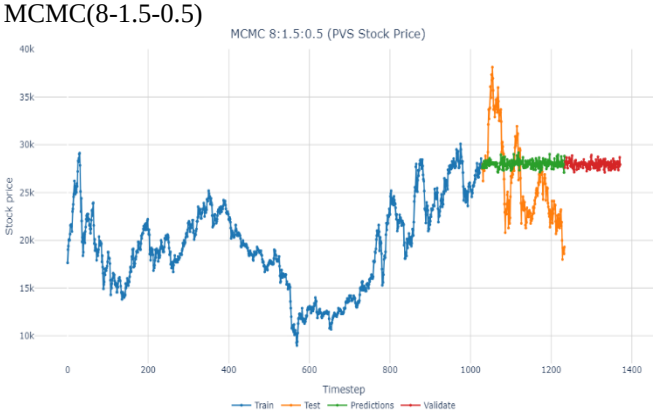


Figure 28. Result of MCMC (8-1.5-0.5)

8) Sequence to Sequence

A Recurrent Neural Network, or RNN, is a network that operates on a sequence and uses its own output as input for subsequent steps.

A Sequence to Sequence network, or seq2seq network, or Encoder Decoder network, is a model consisting of two RNNs called the encoder and decoder. The encoder reads an input sequence and outputs a single vector, and the decoder reads that vector to produce an output sequence.

Applying the Seq2Seq algorithm to the VNM dataset:

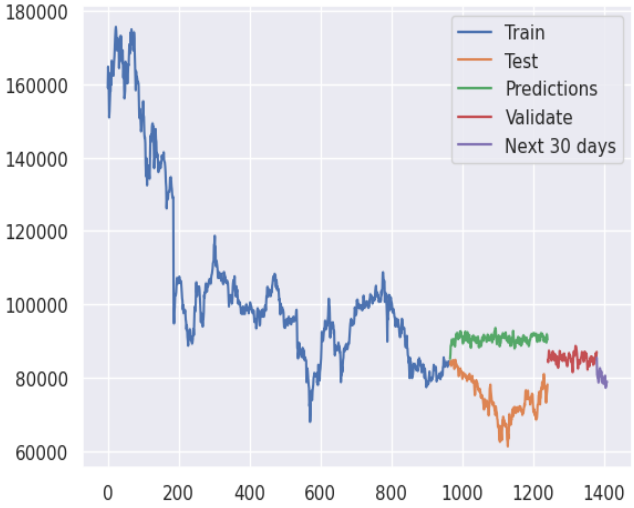


Figure 29. Result of Seq2Seq (7-2-1)

Figure 30. Result of Seq2Seq (7.5-1.5-1)

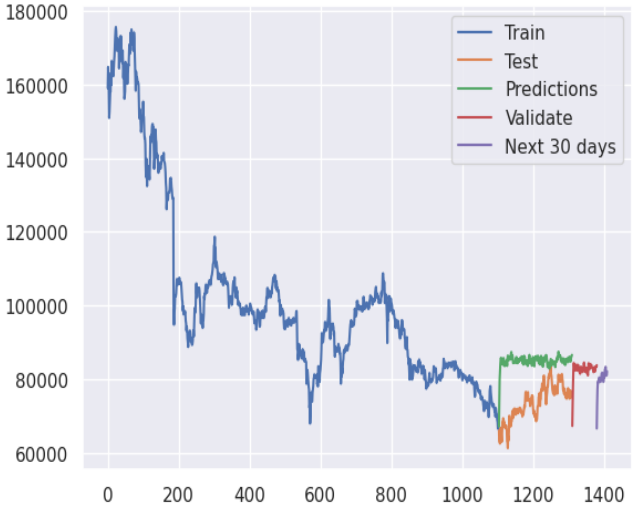


Figure 31. Result of Seq2Seq (8-1.5-0.5)

9) Fully Convolutional Neural Networks (FCN)

Fully Convolutional Neural Networks (FCN) were first proposed in Wang et al. (2017b) for classifying univariate time series and validated on 44 datasets from the UCR/UEA archive. FCN 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. [7]

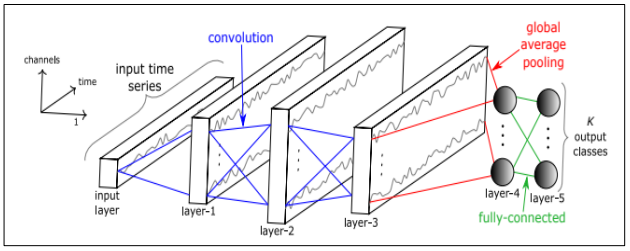


Figure 32. Fully Convolutional Neural Network architecture

Result of FCN model with:

