VN-INDEX Prediction by using Statistical Methods and Machine Learning Algorithms

Name: Nguyễn Phi Hùng
University of Information Technology
ID student: 20521425
Email: 20521425@gm.uit.edu.vn

***Abstract*—The stock market index is usually affected by many factors. Because of the random nature of stock market, the stock index is volatile and non-linear which challenging task for predicting. This article represents the application of LinearRegression, Non- LinearRegression, XGBoots, LSTM, and Random Forest Regression on forecasting VN-Index, the index representing all stock listed and traded on HOSE (Ho Chi Minh Stock Exchange).**

Keywords— Random Forest, LSTM, LinearRegression, Non-LinearRegression, XGBoots, MAE, MSE, MAPE, R2

I. INTRODUCTION

Ho Chi Minh Stock Exchange or HoSE was established in 2000, is a securities exchange licensed to operate by the Commission. State Securities Commission. It is the one of most reputable, long-standing, and influential stock exchanges on the Vietnamese stock market. With the inaugural of HoSE, is the release of the VN-Index, one of the most common Stock Index of Vietnam. Although there are other Indexes on HoSE like VN100, VN30, VNALL, etc., and several stock exchanges with individual Index, VN-Index is still the index that accounts for the majority of Vietnam's stock market. It is not only just in the stock market but it is also considered as a measure for assessing economic growth, like when the more business activities of companies develop, the more the VN-Index will increase, and in some bad situations like recession, VN-Index also fluctuates and stays at a low level [1]. From the application of some Forecasting Models for the prediction of the VN-Index, the investors know about the general of the stock market, and investment psychology in the future so that they can decide when is the right time to buy and sells, furthermore we can know about the development of Viet Nam economic in upcoming, how the Viet Nam economic will be.

II. RELATED WORK

This chapter will discuss about the previous paper work that was researched on stock price forecasting using regression, machine learning, and deep learning domain.

In the last decade, machine learning methods have exhibited distinguished development in financial time series prediction. Dinesh Bhuriya et al [11] 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.

Tianqi Chen [12] who developed XGBoost released in 2014 but Tree boosting is a highly effective and widely used machine learning method. In this paper, we describe a scalable end to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges.[13]

It is an efficient and scalable implementation of gradient boosting framework by (Friedman, 2001) (Friedman et al., 2000). At the moment it's the de facto standard algorithm for getting accurate results from predictive modeling with machine learning. It's the fastest gradient-boosting library for R, Python, and C++ with very high accuracy.[14]

LSTM neural network was firstly proposed by Hochreiter and Schmidhuber in 1997.Learning to store information over extended time intervals by recurrent backpropagation takes a very long time, mostly because of insufficient, decaying error backflow. We briefly review Hochreiter's (1991) analysis of this problem, then address it by introducing a novel, efficient, gradient-based method called long short-term memory (LSTM) Moreover, Krauss et al. (2017) compare different deep learning methods such as deep neural networks, gradient-boostedtrees and random forests. In a single-feature setting, the daily returns with respect to the closing prices of the S&P 500 from December 1992 until October 2015 are provided to forecast one-day-ahead for every stock the probability of outperforming the market. As trading strategy, the 10 stocks with the highest probability are bought and the 10 stocks with the lowest probability are sold short – all with equal monetary weight. It turns out that random forests achieve the highest return of each of the above deep learning methods with returns of 0.43% per day, prior to transaction costs. Fischer & Krauss (2018) continue the study of Krauss et al. (2017) by employing LSTM networks as deep-learning methodology and obtain returns of 0.46% per day prior to transaction costs, therefore outperforming all the memory-free methods in Krauss et al. (2017).[15]

In theory, classic (or "vanilla") RNNs can keep track of arbitrary long-term dependencies in the input sequences. The

problem with vanilla RNNs is computational (or practical) in nature: when training a vanilla RNN using back-propagation, the long-term gradients which are back-propagated can "vanish" (that is, they can tend to zero) or "explode" (that is, they can tend to infinity)[8], which use finite-precision numbers. RNNs using LSTM units partially solve the vanishing gradient problem, because LSTM units allow gradients to also flow unchanged. However, LSTM networks can still suffer from the exploding gradient problem.[9]
In such a manner that, when applying, I have optimized models which can be effective with the data in this article.

