VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF INFORMATION TECHNOLOGY

INFORMATION SYSTEM FACULTY

SUBJECT: BUSINESS ANALYSIS

FINAL PROJECT REPORT
FORECASTING VIETNAMESE BANKS STOCKS PRICE BY STATISTICAL MODELS AND MACHINE LEARNING ALGORITHMS

Lecturer: Assoc. Prof. Nguyen Dinh Thuan
Mr. Nguyen Minh Nhut
Ms. Nguyen Thi Viet Huong

Class: IS403.N21.HTCL

Student performance:

Bui Duc Duy	20521228
Nguyen Thi Cam Van	20522145
Ho Bao An	20520876

Ho Chi Minh City, June 2023

VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF INFORMATION TECHNOLOGY

INFORMATION SYSTEM FACULTY

SUBJECT: BUSINESS ANALYSIS

FINAL PROJECT REPORT FORECASTING VIETNAMESE BANKS STOCKS PRICE BY STATISTICAL MODELS AND MACHINE LEARNING ALGORITHMS

Lecturer: Assoc. Prof. Nguyen Dinh Thuan
Mr. Nguyen Minh Nhut
Ms. Nguyen Thi Viet Huong

Class: IS403.N21.HTCL

Student performance:

Bui Duc Duy	20521228
Nguyen Thi Cam Van	20522145
Ho Bao An	20520876

Ho Chi Minh City, June 2023

TEACHER'S COMMENTS

TABLE OF CONTENTS

TEACHER'S COMMENTS..3
TABLE OF CONTENTS..4
CHAPTER I : INTRODUCTION..5
CHAPTER II: RELATED WORK..6
CHAPTER III: MATERIALS AND METHODS..8
3.1 Data Overview..8
3.2 Algorithm..11
3.2.1 Linear Regression Model..11
3.2.2 ARIMA Model..11
3.2.3 SARIMAX Model..13
3.2.4 K-Nearest Neighbors Model..13
3.2.5 Gradient Boosting Tree Model..14
3.2.6 Bayesian Dynamic Linear Model..15
3.2.7 Recurrent Neural Network Model..16
3.2.8 Long short-term Memory Model..18
3.2.9 Gated Recurrent Unit Model..20
CHAPTER IV: EVALUATION METHODOLOGY..21
4.1 Mean Absolute Error (MAE)..21
4.2 Root Mean Squared Error (RMSE)..21
4.3 Mean Absolute Percent Error (MAPE)..21
CHAPTER V: RESULT..22
CHAPTER VI: CONCLUSION..30
TASK ASSIGNMENT TABLE..31
REFERENCES..32

CHAPTER I: INTRODUCTION

The stock market is one of the factors that have a great influence on the development of a country's market economy, and Vietnam is no exception.

For the Government, this is a tool to raise capital and enrich the budget in a sustainable way through: stocks, bonds, fund certificates, etc. Securities transaction costs will be collected by the Stock Exchange. for the purpose of increasing the state budget.

The stock market is also an extremely effective place to raise capital for businesses by selling shares of company ownership. The amount of money collected will be used as investment capital for business, production, etc.

For investors, the stock market provides the public with diverse and rich investment channels so that investors can choose the right type for themselves and come up with an effective investment strategy.

In the field of investment, forecasting stock prices is an important and challenging task. To help investors and professionals make smart investment decisions, the use of predictive algorithms and models has become a popular and useful method.

Using stock price forecasting algorithms has several important benefits. First, it can help identify market trends and volatility, thereby helping investors make decisions to buy, sell, or hold stocks. Second, it provides a means of assessing the return and risk potential of particular stocks. Third, stock price forecasting algorithms also help enhance risk management and personal or business financial planning. However, forecasting stock prices is not an easy task and can be fraught with challenges, uncertainties and risks that are always present in predicting stock prices. Therefore, using algorithms and models to forecast stock prices is a useful tool in evaluating and making investment decisions.

In this article, we will use **Linear Regression, ARIMA, SARIMAX, LSTM, GRU, RNN, BDLM, GBT and KNN** models to predict the stock prices of 3 companies in the next 30 days. From there, evaluate and compare the above 9 models to choose the most suitable model.

CHAPTER II: RELATED WORK

Bitcoin is the most popular and valuable cryptocurrency in the financial market which attracts traders for investment and opens new research opportunities for researchers. Many research works have been done on bitcoin price prediction with different machine learning prediction algorithms.

Yang Si (2022) [1] used the ARIMA model to analyze and forecast the adjusted closing price of bitcoin. She used two models, namely ARIMA (5, 2, 1) and ARIMA (0, 2, 2) are selected and comparations between them are made. When comparing ARIMA (5, 2, 1) - RMSE = 0.275 and ARIMA (0, 2, 2) - RMSE = 0.299, it can be seen that ARIMA (5, 2, 1) is better than ARIMA (0, 2, 2) when the forecasting period is 5-day.

To conclude, ARIMA model perform better while making short run predictions than in the long run.

Mohammad Ali and Swakkhar Shatabda (2020) [2] used the LR model to Predict Bitcoin Price. Predict bitcoin price in 7 days with linear regression model. When they trained the linear regression model with a suitable chunk of data determined by their methods, they looked for acceptable prediction results. The percentage error method was applied to calculate the error with an accuracy of 96.97%.

Namrata Hemraj Gawali (2021) [3] used Neural Network to predict price of bitcoin by reviewing several aspects that affect the bitcoin price. Stock price data as an exogenous variable to build SARIMAX model. When the result of designed models were compared, it was observed that the LSTM – RMSE = 3310.35 model outperformed all SARIMA – RMSE = 5986.31, SARIMAX – RMSE = 4944.03 and RNN – RMSE = 8008.36.

Md. Ebtidaul Karim, Md. Foysal and Sunanda Das [4] used Bi-LSTM and GRU to Stock Price Prediction based Hybrid Deep Learning Approach. The test results show that the proposed Bi-LSTM-GRU model achieves the lowest error values of RMSE 0.0042403, MAE 0.0028890, MAPE 28,058 with higher performance than individual models.

Malti Bansal, Apoorva Goyal, Apoorva Choudhary (2022) [5] using 5 algorithms namely K-Nearest Neighbors, Linear Regression, Support Vector Regression, Decision Tree

Regression, and Long Short-Term Memory for predicting stock prices of 12 leading companies of the Indian stock market and inferred that the DL algorithm outperforms all the other algorithms for stock price or time series prediction and provides results with extensive accuracy specifically LSTM algorithm is the best choice among the given algorithms for time series prediction with RMSE(22.55) while knn and linear regression are lower than rmse by 56.44 and 51.20 respectively.

The authors, Ge Chenghan and Wang Tao (2020) [6] conducted research on the "Improvement of Bayesian Dynamic Linear Model for Predicting Missing Data of Bridges." In this paper, they employed the Bayesian Dynamic Linear Model to predict missing data in bridge-related studies. By comparing the predicted result with the observed value, it is found that the absolute error is less than 14.05Hz and the relative error is less than 1.82% when the training frequency value varies from 756 Hz to 773.4 Hz.

CHAPTER III: MATERIALS AND METHODS

3.1 Data Overview

Bitcoin is the most popular and valuable cryptocurrency in the financial market which attracts traders for investment and opens new research opportunities for researchers. Many research works have been done on bitcoin price prediction with different machine learning prediction algorithms.

We extract data sets on stock market prices of some large banks in Vietnam including Joint Stock Commercial Bank for Investment and Development of Vietnam (BID), Sai Gon Thuong Tin Commercial Joint Stock Bank (STB) and Vietcombank (VCB). The historical data of VN 30 has been collected from Investing.com [7]. The dataset retrieved from December 1, 2017 to June 9, 2023 has 1376 rows and 7 attribute columns including Date, Price, Open, High, Low, Vol., Change %.

In this article, we just use the Close as an input for predicting, in other that the table has two rows which are Price and Date. The dataset is divided into three subsets: training, testing and validating dataset that we split into 3 different scales: 75/15/10, 70/20/10 and 65/25/10.

The figure below uses the library matplotlib.pyplot to help us visualize the correlation between stock prices and date.

