Predicting Stock Performance of Technology Giants: A Scientific Research Study

1st Nguyen Hien Duc
Faculty of Information Systems
University of
Information Technology
Ho Chi Minh City, Vietnam
20520450@gm.uit.edu.vn

2nd Nguyen Bao Anh
Faculty of Information Systems
University of
Information Technology
Ho Chi Minh City, Vietnam
20521068@gm.uit.edu.vn

3rd Vo Nu Diem Trang
Faculty of Information Systems
University of
Information Technology
Ho Chi Minh City, Vietnam
20521013@gm.uit.edu.vn

4th Tran Gia Phong
Faculty of Information Systems
University of
Information Technology
Ho Chi Minh City, Vietnam
20521748@gm.uit.edu.vn

5th Le Nguyen Minh Trung
Faculty of Information Systems
University of
Information Technology
Ho Chi Minh City, Vietnam
19521061@gm.uit.edu.vn

Abstract – ***The stock market allows buyers and sellers to interact and transact shares that represent their ownership of a business. The creation of trustworthy prediction models for the equities market enables investors to make better choices. This research aims to predict the future stock prices of some large businesses (MSFT, GOOGL, AMZN) in the world. In addition. Machine learning and deep learning consists of making computer tasks using human intelligence is currently the top trending technique. It is now a potent analytical tool for managing investments effectively in the financial markets. A novel method that can assist investors in making better investment and management decisions to achieve improved performance of their securities investments has been made possible by the widespread use of ML in the financial sector. In this research, we first review the shares prices of the above business in recent years. After that, we use data and combine some machine learning algorithms (Linear Regression, ARIMA, GRU, LSTM, VAR, SSA, DeepAR, Seq2Seq) and price stock prediction in the future (30 days).***

Keywords **– Stock Prices, Predict,** ***Linear regression, Arima, GRU, VECM, LSTM, VAR, SSA, Seq2Seq, FCN, DeepAR.***

I. Introduction

Investors, publicly traded firms, and governments are all clearly interested in forecasting stock price changes. The question of whether the market can be forecast has been up for dispute. According to the Random Walk Theory (Malkiel, 1973), prices are established arbitrarily, making it impossible to outperform the market. But with AI advancements, it has been empirically demonstrated that stock price movement is predictable.

The Stock Market is a highly complex system, where huge chunks and volumes of information. Data is generated instantaneously and constantly changes in small proportions with different factors and diversity. Since the stock market is primarily dynamic, nonlinear, complex, nonparametric, and chaotic in character, stock market prediction is regarded as a tough task of the financial time series prediction process. The stock market is additionally influenced by a variety of macroeconomic factors, including political developments, corporate policies, general economic conditions, investor expectations, institutional investment preferences, movement in other stock markets, investor psychology, etc.

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

II. Related Work

The prediction of stock prices has been a challenging and extensively studied problem in the field of finance and machine learning. Researchers have explored various techniques and methodologies to improve the accuracy of stock price predictions. In this study, we will utilize the following 9 algorithms: ARIMA, GRU, LSTM, Linear Regression, RNN, VAR, SSA, Seq2seq, DeepAR and VECM to predict the stock prices of Technology Giants: Microsoft, Amazon and Alphabet.

Changchun Cai, Yuan Tao, Qiwen Ren and Gang Hu with their paper "Short-term load forecasting based on MB-LSTM neural network" [1]. In this paper, a short-term load forecasting framework based on multi-layer stacked bidirectional long short-term memory (MB-LSTM) is proposed. The goal of the model is to solve the problem of error accumulation caused by traditional LSTM and improve the forecasting accuracy.

S. Bayraci, Y. Ari, and Y. Yildirim with their paper "A VECTOR AUTO-REGRESSIVE (VAR) MODEL FOR THE TURKISH FİNANCIAL MARKETS" [2]. In this paper, they develop a vector autoregressive (VAR) model of the Turkish financial markets for the period of June 15, 2006 – June 15 2010 and forecasts ISE100 index, TRY/USD exchange rate, and short-term interest rates. The out-of-sample forecast performance of the VAR model is compared with the results from the univariate models.

The paper "Short-time multi-energy load forecasting method based on CNN-Seq2Seq model with attention mechanism" [3] written by Ge Zhang, Xiaoqing Bai, Yuxuan Wang… This paper proposes a CNN-Seq2Seq model with an attention mechanism based on a multi-task learning method for a short-time multi-energy load forecasting. In detail, CNN is used to extract useful features of the input data. Then, the short-time multi-energy load is forecasted by using Seq2Seq according to the extracted features. Meanwhile, the attention mechanism and multi-task learning method are introduced to improve the accuracy of load forecasting. The simulation results with the actual data of an IEM validate the effectiveness of the proposed short-time multi-energy load forecasting method.

The paper "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation" [4]. This paper focuses on learning phrase representations within the RNN Encoder-Decoder framework. The authors propose a method to automatically learn continuous distributed representations for phrases by training an RNN Encoder-Decoder model on a large parallel corpus.

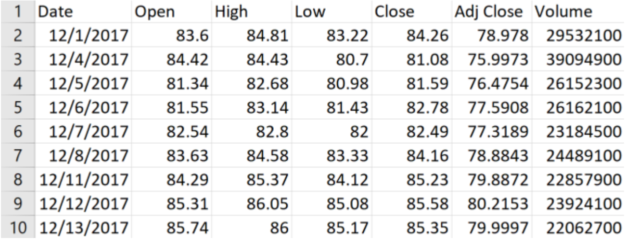
H. Cao, Y. Song, Y. Li, R. Li, H. Shi, J. Yu and M. Hu with their paper "Reduction of Moving Target Time-of-Flight Measurement Uncertainty in Femtosecond Laser Ranging by Singular Spectrum Analysis Based Filtering" [5]. In this paper, they propose a new route to utilizing a powerful singular spectrum analysis (SSA) filtering method to improve femtosecond laser ranging precision for moving targets with acceleration.

D. Salinas, V. Flunkert, J. Gasthaus and T. Januschowski, "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks" [6]. This paper proposes DeepAR, a methodology for producing accurate probabilistic forecasts, based on training an autoregressive recurrent neural network model on a large number of related time series. They demonstrate how the application of deep learning techniques to forecasting can overcome many of the challenges that are faced by widely-used classical approaches to the problem.

III. MATERIALS

A. Microsoft Corporation (MSFT)

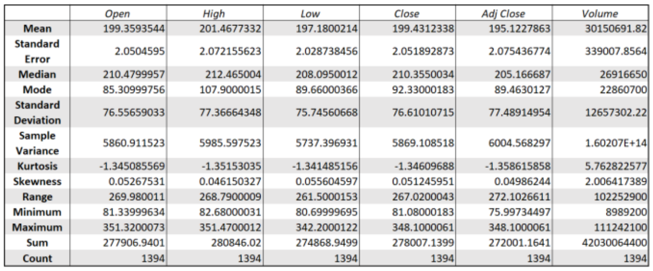
Microsoft Corporation (MSFT) is a leading technology company known for its software, services, and devices. It offers products such as the Windows operating system and Office suite, along with cloud services like Azure. Microsoft has made significant contributions to the IT industry and has a strong market presence. The dataset was gathered from Yahoo! Finanace website, containing 7 columns and 1394 rows of data from December 2017 to the present.



