Tribhuvan University

Institute of Science and Technology

Seminar Report

On

Short Term Stock Price Prediction using RNN with LSTM and GRU: A Comparative Study

Submitted to

Central Department of Computer Science & Information Technology

Tribhuvan University, Kirtipur

Kathmandu, Nepal

Submitted by
Himal Raj Gentil
Roll No.: 6 / 075

In partial fulfillment of the requirement for Master's Degree in Computer Science and Information Technology (M.Sc. CSIT), 2nd Semester

Tribhuvan University

Institute of Science and Technology

Supervisor's Recommendation

I hereby recommend that this Seminar report is prepared under my supervision by **Mr. Himal Raj Gentil** entitled "**Short Term Stock Price Prediction using RNN with LSTM and GRU: A Comparative Study**" be accepted as fulfilment in partial requirement for the degree of Master's of Science in Computer Science and Information Technology. In my best knowledge, this is an original work in computer science.

… … … … … … … … … … … …

Asst. Prof. Bikash Balami

Central Department of Computer Science

and Information Technology

Tribhuvan University
Institute of Science and Technology

LETTER OF APPROVAL

This is to certify that the seminar report prepared by **Mr. Himal Raj Gentil** entitled "**Short Term Stock Price Prediction using RNN with LSTM and GRU: A Comparative Study**" in partial fulfillment of the requirements for the degree of Master's of Science in Computer Science and Information technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Evaluation Committee

……………………………

Asst. Prof. Nawaraj Paudel
Head of Central Department of Computer Science & Information Technology

……………………………

Asst. Prof. Bikash Balami
(Supervisor)
Central Department of Computer Science & Information Technology

..…………………………………

(Internal)

Date:

Abstract

Stock market have its deep root in the economy of today’s era. The fluctuation of the share price has an important role in determining the investors gain. Short term price fluctuation contributes a considerable measure to unpredictability of the stock price changes. At present, prediction and analysis of stock price has got a good role in economic and financial sectors. Deep Neural Networks has outperformed a lots of linear models that were used in past for prediction and analysis of stock prices. As Deep Neural Network is evolving, a wide spectrum of its variants are used in prediction and analysis purposes. This study is based on Recurrent Neural Network variants i.e. LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit). A series of closing price of stock is used as the parameter in the prediction process. The performance analysis of these models are measured in RMSE for both testing and training purposes.

Keywords: Stock Price Prediction, Recurrent Neural Network, Long Short Term Memory, Gated Recurrent Unit

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ARIMA:	Autoregressive Integrated Moving Average
BGRU:	Bidirectional Gated Recurrent Unit
GCP:	Google Cloud Platform
GRU:	Gated Recurrent Unit
IBM:	International Business Machine
LSTM:	Long Short Term Memory
MSE:	Mean Squared Error
RMSE:	Root Mean Squared Error
RNN:	Recurrent Neural Network

1. Introduction

Stock market is a place where shares or stock of firm are traded. For many decades, stock market prediction has been considered as the interesting and very important study area [1]. But, due to its complexity and dynamicity, its proven to be a difficult task [2][3]. In investment and finance predicting the trend and price of stock market are considered as the indispensable aspects. Many researchers have worked and proposed their ideas to forecast the market price to make profit while trading using various techniques such as technical and statistical analysis.

Because of many noises and uncertainties involved, observing and predicting trends of the stock market is challenging. A lot of factors influences the market value in a day such as country's economic change, product value, investor's sentiments, weather, political affairs, etc. [4].

In this seminar project, two variants of RNN (Recurrent Neural Network) i.e. LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit) are used to predict short term stock price prediction. The stock market is mainly affected due to the sentiments of customers or buyers that is their opinion on a particular product or service provided by company is also one of the main additions to the fluctuations in stock prices. The dataset used for this study is stock price data of Apple Inc, Amazon.com Inc. and Ford Motors Company. The dataset is fetched though pandas data reader sub package which allows one to create various data frame from various data sources such as Google Finance, Yahoo Finance and many more [5].

2. Background Study

2.1 RNN

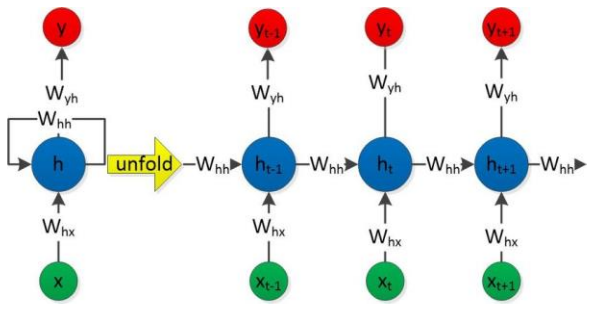


Fig1: Sample RNN structure (Left) and its unfolded representation (Right) [6]

Recurrent Neural Networks(RNN) are the class of neural networks where the units are recurrently connected. This allows them to use their internal memory for processing the sequence of inputs. RNN take input from two sources, one from present and the other from the past. Information from these two sources are used to decide how they react to the new set of data. This is done with the feedback loop where the output at each instant is an input to the next moment. Here, it can be said that the RNN has memory. Each sequence has a plenty of information and this information are stored in the hidden state of neural networks [7]. This hidden information is recursively used in the networks as it sweeps forward to deal with new example.

Given an input time series $X = \{x_1, x_2, \ldots., x_T\}$, the RNN computes the hidden state sequence $h = \{h_1, h_2 \ldots., h_T\}$ as well as the output sequence $y = \{y_1, y_2 \ldots., y_T\}$ iteratively using the following set of equations :

$$h_t = g_n(W_{hx}X_t + W_{hh}h_{t-1} + b_h) \qquad [8]$$

where h_t : hidden layer at t^{th}instant , g_n : activation function , W_{hx} : input to hidden layer weight matrix, W_{hh}: hidden to hidden layer weight matrix, X_t : input at t^{th} instant, h_{t-1} :hidden layer at $t-1^{th}$ instant , b_h :bias or threshold value.

$$y_t = g_n(W_{yh}h_t + b_y) \qquad [8]$$

whereas Y_t :output vector, W_{yh} :hidden to output layer weight matrix, b_y :bias or threshold.

$Sigmoid$, $tanh$ and $ReLU$ are all the most frequently used activation functions in various kinds of neural (deep) networks.

$$sigmod(x) = \frac{1}{1 + e^{-x}}$$

$$tanh\,(x) \; = \; 2sigmoid(2x) - 1$$

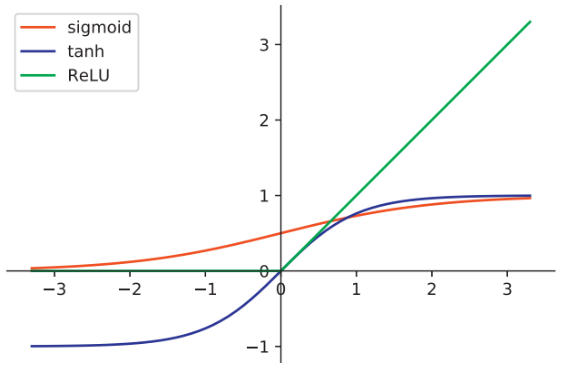


Fig 2: Activation Functions [9]

Since, $sigmoid$ and $tanh$ have similar function curves except the difference between their ranges: the former is $(0,1)$ and the latter is $(-1,1)$. They are generally used to squash the nodes lying the hidden layers near the input layer in order to extract the non-linear features from the raw data. Therefore, they are the candidates of the activation functions from the input layer to the RNN layer and from the RNN layer to the fully connected layers. However, one main drawback of $Sigmoid$ and $tanh$ is that they can cause vanishing or exploding gradients when the gradient-based algorithms are used to train the networks. To overcome this drawback, the function $ReLU = max\{0, x\}$ is applied to squash the nodes in the output layers and the hidden layers near the output layer. When $x \; > \; 0$, ReLU acts as a linear function in order to maintain the gradient stability and when $x \; \leq \; 0$, it eliminates the node output to carry on the non-linear features provided by the nodes in the hidden layers near the input layer [9].

After the RNN outputs the prediction vector y_t, the prediction error E(k) is computed and use the Back Propagation Through Time (BPTT) algorithm to compute the gradient.

$$\frac{\partial E}{\partial W} = \sum\nolimits_{t=1}^{T} \frac{\partial E_t}{\partial W}$$

The gradient is used to update the model parameter by:

$$W \leftarrow W - \alpha \frac{\partial E}{\partial W}$$

The learning process is continued using the Gradient Descent algorithm. On the learning time that included T time steps, the gradient of the error on the k-timestep is given by :

$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial y_k}\frac{\partial y_k}{\partial h_k} \dots\dots \frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W}$$

$$= \frac{\partial E_k}{\partial y_k}\frac{\partial y_k}{\partial h_k}\left(\prod_{t=2}^{k} \frac{\partial h_t}{\partial h_{t-1}}\right)\frac{\partial h_1}{\partial W} \dots\dots (1)$$

Since, $h_t = g_n(W_{hx}X_t + W_{hh}h_{t-1} + b_h)$

Derivative of h_t is given by:

$$\frac{\partial h_t}{\partial h_{t-1}} = g_n{}'(W_{hx}X_t + W_{hh}h_{t-1})\frac{\partial}{\partial h_{t-1}}(W_{hx}X_t + W_{hh}h_{t-1})$$

$$= g_n{}'(W_{hx}X_t + W_{hh}h_{t-1})W_{hh} \dots\dots (2)$$

From equation (1) and (2):

$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial y_k}\frac{\partial y_k}{\partial h_k}\left(\prod_{t=2}^{k} g_n{}'(W_{hx}X_t + W_{hh}h_{t-1})W_{hh}\right)\frac{\partial h_1}{\partial W} \qquad [10]$$

The last expression i.e. W_{hh} tends to vanish when k is large, this is due to the derivative of the activation function i.e sigmoid or tanh , which is smaller than 1. The product of derivatives can also explode if the weights W_{hh} are large enough to overpower the smaller tanh derivative, this is known as the exploding gradient problem [10].