III. DATA

A. *Data Collecting*

The historical data of VN-Index has been collected from investing.com [2]. The dataset includes 13 years from 02/01/2009 to 12/12/2022. The data contains information about the VN-Index such as High, Low, Open, Close, Vol, and % Change. But in this article, we just use the Close as an input for predicting, in other that the table has two rows which are Close and Date.

Information	Description
Name of data	VN-Index
Source	investing.com
Time Interval	02/01/2009 - 12/12/2022

B. *Data description*

The data of Close about VN-Index from the past, there are 3481 samples corresponding with 3481 days from 02/01/2009 - 12/12/2022. There are some Date have a NULL value which does not affect the forecast because that Date is a day off, and the stock market doesn't have any exchange. The figure below shows the value of the VN-Index in the Data

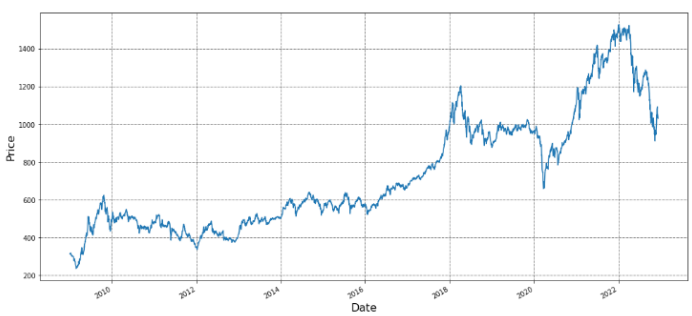


Figure 1: VN-Index stock data

The value is going up over time but in general, the data is highly randomly

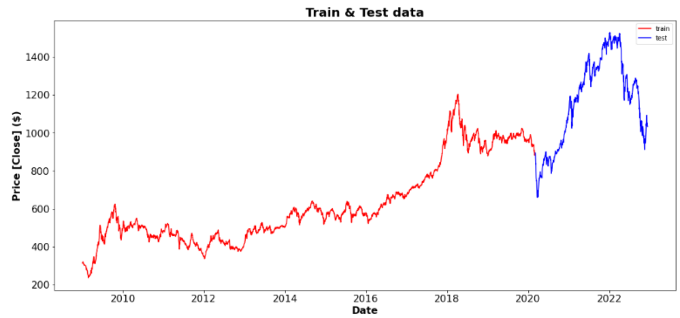
C. *Training and Test Data*

For three models which are Random Forest, LSTM, and XGBoots, the data set was divided into training and testing where 80% of each data set was used for training and 20% of each data set was used for testing the accuracy of models. This is the common ratio in every prediction because it has the best performance, especially when the size of the data is not very large [3].

Figure 2: The train and test data of three models

IV. METHODOLOGY

A. *Model*



1) *Linear and Nonlinear regression*

a) About Regression

Regression is a statistical method for predictive analysis, it is a basic and common algorithm in analysis. The main goal of Regression is to find the relationship between variables of a certain problem. One is regarded as an explanatory variable or independent and the other as a dependent variable. In next sections about Linear and Non-Linear will give a specific example of the explanatory variable, dependent variable, and the relationship between them.
Before building a linear model fit with the given data, the first is to determine whether or not there is a relationship between the variables of interest [3]. The strength of the relationship between two variables is shown by a scatterplot, showing the increase or decrease trends.

b) Linear Regression

Linear is mean straight, flat. When visualized, Linear is a function if it graph is in the form of a straight line. In statistics, Linear Regression determines the strength of the association between a dependent variable and one or multiple independent variables in a linear way. It can be created into models to predict something, and in this term, it is used for predicting VN-Index. Linear Regression is followed by the formula:

Simple Linear Regression :

$$Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \varepsilon$$

Multiple Linear Regression :

$$Y = \beta_0 + \beta_1 * X_1 + \ldots + \beta_n * X_n + \varepsilon$$

Where
Y is the dependent variable
X is the independent variable
ε represent a random error
β represent a parameter