1	Date	Open	High	Low	Close	Adj Close	Volume
2	12/1/2017	83.6	84.81	83.22	84.26	78.978	29532100
3	12/4/2017	84.42	84.43	80.7	81.08	75.9973	39094900
4	12/5/2017	81.34	82.68	80.98	81.59	76.4754	26152300
5	12/6/2017	81.55	83.14	81.43	82.78	77.5908	26162100
6	12/7/2017	82.54	82.8	82	82.49	77.3189	23184500
7	12/8/2017	83.63	84.58	83.33	84.16	78.8843	24489100
8	12/11/2017	84.29	85.37	84.12	85.23	79.8872	22857900
9	12/12/2017	85.31	86.05	85.08	85.58	80.2153	23924100
10	12/13/2017	85.74	86	85.17	85.35	79.9997	22062700

Figure 1. MSFT dataset

Calculate the values of Descriptive Statistics for the attributes in the MSFT dataset:

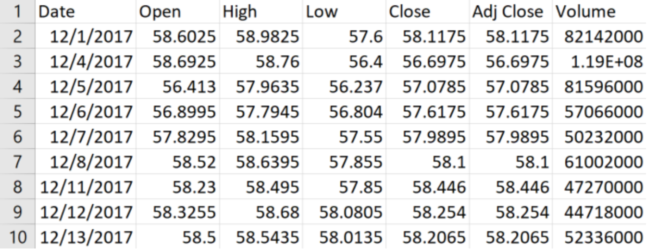


	Open	High	Low	Close	Adj Close	Volume
Mean	199.3593544	201.4677332	197.1800214	199.4312338	195.1227863	30150691.82
Standard Error	2.0504595	2.072155623	2.028738456	2.051892873	2.075436774	339007.8564
Median	210.4799957	212.465004	208.0950012	210.3550034	205.166687	26916650
Mode	85.30999756	107.9000015	89.66000366	92.33000183	89.4630127	22860700
Standard Deviation	76.55659033	77.36664348	75.74560668	76.61010715	77.48914954	12657302.22
Sample Variance	5860.911523	5985.597523	5737.396931	5869.108518	6004.568297	1.60207E+14
Kurtosis	-1.345085569	-1.35153035	-1.341485156	-1.34609688	-1.358615858	5.762822577
Skewness	0.05267531	0.046150327	0.055604597	0.051245951	0.04986244	2.006417389
Range	269.980011	268.7900009	261.5000153	267.0200043	272.1026611	102252900
Minimum	81.33999634	82.68000031	80.69999695	81.08000183	75.99734497	8989200
Maximum	351.3200073	351.4700012	342.2000122	348.1000061	348.1000061	111242100
Sum	277906.9401	280846.02	274868.9499	278007.1399	272001.1641	42030064400
Count	1394	1394	1394	1394	1394	1394

Figure 2. Descriptive Statistics in the MSFT dataset

B. Amazon.com Inc. (AMZN)

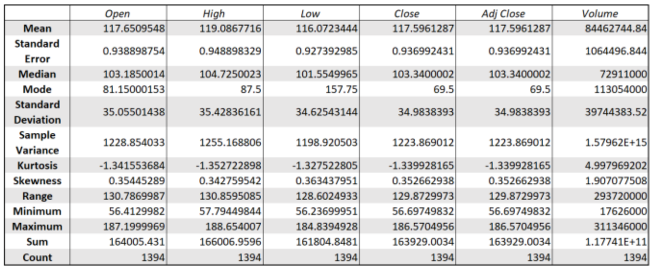
Amazon is a global technology company and the world's largest online retailer. It provides a wide range of goods and services through its e-commerce platform and has expanded into areas such as cloud services (AWS), media streaming (Amazon Prime Video), and artificial intelligence (Alexa). Amazon has revolutionized the way people shop and access products online. The dataset was gathered from Yahoo! Finanace website, containing 7 columns and 1394 rows of data from December 2017 to the present.



1	Date	Open	High	Low	Close	Adj Close	Volume
2	12/1/2017	58.6025	58.9825	57.6	58.1175	58.1175	82142000
3	12/4/2017	58.6925	58.76	56.4	56.6975	56.6975	1.19E+08
4	12/5/2017	56.413	57.9635	56.237	57.0785	57.0785	81596000
5	12/6/2017	56.8995	57.7945	56.804	57.6175	57.6175	57066000
6	12/7/2017	57.8295	58.1595	57.55	57.9895	57.9895	50232000
7	12/8/2017	58.52	58.6395	57.855	58.1	58.1	61002000
8	12/11/2017	58.23	58.495	57.85	58.446	58.446	47270000
9	12/12/2017	58.3255	58.68	58.0805	58.254	58.254	44718000
10	12/13/2017	58.5	58.5435	58.0135	58.2065	58.2065	52336000

Figure 3. AMZN dataset

Calculate the values of Descriptive Statistics for the attributes in the AMZN dataset:

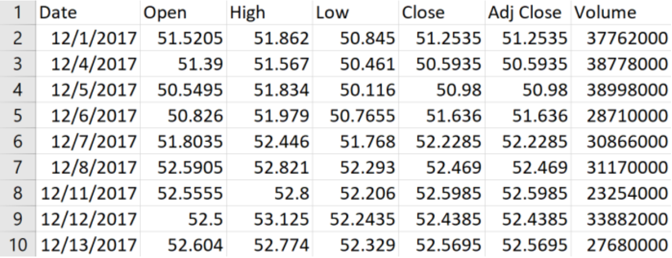


	Open	High	Low	Close	Adj Close	Volume
Mean	117.6509548	119.0867716	116.0723444	117.5961287	117.5961287	84462744.84
Standard Error	0.938898754	0.948898329	0.927392985	0.936992431	0.936992431	1064496.844
Median	103.1850014	104.7250023	101.5549965	103.3400002	103.3400002	72911000
Mode	81.15000153	87.5	157.75	69.5	69.5	113054000
Standard Deviation	35.05501438	35.42836161	34.62543144	34.9838393	34.9838393	39744383.52
Sample Variance	1228.854033	1255.168806	1198.920503	1223.869012	1223.869012	1.57962E+15
Kurtosis	-1.341553684	-1.352722898	-1.327522805	-1.339928165	-1.339928165	4.997969202
Skewness	0.35445289	0.342759542	0.363437951	0.352662938	0.352662938	1.907077508
Range	130.7869987	130.8595085	128.6024933	129.8729973	129.8729973	293720000
Minimum	56.4129982	57.79449844	56.23699951	56.69749832	56.69749832	17626000
Maximum	187.1999969	188.654007	184.8394928	186.5704956	186.5704956	311346000
Sum	164005.431	166006.9596	161804.8481	163929.0034	163929.0034	1.17741E+11
Count	1394	1394	1394	1394	1394	1394

Figure 4. Descriptive Statistics in the AMZN dataset

C. Alphabet Inc. - Class A (GOOGL)

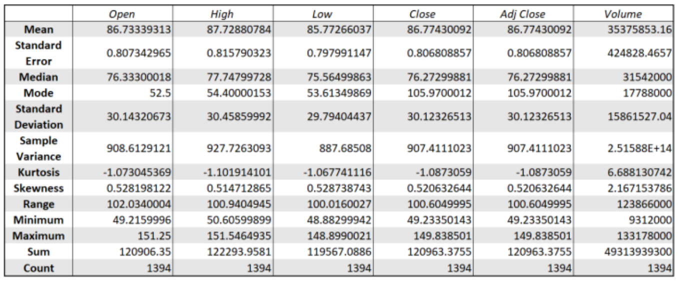
Alphabet is the parent company of Google, a leading technology company recognized for its online search, advertising, and mobile operating system Android. Alphabet also offers services like YouTube, Google Maps, and Google Cloud. It invests in emerging technologies such as artificial intelligence, self-driving cars, and renewable energy. Alphabet's innovation and influence span across various sectors, shaping our interactions with technology. The dataset was gathered from Yahoo! Finanace website, containing 7 columns and 1394 rows of data from December 2017 to the present.



1	Date	Open	High	Low	Close	Adj Close	Volume
2	12/1/2017	51.5205	51.862	50.845	51.2535	51.2535	37762000
3	12/4/2017	51.39	51.567	50.461	50.5935	50.5935	38778000
4	12/5/2017	50.5495	51.834	50.116	50.98	50.98	38998000
5	12/6/2017	50.826	51.979	50.7655	51.636	51.636	28710000
6	12/7/2017	51.8035	52.446	51.768	52.2285	52.2285	30866000
7	12/8/2017	52.5905	52.821	52.293	52.469	52.469	31170000
8	12/11/2017	52.5555	52.8	52.206	52.5985	52.5985	23254000
9	12/12/2017	52.5	53.125	52.2435	52.4385	52.4385	33882000
10	12/13/2017	52.604	52.774	52.329	52.5695	52.5695	27680000

Figure 5. GOOGL dataset

Calculate the values of Descriptive Statistics for the attributes in the GOOGL dataset:



	Open	High	Low	Close	Adj Close	Volume
Mean	86.73339313	87.72880784	85.77266037	86.77430092	86.77430092	35375853.16
Standard Error	0.807342965	0.815790323	0.797991147	0.806808857	0.806808857	424828.4657
Median	76.33300018	77.74799728	75.56499863	76.27299881	76.27299881	31542000
Mode	52.5	54.40000153	53.61349869	105.9700012	105.9700012	17788000
Standard Deviation	30.14320673	30.45859992	29.79404437	30.12326513	30.12326513	15861527.04
Sample Variance	908.6129121	927.7263093	887.68508	907.4111023	907.4111023	2.51588E+14
Kurtosis	-1.073045369	-1.101914101	-1.067741116	-1.0873059	-1.0873059	6.688130742
Skewness	0.528198122	0.514712865	0.528738743	0.520632644	0.520632644	2.167153786
Range	102.0340004	100.9404945	100.0160027	100.6049995	100.6049995	123866000
Minimum	49.2159996	50.60599899	48.88299942	49.23350143	49.23350143	9312000
Maximum	151.25	151.5464935	148.8990021	149.838501	149.838501	133178000
Sum	120906.35	122293.9581	119567.0886	120963.3755	120963.3755	49313939300
Count	1394	1394	1394	1394	1394	1394

Figure 6. Descriptive Statistics in the GOOGL dataset

IV. Method

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. Model

1) Autoregressive Integrated Moving Average (ARIMA)

The Autoregressive Integrated Moving Average (ARIMA) model uses time-series data and statistical analysis to interpret the data and make future predictions. The ARIMA model aims to explain data by using time series data on its past values and uses linear regression to make predictions. [7]

The following descriptive acronym explains the meaning of each of the key components of the ARIMA model:

- The "AR" in ARIMA stands for autoregression, indicating that the model uses the dependent relationship between current data and its past values. In other words, it shows that the data is regressed on its past values.
- The "I" stands for integrated, which means that the data is stationary. Stationary data refers to time-series data that's been made "stationary" by subtracting the observations from the previous values.
- The "MA" stands for moving average model, indicating that the forecast or outcome of the model depends linearly on the past values. Also, it means that the errors in forecasting are linear functions of past errors. Note that the moving average models are different from statistical moving averages.

Each of the AR, I, and MA components are included in the model as a parameter. The parameters are assigned specific integer values that indicate the type of ARIMA model. A common notation for the ARIMA parameters is shown and explained below:

ARIMA (p, d, q)

- The parameter p is the number of autoregressive terms or the number of "lag observations." It is also called the "lag order," and it determines the outcome of the model by providing lagged data points.
- The parameter d is known as the degree of differencing. it indicates the number of times the lagged indicators have been subtracted to make the data stationary.
- The parameter q is the number of forecast errors in the model and is also referred to as the size of the moving average window.

Applying ARIMA algorithm to the MSFT dataset:

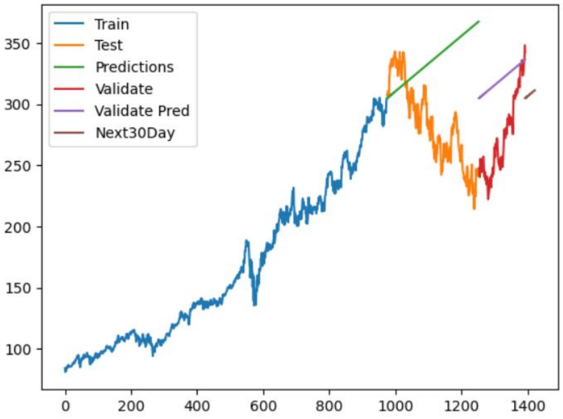


Figure 7. Result of ARIMA (7:2:1)

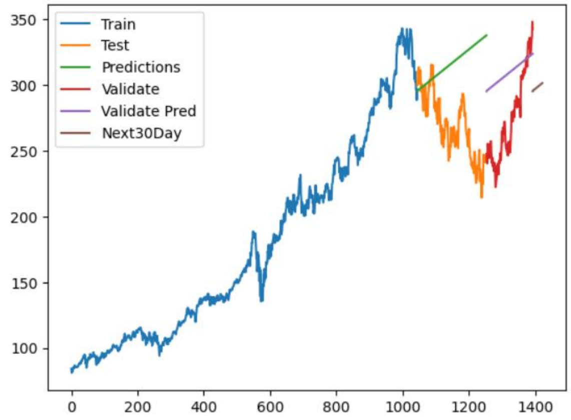


Figure 8. Result of ARIMA (7.5:1.5:1)

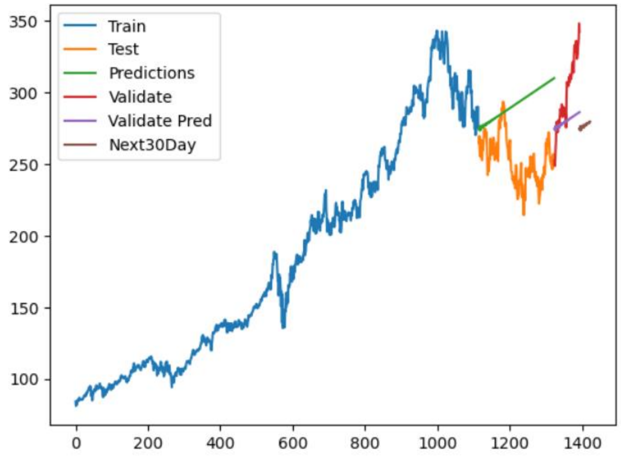


Figure 3. Result of ARIMA (8.5:1.5:0.5)

2) *Long short-term memory (LSTM)*

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network that can learn order dependence in sequence prediction problems. This is a necessary characteristic in complex problem domains such as machine translation, speech recognition, and others. [8]

An LSTM layer is made up of a collection of recurrently connected memory blocks. Each block contains one or more recurrently connected memory cells through three multiplicative units - the input, output, and forget gates. These provide continuous analogs of the cells' write, read, and reset operations. The advent of LSTM networks minimizes the drawback of gradient vanishing in part by allowing information to propagate more directly through the cell state.

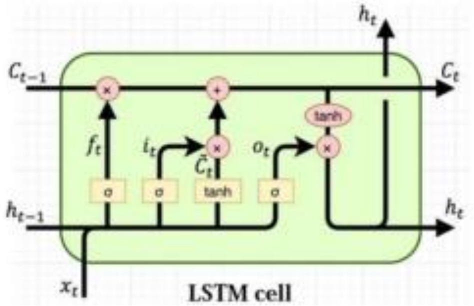


Figure 9. LSTM Architechture Flow Diagram [9]

$$\text{Forget gate}: f_t = \sigma(W_f x_t + U_f h_{t-1})$$

$$\text{Input gate}: i_t = \sigma(W_i x_t + U_i h_{t-1})$$

$$\text{Cell gate}: c_t = \tanh(W_c x_t + U_c h_{t-1})$$

$$\text{Output gate}: o_t = \phi h(W_o x_t + U_o h_{t-1})$$

$$\text{Cell state}: c_t = f_t \times c_{t-1} + i_t \times c_t$$

Figure 10. Calculate in LSTM cell [1]

Applying the LSTM algorithm to the GOOGL dataset:

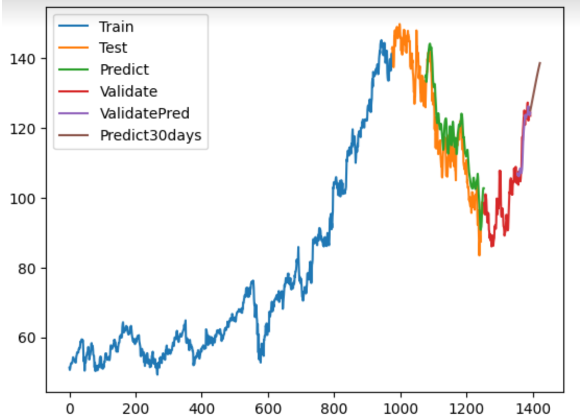


Figure 11. Result of LSTM (7:2:1)

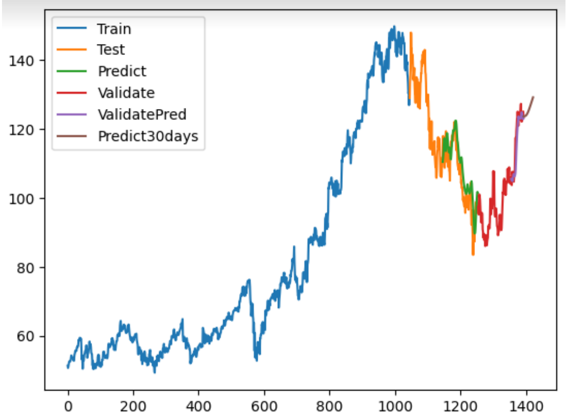


Figure 12. Result of LSTM (7.5:1.5:1)