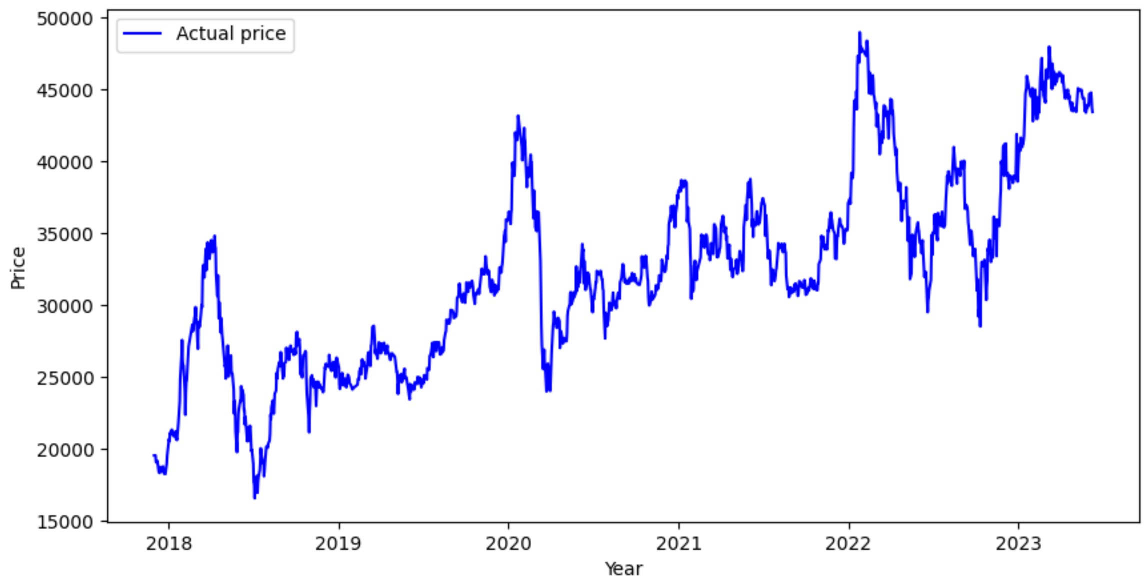

Figure 1. Visualize data of BID

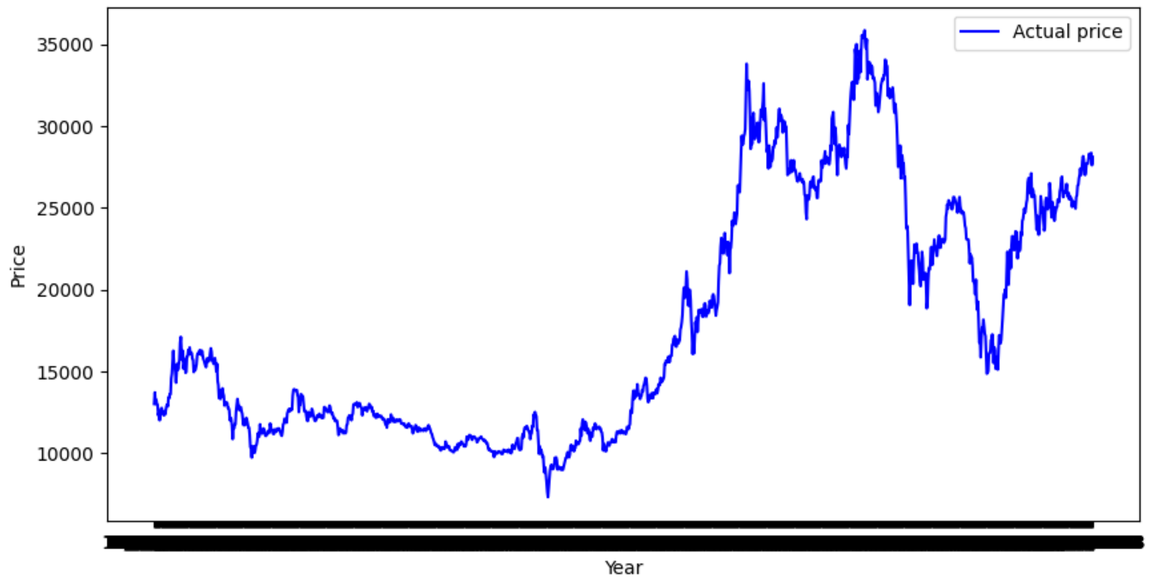

Figure 2. Visualize data of STB

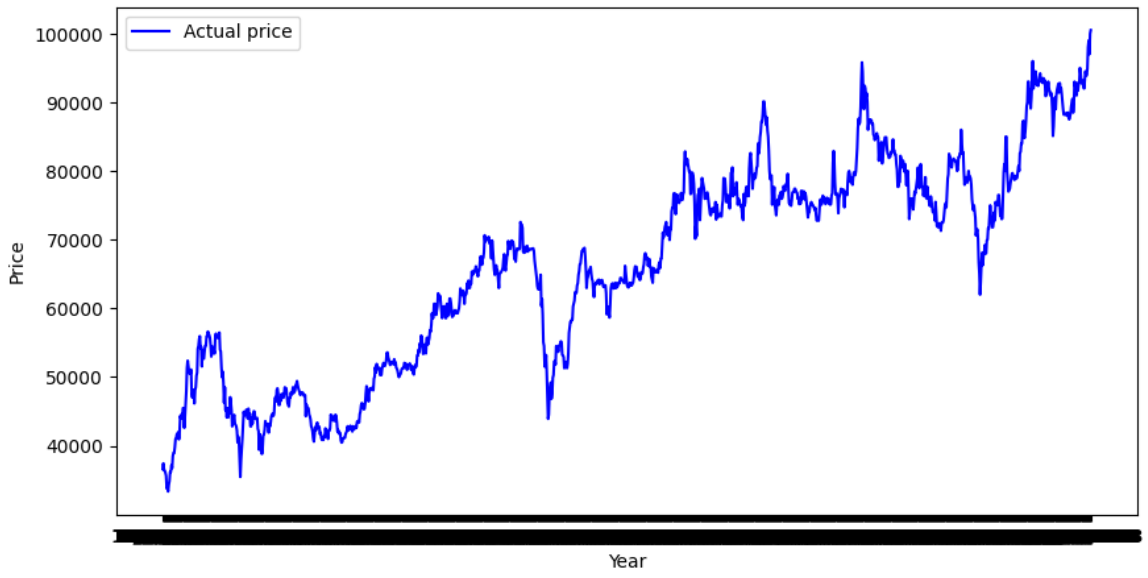

Figure 3. Visualize data of VCB

TABLE 1. DATA SUMMARY

	Price		
	BID	**STB**	**VCB**
count	1376.0	1376.0	1376.0
mean	32278.943	17889.935	66180.947
std	6761.459	7404.637	15338.628
min	16531.4	7300.0	33360.0
25%	26882.325	11500.0	51989.0
50%	31888.950	14950.0	67715.5
75%	36037.225	24950.0	77400.0
max	49000.0	35850.0	100500.0
median	31888.950	14950.0	67715.5
mode	24873.7	11400.0	63738.0
range	32468.6	28550.0	67140.0

3.2 Algorithm

3.2.1 Linear Regression Model

The figure below uses the library matplotlib.pyplot to help us visualize the correlation between stock prices and date.

Linear Regression Linear Regression (LR) is a predictive model that formulates a line of best fit between a scalar dependent variables and multiple explanatory variables. Linear fit occurs by minimizing the mean squared error between the predicted and actual output [8].

The formulation of Simple Linear Regression can be described as below:

$$\boldsymbol{y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 X} + \boldsymbol{\varepsilon}$$

Where,

- $\mathbf{y}$ is the response
- $\boldsymbol{\beta_0}$ is the intercept.
- $\boldsymbol{\beta_1}$ is the regression coefficient.
- $\boldsymbol{X}$ is independent variable.
- $\boldsymbol{\varepsilon}$ is the error of the estimate.

The formulation of Multiple Linear Regression can be described as below:

$$\boldsymbol{y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 X_1} + \boldsymbol{\beta_2 X_2} + \cdots + \boldsymbol{\beta_k X_k} + \boldsymbol{\varepsilon}$$

The parameter $\beta_1, \beta_2, \ldots, \beta_k$ in this model are often called partial regression coefficients.

3.2.2 ARIMA Model

ARIMA (Autoregressive Integrated Moving Average) was created in 1970 by George Box and Gwilyn Jenkins. It is a widely used time series forecasting model in the financial sector. The ARIMA model was created by combining the Moving Average (MA) and Auto-Regressive (AR) model, both of which predict future values using lagged data [1].

AR: Autoregression – The model uses the dependent relationship between current data and its past values (p):

$$\boldsymbol{y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t}$$

Depending on the value of p, there are the following scenarios:

AR(0): If the p parameter is set to zero, there are no autoregressive terms, so this time series is just white noise.

AR(1): With the p parameter set to 1, we are taking into account the previous timestamp adjusted by a multiplier, and then adding white noise.

AR(p): With the p parameter set to 1, we are taking into account the previous timestamp adjusted by a multiplier, and then adding white noise.

