STAT3013.N12.CTTT: TEAM 3 FINAL

PROJECT

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Task** |
| Lê Trí Khoa - Leader | 20521466 | -Directing Project  -Making Report file and Paper  -Code: Linear Regression, Nonlinear Regression, LSTM |
| Nguyễn Hữu Thiên | 20521951 | -Code: Prophet  -Make Presentation file |
| Lâm Lê Phúc Huy | 20521388 | Code: ARIMA |

[I. Project overview 3](#_Toc123745266)

[II. Related Works 3](#_Toc123745267)

[III. Background 5](#_Toc123745268)

[A. Regression 5](#_Toc123745269)

[1. Linear Regression: 5](#_Toc123745270)

[2. Non Linear Regression: 5](#_Toc123745271)

[B. ARIMA 6](#_Toc123745272)

[C. LSTM 8](#_Toc123745273)

[1. Artificial Neural Network (ANN): 8](#_Toc123745274)

[2. Recurrent Neural Network (RNN): 9](#_Toc123745275)

[3. Long Short-Term Memory (LSTM): 9](#_Toc123745276)

[D. Prophet 10](#_Toc123745277)

[IV. DATA PREPARATION AND TOOLS 11](#_Toc123745278)

[A. Data 11](#_Toc123745279)

[B. Data preparation 12](#_Toc123745280)

[C. Tools selection 12](#_Toc123745281)

[D. Assessment Metric: 13](#_Toc123745282)

[V. Implementation 14](#_Toc123745283)

[A. Linear Regression 14](#_Toc123745284)

[B. Nonlinear Regression 18](#_Toc123745285)

[C. ARIMA 20](#_Toc123745286)

[D. LSTM 26](#_Toc123745287)

[E. Prophet 38](#_Toc123745288)

[VI. Result 43](#_Toc123745289)

[A. Preformance Review 43](#_Toc123745290)

[B. 20% Test Forecasting 44](#_Toc123745291)

[C. 10% Test Forecasting 45](#_Toc123745292)

[VII. Conclusion 46](#_Toc123745293)

[VIII. Practical use of LSTM 47](#_Toc123745294)

[IX. References 47](#_Toc123745295)

# Project overview

Project: “Stock Price Prediction Using Statistical Methods”

Link to Project: <https://drive.google.com/drive/folders/1z9Ums3HTcPMJX1AEwMnq10Jic4VcrSg7?usp=sharing>

Objectives: We studied 5 different forecasting methods. Start with 2 basic model that we have learned from the course: Linear Regression and Nonlinear Regression. The last 3 are advanced models: ARIMA, LSTM, Prophet.

The goal is to compare the performance of Linear Regression, Nonlinear

Regression, ARIME, LSTM and Prophet with respect to

minimization achieved in the error rates in prediction. After that, we do a practical study to predict 30 days of stock price that is not in original data set with the best model.

# Related Works

This chapter will discuss the previous work done on stock price forecasting using regression machine learning and deep learning domain.

Dinesh Bhuriya et al [1] used linear regression, polynomial and RBF regression to predict the stock prices using 5 variables and compared the above models and concluded that linear regression is best among all other used.

ARIMA model was first developed by George Box and Gwilym in their textbook Time Series Analysis: Forecasting and Control to engage in Time Series Analysis (TSA) in 1970. They proposed to use ARMA when the sequence is stationary and use ARIMA when the sequence is non-stationary [2].

Many investors then apply the ARIMA model when analyzing their trading strategy and some researchers have found that ARIMA is effective in forecasting stock prices. Nau believes that the ARIMA model is a relatively sophisticated and accurate algorithm for time series forecasting [3].

Zumbo et al. researched that ARIMA is a good method for nonstationary time series prediction that is composed of an autoregressive and a moving average model and was successfully utilized for time series prediction in different areas which includes financial markets [4].

Bollerslev developed the GARCH model which is based on the ARCH model constructed by Robert Engle in 1982 [5]. Shengtantu applied the GARCH model in financial markets and found that it not only accurately depicts and describes whether the impact of positive and negative financial market shocks on stock prices is asymmetric, but also has obvious significance and role in the study of the symmetrical relationship between expected operating returns and expected economic risks of the financial market [6].

Assous et al. researched that the GARCH model can effectively solve the volatility problem of time series since it can accurately describe the basic characteristics of the ”thick tail” of stock prices in financial markets [7].

However, some researchers argued that the accuracy of classic TSF models is not satisfactory. Yang and Wang questioned the accuracy of the ARIMA and GARCH models because financial data have high noise and dynamic characteristics. The flexible relationship between the dependent variable and independent variable limits the further application and expansion of traditional TSF. They proposed that LSTM is a better forecasting method for stock prices [8].

By comparing prediction results of stock prices under the ANN model and the ARIMA model, Milad and Seyed researched that the ANN model has higher accuracy than ARIMA Model [9].

Sima et al. compare the ARIMA model with the LSTM model, arguing that the LSTM model can achieve higher accuracy than the ARIMA model [10].

Lu et al. applied different models to forecast stock prices and the result shows that the CNN-LSTM model has the highest accuracy for the next day stock price [11].

Kumar Jha and S. PandeAs [12] has examined few forecasting models such as- The additive model, the Autoregressive integrated moving average (ARIMA) model, FB Prophet model. From the propsoed research work, it is concluded that, FB Prophet is a better prediction model.

W. -X. Fang, P. -C. Lan [13], in their research LSTM and Prophet are used to predict the trend of time series data, and the prediction trend is combined with the inverse neural network model (BPNN) for prediction. The empirical results show that this method can indeed achieve accurate forecasting trends and reduce errors.

As the reviewed papers above, it can be inferred that studies on forecasting stock prices are still being raised among researchers and it seems that newly proposed models are more advanced than the classic one. However, there still exists debate between different models.