- Train/Test/Validate ratio as 7/2/1

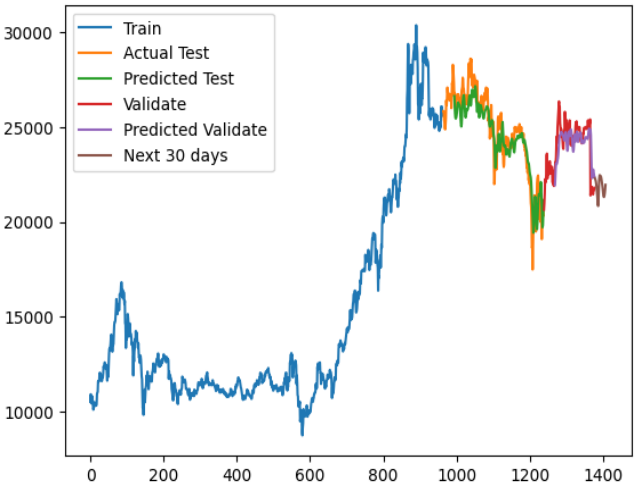


Figure 33. The result of FCN model with data split into the ratio of 7/2/1

- Train/Test/Validate ratio as 7.5/1.5/1

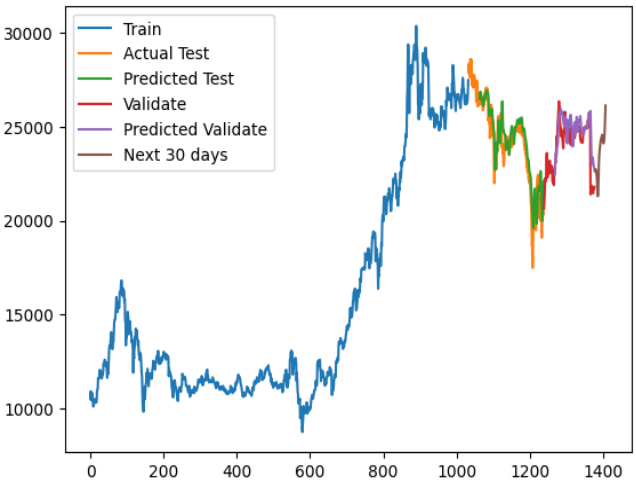


Figure 34. The result of FCN model with data split into the ratio of 7.5/1.5/1

- Train/Test/Validate ratio as 8/1.5/0.5

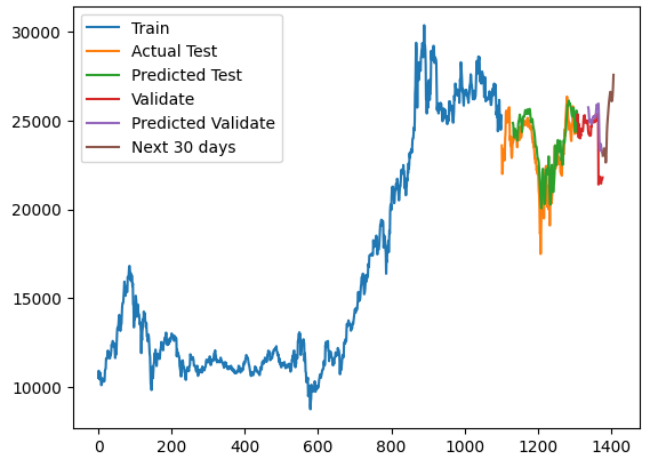


Figure 35. The result of FCN model with data split into the ratio of 8/1.5/0.5

B. Model Evaluation

1) Mean Absolute Percentage Error (MAPE)

MAPE is the mean absolute percentage error, which is a relative measure that essentially scales MAD to be in percentage units instead of the variable's units. Mean absolute percentage error is a relative error measure that uses absolute values to keep the positive and negative errors from canceling one another out and uses relative errors to enable you to compare forecast accuracy between time-series models.

Formula

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad [10]$$

Where:

- n is the number of fitted points
- A_t is the actual value
- F_t is the forecast value

2) Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.

Formula

$$RMSE = \sqrt{(f-o)^2} \quad [11]$$

Where:

- f = forecasts (expected values or unknown results)
- o = observed values (known results)

3) Mean Directional Accuracy (MDA)

The Mean Directional Accuracy (MDA) is a statistical measure used to evaluate the accuracy of a directional model or forecast. It compares the forecast direction to the actual realized direction of a time series, assuming the time series is homogeneous and the same size as the forecast.

Syntax: MDA (X, F) [12]

With:

- X is the eventual outcome time series sample data (a one-dimensional array of cells e.g., row or column).
- F is the forecast time series data (a one-dimensional array of cells e.g., row or column).

Observations with missing values in Y or F are excluded from the calculation.