I: Integrated - To make the time series stationary by measuring the differences of observations at different time (d).

$$\nabla \boldsymbol{y_t} = \boldsymbol{y_t} - \boldsymbol{y_{t-1}}$$

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).

$$\boldsymbol{y_t = c + \theta_1 y_{t-1} + \theta_2 y_{t-2} + \cdots + \theta_q y_{t-q} + \varepsilon_t}$$

The regression equation ARIMA(p, d, q) can be expressed as:

$$\boldsymbol{y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1}}$$
$$\boldsymbol{-\theta_2 \varepsilon_{t-2} - \cdots - \theta_p \varepsilon_{t-p}}$$

Where,

- $\boldsymbol{y_t}$ is the value of the time series at time t
- $\boldsymbol{\varepsilon_t}$ is the error at time t
- $\boldsymbol{\varphi_i, \theta_i}$ are the coefficients

ARIMA models may be defined as [9]:

- **ARIMA(1,0,0)** – first-order autoregressive model
- **ARIMA(0,1,0)** – random walk model
- **ARIMA(1,1,0)** – differenced first-order autoregressive model
- **ARIMA(0,1,1)** – simple exponential smoothing
- **ARIMA(0,2,1)** or ARIMA**(0,2,2)** – linear exponential smoothing
- **ARIMA(1,1,2) -** damped-trend linear exponential smoothing

3.2.3 SARIMAX Model

The SARIMAX model is an improved version of the SARIMA model, with exogenous factors (X) as external feature parameters for enhancing the model's performance, reducing the prediction errors, overcoming the autocorrekation issues, and improving the prediction results.

The SARIMAX model consists of both seasonl effects and exogenous factors that can be used as **SARIMAX(p,d,q)(P,D,Q)** while the exogenous factors are optional parameters.

The exogenous factors can be external parallel time series daraa such as wind speed or temperature values that have the same correlation with the original data which need to be predicted [10].

- **p** is the order (number of time lags) of the autoregressive part of the model
- **d** is the degree of differencing (the number of times the data have had past values subtracted)
- **q** is the order of the moving-average part of the model
- **P** is the order (number of time lags) of the seasonal part of the model
- **D** is the degree of differencing (the number of times the data have had past values subtracted) of the seasonal part of the model
- **Q** is the order of the moving-average part of the seasonal part of the model

3.2.4 K-Nearest Neighbors Model

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another [11].

K in KNN algorithms is the number of neighborhood points to a query point. The assignment of a class label to a query point depends on the hyperparameter K.i.e, if we assign a value of 5 for K, then we must consider 5 nearest neighbor data points to the query point. The class label will be assigned based on the majority class label out of the 5 neighborhood points. K can be assigned to any positive integer value, but it is more likely to assign odd positive integers to avoid the tie arise while taking a majority vote to predict class label.

The shortest distance between two data points is called Euclidean distance, In 2 dimension, the distance between **A(x_1, y_1)** and **B(x_2, y_2)** is:

$$\sqrt{[(\mathbf{x}_2 - \mathbf{x}_1)^2 - (\mathbf{y}_2 - \mathbf{y}_1)^2]}$$

In this approach, the points which are closer are considered as more similar and grouped into the same category. The points which are farther away are treated as points belonging to different categories.

In the case we discussed with K = 5, the KNN algorithm analyzes 5 data points which are at the shortest distance from the query point using Euclidean distance measure.

The algorithm predicts the class label of the query point as the same as that of the majority data points in the neighborhood. It estimates out of the nearest neighbors, to which class the majority of the neighboring points belongs.

3.2.5 Gradient Boosting Tree Model

Gradient boosting is a type of algorithm that belongs to the field of machine learning, based upon the ensemble of multiple models. In fact, boosting is considered one of the most influential ideas introduced in the last thirty years in machine learning. In contrast to the most frequent approaches to data-driven modelling where a single "strong" predictive model is fitted, the boosting algorithm depends on merging several relatively "weak" predictive models to get a stronger ensemble estimator. However, unlike other popular machine-learning ensembles techniques, such as Random Forest which is based upon directly averaging the predictions of the predictive models, boosting is grounded on a forward stagewise strategy where a new predictive model is added in sequence [12].

Traditionally, modelers would try to find a function that can accurately describe the data. However, the function can only be an approximation of the data distribution and there must be errors:

$$\boldsymbol{y_i} = \mathbf{F1}(\boldsymbol{X_i}) + \mathbf{error}_{\mathbf{1}\boldsymbol{i}}$$ [13]

where y_i is the outcome variable and X_i is a vector of predictors.

Suppose that the F1(X_i) function is a weak learner and the relationship between X and y is not fully described. In this situation, the error or residual is not white noise but has some correlation with y. We may want to train a second model on the error term:

$$\boldsymbol{y_i} - \mathbf{F1}\ (\boldsymbol{X_i}) = \mathbf{error}_{\boldsymbol{1i}} = \boldsymbol{h_1}\ (\boldsymbol{X_i}) + \mathbf{error}_{\boldsymbol{2i}}\ [14]$$

The updated model will look something like:

$$\mathbf{F2}\ (\boldsymbol{X_i}) = \mathbf{F1}\ (\boldsymbol{X_i}) + \boldsymbol{h_1}\ (\boldsymbol{X_i})\ [15]$$

After performing this procedure iteratively we finally can fit a model at the M^{th} step:

$$\mathbf{FM}\ (\boldsymbol{X_i}) = \boldsymbol{F_{M-1}}\ (\boldsymbol{X_i}) + \boldsymbol{h_{M-1}}\ (\boldsymbol{X_i})\ [16]$$

Gradient boosting allows one to plug in any class of weak learners hm (X_i) to improve predictive accuracy. In other words, the hm (X_i) is a weak learner that can take any functional form such as a GLM, a neural network or a decision tree. Although there is no requirement for $h_M(X_i)$ to be a specific function, it is usually a tree-based learner in practice.

3.2.6 Bayesian Dynamic Linear Model

Bayesian dynamic linear model is a promising method for time series data analysis and short-term forecasting. One research issue concerns how the predictive model adapts to changes in the system, especially when shocks impact system behavior. In this study, we propose an adaptive dynamic linear model to adaptively update model parameters for online system state prediction. The proposed method is an automatic approach based on the feedback of prediction errors at each time slot without the needs of external intervention. The experimental study on short-term travel speed prediction shows that the proposed method can significantly reduce the prediction errors of the traditional dynamic linear model and outperform two state-of-the-art methods in the case of major system behavior changes .

In a BDLM, observations y_t at a time t ∈ (1 : T) are modelled by superposing hidden states x_t from several generic components as defined by the observation model.

$$\boldsymbol{y_t} = \boldsymbol{C_t x_t} + \boldsymbol{v_t}, \quad \begin{cases} \boldsymbol{y_t} \sim \mathrm{N}\ (\mathrm{E}[\boldsymbol{y_t}], \boldsymbol{cov}[\boldsymbol{y_t}]) \\ \boldsymbol{x_t} \sim \mathrm{N}\ (\boldsymbol{v_t}, \boldsymbol{\Sigma_t}) \\ \boldsymbol{v_t} \sim \mathrm{N}\ (0, \mathbf{R}_{\boldsymbol{t}}) \end{cases}$$

Where,

- $\boldsymbol{y_t}$ represents the observed variable at time t
- **C** is a matrix that relates the state variable to the observed variable
- $\boldsymbol{x_t}$ represents the state variable at time t
- $\boldsymbol{v_t}$ is the process noise, assumed to be normally distributed with mean zero and covariance matrix R_t
- **E[$\boldsymbol{y_t}$]** is the expected value of y_t
- **cov[$\boldsymbol{y_t}$]** is the covariance matrix of y_t
- $\boldsymbol{X_t}$ is assumed to follow a multivariate normal distribution with mean vector µt and covariance matrix Σ_t

- $\boldsymbol{y_t}$ represents the observed variable at time t
- **C** is a matrix that relates the state variable to the observed variable
- $\boldsymbol{x_t}$ represents the state variable at time t
- $\boldsymbol{v_t}$ is the process noise, assumed to be normally distributed with mean zero and covariance matrix R_t
- **E[$\boldsymbol{y_t}$]** is the expected value of y_t
- **cov[$\boldsymbol{y_t}$]** is the covariance matrix of y_t
- $\boldsymbol{X_t}$ is assumed to follow a multivariate normal distribution with mean vector µt and covariance matrix Σ_t

Where the observation matrix **Ct** indicates how each component from the hidden state vector xt contribute to observations **yt** . The dynamic evolution of the hidden states xt is described by the transition model.

$$\boldsymbol{x_t = A_t x_t - 1 + w_t,\ w_t, \{\, \text{N}\,(0, Q_t)\}}$$

3.2.7 Recurrent Neural Network Model

A recurrent neural network (RNN) is a special type of artificial neural network adapted to work for time series data or data that involves sequences [17]. They're a class of neural networks that allow previous outputs to be used as inputs while having hidden states.