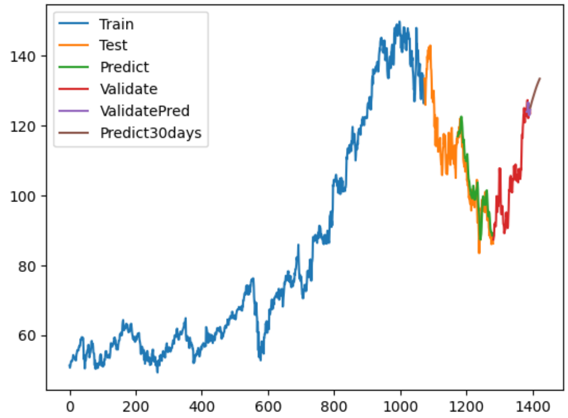


Figure 13. Result of LSTM (8:1.5:0.5)

3) *Gated Recurrent Unit (GRU)*

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

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

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

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

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

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

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

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

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

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

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

Applying Gated Recurrent Unit algorithm to the AMZN dataset:

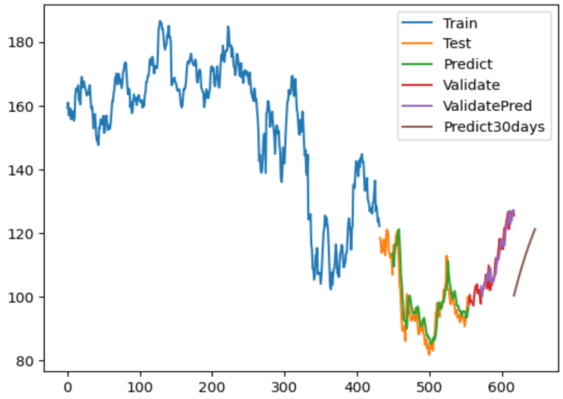


Figure 14. Result of Recurrent Neural Network (7:2:1)

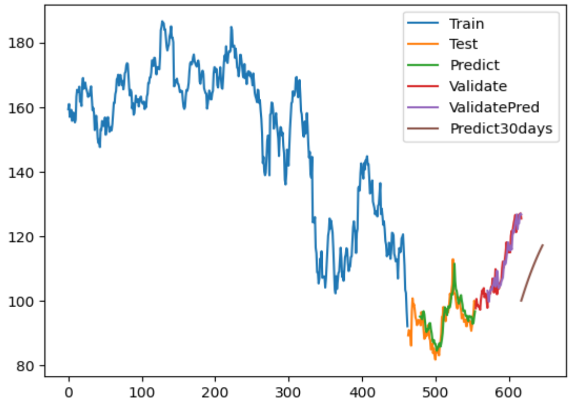


Figure 15. Result of Recurrent Neural Network (7.5:1.5:1)

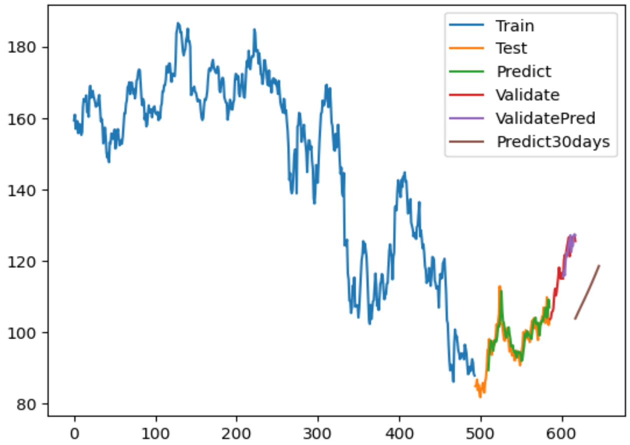


Figure 16. Result of Recurrent Neural Network (8:1.5:0.5)

4) *Linear Regression*

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

The formula of Linear Regression can present below: [10]

$$y_i = \beta_0 + \beta_1 x_{i1} + \varepsilon$$

- y_i : dependent variable (Target Variable)
- x_{i1} : independent variable
- β_0 : intercept of the line
- β_1 : linear regression coefficient
- ε : random error

Applying Linear Regression algorithm to the GOOGL dataset:

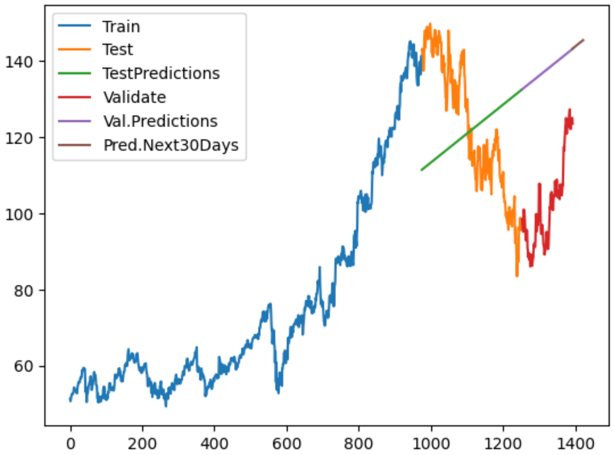


Figure 17. Result of Linear Regression (7:2:1)

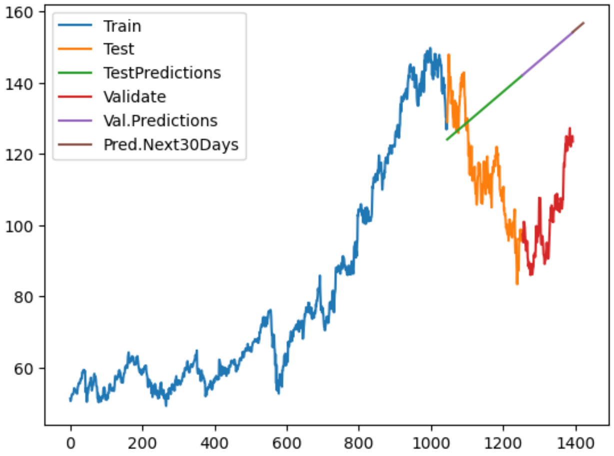


Figure 18. Result of Linear Regression (7.5:1.5:1)

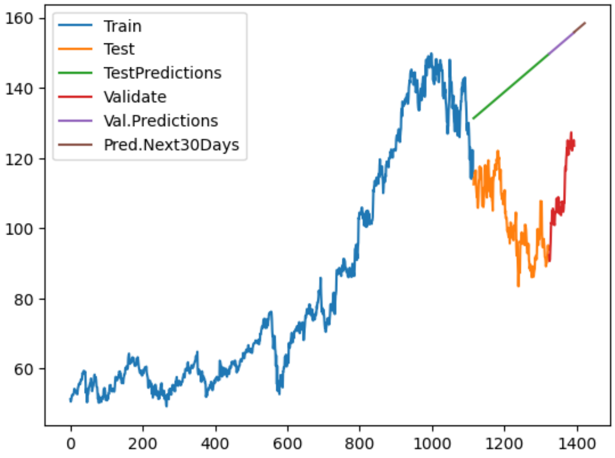


Figure 19. Result of Linear Regression (8:1.5:0.5)

5) Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is a neural network that consists of a hidden state h and an optional output y which operates on a variable length sequence $x = (x_1, ..., x_T)$. At each time step t, the hidden state hhti of the RNN is updated by

$$h_{<t>} = f(h_{<t-1>}, x_t)$$

where f is a non-linear activation function. f may be as simple as an elementwise logistic sigmoid function and as complex as a long short-term memory (LSTM) unit (Hochreiter and Schmidhuber, 1997).