The mean directional accuracy is given by:

$$\mathrm{MDA} = \frac{1}{N}\sum_{t}\mathbf{1}_{sign(X_t - X_{t-1}) == sign(F_t - X_{t-1})} \quad [12]$$

Where:

- $\{X_i\}$ is the actual observations time series.
- $\{F_i\}$ is the estimated or forecast time series.
- N is the number of non-missing data points.
- sign(·) is sign function.
- 1 is the indicator function.

4) Evaluate the model on the validation set

Mean Directional Accuracy (MDA)

With PVS Dataset:

Model	Train-Test-Val	RMSE	MAPE (%)	MDA (%)
ARIMA	7:2:1	4349.145	14.447	4.762
	7.5:1.5:1	4669.692	15.906	4.902
	8:1.5:0.5	2577.805	8.545	5.882
LSTM	7:2:1	1127.151	3.257	41.322
	7.5:1.5:1	1110.34	3.418	43.353
	8:1.5:0.5	848.728	2.758	44.509
MCMC	7:2:1	5201.783	18.476	43.59
	7.5:1.5:1	5193.602	18.668	43.627
	8:1.5:0.5	4804.443	16.58	45.588

Table 1. Evaluate models on the test set with PVS Dataset

With VNM dataset:

Model	Train-Test-Val	RMSE	MAPE (%)	MDA (%)
Linear	7:2:1	16954.952	22.149	42.182
	7.5:1.5:1	10324.827	11.837	50.0
	8:1.5:0.5	15274.755	17.471	46.602
SSA	7:2:1	11573.929	12.511	48.364
	7.5:1.5:1	12433.374	12.29	47.272
	8:1.5:0.5	31132.573	34.892	49.029
Seq2Seq	7:2:1	16954.952	22.149	42.182
	7.5:1.5:1	16325.868	22.149	59.223
	8:1.5:0.5	12265.811	15.812	50.485

Table 2. Evaluate models on the test set with VNM Dataset

With ACB Dataset

Model	Train - Test - Val	RMSE	MAPE (%)	MDA (%)
GRU	7:2:1	549.326	1.659	43.621
	7.5:1.5:1	568.937	1.778	45.402
	8:1.5:0.5	556.454	1.866	48.851
VAR	7:2:1	4109.376	14.099	48.175
	7.5:1.5:1	3827.133	13.193	46.829
	8:1.5:0.5	2399.438	8.16	48.293

FCN	7:2:1	918.105	3.003	51.639
	7.5:1.5:1	882.079	2.99	47.429
	8:1.5:0.5	993.943	3.662	46.857

Table 2. Evaluate models on the test set with ACB Dataset

IV. CONCLUSION

A. Challenges Encountered

During the implementation of the research project "Predicting Stock Prices in Vietnam Using Statistical and Machine Learning Models," we encountered several challenges, including:

- Difficulty in data processing: Stock market data is often complex and diverse, requiring scientific and accurate data processing methods to ensure feasibility and accuracy of prediction models.
- Difficulty in building prediction models: Stock market prediction models are often complex and require in-depth knowledge of this field. We had to make critical decisions in selecting and building prediction models, such as which algorithm to use, how to process data, and which important variables to include in the model.
- Difficulty in evaluating model effectiveness: To evaluate the effectiveness of prediction models, we used several different algorithmic and statistical indicators, but the results showed that the accuracy of the models was still not high.

In the future, we will try to address the above challenges and provide better solutions for predicting stock prices by:

- Improving skills in selecting and processing data: We will continue to research and apply the most advanced methods in selecting and processing data to ensure the feasibility and accuracy of prediction models.
- Applying advanced prediction models: We will continue to research and apply the most advanced prediction models such as Deep Learning and Reinforcement Learning to enhance the effectiveness and accuracy of prediction models.
- Strengthening model effectiveness evaluation: We will continue to research and apply the latest and widely accepted indicators in the field of stock price prediction, such as Mean Absolute Scaled Error (MASE), Mean Absolute Error Percentage (MAPE), and Symmetric Mean Absolute Percentage Error (SMAPE), to evaluate the effectiveness of prediction models.
- Enhancing cooperation and sharing experience: We will continue to seek and participate in communities and forums specializing in stock price prediction to share experiences and learn from experts in this field.

With the above solutions, we believe that we can improve the effectiveness and accuracy of stock price prediction models in the future.

B. Conclusion

After comparing models with different splitting ratios on 3 datasets PVS, ACB, VNM, we have obtained the model that give the best forecast result for each dataset as follows:

- For PVS dataset: The best model is LSTM with the splitting ratio of Train/Test/Validate as 8/1.5/0.5, RMSE value as 848.728, MAPE as 2.758%, MDA as 44.509%.
- For VNM dataset: The best model is Linear Regression with the splitting ratio of Train/Test/Validate as 7.5/1.5/1, RMSE value as 10324.827, MAPE as 11.837%, MDA as 50.0%.
- For ACB dataset: The best model is GRU with the splitting ratio of Train/Test/Validate as 7/2/1, RMSE value as 549.326, MAPE as 1.659%, MDA as 43.621%.