The RNN models have a recurrent hidden state as in:

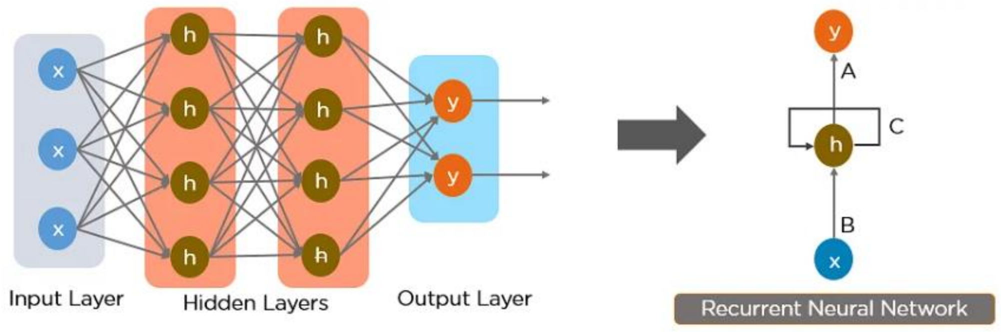

Figure 3. Structure of Simple Recurrent Neural Network

Here, x is the input layer, h is the hidden layer, and y is the output layer. A, B, and C are the network parameters used to improve the output of the model. At any given time t, the current input is a combination of input at x(t) and x(t-1).The output at any given time is fetched back to the network to improve on the output [18].

$$\mathbf{h(t) = f_c(h(t-1), x(t))}$$

Where:

- **h(t) –** new state
- **fc** – funcion with parameter c
- **h(t-1)** – old state
- **x(t)** – input vector at time step t

The input layer x takes in the input to the neural network and processes it and passes it onto the middle layer.

The middle layer h can consist of multiple hidden layers, each with its own activation functions and weights and biases.

The Recurrent Neural Network will standardize the different activation functions and weights and biases so that each hidden layer has the same parameters. Then, instead of creating multiple hidden layers, it will create one and loop over it as many times as required[18].

We can use any activation function we like in the recurrent neural network that's usually sigmoid, tanh or relu [17].

Sigmoid function: $\frac{1}{1+e^{-x}}$

Tanh function: $\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

Relu function: max(0,z)

3.2.8 Long short-term Memory Model

S. Hochreiter et al. [19] came up with the idea by introducing a novel, efficient, gradient based method called long short-term memory (LSTM) to addresses the issue of vanishing gradients in traditional Recurrent Neural Networks.

The LSTM network architecture consists of three parts, these three parts of an LSTM unit are known as gates. They control the flow of information in and out of the memory cell or lstm cell. The first gate is called Forget gate, the second gate is known as the Input gate, and the last one is the Output gate. An LSTM unit that consists of these three gates and a memory cell or lstm cell can be considered as a layer of neurons in traditional feedforward neural network, with each neuron having a hidden layer and a current state [20].

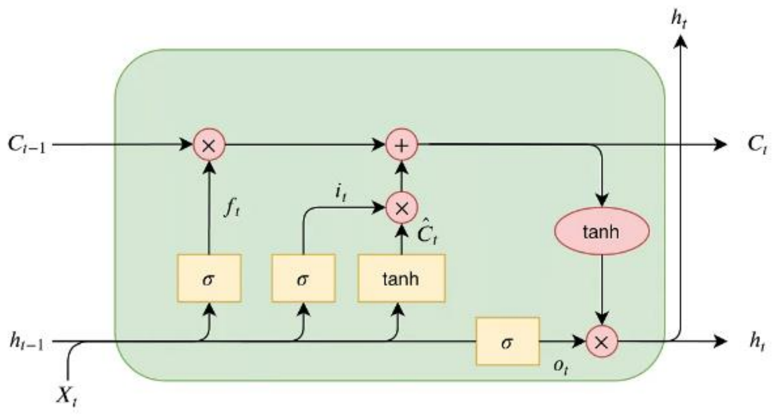

Figure 4. Structure of LSTM

Forget Gate: the first stage in architecture is Forget Gate. In this stage, the LSTM neural network will determine which elements of the cell state (long-term memory) are relevant based on the previous hidden state and the new input data.

Calculate the forget gate according to the following formula:

$$\mathbf{f_t = \sigma(W_f[h_{t-1}, X_t] + b_f)}$$

These output values are then multiplied element-wise with the previous cell state (C_{t-1}).

***Input Gate**:* the following stage involves the input gate and the new memory network. The objective of this step is to identify what new information should be incorporated into the network's long-term memory (cell state), based on the previous hidden state and the current input data.

$$\mathbf{i_t = \sigma(W_i[h_{t-1}, X_t] + b_i)}$$

The new memory network is a neural network that uses the tanh activation function and has been trained to create a "new memory update vector" by combining the previous hidden state and the current input data. Calculate the new memory network according to the following formula:

$$\mathbf{\widehat{C}_t = tanh(W_c[h_{t-1}, X_t] + b_c)}$$

The updated cell state is then passed through a tanh activation to limit its values to [-1,1] before being multiplied pointwise by the output of the output gate network to generate the final new hidden state.

The updated cell state represents the updated long-term memory of the network. The internal state is updated with this rule:

$$\mathbf{C_t = i_t\widehat{C}_t + f_t C_{t-1}}$$

***Output Gate**:* In the final stage of an LSTM, the new hidden state is determined using the newly updated cell state, previous hidden state, and new input data. The output gate performs this decision-making process.

$$\mathbf{o_t = \sigma(W_o[h_{t-1}, X_t] + b_o)}$$

The updated cell state is then passed through a tanh activation to limit its values to [-1,1] before being multiplied pointwise by the output of the output gate network to generate the final new hidden state.

$$\mathbf{h_t = O_t tanh(C_t)}$$

These output values are then multiplied element-wise with the previous cell state (C_{t-1}). The LSTM cell uses weight matrices and biases in combination with gradient-based optimization to learn its parameters. These parameters are connected to each gate, as in any other neural network. The weight matrices can be identified as W_f, b_f, W_i, b_i, W_o, b_o, and W_C, b_C respectively in the equations above.

3.2.9 Gated Recurrent Unit Model

Get the idea from LSTM architecture, Gated Recurrent Unit inherited some its necessary features. The main difference between LSTM and GRU is that GRU has two gating called the reset gate and the update gate and use gating mechanisms to selectively update the hidden state of the network at each time step. By J. Chung et al. [21], the result showed that GRU RNN are much more advance than LSTM in most cases.

The reset gate determines how much of the previous hidden state should be forgotten, while the update gate determines how much of the new input should be used to update the hidden state. The output of the GRU is calculated based on the updated hidden state.

The equations used to calculate the reset gate, update gate, and hidden state of a GRU are as follows:

Reset Gate:

$$\mathbf{r_t = \sigma(W_r[h_{t-1}, X_t] + b_r)}$$

Update Gate:

$$\mathbf{z_t = \sigma(W_z[h_{t-1}, X_t] + b_z)}$$

Candidate hidden state:

$$\mathbf{h'_t = \tanh(W_h[r_t h_{t-1}, X_t] + b_h)}$$

Hidden state:

$$\mathbf{h_t = (1-z_t)h_{t-1} + z_t h'_t}$$

CHAPTER IV: EVALUATION METHODOLOGY

In the research, predictive models are evaluated according to three criteria: **MAE, MAPE and RMSE.**

In the following formulas, Xi is the predicted ith value, and the Yi element is the actual ith value. The regression method predicts the Xi element for the corresponding Yi element of the ground truth dataset [22].

4.1 Mean Absolute Error (MAE)

MAE (the mean absolute error) is a measure of a predictive model's performance or accuracy measured in by averaging the absolute value of the error between the actual value and the predicted value.

$$\boldsymbol{MAE = \frac{1}{m}\sum_{i=1}^{m}|X_i - Y_i|}$$

4.2 Root Mean Squared Error (RMSE)

MAPE (Mean Absolute Percent Error) is a measure of the performance or accuracy of a predictive model measured by taking the average percentage of the absolute value of error between the actual value and the predicted value.

$$\boldsymbol{MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{Y_i - X_i}{Y_i}\right|}$$

4.3 Mean Absolute Percent Error (MAPE)

RMSE (Root Mean Square Error) is a measure of the performance or accuracy of a predictive model measured by calculating the square root of the mean square error between the predicted values and the actual values.