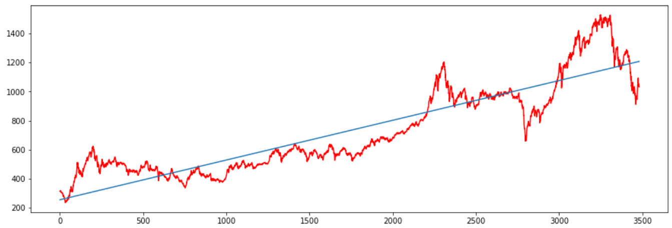


Figure 3: VN-Index Close Stock Price Forecasting with Linear Regression

With data that has a lot of random factors such as the stock market, which VN-Index is high or low followed by the market. Linear Regression cannot be good for fitting the prediction with the actual value test data, it also can be evaluated in the table in section VI (Note: Data tested on Linear and Non-Linear Regression is not split into two data set).

c) Non-Linear Regression

Nonlinear regression is a kind based on Linear Regression. From the Linear Regression formula, if one or more independent variables become Non-Linear, then there will become a Non-Linear Regression. In a graph, Non-Linear is shown in a curve line whose slope changes as the value of one of the variables changes. Depending on the problem, we will choose which formula or which variables will be Non–Linear. For VN-Index, I will choose the Quadratic model which has the formula is:

$$Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2^2 + \varepsilon \qquad [4]$$

Where

y is the dependent variable
x is the independent variable
ε represent a random error
β represent a parameter

Figure 4:VN-Index Close Stock Price Forecasting with Nonlinear Regression

d) Evaluate both statistical models

From visualizing data, we can see that the random factor can make the data have so much noise, while both Linear and Non-Linear just show the changes of values in uptrend or downtrend in general. Of course, the prediction of both are not fit well with actual value, that is why we may look up to new algorithms or models that can make a better prediction.

2) Random Forest

Random Forest is a machine learning ensemble method that is widely used because of its flexibility, simplicity, and often quality results. Before talking about Random Forest algorithms, I would like to present about Decision Tree algorithm first, because Random Forest (and also XGBoots) is based on and improved from the Decision Tree model.

In essence, Decision Trees is a flowlike chart structure where each node of the tree is used to test a particular attribute of the object. For example, imagine I have a person which will represent our object. We then test certain attributes of this person object. For example, one test would be whether they are male or female. The test will represent a "Decision Node" in our tree, and each of the possible outcomes "Male" or "Female" will represent a leaf node. The first "Decision Node" in our Decision Tree will be our "Root Node"[17].

Here is the illustration for Construction of Decision Tree:

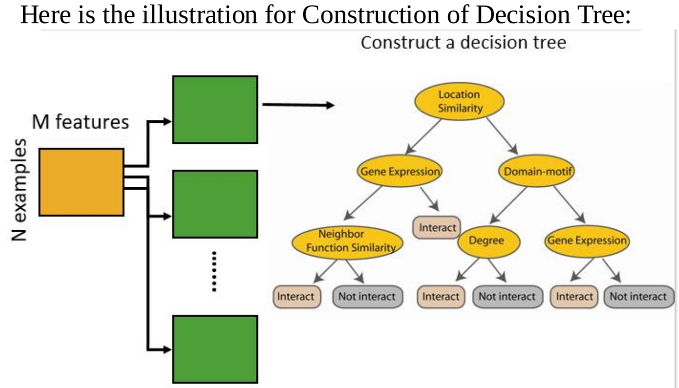


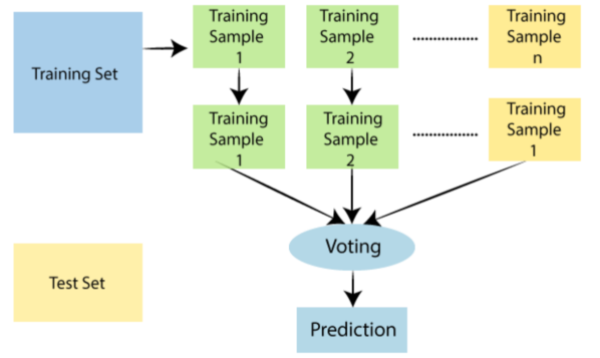
Figure 5: Illustration for Construction of the Decision Tree

Root Node: Represents the entire population or sample and this further gets divided into two or more homogeneous sets. Our starting point.[17]

Random Forest is an ensemble learning algorithm that can solve both classification and regression problems. **Ensemble learning algorithms** combine multiple machine learning algorithms to obtain a better model. Random Forest will build many decision trees with the Decision Tree algorithm, but each decision tree will be different, it came from sampling data randomly, which is why it was named ‘Random’. From the explanation, Random Forest may be a good choice for data that has a lot of random factors like VN-Index[17]. Here is how Random Forest works in four steps:

1. Select random samples from the given dataset
2. Set up a decision tree for each sample and get predicted results from each decision tree.
3. Then vote for each prediction result
4. Choose the most predicted result as the prediction’s final

Here is the flow chart of Working in Random Forest:

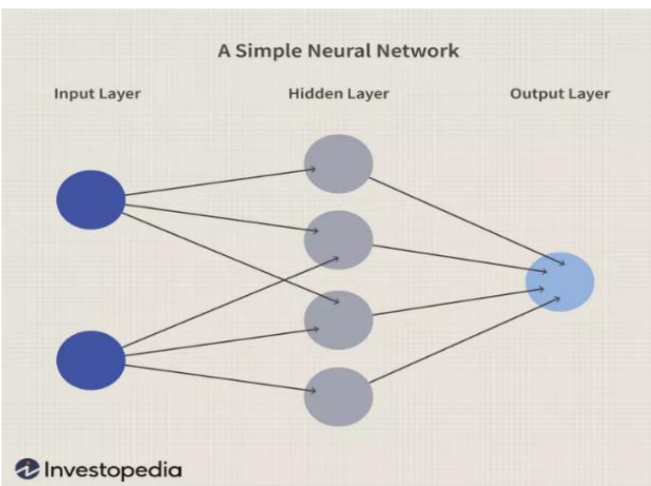


3) LSTM

Before stepping into the LSTM, Neural Network and RNN should be explained first.

a) What is Neural Network?

Neural Network is a type of machine learning process that uses nodes or neurons linked together in a layered structure similar to the human brain. This approach creates an adaptive system used by computers to learn from past data and continuously improve to give the best output.[7]



[5] *Figure 6: Neural Network Architecture*

Neural Network Architecture:

A Neural Network usually includes 3 specific layers as follows:

Input Layer: The first layer that input information comes to. Input nodes process the data, analyze or classify, and then pass the data to the next layer.

Hidden Layer: Located between the input and output layers, represents the inference and information processing process of the system. A neural network can have a large number of hidden layers which is dependent on how large the data has to train.

Output Layer: The output layer gives the result of all the data processed by the neural network.

There are many types of Neural networks which are used for different purposes. For making predictions about future outcomes such as stock data, Recurrent Neural Network (RNN) will be the best fit.

b) Recurrent Neural Network

Recurrent Neural Network (RNN) is a class of artificial neural networks where there is a connection between the output and input in the same segment, the output becomes the next input of that segment creating a loop.

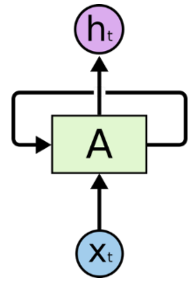


Figure 7: Recurrent Neural Networks have loops.

The figure above depicts a segment of the recurrent neural network A with input x_t and output h_t. A loop allows information to be passed from step to step in the neural network.

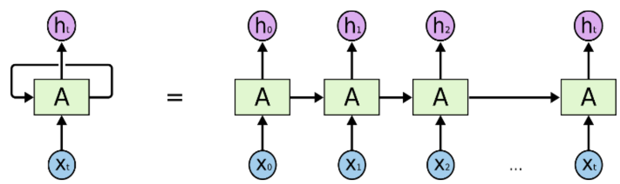


Figure 8: An unrolled recurrent neural network.

If unfolding that loop of neural network A, we can see a recurrent neural network can be thought of as multiple replicas of the same network, where each output of one network is an input of another [30]. This special RNNs memory is called recurrent hidden states and gives the RNNs the ability to predict what input is coming next in the sequence of input data.

But with long sequences like we want to predict VN-Index in long-time sequences, RNN cannot do well. One of the common problems of RNNs is called ”vanishing gradients”. The gradients carry information used in the RNN, and when the gradient becomes too small, the parameter updates become insignificant. This makes the learning of long data sequences difficult. Another problem is Exploding Gradient is when the slope tends to grow exponentially. This problem arises when large error gradients accumulate, resulting in very large updates during the training process.

c) Long Short-Term Memory (LSTM) Models

Long Short-Term Memory (LSTM) is an improvement from the RNNs, which is able to solve the Gradient Problem. The LSTM models essentially extend the RNNs’ memory to enable them to keep and learn long-term dependencies of inputs. The figure below will show the LSTM Architecture