# Background

## Regression

### Linear Regression:

Linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables. In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data which are called linear models. Linear regression model follows a very particular form, a regression model is linear when all terms in the model have a constant or a parameter multiplied by an independent variable. And buy adding the terms together, the equation is formed:

Where Y is a dependent variable, X are independent variables, βi is the parameter and ϵ are errors.

### Non Linear Regression:

Nonlinear regression is a form of regression analysis in which observational data are modeled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables . In statistic, nonlinear regression can be demonstrated by the equation:

Y = F(X, β) + ϵ

Where X is a vector of P predictors, β is a vector of k parameters, F is the known regression function. Systematic error may be present in the independent variables, but its treatment is outside the scope of regression analysis. In general, there is no closed-form expression for the bestfitting parameters, as there is in linear regression. In contrast, there may be many local minima of the function to be optimized and even the global minimum may produce a biased estimate.

In practice, estimated values of the parameters are used, in conjunction with the optimization algorithm, to attempt to find the global minimum of a sum of squares.

## ARIMA

Autoregressive Integrated Moving Average Model (ARIMA) 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. The reason for stationarity is that ARIMA can only be applied to stock price prediction if the time series is not white noise and not seasonal . As acronym indicates, ARIMA (p, d, q) captures the key elements of the model:

* AR: Autoregression - A regression model that uses the dependencies between an observation and the 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:

Where xt is the stationary variable, c is constant, the terms in θi are autocorrelation coefficients at lags 1, 2, p and t, the residuals, are the Gaussian white noise series with mean zero and variance σ 2 . An MA model of order q, i.e., MA(q), can be written in the form:

Where xt is the stationary variable, c is constant, the terms in θi are autocorrelation coefficients at lags 1, 2, p and t, the residuals, are the Gaussian white noise series with mean zero and variance σ 2 . An MA model of order q, i.e., MA(q), can be written in the form:

Where µ is the expectation of xt (usually assumed equal to zero), the θi terms are the weights applied to the current and prior values of a stochastic term in the time series, and θ0 = 1. We assume that t is a Gaussian white noise series with mean zero and variance σ 2 . We can combine these two models by adding them together and form an ARIMA model of order (p, q):

Where φi ̸= 0, θi ̸= 0, and σ 2 > 0. The parameters p and q are called the AR and MA orders. ARIMA forecasting, also known as Box and Jenkins forecasting, can deal with nonstationary 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).

With seasonal time series data, it is likely that short runnonseasonal components contribute to the model. Therefore, we need to estimate seasonal ARIMA model, which incorporates both non-seasonal and seasonal factors in a multiplicative model. The general form of a seasonal ARIMA model is denoted as ARIMA (p, d, q) × (P, D, Q)S, where p is the non-seasonal AR order, d is the non-seasonal differencing, q is the non-seasonal MA order, P is the seasonal AR order, D is the seasonal differencing, Q is the seasonal MA order, and S is the time span of repeating seasonal pattern.

The most important step in estimating seasonal ARIMA model is to identify the values of (p, d, q) and (P, D, Q). Based on the time plot of the data, if for instance, the variance grows with time, we should use variance-stabilizing transformations and differencing. Then, using autocorrelation function (ACF) to measure the amount of linear dependence between observations in a time series that are separated by a lag p, and the partial autocorrelation function (PACF) to determine how many autoregressive terms q are necessary and inverse autocorrelation function (IACF) for detecting over differencing, we can identify the preliminary values of autoregressive order p, the order of differencing d, the moving average order q and their corresponding seasonal parameters P, D and Q. The parameter d is the order of difference frequency changing from non-stationary time series to stationary time series.

## LSTM

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) with the capability of remembering the values from earlier stages for the future usage. Before getting into LSTM, it is essential to have a look of what a neural network looks like.

### Artificial Neural Network (ANN):

A neural network consists of at least three layers: an input layer (1), hidden layers (2), and an output layer (3). The number of features of the data set determines the dimensionality or the number of nodes in the input layer. These nodes are connected through links called “synapses” to the nodes created in the hidden layer(s). The synapses links carry some weights for every node in the input layer. The weights play the role of a decision maker to decide which signal, or input, can pass through and which cannot. The weights also show the strength or extent to the hidden layer. A neural network learns by adjusting the weight for each synopsis.

In the hidden layers, the nodes apply an activation function on the weighted sum of inputs to transform the inputs to the outputs or predicted values. The output layer generates a vector of probabilities for the various outputs and selects the one with minimum error rate or cost, which minimizes the differences between expected and predicted values, which also known as the cost, using a function called SoftMax.

The assignments to the weights vector and thus the errors obtained through the network training for the first time might not be the best. In order to find the most optimal values for errors, the errors are “back propagated” into the network from the output layer towards the hidden layers and as a result the weights are adjusted. The procedure is repeated several times with the same observations and the weights are re-adjusted until there is an improvement in the predicted values and subsequently in the cost. When the cost function is minimized, the model is trained.

### Recurrent Neural Network (RNN):

A Recurrent Neural Network (RNN) is a special case of neural network where the objective is to predict the next step in the sequence of observations with respect to the previous steps observed in the sequence. In fact, the idea behind RNNs is to make use of sequential observations and learn from the earlier stages to forecast future trends. As a result, the earlier stages data need to be remembered when guessing the next steps. In RNNs, the hidden layers act as internal storage for storing the information captured in earlier stages of reading sequential data. RNNs are called recurrent because they perform the same task for every element of the sequence, with the characteristic of utilizing information captured earlier to predict future unseen sequential data. The major challenge with a typical generic RNN is that these networks remember only a few earlier steps in the sequence and thus are not suitable to remembering longer sequences of data. This challenging problem is solved using the memory line introduced in the Long Short-Term Memory (LSTM) recurrent network.

### Long Short-Term Memory (LSTM):

LSTM is a special type 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.