2.2 LSTM

LSTM is a special type of RNN. LSTM was mainly motivated and designed to overcome the vanishing or exploding gradients problem of the standard RNN when dealing with long term dependencies. It was introduced by Hochreiter and Schmidhuber in 1997 [11]. These networks are clearly designed to evade the long term dependency problem, but remembering information for a long time period back is their normal behavior. LSTM have a different structure compared to other neural networks. Conventional RNN has a very simple neural network with a feedback loop but LSTM consists of a memory block or cells instead of a single neural network layer. Each cell or block has 3 gates which are input gate, forget gate and output gate and a cell state tend to regulate the flow of data information through the cells.

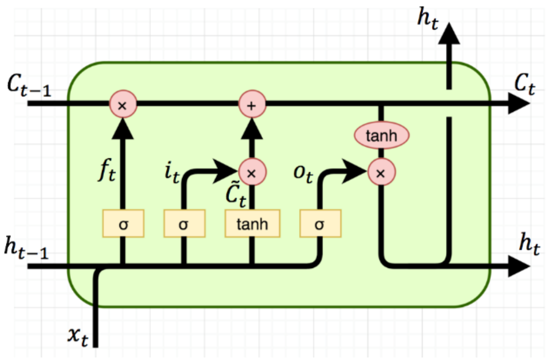


Fig 3: LSTM Cell [7]

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

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

$$\tilde{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_c)$$

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

$$O_t = \sigma(W_0.[h_{t-1}, x_t] + b_0)$$

$$h_t = O_t * \tanh(C_t)$$

Here C_{t-1} : *old cells state*, C_t : *present cell state*, h_{t-1}: *output of previous cell*, h_t : *Output of present cell*, i_t : *Input gate layer*, f_t : *forget gate layer*, O_t : *output sigmoid gate layer*[7].

In the above figure, horizontal line passing through the top of the diagram is known as cell state (C_{t-1}, C_t). It acts like a conveyor belt that runs over the entire network. It carries the information from the previous cell to the present and so on. The decision for storing information in cell state is taken by forget gate layer (f_t) which is also known as sigmoid layer. The output from forget gate is added to cell state using a point-wise multiplication operation. Next is input gate which comprises of a sigmoid layer (i_t) and $tanh$ layer. Input gate combines these two into the cell state.

Here, $\tilde{C}_t$ are the new values created by $tanh$ layer. Output (h_t) is formed by a point-wise multiplication of sigmoid gate O_t and $tanh$.

2.3 GRU

A Gated Recurrent Unit, or GRU, is a type of recurrent neural network. It is similar to an LSTM, but only has two gates - a reset gate and an update gate - and notably lacks an output gate. Fewer parameters mean GRUs are generally easier and faster to train than their LSTM counterparts. It largely mitigates the problem of vanishing gradient of RNNs through the gating mechanism and make the structure simpler while maintaining the effect of LSTM [12].

The gated recurrent unit is a special case of LSTM proposed by Kyunghyun Cho in 2014 [13]. It has shorter training time than traditional LSTM and has fewer parameters than LSTM, as they lack an output gate.

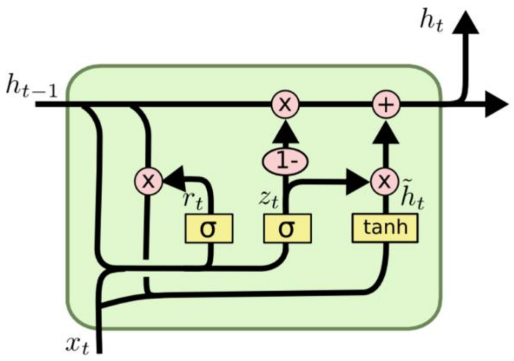


Fig 4: GRU Cell [14]

Here, update gate is introduced, to decide whether to pass Previous Output (h_{t-1}) to next cell as (h_t) or not. There are two input features at each time, which include the input vector x(t) and previous output vector (h_{t-1}). The output of each gate can be obtained through logical operation and nonlinear transformation of input.

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

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

$$\tilde{h}_t = tanh(W_c.[h_{t-1}, x_t] + b_h)$$

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

Here h_{t-1}: *output of previous cell,* h_t : *Output of present cell,* z_t : *update sigmoid gate layer,* r_t : *reset sigmoid gate layer* [14].

In the above figure, both update gate (z_t) and reset gate (r_t) are related to input sequence (x_t) and output value of memory cell from previous time point. The reset gate (r_t) is used to control the influence of h_{t-1} i.e. information of the unit at previous time step on the current information (x_t). If h_{t-1} is not important to x_t, then reset gate (r_t) is activated, making h_{t-1} does not affect x_t. The update gate (z_t) specifies whether to ignore the current information (x_t).

2.4 Sliding Window Method:

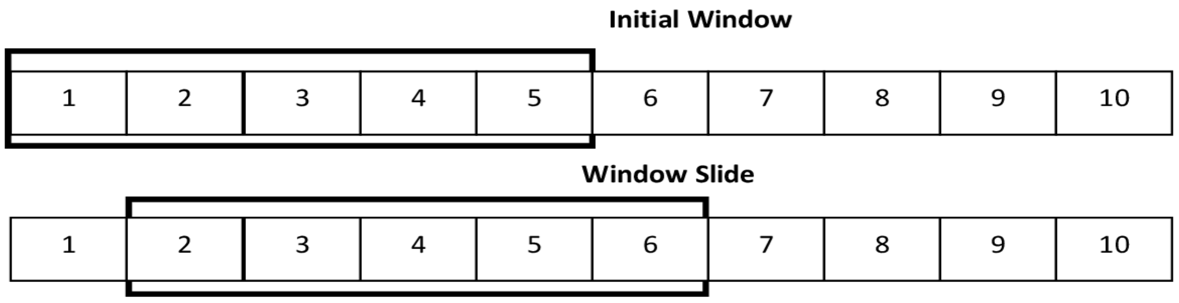


Fig 5: Sliding Window [35]

Sliding window is the way to restructure a time series dataset as a supervised learning problem. The sliding window method converts the sequential supervised learning problem into the classical supervised learning problem. It constructs a window model h_w , that maps an input window of width w into an individual output value y. Specifically, let $d = (w-1)/2$ be the "half-width" of the window. Then h_w predicts $y_{i,t}$ using the window $(x_{i,t-d}, x_{i,t-d+1}, \dots \dots . x_{i,t}, \dots \dots x_{i,t-d-1}, x_{i,t+d})$ [31].

The window classifier h_w is trained by converting each sequential training example (x_i, y_i) into windows and then applying a standard supervised learning algorithm. A new sequence x is modeled by converting it to windows, applying h_w to predict each y_t and then concatenating the y_t's to form the predicted sequence y.

3. Literature Review

During the pre-deep learning era, Financial Time Series modelling has mainly concentrated in the field of linear models. Linear models refer to the models with fixed structure based on some assumptions and that parameter can be computed with empirical data. The most commonly used linear model, ARIMA model was proposed by Box and Jenkins [15] in 1976. These models are based on the assumption of stationary variance and mean of the time series.

Linear models have several merits. First of all, they are usually simple models and explicit to understand. Secondly, the solution of them is usually easier than non-linear models and takes less time. These models use some predefined equations to fit a mathematical model to univariate time series. The main disadvantage of these models is that, they do not account the latent dynamics existing in the data. Since, they consider only univariate time series, the interdependencies among the various features of stocks are not identified by these models. Due to this reasons, it is not possible to identify the patterns or dynamics present in the data as whole [16][17][18].

In the case of stock market, the data generated is enormous and is highly non-linear. To model such kind of dynamical data, models that can analyze the hidden patterns and underlying dynamics are needed. Deep learning models are capable of identifying and exploiting the interactions and patterns existing in the data through a self-learning process. Unlike other models, deep learning models can effectively model these type of data and can have a good prediction by analyzing the interactions and hidden patterns within the data. The first attempt to model a financial time series using a neural network model was introduced in [19]. This work made an attempt to model a neural network model for decoding the non-linear regularities in asset price movement for IBM. The applications of Deep Learning in different financial domains was explained by J. B. Heaton and his colleagues [20]. Their study discussed a few prediction problems in the financial domain. It also stated a few advantages deep learning predictors have over traditional predictors. A few of them being, over fitting can be easily avoided and correlation in input data can also be handled easily.