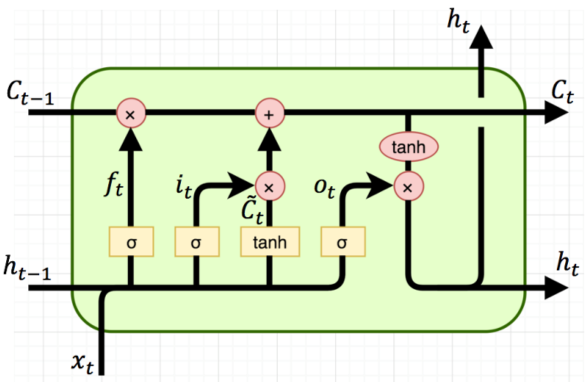


Figure 9: LSTM Architecture

In this Architecture of t state, there are:

Output: h_t,C_t

- where C is the cell state
- h is the hidden state.

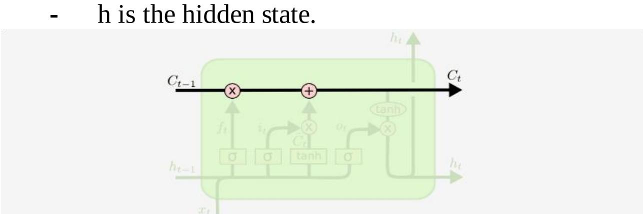
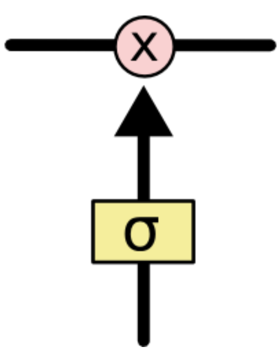


Figure 10: Cell State LSTM

Cell state is basically the aggregate memory of the LSTM network overall time-steps.
This figure (*Figure 10*), the horizontal line on the top is Cell State (from old Cell State C_{t-1} to Cell State C_t, it runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow and avoid change.

Input: C_{t-1}, h_{t-1}, x_t

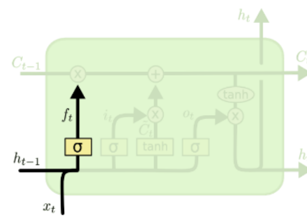
- x_t is the input in the t state of the model.
- C_{t-1}, h_{t-1}, are the output of the previous layer.



The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means "let nothing through," while a value of one means "let everything through!"

Step 1

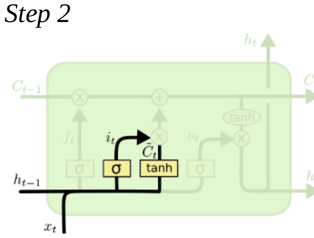


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 12: forget gate layer

The first step in is to decide what information will going to throw away from the cell state, a sigmoid layer called the "forget gate layer." It takes input at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the previous cell state C_{t-1},. The output 1 means the information is kept, 0 means the information is thrown away.

Step 2

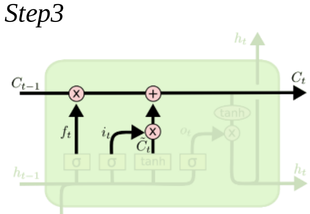


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 13: Updating the value for the state cell by combining the two results from the input layer and the tanh hidden layer

Next step is to decide which new information will be added to LSTM memory. There are two parts: a "sigmoid" layer, and a "tanh" layer. A sigmoid layer called the "input gate layer" decides which value needs to be updated, and the tanh creates a new vector of new candidate values $\tilde{C}_t$ which could be added into the state. The outputs of these two layers are computed by i_t and $h_t$formula above. i_t represents whether the value needs to be updated or not, and $\tilde{C}_t$indicates a vector of new candidate values that will be added to the LSTM memory. The combination of these two layers creates an update to update the old cell state C_{t-1}, into the new cell state C_t .

Step3



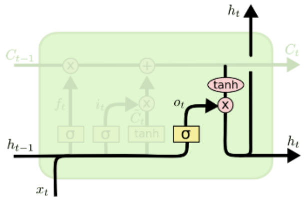
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 14: New status box

Multiplying the old state C_{t-1}, by f_t, where it results at the forget gate. Then add it with $i_t * \tilde{C}_t$.