The results show that the LSTM, GRU, and Linear Regression algorithms had good performance in predicting stock prices in Vietnam. This suggests that these algorithms can be used in practical applications for stock price forecasting.

However, to develop stock price forecasting technology, we need to focus on the following issues:

- Improving the accuracy of the model: Although the LSTM, GRU, and Linear Regression algorithms have shown good results in predicting stock prices, we still need to improve the accuracy of the model to ensure more accurate forecasting results.
- Algorithm optimization: We need to search for and use optimization algorithms to enhance the performance and accuracy of the forecasting model.
- Developing real-world applications: Stock price forecasting technology can be applied in other fields such as finance, business, and investment. We can search for and develop real-world applications to enhance the value and application of this technology.
- Researching new forecasting models: As new forecasting algorithms and models are being researched and developed, we need to study new algorithms and models to improve the forecasting ability of stock prices.

In summary, to develop stock price forecasting technology, we need to focus on improving the accuracy of the model, optimizing algorithms, developing real-world applications, and researching new forecasting models. These technologies will help us improve the ability to forecast stock prices and meet the needs of the financial market.

ACKNOWLEDGMENT

First of all, we sincerely express our gratitude to Assoc. Prof. Dr. Nguyen Dinh Thuan and TA. Nguyen Minh Nhut for providing us with the expertise necessary to complete this project, as well as for your enthusiastic and sincere guidance and assistance. I believe the group's report would be extremely difficult to finish without your passionate supervision.

This is also an opportunity for each team member to collaborate, improve their cooperation abilities, learn from one another, and, most importantly, implement products during the course.

During the implementation of the project, the team applied the knowledge they had been taught and used new things, with the desire to be able to complete the work most perfectly. However, with limited time, knowledge, and experience, shortcomings cannot be avoided, so the group is looking forward to receiving valuable suggestions from you to help

the group supplement and improve their knowledge to better serve future projects and actual work.

Finally, my team wishes you a lot of health to continue to carry out your noble mission of imparting knowledge to future generations

REFERENCES

[1]D. Spade, “Markov chain Monte Carlo methods: Theory and practice,” in *Handbook of Statistics*, 2020. doi: 10.1016/bs.host.2019.06.001.

[2]S. S. Namin and A. S. Namin, “1. Graduate Research Assistant and Ph.D. Student, Department of Agricultural and Applied Economics”.

[3]A. Osmanzade, “Singular spectrum analysis forecasting for financial time series,” Thesis, Tartu Ülikool, 2017. Accessed: Jun. 17, 2023. [Online]. Available: https://dspace.ut.ee/handle/10062/57101

[4]G. Zhang, X. Bai, and Y. Wang, “Short-time multi-energy load forecasting method based on CNN-Seq2Seq model with attention mechanism,” *Mach. Learn. Appl.*, vol. 5, p. 100064, Sep. 2021, doi: 10.1016/j.mlwa.2021.100064.

[5]J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.” arXiv, Dec. 11, 2014. Accessed: Jun. 17, 2023. [Online]. Available: http://arxiv.org/abs/1412.3555

[6]S. Bayraci, Y. Ari, and Y. Yildirim, “A VECTOR AUTO-REGRESSIVE (VAR) MODEL FOR THE TURKISH FİNANCIAL MARKETS”.

[7]H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “Deep learning for time series classification: a review,” *Data Min. Knowl. Discov.*, vol. 33, no. 4, pp. 917–963, Jul. 2019, doi: 10.1007/s10618-019-00619-1.

[8]“Linear Regression in Machine learning - Javatpoint.” https://www.javatpoint.com/linear-regression-in-machine-learning (accessed Jun. 17, 2023).

[9]T. Kim and H. Y. Kim, “Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data,” *PLOS ONE*, vol. 14, no. 2, p. e0212320, Feb. 2019, doi: 10.1371/journal.pone.0212320.

[10]E. I. D. Team, “MAPE (Mean Absolute Percentage Error),” *Oracle Help Center*. https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusu/insights_metrics_MAPE.html#GUID-C33B0F01-83E9-468B-B96C-413A12882334 (accessed Jun. 17, 2023).

[11]“RMSE: Root Mean Square Error,” *Statistics How To*. https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/ (accessed May 14, 2023).

[12]S. F. info@numxl.com, “MDA - Mean Directional Accuracy,” *Help center*, Jun. 14, 2019. https://support.numxl.com/hc/en-us/articles/360029220972-MDA-Mean-Directional-Accuracy (accessed Jun. 17, 2023).