In 1996, J. Roman and A. Jameel used back propagation and RNN models for the prediction of stock index for five different stock markets [21]. LSTMs had been conventionally proven successful for time series prediction. Hengjian Jia found that LSTMs learn patterns effective for stock market prediction and he obtained decent RMSEs with different architectures of LSTM [22].

This study helped in realization of this problem as a time-series problem, and gave an insight to solve this problem with a sliding window approach. H.B. Hashemi with his colleagues, in their research work stated that MLPs perform superior to LSTMs at prediction stock prices [23]. Their study focused on inter day price prediction. A. Dutta with his team have applied GRU for the bitcoin price prediction [24]. They have obtained quite promising results with train RMSE of 0.011 and Test RMSE of 0.017. RNN with Gated Recurrent Units (GRU) have been applied for stock price movement prediction in some recent works by D. Duong with his colleagues [25] and have shown promising success. In [25], Bidirectional Gated Recurrent Unit (BGRU) have been used to predict stock price movements, and their model achieved accuracy of nearly 60% in S&P500 index prediction and the individual stock prediction is over 65%.

From the motivation of success of deep RNN with LSTM and GRU units in different areas to model sequence data, in this study, it is attempted to experiment with RNN with LSTM and GRUs to predict future stock price based on previous days data. The proposed method focuses on predicting stock price for Apple Inc., Amazon.com Inc and Ford Motors Company. The approach adopted is a sliding window approach with data overlap. The data set contains daily stock price data of Apple Inc., Amazon Inc. and Ford Motors Company. Here, it is tried to obtain a generalized model of RNN with LSTM and GRU for the purpose of prediction which can use daily stock data as input and comparing the performances of both models.

4. Methodology

In machine learning, most of the times it has been found that the simplest of algorithms give astounding results than the complex algorithms. It has been adopted this line of thinking and for the same reason. So, this study will be based on RNN with LSTM and GRU. The two models would be trained on the same data and would be made to predict short-term stock prices and their results are compared and analyzed.

4.1 Implementation Details

Our network is implemented in python platform using Keras library [26]. LSTM and GRU networks are by default implemented in Keras, which help us a lot to customize the network. Complete execution of model training and text stock price prediction was done with the help of deep learning technology and code implementation on Keras. Implementation process of the full project in detail can be viewed from following figure:

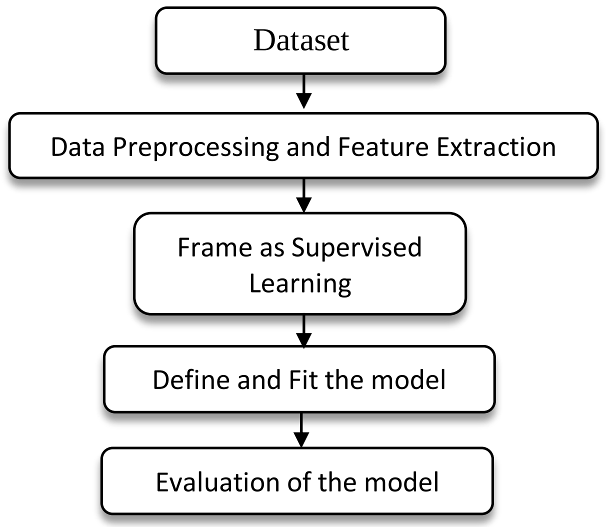

Dataset
Data Preprocessing and Feature Extraction
Frame as Supervised Learning
Define and Fit the model
Evaluation of the model

Fig 6: Methodology

4.2 Dataset

Dataset is taken from highly traded stocks of three different sectors which are Tech, E-commerce and Manufacturing sector. The corresponding stocks from these sectors are Apple Inc., Amazon.com Inc., and Ford Motors Coompany. Each contains information like trading day date, Open, High, Low, Close, Adj.Close, and Volume. The data sets contain the data from 2012-Jan-03 to 2020-Nov-24. The data sets were made available from yahoo finance api [28] and fetched using pandas datareader python library [29].

4.3 Data Preprocessing and Feature Selection

The data was normalized into the range of [0, 1] using a python library MinMaxScaler[30], since neural networks are known to be sensitive to non-normalized data.

$$X_{std} = \frac{X_{std} - X_{min}}{X_{max} - X_{min}}$$

$$X_{scaled} = X_{std} * (max - min) + min$$

$$Where\ min = 0\ and\ max\ = 1\ [32].$$

In this study, feature selection process, followed a heuristic method of study and analysis, which involved a lot of review of the related work as well as a lot of understanding of what stock market analysts use in real life. Choosing appropriate features is a very crucial factor in any prediction model. The correlation between the target output and the features were tested for finding which features resemble most with the target output. The correlation within the features were also tested to find the redundant features. Pearson Correlation Coefficient was used for finding the correlation between data.

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Where, n is the sample size ,x_i, y_i are individual sample points indexed with i, $\bar{x}$ is the sample mean [33].

Before performing the correlation test, the data frame was modified so that one extra column Close_Target can be added which is the target output. The correlation test was performed using the method of python library called pandas datareader [29]. The input features High, Low, Open, Close, Adj Close are highly correlated within themselves and with the target output i.e. Close_Target. But the Volume is negatively correlated with Close_Target and other input features for Apple Inc. and Ford Motors Company whereas uncorrelated for Amazon.com Inc. Input features are highly correlated within themselves indicates that all input features possess the similar characteristics, which also indicates redundant data features. So, closing price of the stock i.e. Close was selected, since it is highly correlated with Close_Target for all three datasets. Volume was selected for Apple Inc. and Ford Motors Company, since it is negatively correlated with Close_Target because negatively correlated data also possess some relation between them. Volume was not selected for Amazon.com Inc., since it is uncorrelated with the Close_Target.

4.4 Framing as Supervised Learning

The re-framing of your time series data as a supervised learning problem allows us access to the suite of standard linear and nonlinear machine learning algorithms. The dataset is split into training and testing set. For training set first 80% of the dataset is allocated and for testing rest 20% of the dataset is chosen. Multivariate and single-step forecasting time series can also be framed as supervised learning using the sliding window method. In this study, dataset was framed as per the sliding window method and single step forecasting models of RNN with LSTM and GRU are adopted for prediction. In single step forecasting historical observations (t-1, t-2, … t-n) forecast t. Window size of 60 is taken for the sliding window.

4.5 Model Architecture and Fitting the model

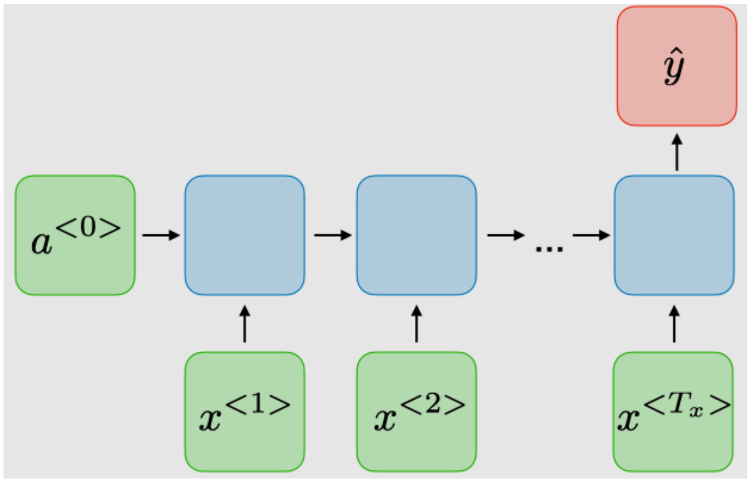


Fig 7: Many to one architecture of RNN [34]

The model architecture for both LSTM and GRU are same. Many to one RNN architecture is used for both models LSTM and GRU. Both models have two LSTM/GRU layers with 50 neurons and two Dense layers, one with 25 neurons and the other with 1 neuron. Moreover, Adam optimizer [27] is used, learning rate is set to 0.001. Appropriate epochs are used to train the models on Google Colab, running over Google Cloud Platform (GCP) to obtain the best fit for each model. The MSE (Mean Square Error) is used for error computation during training for each epochs and RMSE (Root Mean Square Error) is used for error computation during the testing phase.

4.6 Results and Analysis

Company Name	**LSTM**		**GRU**	
	Epoch	**RMSE**	**Epoch**	**RMSE**
Apple Inc.	30	4.1033274	4	2.0598261
Amazon.com Inc.	5	57.8884384	2	53.8340705
Ford Motors Company	25	0.2257657	4	0.2323217
Average	20	20.7391771	3.333	18.70873943

Table 1: Results and Analysis

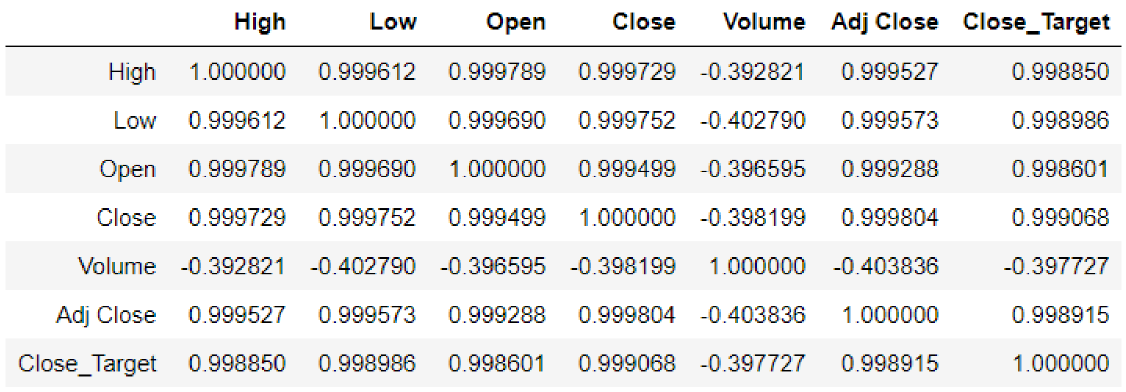
In this study, both models LSTM and RNN were trained, until the best fit within the proposed RNN architecture was found. The performances of the models are measured in number of epochs used during training phase and RMSE for testing phase. The average epochs used for training LSTM was 20 and average RMSE for LSTM during prediction was 20.7391771 whereas the average epochs used for training was 3.333 and average RMSE for GRU during prediction was 18.70873943. From the above result, it can be analyzed that GRU performs better than LSTM for short term stock price prediction, since it takes less epochs during training phase and also has less RMSE than LSTM during the testing phase.