Final step

First a sigmoid layer to decide what part of the LSTM memory contributes to the output. Then, putting cell state through a non-linear tanh function to push the values between −1 and 1. Finally, the result is multiplied by the output of a sigmoid layer.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 15: Adjusting the output information through the tanh . function

Where :

- o_t t is the output value
- h_t is its representation as a value between −1 and 1.

4) *XGBoots Model*

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library.

Decision trees are generated sequentially in this approach. Weights are significant in XGBoost. Each independent variable is given a weight before being fed into the decision tree that forecasts outcomes. Variables that the tree incorrectly predicted are given more weight before being placed into the second decision tree. These distinct classifiers/predictors are then combined to produce a robust and accurate model. Regression, classification, ranking, and user-defined prediction issues are all applicable.[10]

B. *Evaluation methods*

To evaluate the effectiveness of models, and compare between models, Mean Squared Error, Mean Absolute Error, Mean Absolute Percent Error, and R2 Score are used. These are common metrics for evaluating the model.

1) *Mean Squared Error:*

In statistics, Mean Squared Error (MSE) calculates the average of the square of the error between the actual value and the predicted value

$$\text{MSE} = \frac{\sum_{t=1}^{n} \varepsilon_t^2}{n} = \frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)}{n}$$

[26]

2) *Mean Absolute Error:*

Mean Absolute Error (MAE) evaluates a model by averaging the absolute value of the error between the actual value and the predicted value.

$$\text{MAE} = \frac{\sum_{t=1}^{n} |\varepsilon_t|}{n} = \frac{\sum_{t=1}^{n} |Y_t - \hat{Y}_t|}{n}$$

[9]

3) *Mean Absolute Percent Error:*

It calculates the average of the absolute percentage errors of each entry in a dataset in other that to know how accurate the forecasted quantities were in comparison with the actual quantities.

$$\text{MAPE} = \frac{\sum_{t=1}^{n} \frac{|\varepsilon_t|}{Y_t}}{n} = \frac{\sum_{t=1}^{n} \frac{|Y_t - \hat{Y}_t|}{Y_t}}{n}$$

[29]

4) *R2 Score :*

R2 is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, r-squared shows how well the data fit the regression model. The R-squared value R2 is always between 0 and 1 inclusive, the model is effective when the value of R2 is higher than 0.5. The higher the R2 value, the closer the relationship between the independent variable and the dependent variable is, which means the model is more effective.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

[10]

V. HYPER-PARAMETER SETTINGS

There is no specific rule when setting hyperparameters for each model. In order that finds out the hyperparameter is a search process. For Random Forest, I used the tuning hyperparameters for Random Forest. About LSTM, I found that the model would be effective with the number of neurons is between input and output in each layer, while the value of batch size would be a number that is divisible by 2 when we choose the value of epoch from 100 above. XGBoots is the only model which not optimized because the best hyperparameters that hyperparameters tunning found for XGBoots result in worse than the initial model, because the limited time of the project did not allow me to search and test to find out the optimized hyperparameters for the model. So that with XGBoots, I used the initial model.

VI. RESULT OF THE PREDICTED VALUE ON TEST DATA

A. *XGBOOST Model*

Figure 20: Prediction VN-Index of XGBoots

B. *LSTM Model*

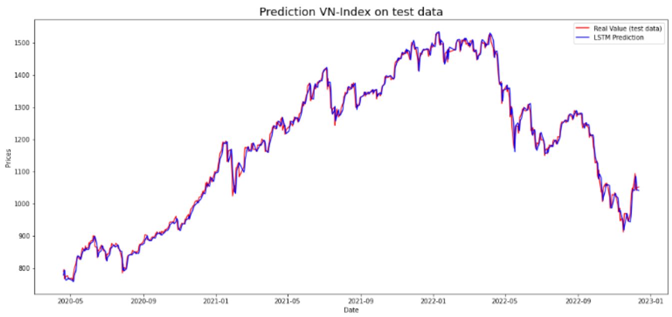


Figure 21: Prediction VN-Index of LSTM

C. *Random Forest*

Figure 22: Prediction VN-Index of Random Forest

VII. ERROR ANALYSIS

We calculate Mean Squared Error(MSE), Mean Absolute Error(MAE), Mean Absolute Percent Error(MAPE), R2 between the Predict Value and Actual Value to evaluate and compare between models.