$$\boldsymbol{RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2}}$$

CHAPTER V: RESULT

TABLE 2. EVALUATION METRIC

Stock	Model	Train – Test - Valid	MAE	MAPE	RMSE
VCB	Linear Regression	65-25-10	7359.93	9.76%	9411.65
		70-20-10	8108.67	10.78%	9996.63
		75-15-10	8684.14	11.56%	10463.08
	ARIMA	65-25-10	19796.94	25.97%	21454.81
		70-20-10	4804.57	5.94%	6324.09
		75-15-10	9313.59	12.40%	10714.68
	SARIMAX	65-25-10	32115.79	47.39%	33301.85
		70-20-10	32881.71	41.91%	33439.45
		75-15-10	32513.08	40.79%	34035.11
BID	RNN	65-25-10	772.57	2.08%	1046.07
		70-20-10	862.56	2.42%	1145.01
		75-15-10	844.29	2.41%	1144.18
	LSTM	65-25-10	914.41	2.45%	1202.94
		70-20-10	785.11	2.23%	1027.63
		75-15-10	757.23	2.22%	990.19
	GRU	65-25-10	742.97	2.02%	992.65
		70-20-10	733.30	2.06%	984.91
		75-15-10	786.80	2.24%	1022.70
STB	KNN	65-25-10	26958.76	32.47%	27302.85
		70-20-10	26843.30	99.99%	5425.17
		75-15-10	23973.36	22.95%	24222.60
	GBT	65-25-10	26921.94	37.40%	27279.79
		70-20-10	25728.82	22.40%	27278.79
		75-15-10	23274.72	38.30%	0.2995
	BDLM	65-25-10	0.1481	21.09%	0.1789
		70-20-10	0.2602	38.60%	0.3004
		75-15-10	0.259.4	38.30%	0.2995

From Table II is the result after training data from stock prices of three different banks in VietNam, each bank will be trained and predicted from 3 different models with rates 65/25/10, 70/20/10 and 75/15/10 to determine a good stock price prediction model for each bank, that bank and evaluated by the MAE, MAPE and RMSE measures. Here I will take the MAPE measure as the main metric for comparison and prediction the next 30 days.

Through the results, we see that for VCB's stock price when it is predicted through 3 models Linear Regression, ARIMA, SARIMAX, in general, the Linear Regression model gives a fairly high and uniform prediction rate at all 3 ratios. The ratio in which the best accuracy is the ratio 65/25/10 with the measures determined as follows MAE = 7359.93, MAPE = 9.76% and RMSE = 9411.65.

Next is ARIMA with the best accuracy determined at the rate of 70/20/10 with the measurements determined respectively as follows: MAE = 4804.57, MAPE = 5.94% and RMSE = 6324.09. And SARIMAX gave the worst results both above 40% on MAPE worst at 65/25/10 with MAPE = 47.39%.

Similar to BID's stock price when predicted with LSTM, RNN and GRU models, the accuracy rate is quite high in all 3 models. In which the best is GRU at 65/25/10 with measures of MAPE = 2.02%, with best RNN at rate 65/25/10 with MAPE = 2.08%, with best LSTM at ratio 75/15/10 with MAPE = 2.22%.

As for the share price of STB bank, when predicting with 3 models KNN, GBT and BDLM, all of them gave not very good results, the best is BDLM at the rate of 65/25/10 with MAPE = 21.09%, followed by BDLM at the rate of 65/25/10 with MAPE = 21.09%. GBT at rate 70/20/10 with MAPE = 22.4%, worst is KNN at rate 70/20/10 MAPE = 99.99%.

I will show with the best MAPE ratio for each model and will take out the best 3 models in each category and predict the price for the next 30 days.