LSTM is a special type of RNNs with additional features to memorize the sequence of data. Each LSTM is a set of cells, or system modules, where the data streams are captured and stored. The cells resemble a transport line (the upper line in each cell) that connects out of one module to another one conveying data from past and gathering them for the present one. Due to the use of some gates in each cell, data in each cell can be disposed, filtered or added for the next cells. Thus, the gates which are based on sigmoidal neural network layer enable the cells to optionally let data pass through or disposed.

Each sigmoid layer yields numbers in the range of zero and one, depicting the amount of every segment of data ought to be let through in each cell. More precisely, an estimation of zero value means that let nothing pass through, whereas an estimation of one indicates that let everything pass through. There are three types of gates 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 indicates completely keep this, whereas 0 indicates completely ignore this.

- Memory Gate: chooses which new data need to be stored in the cell. Firstly, a sigmoid layer, called the input door layer chooses which values will be modified. Next, a tan 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.

## Prophet

Facebook Prophet is a model and a library that provides features both from generalized linear models (GLM) and additive models (AM), mainly extending GLM by using nonlinear smoothing functions. It was specified by Taylor and Letham [18] in 2017.

The main difference between Prophet and other statistical methods is the analyst-in-the-loop approach. This approach allows the the analyst to apply their domain knowledge about the data to the forecasting algorithm, without having any knowledge of the statistical methods working from within. This approach, therefore, tries to take advantage from both the statistical forecasting and the judgmental forecasting, the latter being the forecasting methods based on human experts decisions.

The general function to define the time series is the following:

y(t) = g(t) + s(t) + h(t) + ϵt

Where g(t) represents the non-periodic changes in the value of the time series, s(t) model seasonality (which can be daily, weekly, monthly, yearly or any other), h(t) represents the effects of holidays and ϵt is the error term.

# DATA PREPARATION AND TOOLS

## Data

The authors extracted daily VinaCapital Vietnam Opportunity Fund Limited (VOF.L) index for the period between 12/28/2017 - 12/23/2022 from the Yahoo finance Website [14].

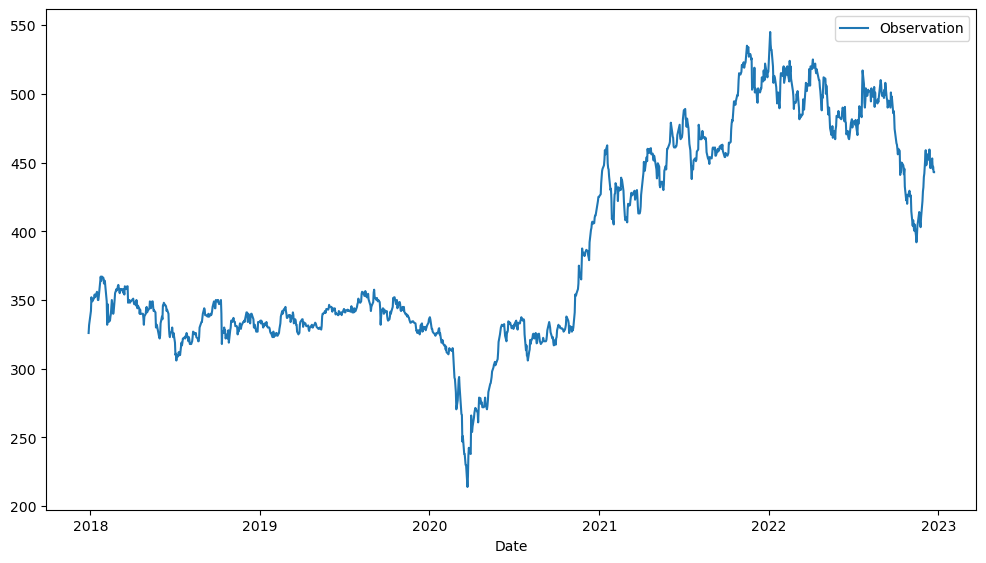


Figure Visualization for Close stock price of VOF.L

The data set consists of 1262 observations. Fig 1. shows the visualization of the data for the period while Fig 2. shows the description of the data.

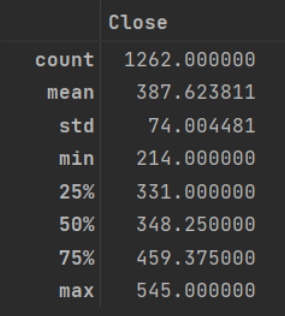


Figure Data Description

## Data preparation

Table

Description automatically generatedEach financial time series data set features a number of variables: Open, High, Low, Close, Adjusted Close and Volume. The authors chose the “Close” variable as the only feature of financial time series to be fed into each models. The vinacapital data set was split into two subsets: training and test datasets where we divide 4 different ratio: 90/10, 80/20, 70/30, and 60/40 . the table below lists the number of time series observations for each ratio.

## Tools selection

The code for the project was written in Python [15] For the implementation of ARIMA, the library pmdarima.arima [16] were used to automatically choose the best hyperparameter for the model without bias. In addition, statsmodels.tsa.arima [17] were used. For LSTM, we used Keras [18]. For the Facebook Prophet model, the implementation in Python was used [19]. Some other libraries were used, the most important of them being: sklearn (used for preprocessing tasks), matplotlib (for plotting), numpy (for array handling) and pandas (for reading and writing the datasets).

## Assessment Metric:

Graphical user interface

Description automatically generated with low confidenceThis research used 3 different performance metric for the models: Root-Mean-Square Error (RMSE), Mean absolute error (MAE), Mean absolute percentage error (MAPE). The Root-Mean-Square Error (RMSE) is a measure frequently used for assessing the accuracy of prediction obtained by a model. It measures the differences or residuals between actual and predicated values. The metric compares prediction errors of different models for a particular data and not between datasets. The formula for computing RMSE is as follows:

Where N is the total number of observations, xi is the actual value; whereas, xˆi is the predicated value. The main benefit of using RMSE is that it penalizes large errors. It also scales the scores in the same units as the forecast values (i.e., per month for this study).