Model	MSE	MAE	MAPE	R2
Non-Linear	12490.834	80.162	11.282	0.867
Linear	18207.858	108.465	16.011	0.806
Random Forest	398.688	14.630	1.270	0.992
LSTM	297.270	12.263	1.063	0.994
XGBoost	683.652	17.382	1.497	0.986

With this result, LSTM is the most effective model of all. The R2 of all five models is very high, while the R2 of LSTM is very close to 1.

VIII. RESULT OF THE FORECAST PRICE IN NEXT 30 DAYS FUTURE

Here is the Predicted values of next 30 days of XGBoost Model:

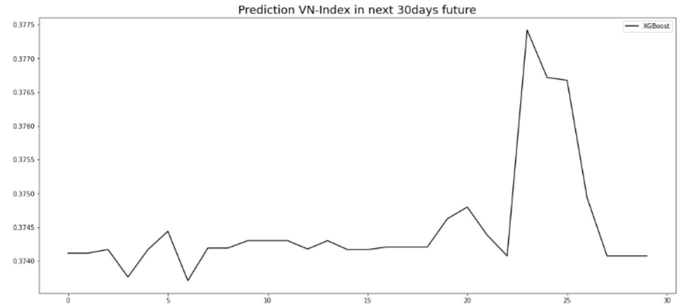


Figure 23: Prediction VN-Index in next 30days future of XGBoots

Here is the Predicted values of next 30 days of LSTM Model:

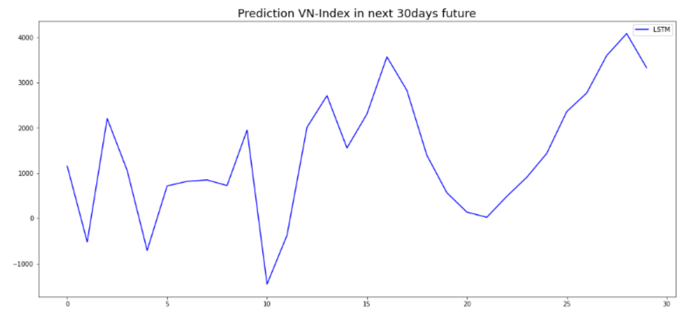


Figure 24: Prediction VN-Index in next 30days future of LSTM

Here is the Predicted values of next 30 days of XGBoost Model:

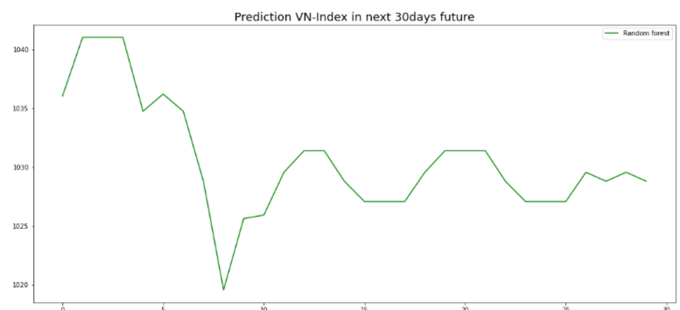


Figure 25: Prediction VN-Index in next 30days future of Random Forest

IX. DEVELOPMENT

Because of limited time, there are some ideas that we have but can not do right at this time. If there is enough time, we will consider developing the model for forecasting by a combination between LSTM and Transformer model to predict the VN-Index, and create a demo for reviewing the model by predicting the percentage of increase or decrease VN-Index and an auto bot make a buy or sell when increase or decrease and calculate which model will bring the most profit.

X. CONCLUSION

In this report, the group presented a number of contents group theory of data usage and five models are Linear Regression, Non - Linear Regression, Random Forest, XGBoost, Long Short Term Memory as well as theory and formula for calculating Mean Squared Error, Mean Absolute Error, Root Squared, and Mean Absolute Percent Error for evaluate and compare between model. In the implementation process, we also searched and tested some hyperparameters on the model to decrease the error between the predicted value and the actual value. We can see that the predictive power of the XGBoost model on the testing with the data 80/20 split ratio is worse than with Random Forest and LSTM models. The result indicated that LSTM model with 80% train data set was superior.

.