- Linear Regression ratio 65/15/10

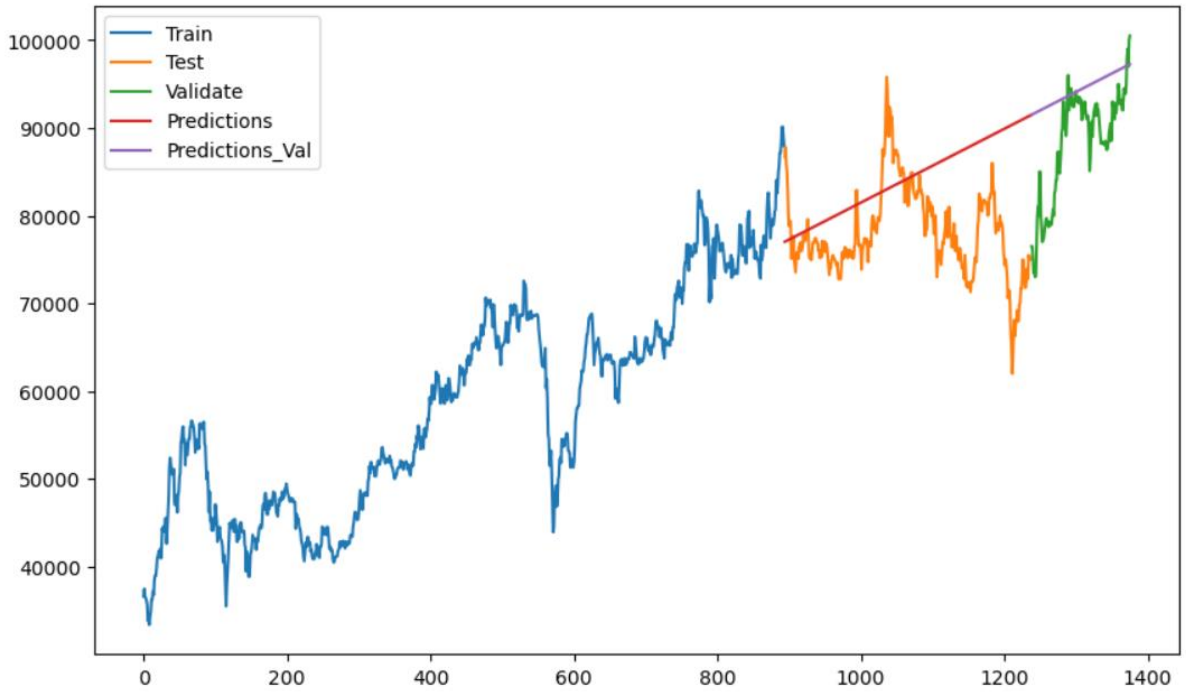

Figure 6. Result of Linear Regression model

- ARIMA ratio 70/20/10

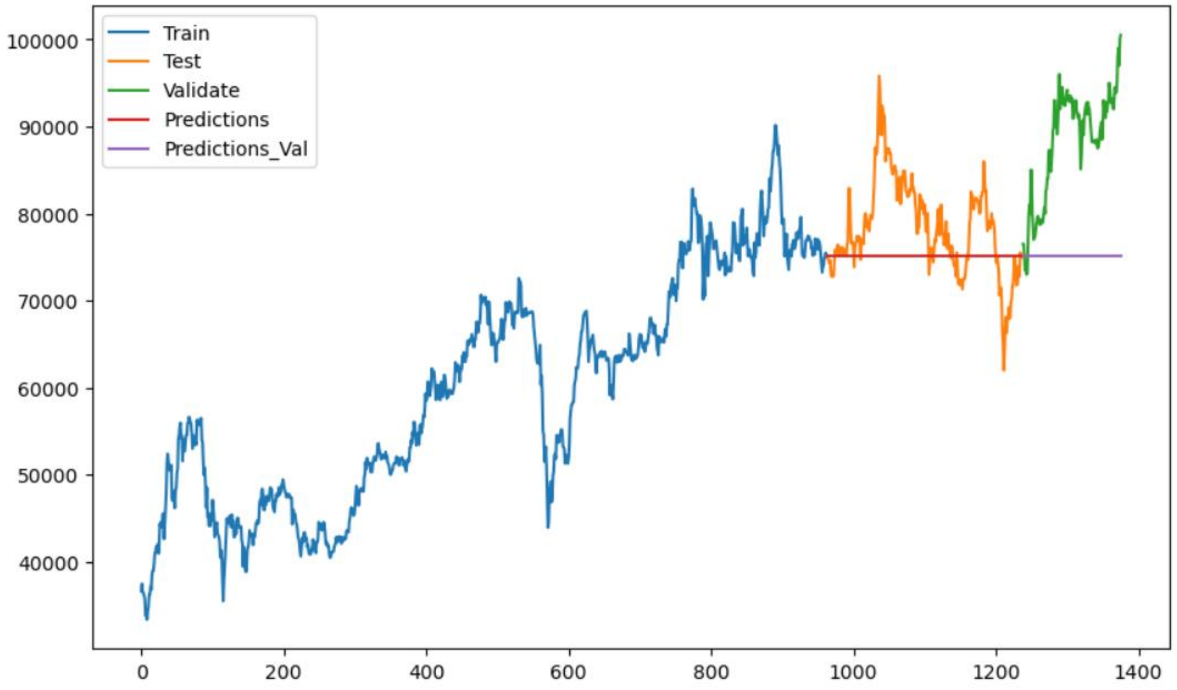

Figure 7. Result of ARIMA model

- SARIMAX ratio 75/15/10

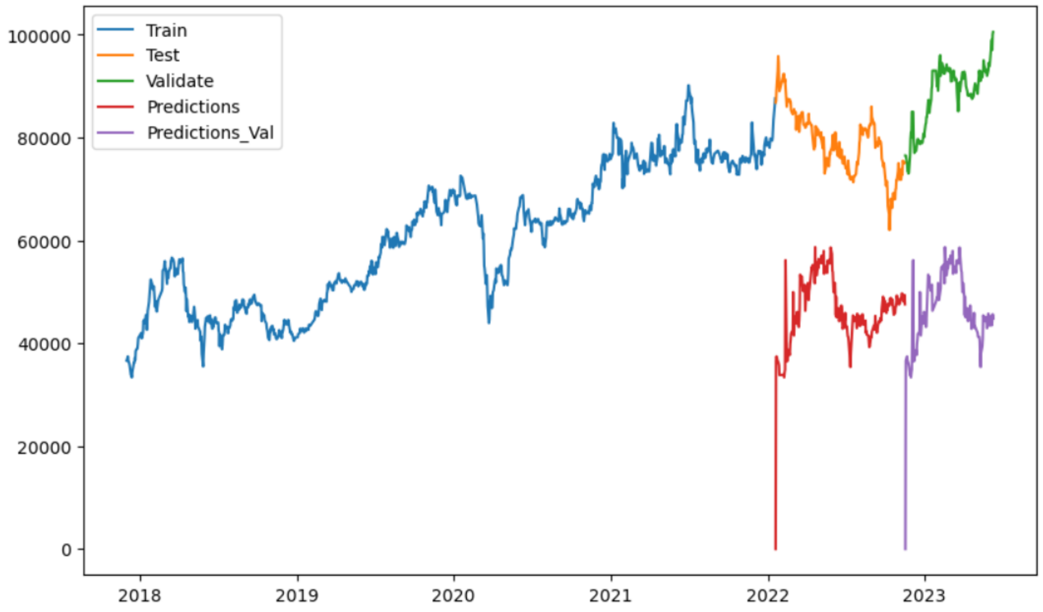

Figure 8. Result of SARIMAX model

- RNN ratio 65/25/10

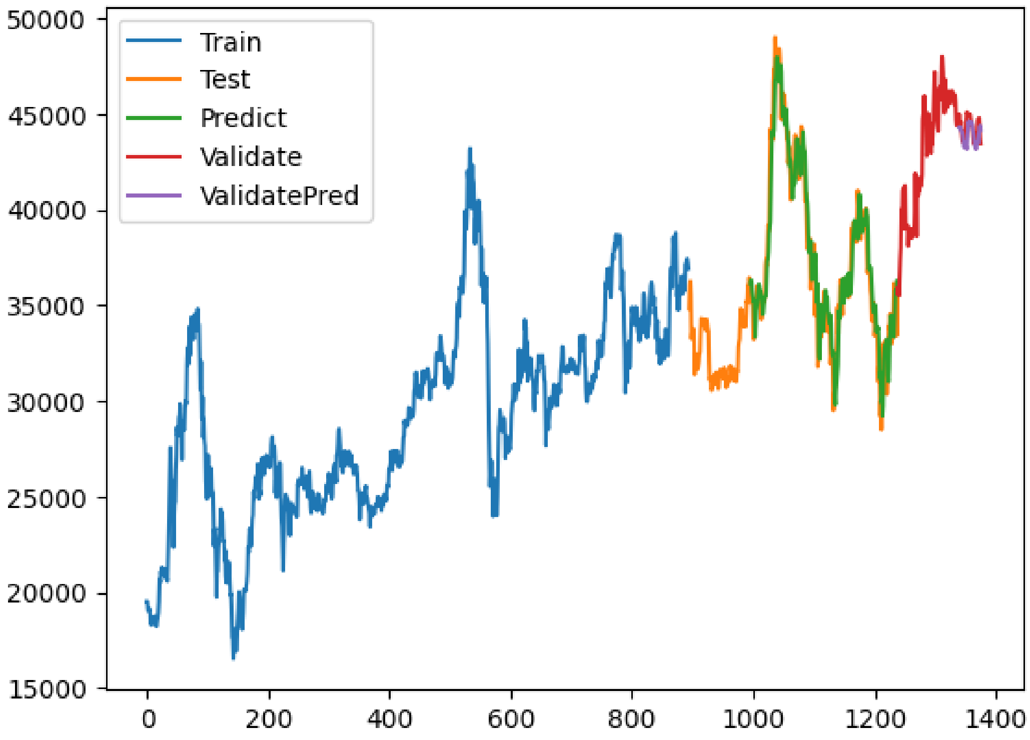

Figure 9. Result of RNN model

- LSTM ratio 75/15/10

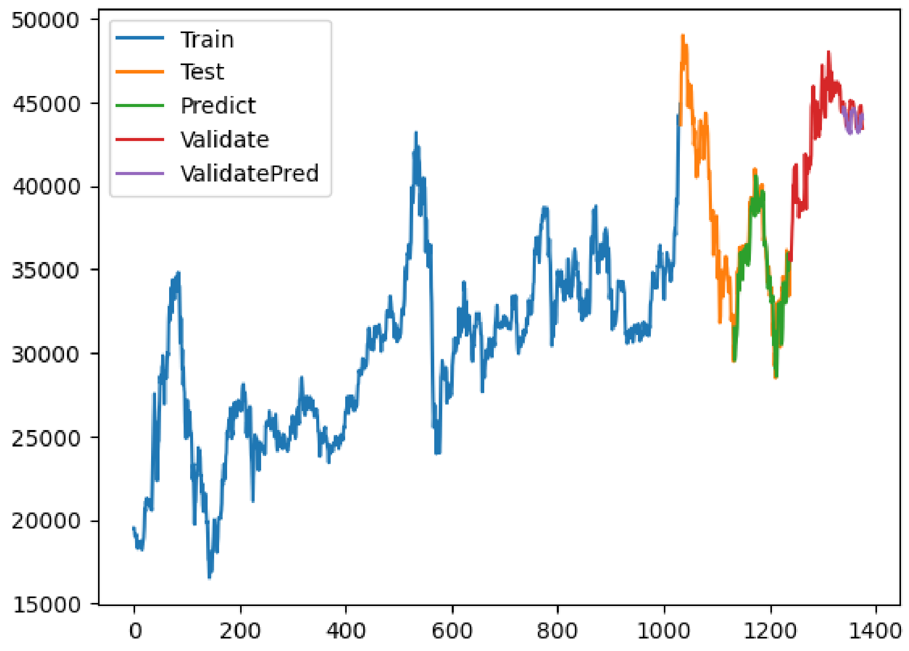

Figure 10. Result of LSTM model