Text

Description automatically generatedMean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. MAE is calculated as the sum of absolute errors divided by the sample size:

A picture containing text, watch, clock

Description automatically generatedThe mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics. It usually expresses the accuracy as a ratio defined by the formula:

Where At is the actual value and Ft is the forecast value. Their difference is divided by the actual value At. The absolute value of this ratio is summed for every forecasted point in time and divided by the number of fitted points n.

# Implementation

## Linear Regression

1. Read the data file, we just need to use The column Close:

Graphical user interface, text

Description automatically generated

2. Split the data into train and test set

Text

Description automatically generated

3. Reshape the x feature to fit into the model. Here we use index as the only x

Text

Description automatically generated

4. Compile and fit model LinearRegression():

Text

Description automatically generated

5. Graph the train set:

Chart

Description automatically generated

6. Graph the test set:

Chart

Description automatically generated

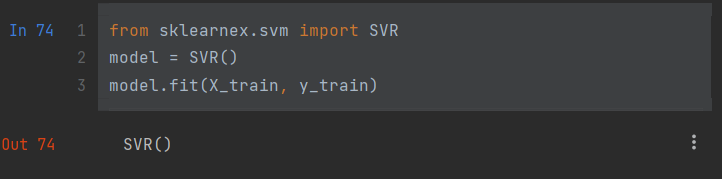
7. Benchmarking:

Text

Description automatically generated

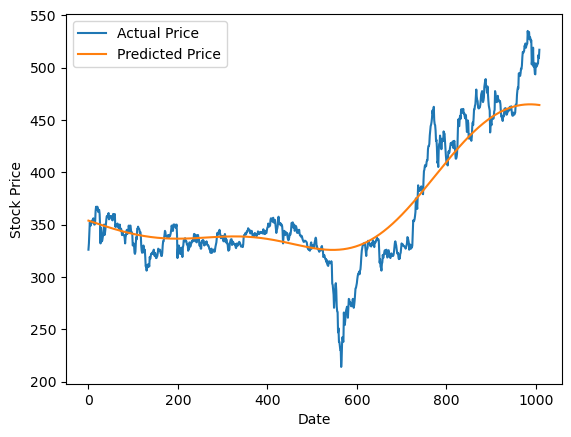
## Nonlinear Regression

For Nonlinear regression, we do the same steps as Linea regression up until we fit the model.

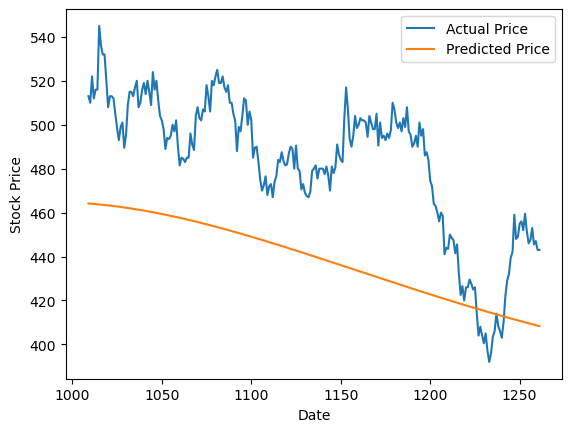


For Nonlinear regression, We use SVR – Support Vector Regression to make a nonlinear model.

We have the train graph:



And the test graph:



Benchmarking:

Text

Description automatically generated

## ARIMA

1. First read the data, we used the Date and Close column



2. Check for the stationary of the data. Data is not stationary

Text

Description automatically generated

3. Divide train set and test set

Text

Description automatically generated

4. Use auto\_arima function to find the best model without bias.

Text

Description automatically generated

Result model ARIMA(0,1,0).