5. Conclusion

This study focuses on comparative analysis of RNN with LSTM and GRU for short term stock prediction. The performances of both LSTM and GRU models are compared and analyzed for three different sector's stock data. From comparison of both models, it is found that GRU outperforms LSTM during both training and testing phase. The GRU model takes 71.43 % less epochs during training than LSTM, can reach global optima much faster to find best fit for datasets and also performs 5.147 % more accurate prediction than LSTM.

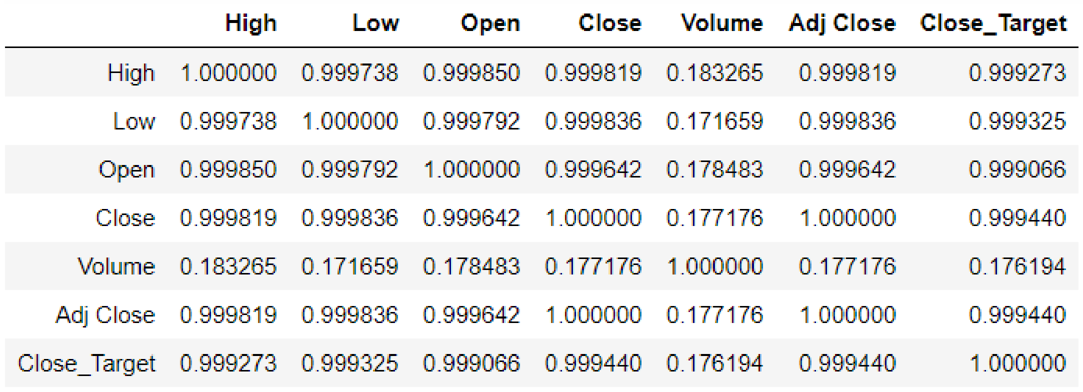
In this study, the datasets taken were of time period of almost 8 years only which is considered as quite short dataset for time series prediction. The further work comprises of comparing and analyzing the performances of LSTM and GRU in quite long ranges of dataset than the range of dataset used in this study. It also remains to be seen whether performances of both models LSTM and GRU can be enhanced by using more hidden layers.

6. Appendix



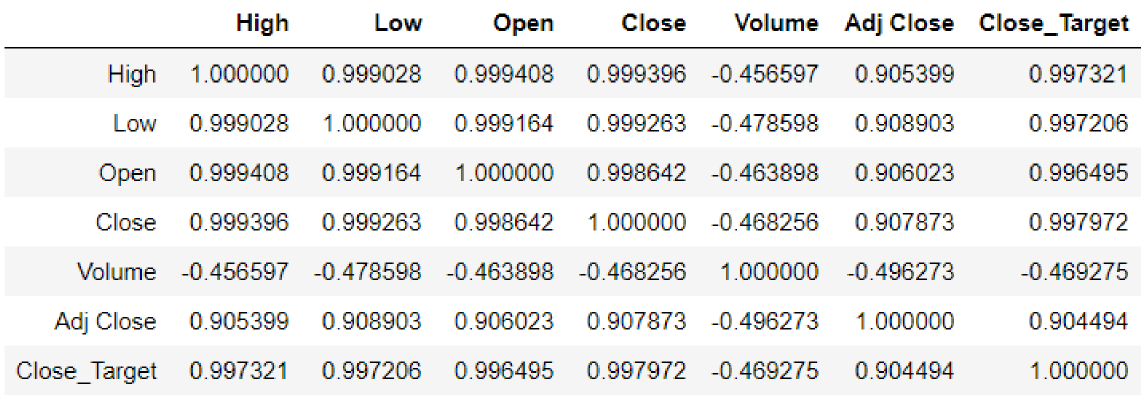
	High	Low	Open	Close	Volume	Adj Close	Close_Target
High	1.000000	0.999612	0.999789	0.999729	-0.392821	0.999527	0.998850
Low	0.999612	1.000000	0.999690	0.999752	-0.402790	0.999573	0.998986
Open	0.999789	0.999690	1.000000	0.999499	-0.396595	0.999288	0.998601
Close	0.999729	0.999752	0.999499	1.000000	-0.398199	0.999804	0.999068
Volume	-0.392821	-0.402790	-0.396595	-0.398199	1.000000	-0.403836	-0.397727
Adj Close	0.999527	0.999573	0.999288	0.999804	-0.403836	1.000000	0.998915
Close_Target	0.998850	0.998986	0.998601	0.999068	-0.397727	0.998915	1.000000

Table 2: Pearson Correlation Coefficient for Apple Inc. Dataset



	High	Low	Open	Close	Volume	Adj Close	Close_Target
High	1.000000	0.999738	0.999850	0.999819	0.183265	0.999819	0.999273
Low	0.999738	1.000000	0.999792	0.999836	0.171659	0.999836	0.999325
Open	0.999850	0.999792	1.000000	0.999642	0.178483	0.999642	0.999066
Close	0.999819	0.999836	0.999642	1.000000	0.177176	1.000000	0.999440
Volume	0.183265	0.171659	0.178483	0.177176	1.000000	0.177176	0.176194
Adj Close	0.999819	0.999836	0.999642	1.000000	0.177176	1.000000	0.999440
Close_Target	0.999273	0.999325	0.999066	0.999440	0.176194	0.999440	1.000000

Table 3: Pearson Correlation Coefficient for Amazon.com Inc. Dataset



	High	Low	Open	Close	Volume	Adj Close	Close_Target
High	1.000000	0.999028	0.999408	0.999396	-0.456597	0.905399	0.997321
Low	0.999028	1.000000	0.999164	0.999263	-0.478598	0.908903	0.997206
Open	0.999408	0.999164	1.000000	0.998642	-0.463898	0.906023	0.996495
Close	0.999396	0.999263	0.998642	1.000000	-0.468256	0.907873	0.997972
Volume	-0.456597	-0.478598	-0.463898	-0.468256	1.000000	-0.496273	-0.469275
Adj Close	0.905399	0.908903	0.906023	0.907873	-0.496273	1.000000	0.904494
Close_Target	0.997321	0.997206	0.996495	0.997972	-0.469275	0.904494	1.000000

Table 4: Pearson Correlation Coefficient for Ford Motors Company Dataset

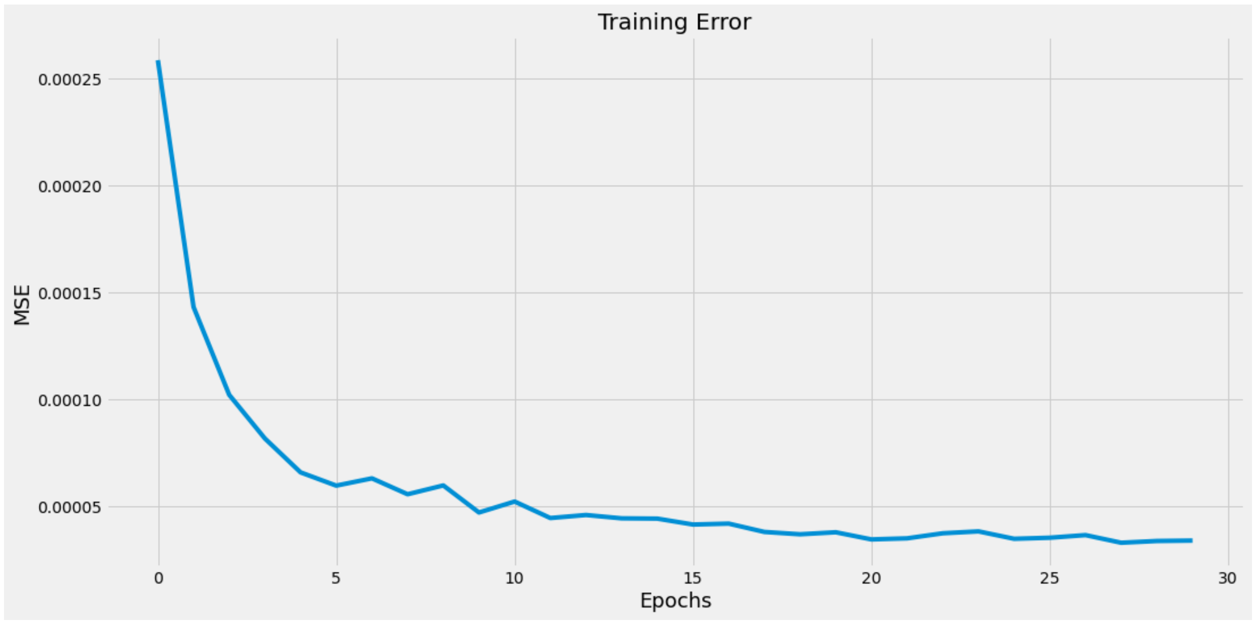


Fig 8: Training Error for LSTM (Apple Inc. Dataset)