- GRU ratio 65/25/10

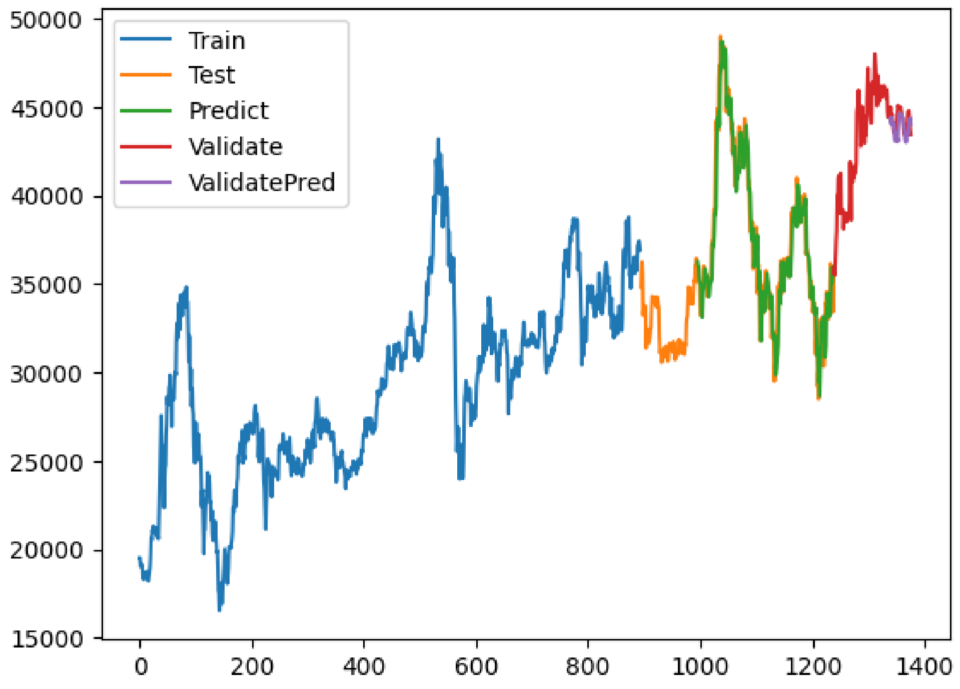

Figure 11. Result of GRU Model

- KNN ratio 75/15/10

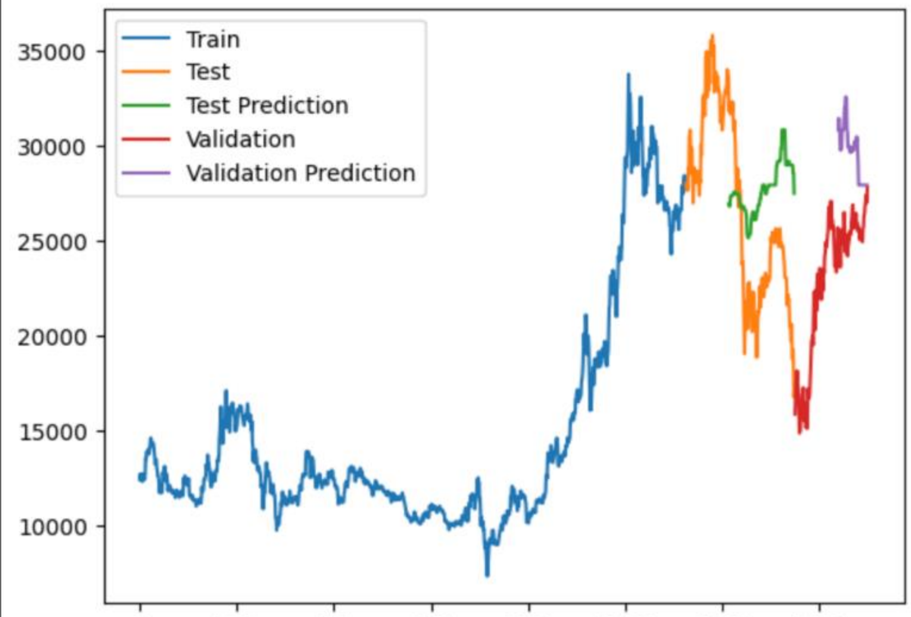

Figure 12. Result of KNN model

- GBT ratio 70/20/10

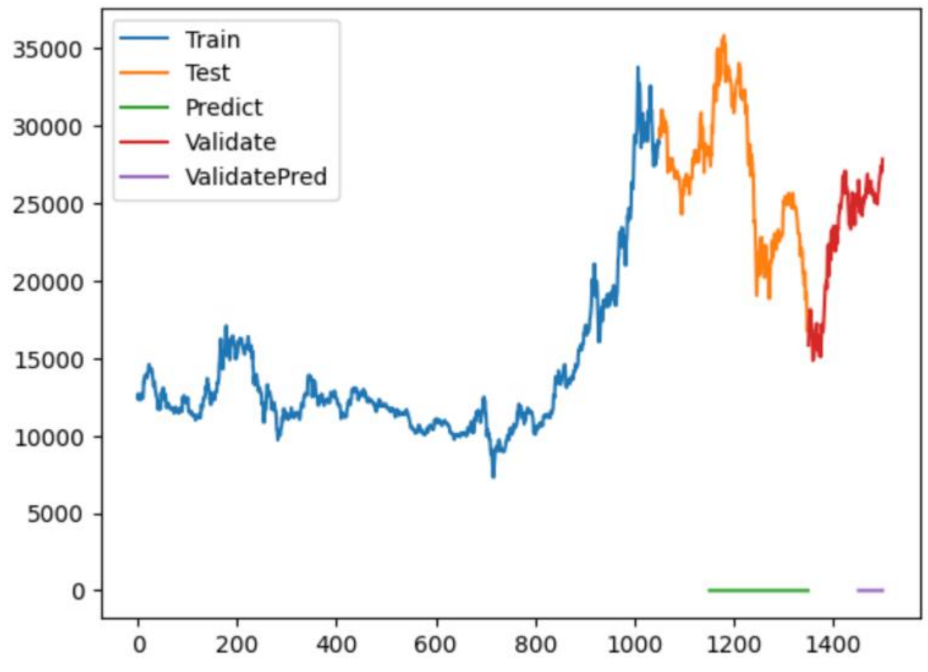

Figure 13. Result of GBT model

- BDLM ratio 65/25/10

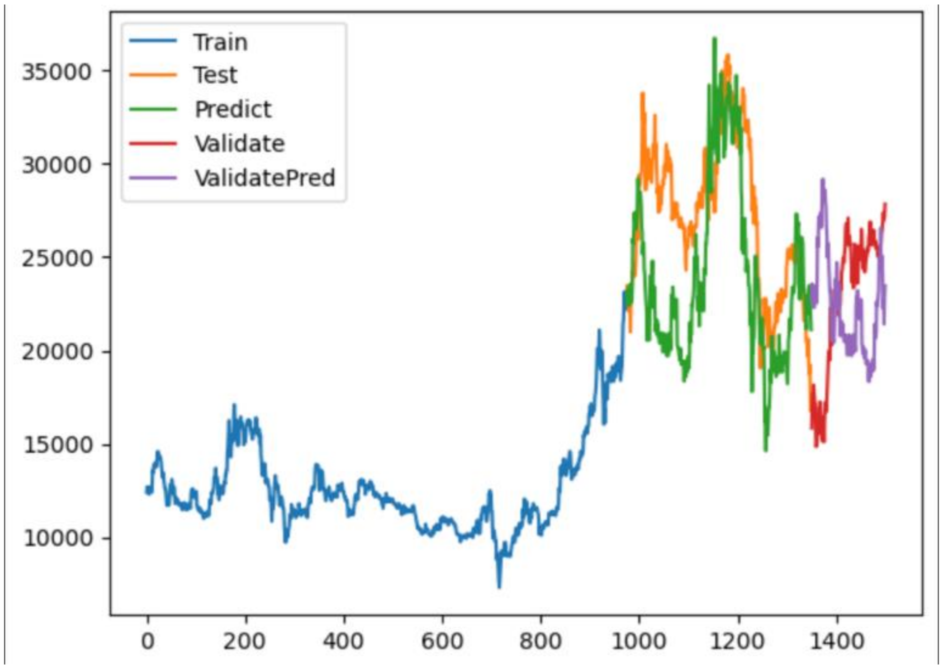

Figure 14. Result of BDLM model

Predict the next 30 days

For VCB stock price, use ARIMA to predict BID will take GRU and STB will take BDLM to predict for the next 30 days.