Text

Description automatically generated

5. Fitting the model

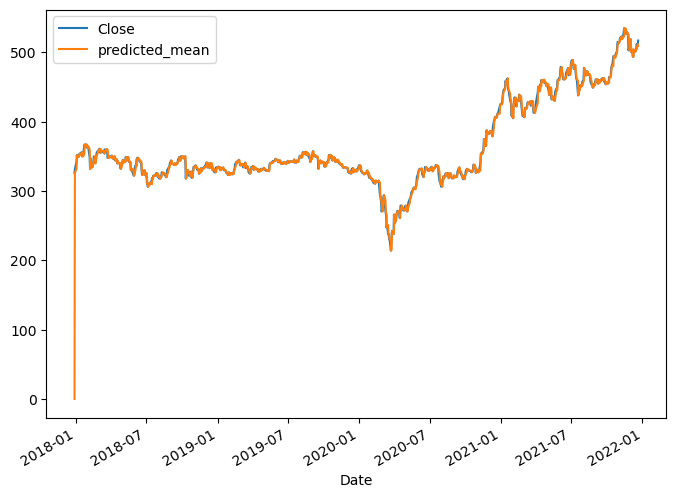
Text

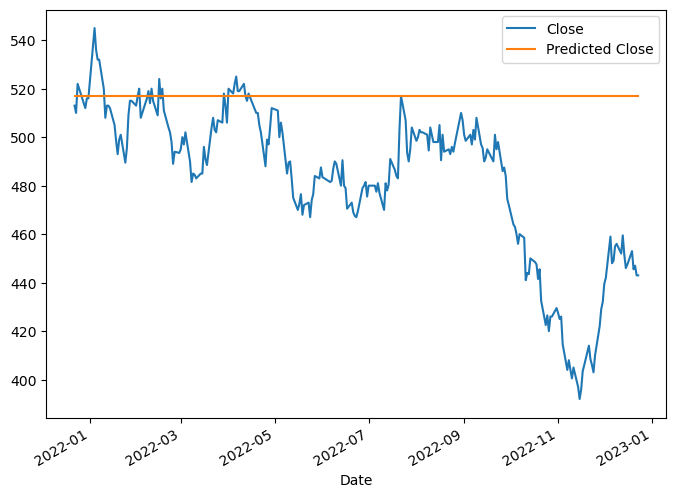
Description automatically generated

6. Create train and test prediction data frame:

Text

Description automatically generated

7. Train graph

8. Test graph

9. Benchmarking

Text

Description automatically generated

## LSTM

1. Import LSTM library

Text

Description automatically generated with medium confidence

2. Read the data

A picture containing text

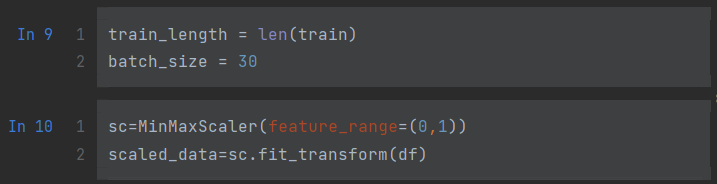
Description automatically generated

3. Split the data

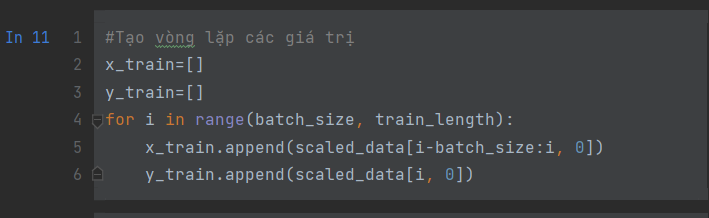
Graphical user interface, text

Description automatically generated

4. Define trai\_length, batch\_size and a scaller from 0 to 1 to normalize data.



5. Reshape the data to fit into the model:



Text

Description automatically generated

6. Complie and fit model

Text

Description automatically generated

Model information:

Table

Description automatically generated

7. Graph of loss fuction:

Chart

Description automatically generated with medium confidence

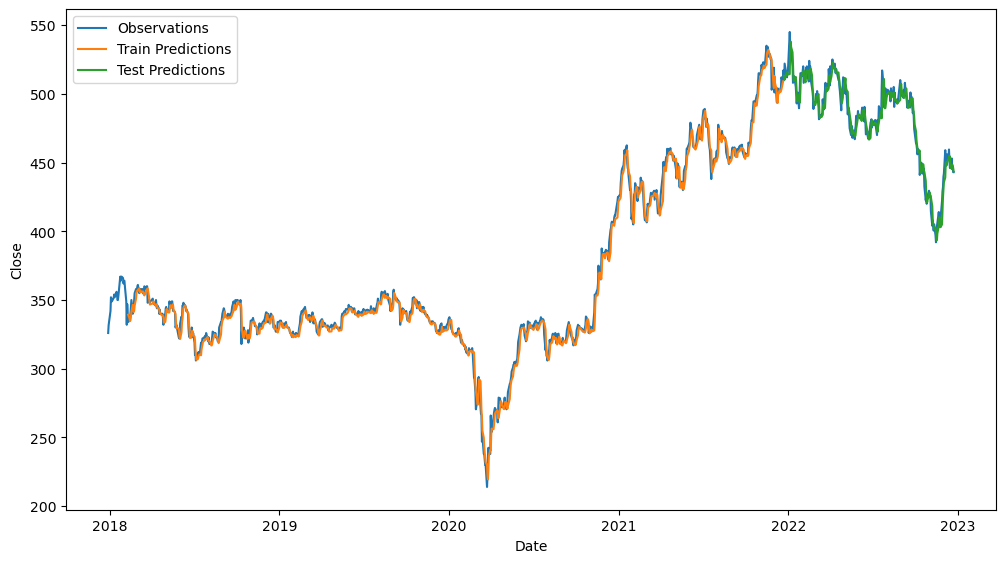
8. Getting predict from the model:

Text

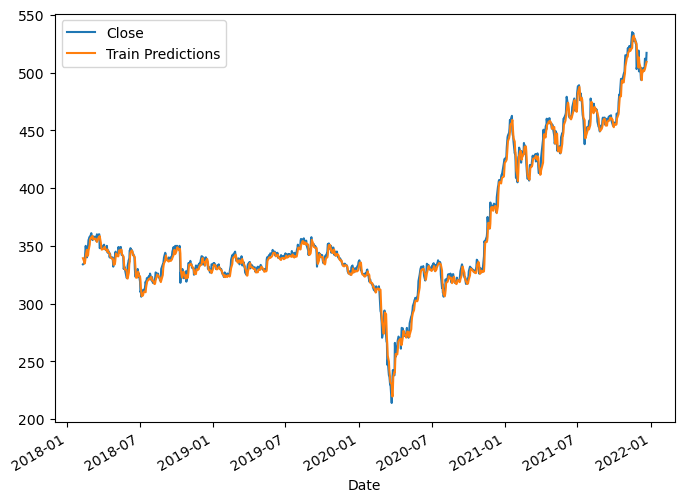
Description automatically generated

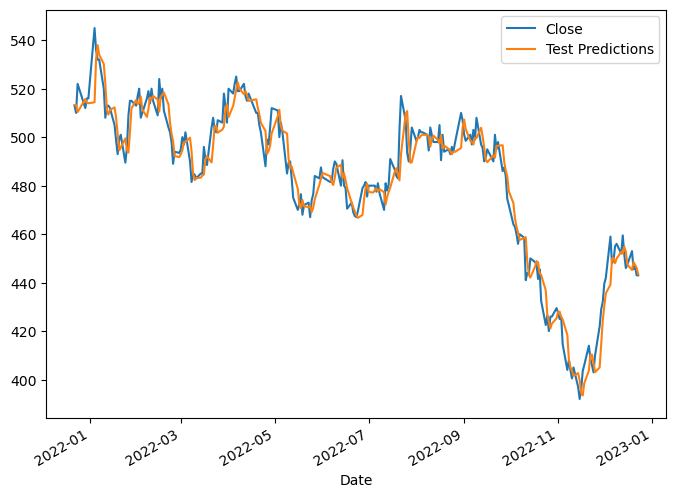
Text

Description automatically generated

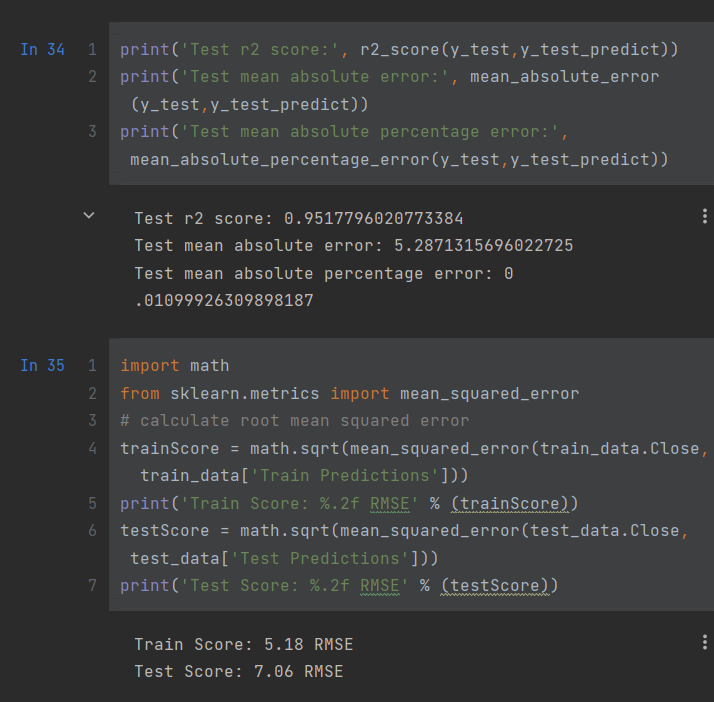
9. Plot data graph

10. Train graph



11. Test graph

12. Benchmarking:



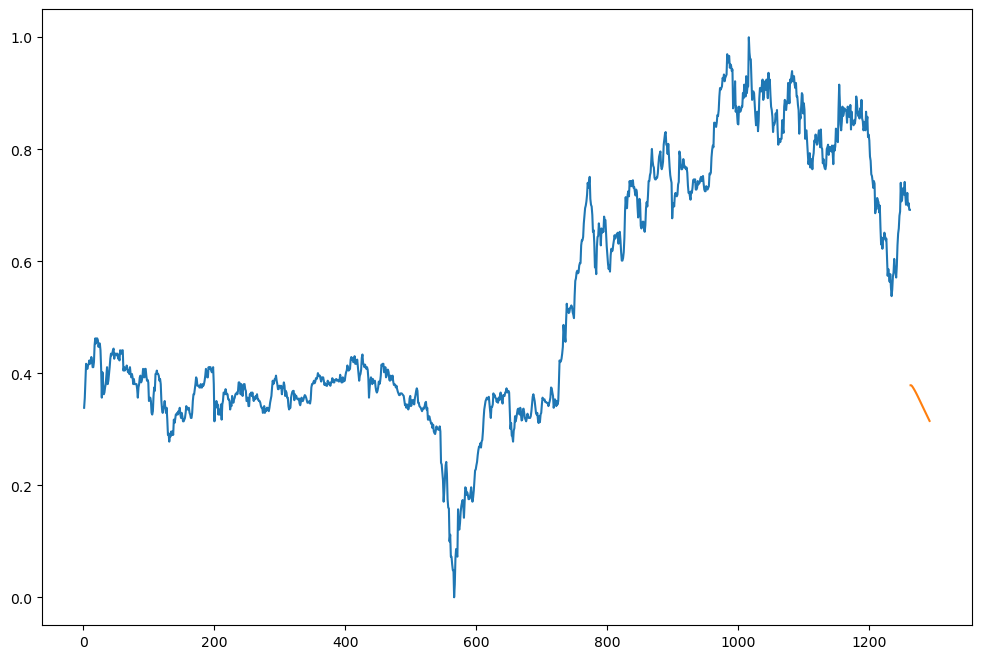
**Practice: predict 30 days using LSTM 80/20 trained model**

Define a function to get fit into LSTM model recursively

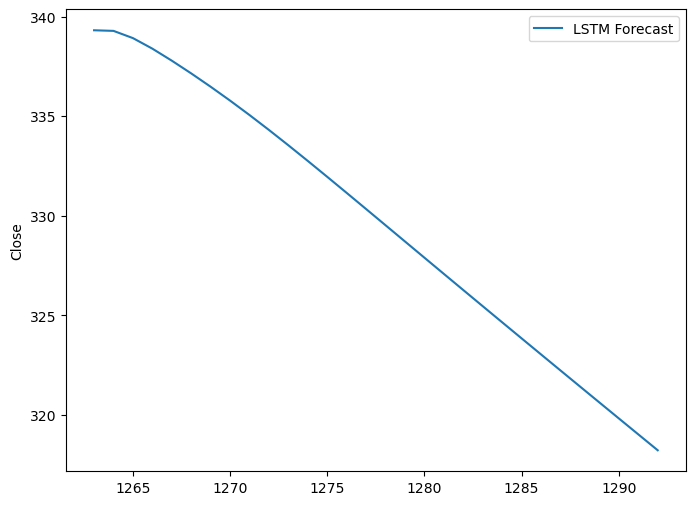
Text

Description automatically generated

Result:

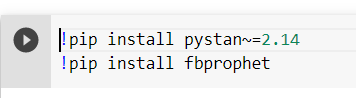


Forecasting graph:



## Prophet

1. Install fb Prophet and import library



Graphical user interface, text, application

Description automatically generated

2. Read data:

A picture containing shape

Description automatically generated

3. Rename dataframe

Graphical user interface, text, application, chat or text message

Description automatically generated

4. Split data set

Chart, scatter chart

Description automatically generated

5. Automatic Hyperparameter tuning

Text

Description automatically generated

Tunning result:



6. Fit model with the tunned parameter

Timeline

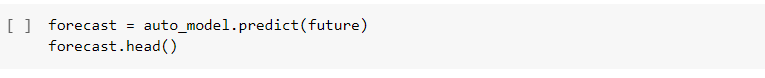
Description automatically generated with medium confidence

7. Create forecast dataframe

Graphical user interface, text, application, Teams

Description automatically generated

8. forecast



9. plot the forecast

Chart, scatter chart

Description automatically generated

10. model components

Chart, line chart

Description automatically generated

11. plot test predict

Chart, line chart

Description automatically generated

12. Benchmarking

Graphical user interface, text, application

Description automatically generated

# Result

## Preformance Review

The Accuracy of 5 models and their respective train/test ratio is presented in below Table. The best performance for each model is highlighted with red color

Table

Description automatically generated

Figure Result table

After analyzing, it was found that most models (Linear Regression, Nonlinear Regression and Prophet) have given the best performance with the more data fitted in. While the ARIMA model work best with 70% trainset and 20% testset. Most notably, the result clearly indicate that the LSTM model outperform other models with a high margin, especially with the train/test ratio of 80/20 with RME value of 7.06

## 20% Test Forecasting

From the result of performance review, it was found that that most of the models have given the best performance for 20% and 10% test dataset. Hence for the next step which is forecasting, therefore, performed forecasting analysis on 20% and 10% test dataset.

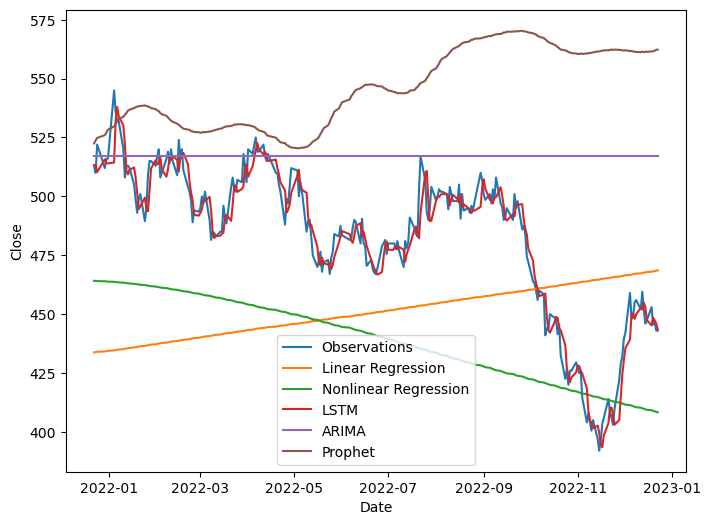
Fig 4. was shown forecasting for 20% test dataset. LSTM, Prophet and non linear regression had shown some trend and patterns for forecasting whereas ARIMA show straight line. The linear regression show wrong trend.

Figure 20% Test graph

## 10% Test Forecasting

Fig 5. was shown forecasting for 10% test dataset. LSTM and Prophet had shown some trend and patterns for forecasting whereas ARIMA, Linear regression and Nonlinear regression model had show straight line.