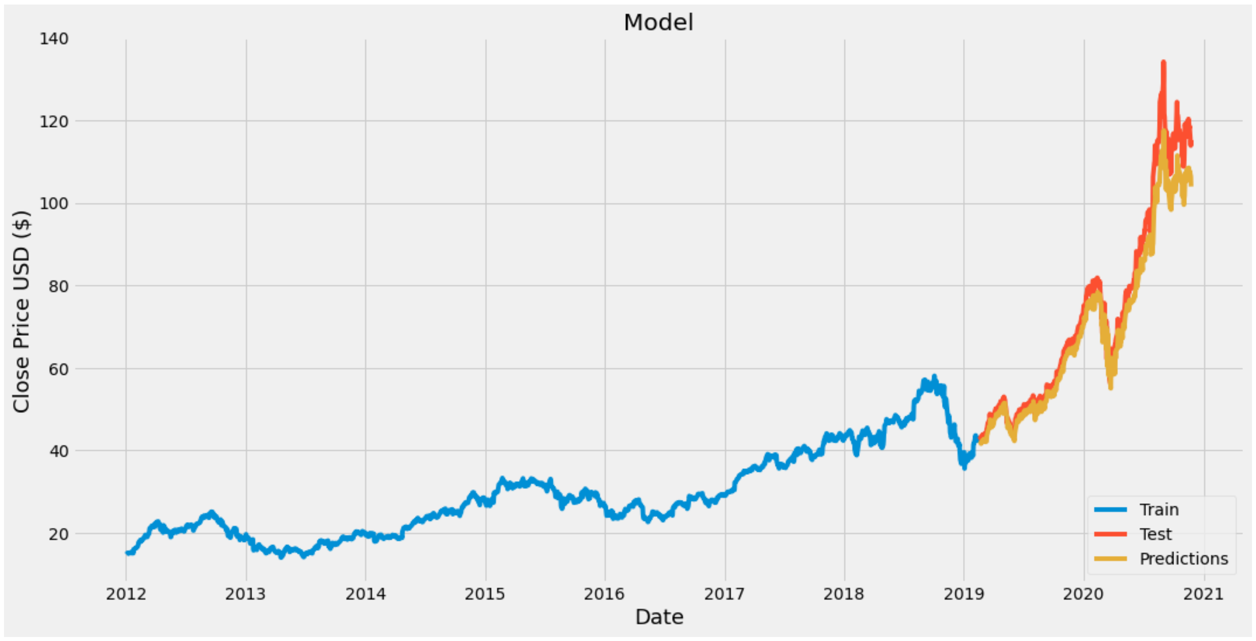


Fig 9: Prediction using LSTM model (Apple Inc. Dataset)

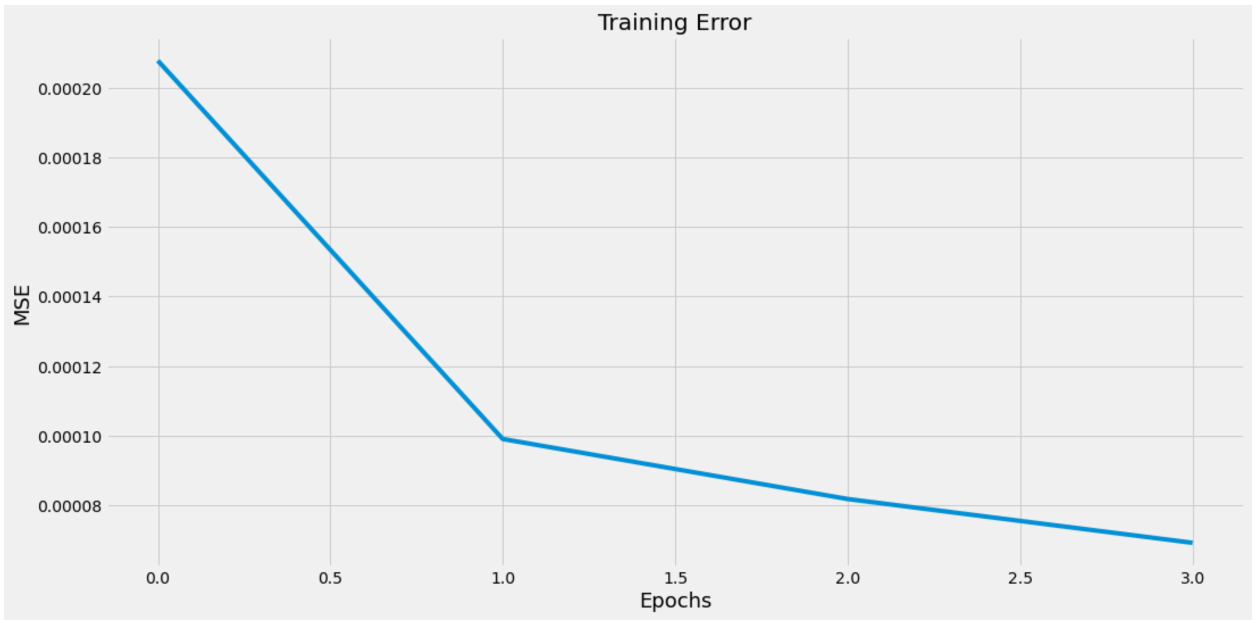


Fig 10: Training Error for GRU (Apple Inc. Dataset)

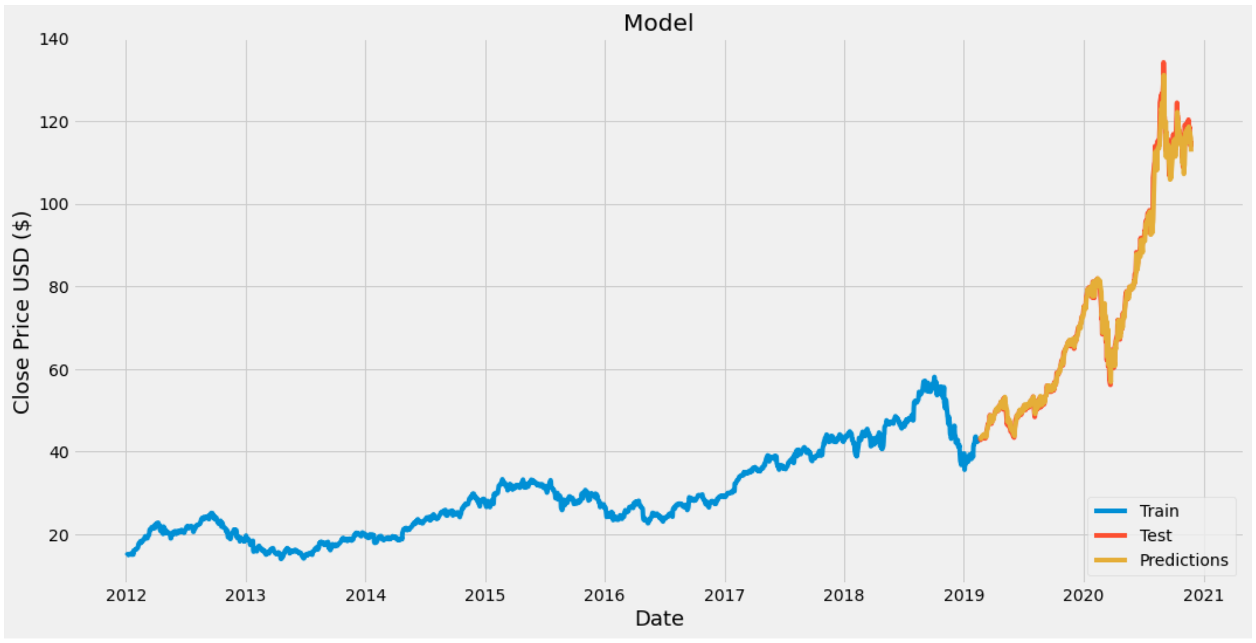


Fig 11: Prediction using GRU model (Apple Inc. Dataset)