- ARIMA

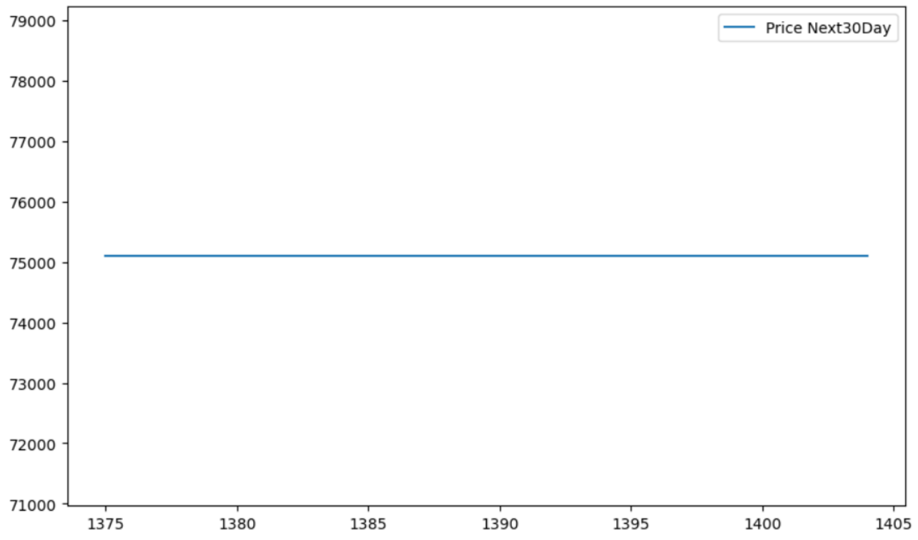

Figure 15. VCB stock price prediction for the next 30 days with ARIMA

- GRU

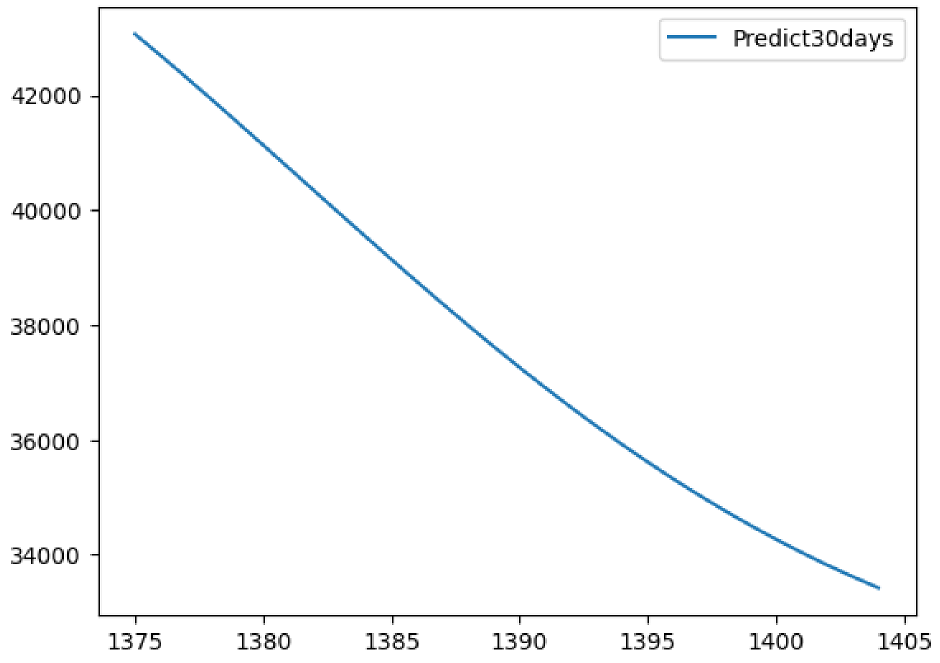

Figure 16. BID stock price prediction for the next 30 days with GRU

- BDLM

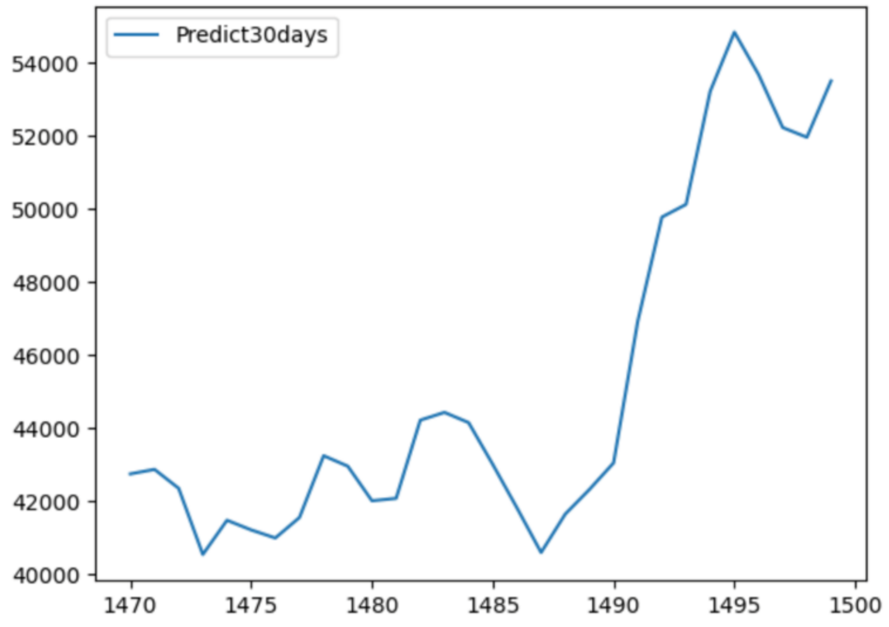

Figure 17. STB stock price prediction for the next 30 days with BDLM

CHAPTER VI: CONCLUSION

Through the above results, we also achieved some high results such as the GRU model giving the best prediction accuracy, in addition, LSTM and GRU also gave high results. Besides, there are also some models with low results such as KNN, GBT or BDLM.

In the future, my team will find a way to help the prediction rate be higher than develop a forecasting model with a combine models like LSTM, RNN and Linear Regression to predict bank stock prices and create bots that automatically trade when they go up or down and calculate which model will bring the most profit.

TASK ASSIGNMENT TABLE

Members / **Task**	**Bui Duc Duy**	**Nguyen Thi Cam Van**	**Ho Bao An**
Introduction	✓	✓	
Related work	✓	✓	✓
Dataset		✓	✓
Visualizing data	✓	✓	✓
Data Analysis	✓	✓	✓
Data preprocessing	✓		
Linear Regression Model		✓	
ARIMA Model		✓	
SARIMAX Model		✓	
RNN Model	✓		
LSTM Model	✓		
GRU Model	✓		
KNN Model			✓
GBT Model			✓
BDLM Model			✓
Evaluation Methodology	✓	✓	✓
Result	✓		
Conclusion		✓	
Writing report	✓	✓	✓

REFERENCES

[1] Y. Si, “Using ARIMA model to analyse and predict bitcoin price,” *BCP Bus. Manag.*, vol. 34, pp. 1210–1216, Dec. 2022, doi: 10.54691/bcpbm.v34i.3161.

[2] M. Ali and S. Shatabda, “A Data Selection Methodology to Train Linear Regression Model to Predict Bitcoin Price,” in *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, Nov. 2020, pp. 330–335. doi: 10.1109/ICAICT51780.2020.9333525.

[3] N. H. Gawali, “Bitcoin Price Prediction using Neural Network”.

[4] M. Karim, M. Foysal, and S. Das, “Stock Price Prediction Using Bi-LSTM and GRU-Based Hybrid Deep Learning Approach,” Nov. 2022, pp. 701–711. doi: 10.1007/978-981-19-3148-2_60.

[5] M. Bansal, A. Goyal, and A. Choudhary, “Stock Market Prediction with High Accuracy using Machine Learning Techniques,” *Procedia Comput. Sci.*, vol. 215, pp. 247–265, Jan. 2022, doi: 10.1016/j.procs.2022.12.028.

[6] G. Chenghan and W. Tao, “Improvement of Bayesian Dynamic Linear Model for Predicting Missing Data of Bridges,” *E3S Web Conf.*, vol. 185, p. 02027, Jan. 2020, doi: 10.1051/e3sconf/202018502027.

[7] “Stock Quotes & Prices,” *Investing.com*. https://www.investing.com/equities/ (accessed Jun. 15, 2023).

[8] A. Greaves and B. Au, “Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin”.

[9] “Introduction to ARIMA models.” https://people.duke.edu/~rnau/411arim.htm (accessed Jun. 16, 2023).

[10] F. Alharbi and D. Csala, “A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach,” *Inventions*, vol. 7, p. 94, Oct. 2022, doi: 10.3390/inventions7040094.

[11] S. Zhang, “Nearest neighbor selection for iteratively kNN imputation,” *J. Syst. Softw.*, vol. 85, no. 11, pp. 2541–2552, Nov. 2012, doi: 10.1016/j.jss.2012.05.073.

[12] “Springer Series in Statistics,” *Springer*. https://www.springer.com/series/692 (accessed Jun. 18, 2023).

[13] “Greedy Function Approximation: A Gradient Boosting Machine on JSTOR.” https://www.jstor.org/stable/2699986?fbclid=IwAR1l9S680LYNNBKcQamT-ZKfCTwvFBtjzaRV4Z-Fa0CfG3h1YLNeNTnXnJ8 (accessed Jun. 18, 2023).

[14] J. Zhao *et al.*, “Learning from Longitudinal Data in Electronic Health Record and Genetic Data to Improve Cardiovascular Event Prediction,” *Sci. Rep.*, vol. 9, no. 1, p. 717, Jan. 2019, doi: 10.1038/s41598-018-36745-x.

[15] R. J. Delahanty, J. Alvarez, L. M. Flynn, R. L. Sherwin, and S. S. Jones, “Development and Evaluation of a Machine Learning Model for the Early Identification of Patients at Risk for Sepsis,” *Ann. Emerg. Med.*, vol. 73, no. 4, pp. 334–344, Apr. 2019, doi: 10.1016/j.annemergmed.2018.11.036.

[16] W. A, Y. At, L. As, G. R, D. Vc, and H. D, "Development and Validation of an Electronic Health Record-Based Machine Learning Model to Estimate Delirium Risk in Newly Hospitalized Patients Without Known Cognitive Impairment," *JAMA Netw. Open*, vol. 1, no. 4, Aug. 2018, doi: 10.1001/jamanetworkopen.2018.1018.

[17] M. Saeed, "An Introduction to Recurrent Neural Networks and the Math That Powers Them," *MachineLearningMastery.com*, Sep. 08, 2022. https://machinelearningmastery.com/an-introduction-to-recurrent-neural-networks-and-the-math-that-powers-them/ (accessed Jun. 18, 2023).

[18] "Recurrent Neural Network (RNN) Tutorial: Types and Examples [Updated] | Simplilearn," *Simplilearn.com*. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn (accessed Jun. 18, 2023).

[19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[20] S. Saxena, "Learn About Long Short-Term Memory (LSTM) Algorithms," *Analytics Vidhya*, Mar. 16, 2021. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/ (accessed Jun. 15, 2023).

[21] 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. 16, 2023. [Online]. Available: http://arxiv.org/abs/1412.3555

[22] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.