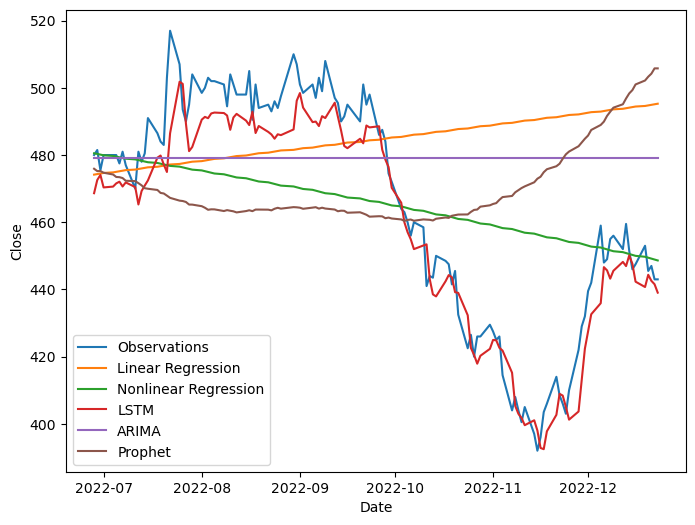


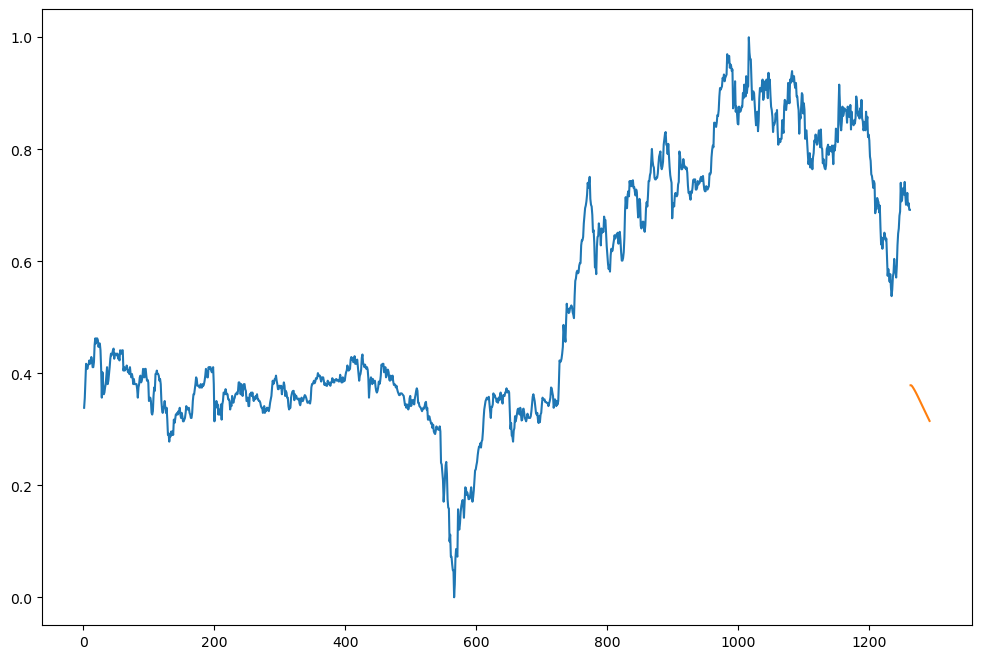
Figure 10% test graph

# Conclusion

With the recent advancement on developing sophisticated machine learning-based techniques and in particular deep learning algorithms, these techniques are gaining popularity among researchers across divers disciplines. The major question is then how accurate and powerful these newly introduced approaches are when compared with traditional methods. This paper compares the accuracy of five different models, as representative techniques when forecasting time series data. The models were implemented and applied on a set of stock data on several train/test split ratio and the results indicated that LSTM with 80% train data set with was superior.

# Practical use of LSTM

We used the 80/20 trained LSTM model to predict the next 30 days that is not included in the data.



From from 24/12/2022 to 4/1/2023, there are only 5 entries for the VOF.L on the yahoo finance website:

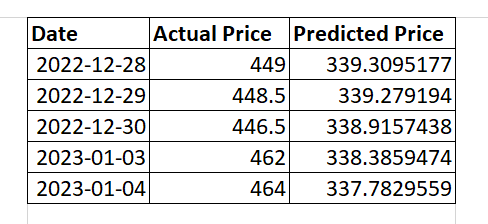


Figure LSTM 80/20 Practical result

From the result the actual price are not as expected as the model forecast. This might be due to our unabillity to correct the code. In the furure, we intend to fixing this problem and make correct forecast of LSTM model.

# References

[1] Dinesh Bhuriya, Ashish Sharma, Upendra Singh. “Stock Market Prediction using Linear Regression,” International Conference on Electronics, Communication and Aerospace Technology ICECA 2017. DOI: 10.1109/ICECA.2017.821271 6, <https://ieeexplore.ieee.org/document/8212716/>

[2] G. E. P. Box, G. M. Jenkins, Time series analysis: forecasting and control. Journal of Time, vol. 31, no. 3, 2010, pp. 190-200.

[3] R. Nau, Mathematical structure of ARIMA models, 2014, pp.1–8. [8] B. D. Zumbo, E. Kroc, Heteroskedasticity in Multiple Regression Analysis: What it is, How to Detect it and How to Solve it with Applications in R and SPSS [Heteroscedasticidad en analisis de regresi ´ on m ´ ultiple: ´ que es, c ´ omo detectarlo y c ´ omo resolverlo con aplicaciones en R y], ´ The Journal of Modern Applied Statistical Methods, vol. 17, 2019.

[4] T. Bollerslev, Generalized Autoregressive Conditional Heteroskedasticity, Journal of Econometrics, 1986, pp. 307–328.

[5] S. T. T. Wang, Research on the Volatility of BYD Stocks Price Based on GARCH Family Model, 1994, pp. 280–284.

[6] H. F. Assous, N. Al-Rousan, D. Al-Najjar, H. AlNajjar, Can international market indices estimate TASI’s movements? The ARIMA model. Journal of Open Innovation: Technology, Market, and Complexity, 2020, pp. 1–17. DOI: <https://doi.org/10.3390/joitmc6020027>

[7] Q. Yang, C. Wang, A study on forecast of global stock indices based on deep LSTM neural network, Statistical Research, 2019, vol. 36, no. 6, pp. 65–77.

[8] S. F. Milad, S. H. R. Hajiagha, Forecasting Stock Price Using Integrated Artificial Neural Network and Metaheuristic Algorithms Compared to Time Series Models, Soft Computing, vol. 25, no. 13, 2021, pp. 483–513

[9] S. Siami-Namini, N. Tavakoli, A.S. Namin, A Comparison of ARIMA and LSTM in Forecasting Time Series, 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 1394–1401.

[10] W.J. Lu, J. Z. Li, Y. F. Li, A. J. Sun, J. Y. Wang, A CNN-LSTM-Based Model to Forecast Stock Prices, Introduction, 2020, pp. 11.

[11] B. Kumar Jha and S. Pande, ”Time Series Forecasting Model for Supermarket Sales using FB-Prophet,” 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 547-554, doi: 10.1109/ICCMC51019.2021.9418033. [17] W. -X. Fang, P. -C. Lan, W. -R. Lin, H. -C. Chang, H. -Y. Chang and Y. -H. Wang, ”Combine Facebook Prophet and LSTM with BPNN Forecasting financial markets: the Morgan Taiwan Index,” 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2019, pp. 1-2, doi: 10.1109/ISPACS48206.2019.8986377.

[12] Taylor SJ, Letham B. Forecasting at scale. PeerJ Preprints. (2017). 5:e3190v2 <https://doi.org/10.7287/peerj.preprints.3190v2>

[13] Yahoo Finance website: https://finance.yahoo.com/ (December 2022) [20] Python official site. https://www.python.org/ (December, 2022)

[14] pmdarima official site. http://alkaline-ml.com/pmdarima/ (December, 2022)

[15] statsmodel official site. https://www.statsmodels.org/dev/about.html (December, 2022) [23] Keras official site. https://keras.io/ (December, 2022)

[16] https://facebook.github.io/prophet/ (December, 2022)