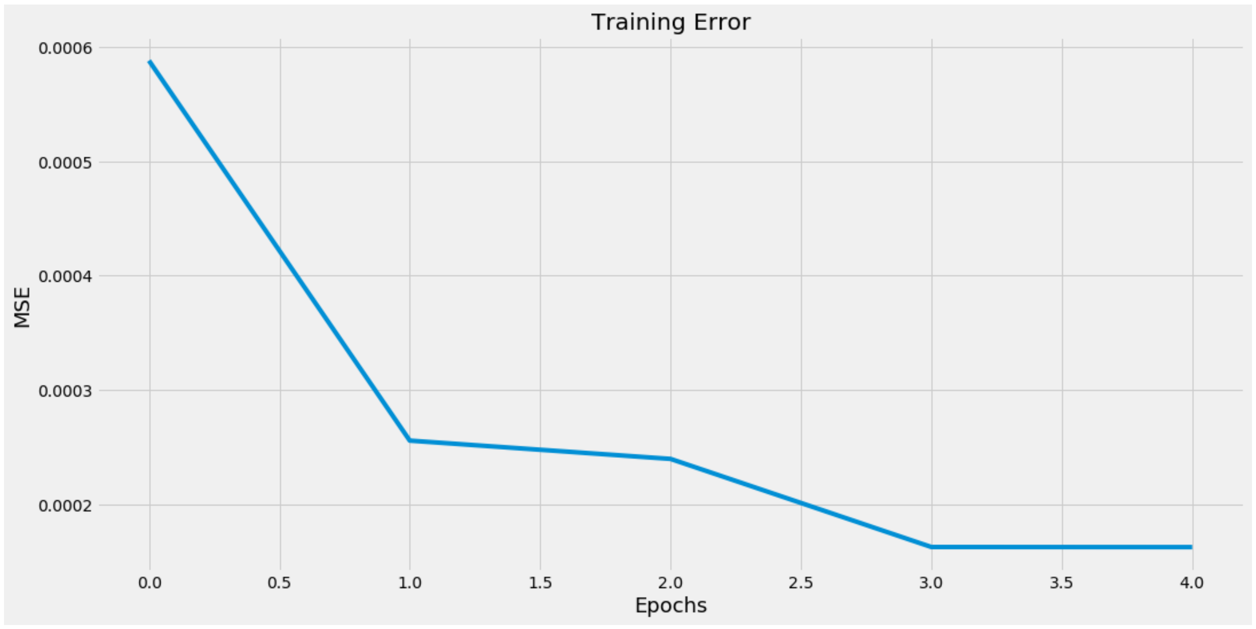


Fig 12: Training Error for LSTM (Amazon.com Inc. Dataset)

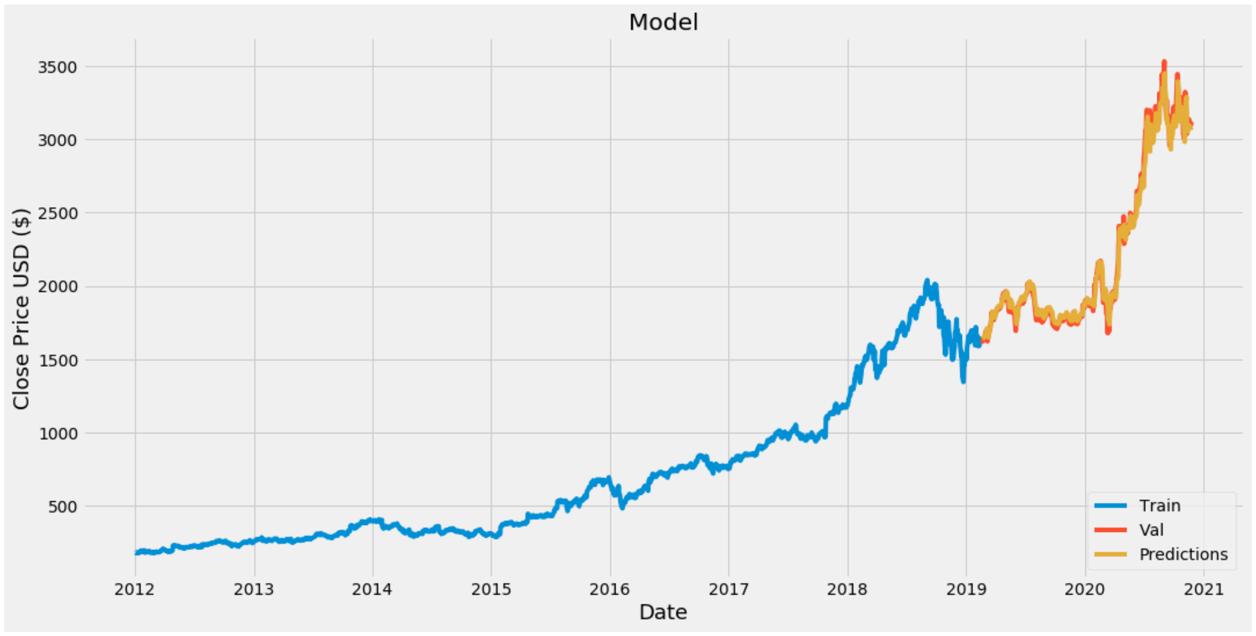


Fig 13: Prediction using LSTM model (Amazon.com Inc. Dataset)

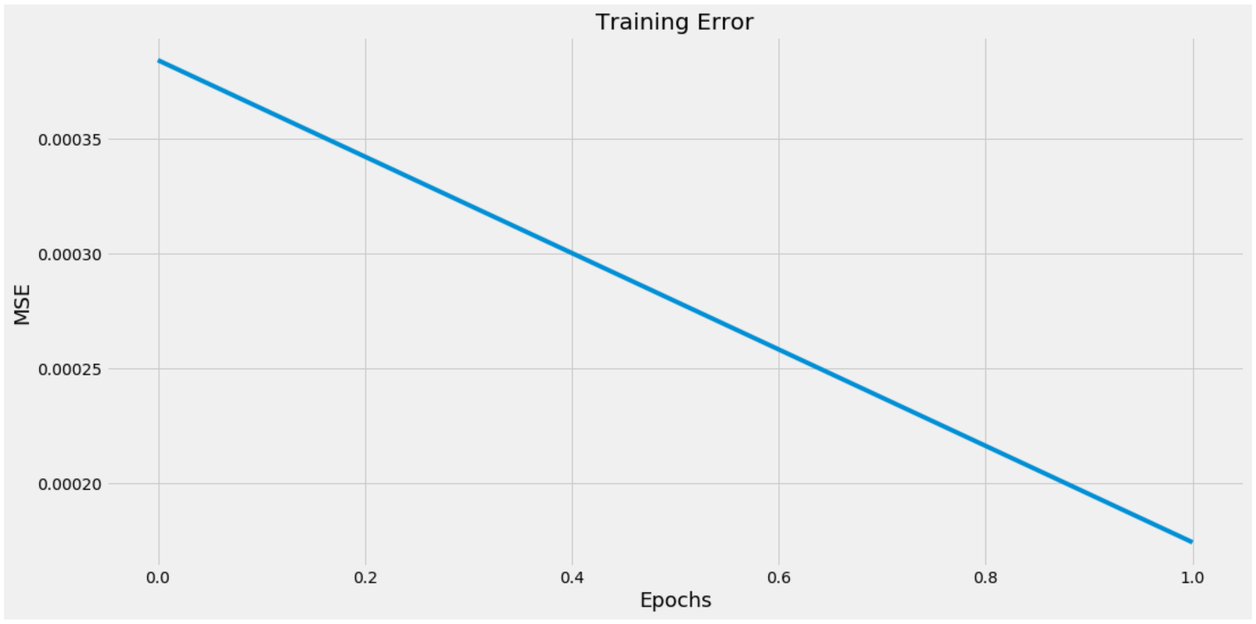


Fig 14: Training Error for GRU (Amazon.com Inc. Dataset)

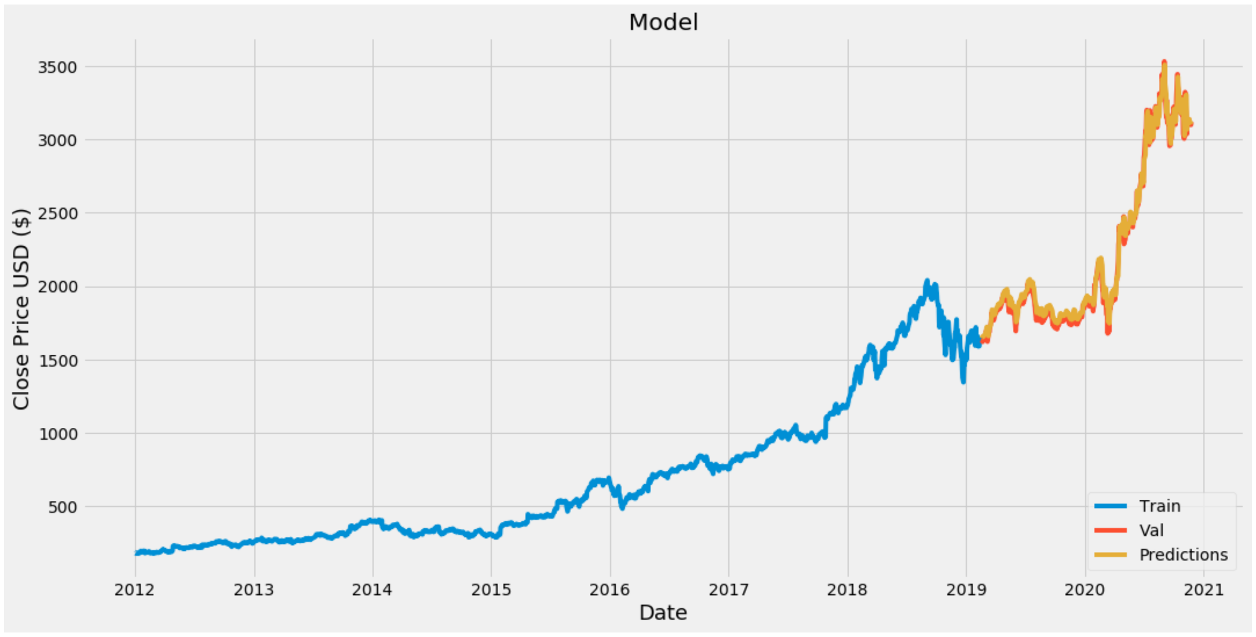


Fig 15: Prediction using GRU model (Amazon.com Inc. Dataset)