An RNN can learn a probability distribution over a sequence by being trained to predict the next symbol in a sequence. In that case, the output at each timestep t is the conditional distribution p(xt | xt−1, . . . , x1). For example, a multinomial distribution (1-of-K coding) can be output using a softmax activation function.

$$p(x_{t,j} = 1 \mid x_{t-1}, \ldots, x_1) = \frac{\exp\left(\mathbf{w}_j \mathbf{h}_{\langle t \rangle}\right)}{\sum_{j'=1}^{K} \exp\left(\mathbf{w}_{j'} \mathbf{h}_{\langle t \rangle}\right)}$$

for all possible symbols j = 1, . . . , K, where wj are the rows of a weight matrix W. By combining these probabilities, we can compute the probability of the sequence x using

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_{t-1}, \ldots, x_1).$$

From this learned distribution, it is straightforward to sample a new sequence by iteratively sampling a symbol at each time step. [4]

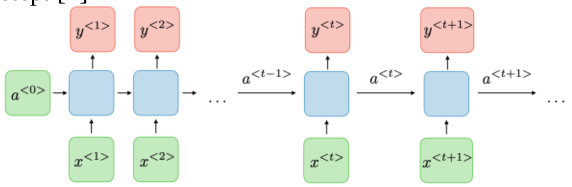


Figure 20. Architecture of a traditional Recurrent Neural Network

Applying Recurrent Neural Network algorithm to the MSFT dataset:

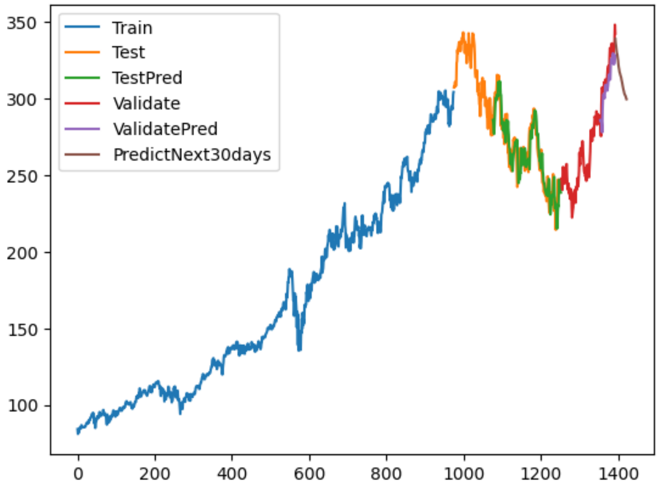


Figure 21. Result of Recurrent Neural Network (7:2:1)

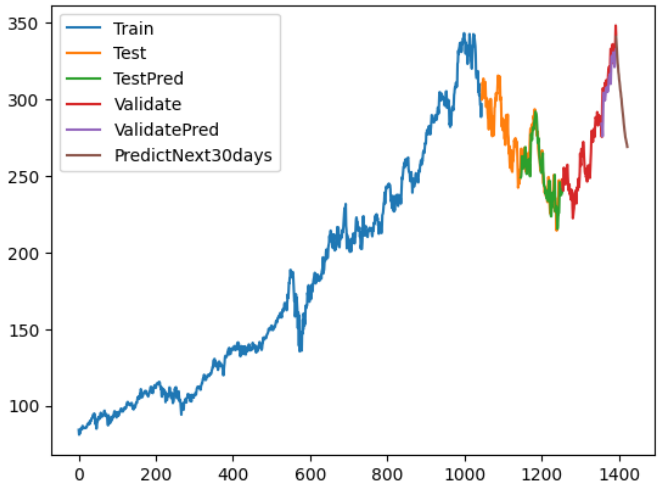


Figure 22. Result of Recurrent Neural Network (7.5:1.5:1)

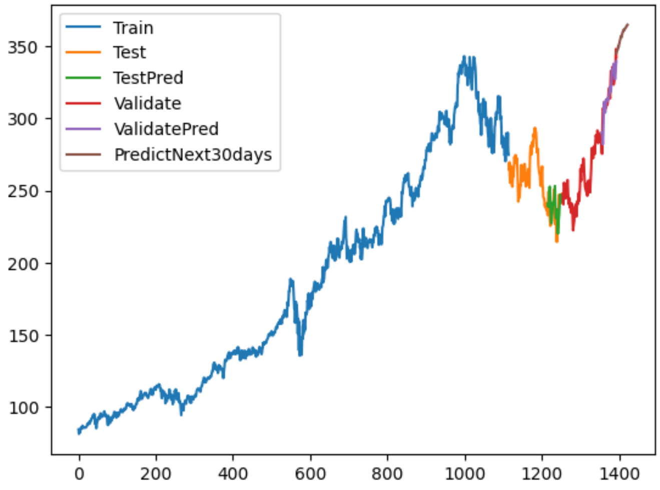


Figure 23. Result of Recurrent Neural Network (8:1:1)

6) Vector Autoregression (VAR)

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

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

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

where f_0 is a k-dimensional vector, and a_t is a sequence of serially uncorrelated random vectors with mean zero and covariance matrix Σ. [2]

Applying Vector Autoregression algorithm to the GOOGL dataset:

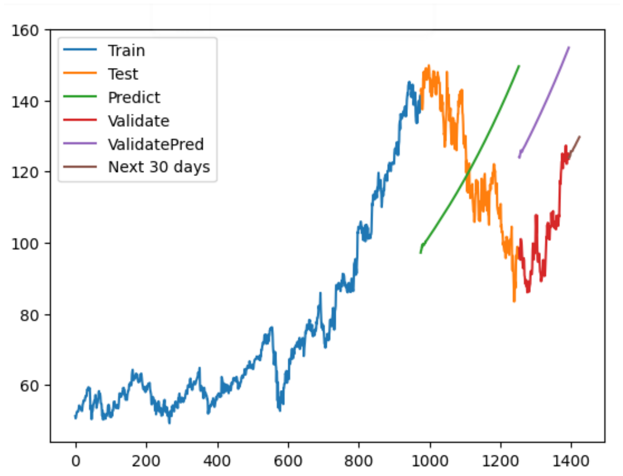


Figure 24. Result of VAR (7:2:1)

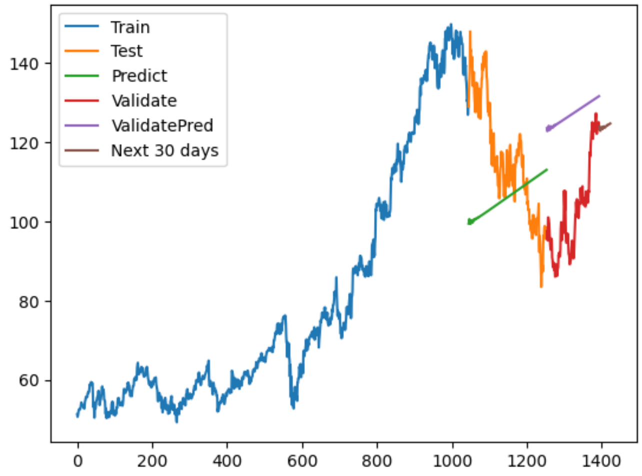


Figure 25. Result of VAR (7.5:1.5:1)

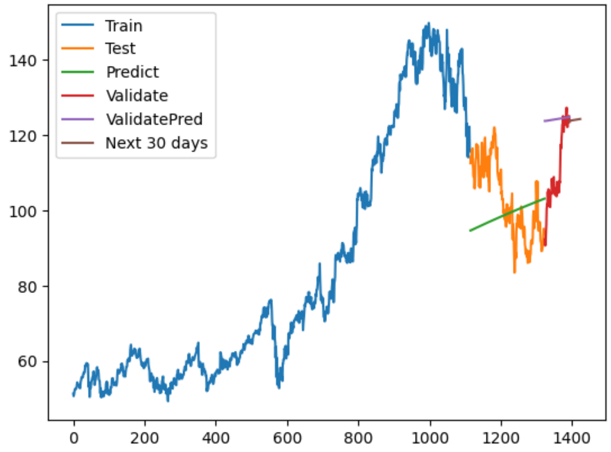


Figure 26. Result of VAR (8:1.5:0.5)

7) Singular Spectrum Analysis (SSA)

Singular spectrum analysis (SSA) is an integrated approach to time series analysis and forecasting that decomposes the original series into interpretable components using techniques from classical time series analysis, multivariate statistics, geometry, dynamical systems, and signal processing. It aims to extract trend, oscillatory components, and noise by performing singular value decomposition on a constructed matrix based on the time series data.

The strategy of basic SSA is to decompose the observed time series into a sum of independent components. The procedure of basic SSA is in two isolated steps: decomposition and reconstruction. Each consists of another two steps separately. The decomposition step is composed of time lagged embedding and singular value decomposition (SVD), while the reconstruction step is composed of diagonal averaging and grouping. A flowchart of basic SSA, consisting of the sub-steps of decomposition and reconstruction.