XI. REFERENCE

[1] "What is VN-Index? Everything about VN-Index" from https://ysedu.yuanta.com.vn/

[2] VN-Index daily data from Investing.com

[3] "Linear Regression" document from http://www.stat.yale.edu/

[4] 1Emad kendory, 2Nazar Abdul Kareem, 3Saleh Mahdi Humadi "Using the nonlinear regression to predict the prices of stocks and its impact on the investor decisions in the financial market"

[5] Image by Sabrina Jiang © Investopedia 2020

[6] Christopher Olah "Understanding LSTM Networks" on colah's blog https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[7] Larry Hardesty | MIT News Office Publication Date:April 14, 2017 "Explained: Neural networks"

[8] "R-Squared in Regression Analysis" from net-informations.com

[9] "Mean Squared Error (MSE)". www.probabilitycourse.com. Retrieved 2020-09-12.

[10] R2 Score in Machine Learning",Aman Kharwal, [Online],Available: 2 Score in Machine Learning (thecleverprogrammer.com)

[11] Hochreiter, Sepp (1991). Untersuchungen zu dynamischen neuronalen Netzen (PDF) (diploma thesis). Technical University Munich, Institute of Computer Science, advisor: J. Schmidhuber because of the computations involved in the process

[12] Calin, Ovidiu (14 February 2020). Deep Learning Architectures. Cham, Switzerland: Springer Nature. p. 555. ISBN 978-3-030-36720-6

[13] XGBoost from https://www.geeksforgeeks.org/

[14] 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

[15] XGBoost: A Scalable Tree Boosting System. https://arxiv.org/pdf/1603.02754.pdf

[16] xgboost: eXtreme Gradient Boosting Tianqi Chen, Tong He Package Version: 0.6-4 January 4, 2017. https://cran.microsoft.com/snapshot/2017-12-11/web/packages/xgboost/vignettes/xgboost.pdf

[17] Long Short-term Memory – Neural Computation 9(8) : 1735-1780, 1997. https://www.researchgate.net/publication/13853244_Long_Short-term_Memory

[18] Forecasting directional movements of stock prices for intraday trading using LSTM and random forests Pushpendu Ghosh a , Ariel Neufeld b , Jajati Keshari Sahoo c aDepartment of Computer Science & Information Systems. https://arxiv.org/pdf/2004.10178v2.pdf

[19] Hochreiter, Sepp (1991). Untersuchungen zu dynamischen neuronalen Netzen (PDF) (diploma thesis). Technical University Munich, Institute of Computer Science, advisor: J. Schmidhuber because of the computations involved in the process

[20] Calin, Ovidiu (14 February 2020). Deep Learning Architectures. Cham, Switzerland: Springer Nature. p. 555. ISBN 978-3-030-36720-6

[21] R.J.Oosterbaan, 1994, Frequency and Regression Analysis. In: H.P.Ritzema (ed.), Drainage Principles and Applications, Publ. 16, pp. 175-224, International Institute for Land Reclamation and Improvement (ILRI), Wageningen, The Netherlands.

[22] "Cây Quyết Định (Decision Tree)," Trí tuệ nhân tạo,[Online], Available: Cây Quyết Định (Decision Tree) - Trí tuệ nhân tạo (trituenhantao.io)

[23] A Gers, J Schmidhuber and F Cummins. Learning to forget continual prediction with LSTM. IEEE Xplore, 10.1049/cp:19991218, 2002revealed

[24] Alex Reed, "Predicting Stock Movement with Random Forest", Github, [Online], Available: : sigma_coding_youtube/random_forest_price_prediction.ipyn b at master · areed1192/sigma_coding_youtube (github.com)

[25] R2 Score in Machine Learning",Aman Kharwal, [Online],Available: 2 Score in Machine Learning (thecleverprogrammer.com)

[26] "Mean Squared Error (MSE)". www.probabilitycourse.com. Retrieved 2020-09-12.

[27] Willmott, Cort J.; Matsuura, Kenji (December 19, 2005). "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance". Climate Research. 30: 79–82. doi:10.3354/cr030079

[28] "2.5 Evaluating forecast accuracy | OTexts". www.otexts.org. Retrieved 2016-05-18.

[29] N Hyndman, R. and Koehler A. (2005). "Another look at measures of forecast accuracy" [21]

[30] According *to the Enterprise Statistics Textbook, National Economics University Publishing House*. "What is Forecast Error?"