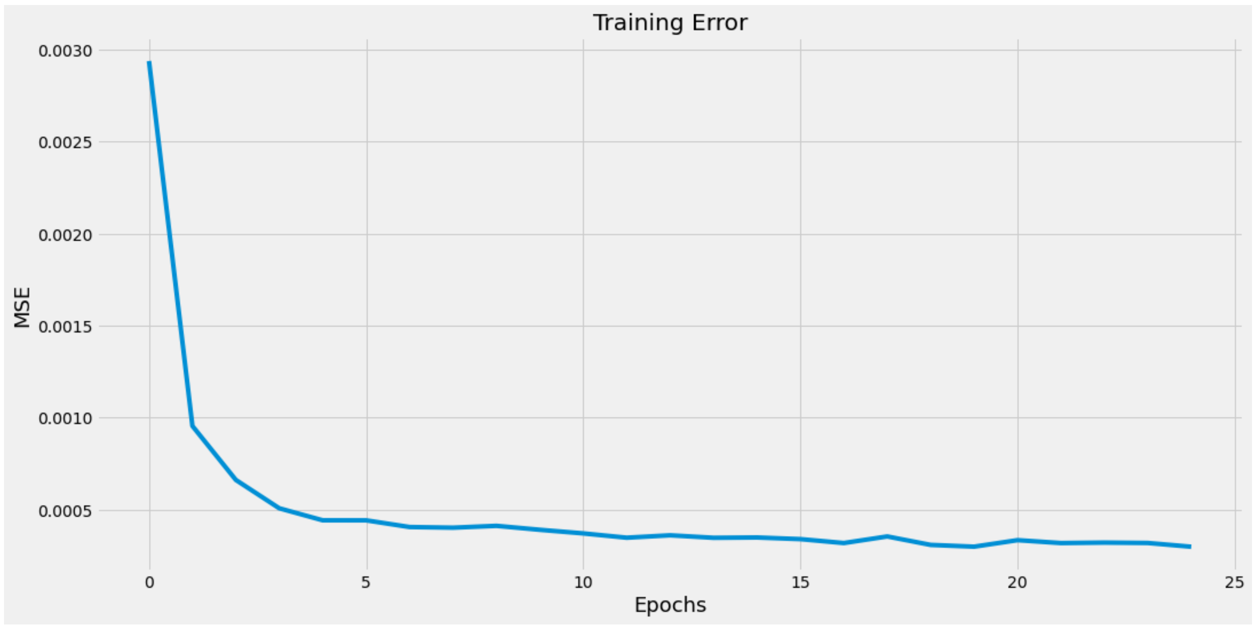


Fig 16: Training Error for LSTM (Ford Motors Company Dataset)

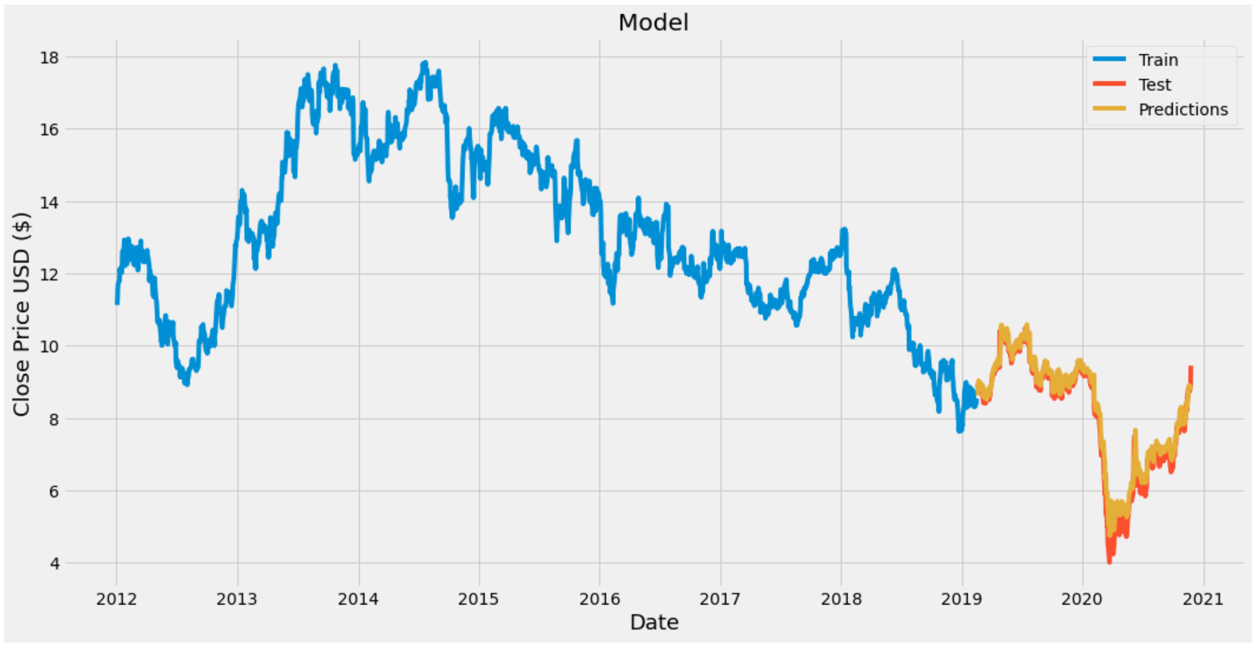


Fig 17: Prediction using LSTM model (Ford Motors Company Dataset)

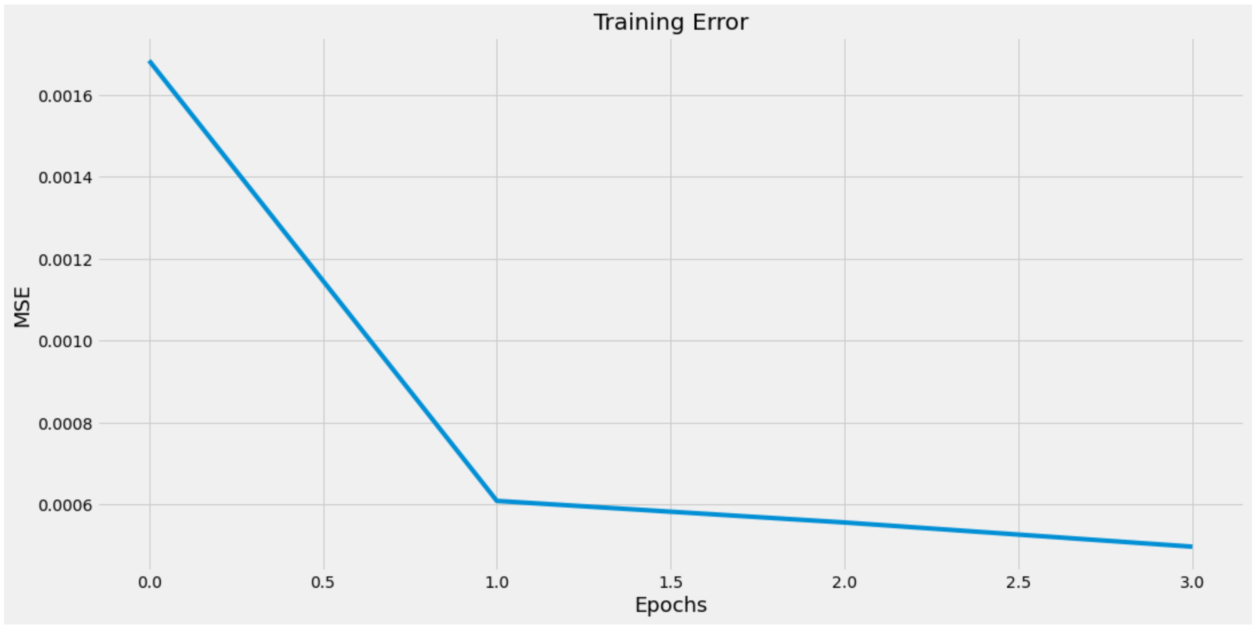


Fig 18: Training Error for GRU (Ford Motors Company Dataset)