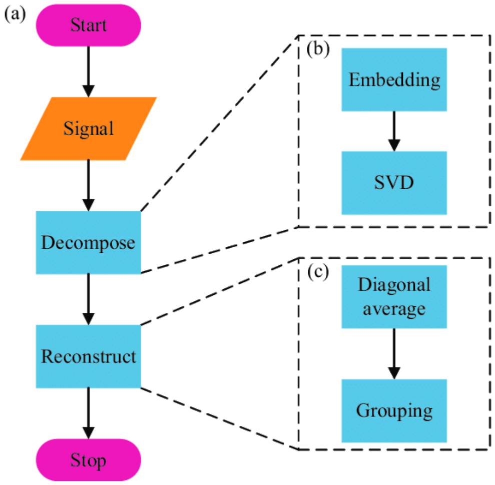


Figure 27. Flowchart of basic singular spectrum analysis (SSA). (a) Procedure of basic SSA; (b) sub-procedure: decomposition; (c) sub-procedure: reconstruction.

Applying Singular Spectrum Analysis algorithm to the MSFT dataset:

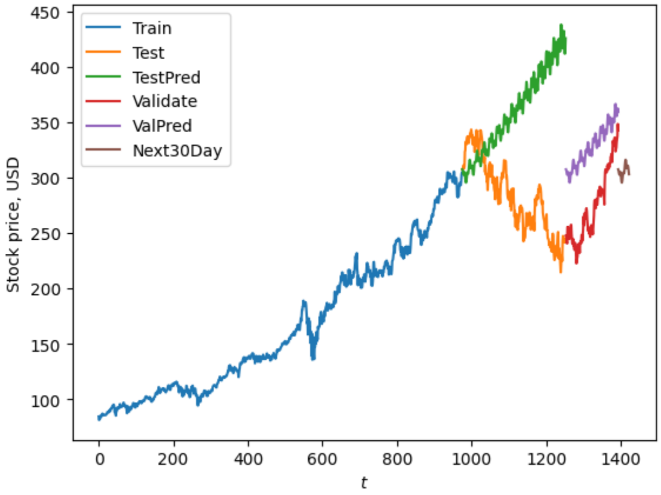


Figure 28. Result of Singular Spectrum Analysis (7:2:1)

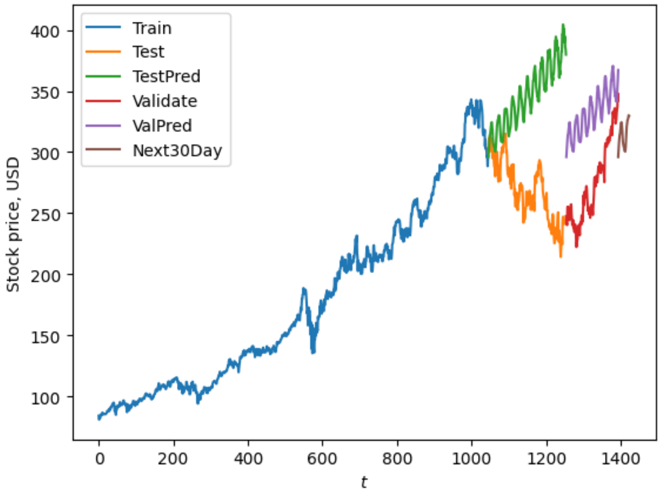


Figure 29. Result of Singular Spectrum Analysis (7.5:1.5:1)

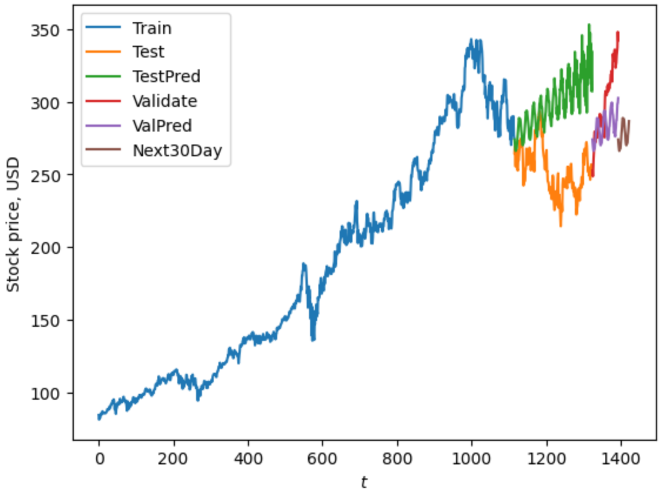


Figure 30. Result of Singular Spectrum Analysis (8:1.5:0.5)

8) Sequence to Sequence (Seq2Seq)

A Seq2Seq model is a model that takes a sequence of items (words, letters, time series, etc) and outputs another sequence of items. The model is composed of an encoder and a decoder.

The encoder captures the context of the input sequence in the form of a hidden state vector and sends it to the decoder, which then produces the output sequence.

Applying Seq2Seq algorithm to the MSFT dataset:

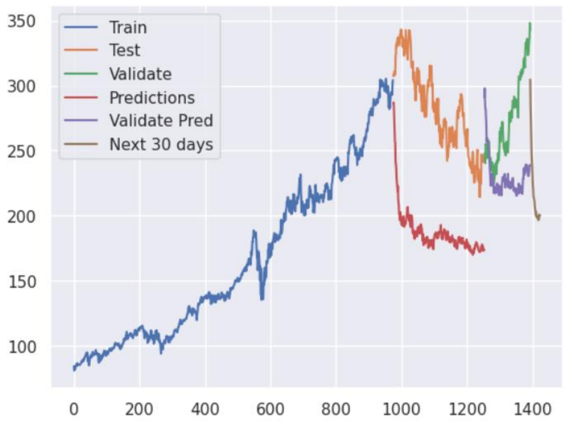


Figure 31. Result of Seq2seq (7:2:1)

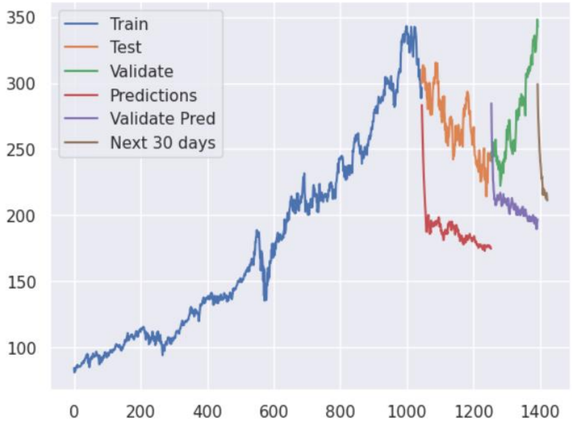


Figure 32. Result of Seq2seq (7.5:1.5:1)

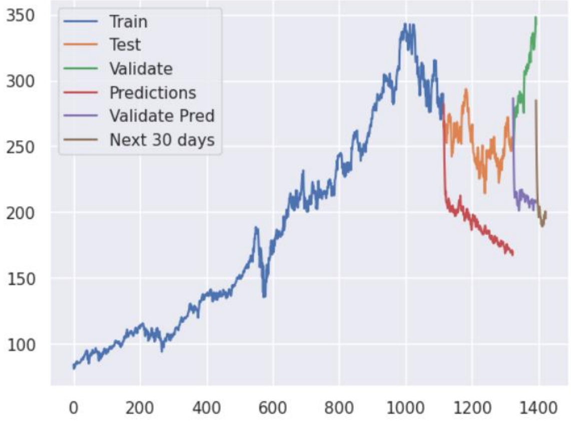


Figure 33. Result of Seq2seq (8:1.5:0.5)

9) *DeepAR*

DeepAR is the first successful model to combine Deep Learning with traditional Probabilistic Forecasting. *DeepAR* uses LSTM networks to create probabilistic outputs. Instead of using LSTMs to calculate predictions directly, *DeepAR* leverages LSTMs to parameterize a Gaussian likelihood function. That is, to estimate the $\theta = (\mu, \sigma)$ parameters (*mean* and *standard* deviation) of the Gaussian function.