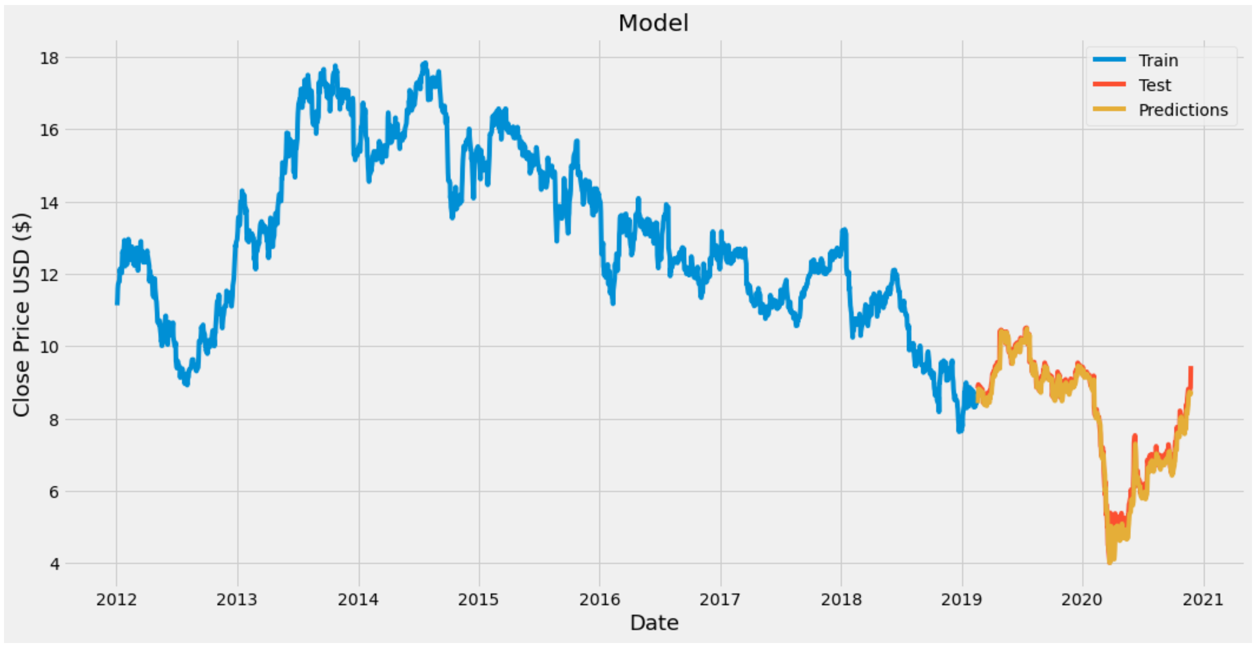


Fig 19: Prediction using GRU model (Ford Motors Company Dataset)

7. References

[1] Li, R., Fu, D., & Zheng, Z. (2017). An analysis of the correlation between internet public opinion and stock market. *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. https://doi.org/10.1109/icisce.2017.41

[2] Tiwari, V., Tiwari, V., Gupta, S., & Tiwari, R. (2010). Association rule mining: A graph based approach for mining frequent itemsets. *2010 International Conference on Networking and Information Technology*. https://doi.org/10.1109/icnit.2010.5508505

[3] Kunal, S., Saha, A., Varma, A., & Tiwari, V. (2018). Textual dissection of live Twitter reviews using naive Bayes. *Procedia Computer Science*, *132*, 307-313. https://doi.org/10.1016/j.procs.2018.05.182

[4] Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, *47*, 552-567. https://doi.org/10.1016/j.najef.2018.06.013

[5] *Pandas-datareader — pandas-datareader 0.9.0rc1+2.g427f658 documentation*. (n.d.). pandas-datareader — pandas-datareader 0.9.0rc1+2.g427f658 documentation. https://pandas-datareader.readthedocs.io/en/latest/

[6] Jian Zheng, Cencen Xu, Ziang Zhang, & Xiaohua Li. (2017). Electric load forecasting in smart grids using long-short-Term-Memory based recurrent neural network. *2017 51st Annual Conference on Information Sciences and Systems (CISS)*. https://doi.org/10.1109/ciss.2017.7926112

[7] M, H., E.A., G., Menon, V. K., & K.P., S. (2018). NSE stock market prediction using deep-learning models. *International Conference on Computational Intelligence and Data Science (ICCIDS 2018),Procedia Computer Science*, *132*, 1351-1362. https://doi.org/10.1016/j.procs.2018.05.050

[8] Zachary C. Lipton, John Berkowitz, & Charles Elkan. (n.d.). *A critical review of recurrent neural networks for sequence learning*. arXiv.org. https://arxiv.org/abs/1506.00019v4

[9] Gao, X., Shi, M., Song, X., Zhang, C., & Zhang, H. (2019). Recurrent neural networks for real-time prediction of TBM operating parameters. *Automation in Construction*, *98*, 225-235. https://doi.org/10.1016/j.autcon.2018.11.013

[10] Pascanu, R., Mikolov, T., & Bengio, Y. (n.d.). *On the difficulty of training recurrent neural networks*. arXiv.org. https://arxiv.org/abs/1211.5063v2

[11] Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, *38*(8), 10389-10397. https://doi.org/10.1016/j.eswa.2011.02.068

[12] Shen, G., Tan, Q., Zhang, H., Zeng, P., & Xu, J. (2018). Deep learning with gated recurrent unit networks for financial sequence predictions. *Procedia Computer Science*, *131*, 895-903. https://doi.org/10.1016/j.procs.2018.04.298

[13] Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://doi.org/10.3115/v1/d14-1179

[14] Wang, Y., Liao, W., & Chang, Y. (2018). Gated recurrent unit network-based short-term photovoltaic forecasting. *Energies*, *11*(8), 2163. https://doi.org/10.3390/en11082163

[15] Box, G. Box and jenkins: Time series analysis, forecasting and control. *In A Very British Affair: Six Britons and the Development of Time Series Analysis during the 20th Century*; Palgrave Macmillan: London, UK, 2013; pp. 161–215, ISBN 978-1-137-29126-4.

[16] De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, *22*(3), 443-473. https://doi.org/10.1016/j.ijforecast.2006.01.001

[17] Menon, V. K., Chekravarthi Vasireddy, N., Jami, S. A., Pedamallu, V. T., Sureshkumar, V., & Soman, K. P. (2016). Bulk price forecasting using spark over NSE data set. *Data Mining and Big Data*, 137-146. https://doi.org/10.1007/978-3-319-40973-3_13

[18] Wilson, G. T. (2016). Time series analysis: Forecasting and control, 5th edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1. *Journal of Time Series Analysis*, *37*(5), 709-711. https://doi.org/10.1111/jtsa.12194

[19] White, H.(1988). *Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns*, ser. Discussion paper- Department of Economics University of California San Diego. Department of Economics, University of California.

[20] Heaton, J., & Polson, N. (2016). Deep learning for finance: Deep portfolios. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2838013

[21] Roman, J., & Jameel, A. (1996). Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns. *Proceedings of HICSS-29: 29th Hawaii International Conference on System Sciences*. https://doi.org/10.1109/hicss.1996.495431

[22] Hengjian, J., *Investigation Into The Effectiveness Of LongShort Term Memory Networks For Stock Price Prediction*, arXiv:1603.07893v3 [cs.NE].

[23] Naeini, M. P., Taremian, H., & Hashemi, H.B. (2010). Stock market value prediction using neural networks. *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*. https://doi.org/10.1109/cisim.2010.5643675

[24] Dutta, A., Kumar, S., & Basu, M. (2020). A gated recurrent unit approach to bitcoin price prediction. *Journal of Risk and Financial Management*, *13*(2), 23. https://doi.org/10.3390/jrfm13020023

[25] H. D. Huynh, L. M. Dang, and D. Duong, A new model for stock price movements prediction using deep neural network. *Proceedings of the Eighth International Symposium on Information and Communication Technology*. ACM, 2017, pp. 57–62.

[26] Keras Team. (n.d.). *Keras documentation: Keras API reference*. Keras: the Python deep learning API. https://keras.io/api/

[27] D. P. Kingma and J. Ba., *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, 2014.

[28] (n.d.). Yahoo Finance - Stock Market Live, Quotes, Business & Finance News. https://finance.yahoo.com/

[29] *Pandas-datareader — pandas-datareader 0.9.0rc1+2.g427f658 documentation*. (n.d.). pandas-datareader — pandas-datareader 0.9.0rc1+2.g427f658 documentation. https://pandas-datareader.readthedocs.io/en/latest/

[30] *Sklearn.preprocessing.MinMaxScaler — scikit-learn 0.24.1 documentation*. (n.d.). scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation. Retrieved February 13, 2021,from https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[31] Dietterich, T. G. (2002). Machine Learning for Sequential Data: A Review. *Oregon State University, Corvallis, Oregon, USA,*. : http://www.cs.orst.edu/~tgd

[32] Khare, K., Darekar, O., Gupta, P., & Attar, V. Z. (2017). Short term stock price prediction using deep learning. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. https://doi.org/10.1109/rteict.2017.8256643

[33] Halkjaer, S., & Winther, O. (n.d.). The effect of correlated input data on the dynamics of learning. *The Niels Bohr Institute Blegdamsvej 17 2100 Copenhagen, Denmark* . halkjaer,winther@connect.nbi.dk

[34] *CS 230 - Recurrent neural networks Cheatsheet*. (n.d.). Stanford University. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[35] *Time series data prediction using sliding window based RBF neural network*. (n.d.). Semantic Scholar | AI-Powered Research Tool. https://www.semanticscholar.org/paper/Time-Series-Data-Prediction-Using-Sliding-Window-Hota-Handa/91037f01fd4b845eadca0b53f5dc00d9f61ac493