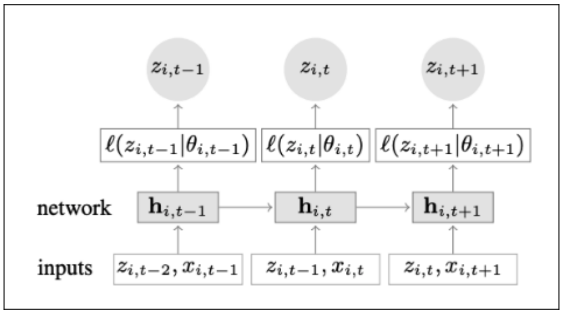


Figure 34. Architecture of DeepAR

Applying DeepAR algorithm to the MSFT dataset:

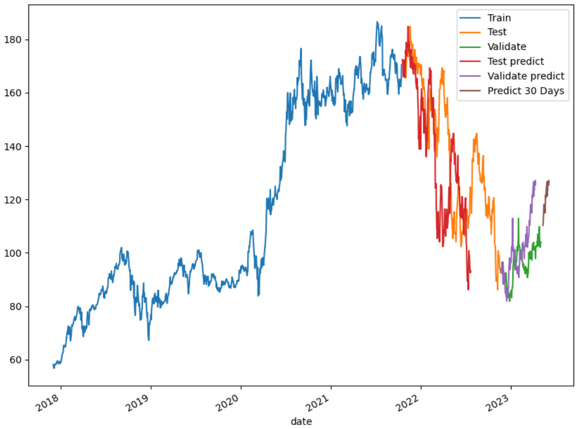


Figure 35. Result of DeepAR (7:2:1)

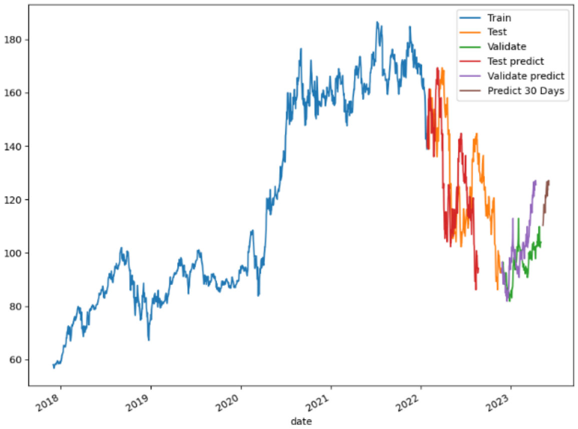


Figure 36. Result of DeepAR (7.5:1.5:1)

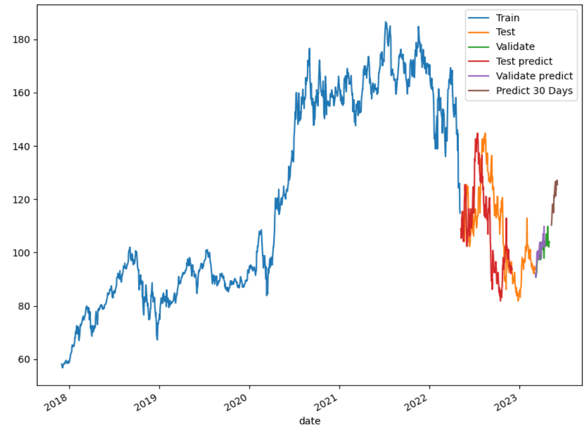


Figure 37. Result of DeepAR (8:1.5:0.5)

10)Vector Error Correction Model (VECM)

VECM is a kind of VAR model used with co-integration restrictions. VAR system was made according to empirical rules and statistic information. Lucas (1976)[1] and Sims (1980)[2], critics of traditional econometrics model, developed VAR model. In VAR model all of the variables are endogenous and similar to simultaneous equation. General form of VAR model: [11]

$$Yt = A_1.Y_{t-1} + A_2.Y_{t-2} + \dots + A_p.Y_{t-p} + U_t$$

- K: number of endogenous variables
- Y: vector of variables
- p: number of lags

VECM contains both long-run and short-run relations among variables set in vector Y. General form of VECM[3] is:

$\Delta Yt = B1.\Delta Yt - 1 + ... + Bp - 1.\Delta Yt - p + 1 + \Pi.Yt - p + Ut$

$\Pi = \alpha . \beta'$

- B_i is the matrix of parameters.
- Π contains long-run information.
- α is the matrix of error correction coefficients.
- The α parameters measure the speed at which the variables adjust to restore a long-run equilibrium.
- Matrix β is long-run coefficients.
- The error correction terms, β′Yt-1, are the mean reverting weighted sums of cointegrating vectors and data dated t-1.

Applying VECM algorithm to the GOOGL dataset:

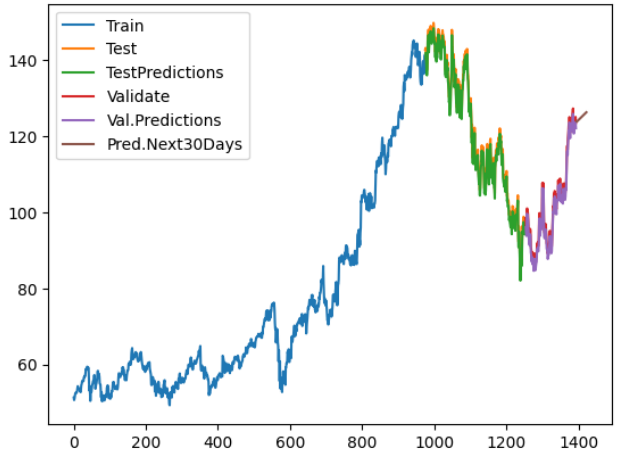


Figure 38. Result of VECM (7:2:1)

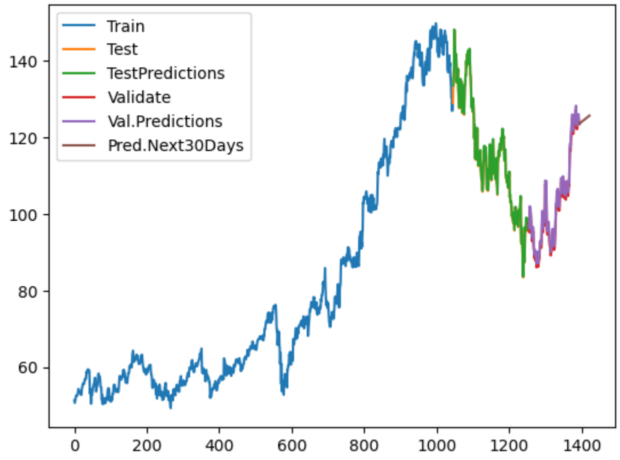


Figure 39. Result of VECM (7.5:1.5:1)

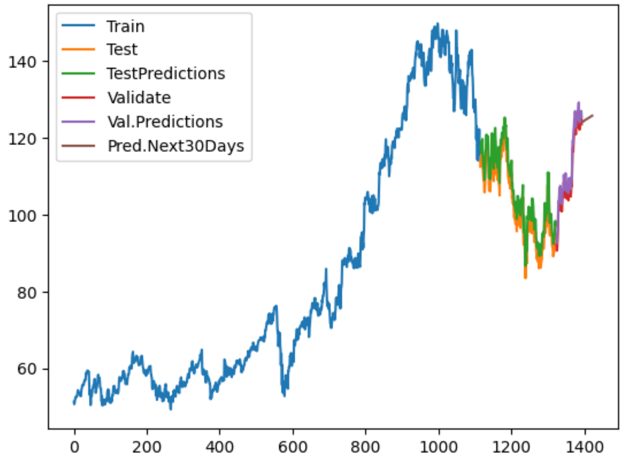


Figure 40. Result of VECM (8:1.5:0.5)

B. Evaluation

1) Root Mean Square Error (RMSE)

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

$$RMSE = \sqrt{(f-o)^2}$$ [12]

Where:

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

2) Mean Absolute Percentage Error (MAPE)

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

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$ [13]

Where:

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

3) Mean Absolute Error (MAE)

Mean 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:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n}.$$ [14]

Where:

- y_i is the actual value.
- y_p is the predicted value.
- n is the number of observations/ rows.

4) Mean Squared Error (MSE)

The mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$ [15]

Where:

- n: total number of data points
- Y_i = actual value
- $\hat{Y}_i$ = predict value

5) Evaluate the model on the test & validation set

With GOOGL Dataset:

Model	Train:Test: Val	RMSE	MAPE (%)	MAE (%)	MSE (%)
LSTM	7:2:1	116.047	191.551	115.382	13463.909
	7.5:1.5:1	105.236	192.121	105.236	11149.198
	8:1.5:0.5	102.528	205.826	102.072	10510.284
Linear	7:2:1	23,341	17,446	20,775	544,806
	7.5:1.5:1	26,112	20,943	22,250	681,853
	8:1.5:0.5	40,387	38,337	37,682	1631,070
VAR	7:2:1	31.852	23.664	28.157	1014.513
	7.5:1.5:1	20.472	13.651	16.442	419.111
	8:1.5:0.5	12.576	10.134	10.657	158.157
VECM	7:2:1	1,410	1,176	1,410	1,988
	7.5:1.5:1	0,228	0,201	0,228	0,052
	8:1.5:0.5	3,235	3,178	3,235	10,468

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

With MSFT Dataset:

Model	Train:Test: Val	RMSE	MAPE (%)	MAE (%)	MSE (%)
ARIMA	7:2:1	73.133	17.62	61.058	5348.436
	7.5:1.5:1	59.474	15.366	49.728	3537.203
	8:1.5:0.5	87.315	31.317	84.915	7623.945
RNN	7:2:1	273.343	40440.948	270.607	74716
	7.5:1.5:1	0.101	12.73	0.081	0.01
	8:1:1	237.36	41082.245	237.266	56339.705
SSA	7:2:1	103.512	32.943	85.405	10714.642
	7.5:1.5:1	90.916	30.872	78.327	8265.703
	8:1.5:0.5	54	18.96	46.34	2916.07
Seq2Seq	7:2:1	94.216	30.926	89.439	8876.677
	7.5:1.5:1	80.902	29.249	79.082	6545.263
	8:1.5:0.5	58.314	21.975	56.089	3400.6091

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

With AMZ Dataset:

Model	Train:Test:Val	RMSE	MAPE (%)	MAE (%)	MSE (%)
GRU	7:2:1	3.705	0.019	2.837	13.728
	7.5:1.5:1	3.461	0.017	2.516	11.979
	8:1.5:0.5	3.576	0.018	2.626	12.791
DeepAR	7:2:1	11.813	0.104	9.425	139.55
	7.5:1.5:1	9.227	0.0783	7.362	85.143
	8:1.5:0.5	4.618	0.034	3.685	21.328

Table 3. Evaluate models on the test set with AMZ Dataset

V. Conclusion

A. Challenges Encountered

During the implementation of the research project " Predicting Stock Performance of Technology Giants: A Scientific Research Study" we encountered several challenges, including:

- Data processing complexity: Stock data is intricate and diverse, necessitating the use of scientific and accurate methods to process the data. Our aim was to ensure the feasibility and accuracy of our prediction models.
- Building intricate prediction models: Constructing prediction models for the stock requires deep domain knowledge. We had to make crucial decisions, such as selecting appropriate algorithms, determining data processing techniques, and identifying significant variables to incorporate in the models.
- Evaluating model effectiveness difficulty: We employed various algorithmic and statistical indicators to evaluate the effectiveness of our prediction models. However, the results revealed that the accuracy of the models was still unsatisfactory.

In the future, we will address these challenges by implementing the following solutions to improve the prediction of stock prices:

- Enhancing data selection and processing skills: We will continue to research and apply state-of-the-art methods to select and process data. This approach will ensure the feasibility and accuracy of our prediction models.
- Utilizing advanced prediction models: We will explore and implement advanced prediction models such as Deep Learning and Reinforcement Learning. These models have the potential to enhance the effectiveness and accuracy of our predictions.
- Strengthening the evaluation of model effectiveness: We will conduct further research and incorporate widely accepted indicators in the field of stock price prediction, such as Mean Absolute Scaled Error (MASE), Mean Directional Accuracy (MDA), and Symmetric Mean Absolute Percentage Error (SMAPE). These indicators will provide a comprehensive evaluation of the effectiveness of our prediction models.
- Fostering cooperation and knowledge sharing: We will actively seek out and engage with communities and forums dedicated to stock price prediction.

By sharing experiences and learning from experts in the field, we can enhance our expertise and improve our models.

B. Conclusion

VI. Acknowledgment

We would like to express our heartfelt gratitude to Assoc. Prof. Dr. Nguyen Dinh Thuan and TA. Nguyen Minh Nhut for their invaluable expertise, enthusiastic guidance, and sincere assistance throughout the completion of this project. Without your passionate supervision, it would have been extremely challenging for our group to accomplish this report.

This project has provided us with an opportunity to collaborate, enhance our cooperative skills, learn from one another, and most importantly, apply what we have learned during the course to practical endeavors.

Throughout the project implementation, our team applied the knowledge imparted to us and ventured into new territories, aiming to achieve the highest level of excellence in our work. However, due to limited time, knowledge, and experience, we acknowledge that there may be shortcomings. Therefore, we eagerly await your valuable suggestions to help us supplement and improve our knowledge, enabling us to better serve future projects and real-world scenarios.

Lastly, we would like to wish you good health as you continue your noble mission of imparting knowledge to future generations.

VII. References

[1] C. Cai, Y. Tao, T. Zhu and Z. Deng, "Short-Term Load Forecasting Based on Deep Learning Bidirectional LSTM Neural Network," *Applied Sciences,* 2021.

[2] S. Bayraci, Y. Ari and Y. Yildirim, "A Vector Auto-Regressıve (VAR) Model for the Turkish Financial Markets," pp. 405-422, 2011.

[3] G. Zhang, X. B. (Ph.D.) and Y. Wang, "Short-time multi-energy load forecasting method based on CNN-Seq2Seq model with attention mechanism," *Machine Learning with Applications,* vol. 5, no. 100064, 2021.

[4] K. Cho, B. v. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* p. 1724–1734, 2014.

[5] H. Cao, Y. Song, Y. Li, R. Li, H. Shi, J. Yu and M. Hu, "Reduction of Moving Target Time-of-Flight Measurement Uncertainty in Femtosecond Laser Ranging by Singular Spectrum Analysis Based Filtering," *Applied Sciences,* vol. 8, no. 9, p. 1625, 2018.

[6] D. Salinas, V. Flunkert, J. Gasthaus and T. Januschowski, "DeepAR: Probabilistic Forecasting

with Autoregressive Recurrent Networks," *International Journal of Forecasting,* vol. 36, no. 3, pp. 1181-1191, 2020.

[7] CFI Team, "Autoregressive Integrated Moving Average (ARIMA) - Applications," Corporate Finance Institute, 28 December 2022. [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/autoregressive-integrated-moving-average-arima/. [Accessed 18 June 2023].

[8] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks,* vol. 18, no. 5-6, pp. 602-610, 2005.

[9] S. Varsamopoulos, K. Bertels and C. G. Almudever, "Decoding surface code with a distributed neural network–based decoder," *Quantum Machine Intelligence,* vol. 2, no. 3, 2020.

[10] JavatPoint Team, "Linear Regression in Machine Learning - Javatpoint," JavatPoint, [Online]. Available: https://www.javatpoint.com/linear-regression-in-machine-learning. [Accessed 17 June 2023].

[11] M. Sadeghi and S. Y. Alavi, "Modeling the impact of money on GDP and inflation in Iran: Vector-error-correction-model (VECM) approach," *African Journal,* vol. 7, no. 35, pp. 3423 - 3434, 2013.

[12] "RMSE: Root Mean Square Error,” Statistics How To.," [Online]. Available: www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error. [Accessed 17 June 2023].

[13] Oracle, "MAPE (Mean Absolute Percentage Error)," Oracle, [Online]. Available: https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusu/insights_metrics_MAPE.html. [Accessed 15 May 2023].

[14] "Mean absolute error - Wikipedia," Wikipedia, [Online]. Available: https://en.m.wikipedia.org/wiki/Mean_absolute_error?fbclid=IwAR2dUYFREckq_FFzasXPFLrOMjQx7ezdsBz67Mai2t-D7WuCPnYjIdz6KIQ. [Accessed 17 June 2023].

[15] "Mean squared error - Wikipedia," Wikipedia, [Online]. Available: https://en.m.wikipedia.org/wiki/Mean_squared_error?fbclid=IwAR2dUYFREckq_FFzasXPFLrOMjQx7ezdsBz67Mai2t-D7WuCPnYjIdz6KIQ. [Accessed 17 June 2023].

[16] C. A. Sims, "Comparison of Interwar and Postwar Business Cycles: Monetarism Reconsidered," *The American Economic Review,* vol. 70, no. 2, pp. 250-257, 1980.