**Stock Analysis and Prediction Based on API using LSTM**

***Abstract :***  Stock price prediction serves as a interesting and challenging field for most of the researchers . Prices of stock being fluctuating in nature requires proper acquisition of the predicting model . Taking into consideration the trends of the stock market , we have designed a model based on LSTM ( Long Short-Term Memory ) , that can be used to analyse and forecast the trends of stock values for any company . The model is trained and tested using LSTM that makes use of time series for prediction . LSTM advances the concept of RNN ( recurrent neural network ) by offering memory advantages i.e it provides a feedback mechanism to ensure security of data . The proposed model is able to intake various fluctuating data values either by manual datasets or via API ( Application Programming Interface ) and has been tested with various single day values and consecutive day values for prediction . However the accuracy over time series of two days gave the maximum accuracy . The experimental results thus obtained reveal that the LSTM model proposed in this paper can efficiently analyze and extract the features of the feeding input and can perform plenty good over any sort of fluctuating data .

***Keywords :*** LSTM ; GRU ; deep learning ; stock price prediction ; Yahoo! Finance.

1. ***Introduction :***

A place where trading of shares for any company takes place is known as stock market . Stock markets have been known to grab a huge attention of youth as well as various investors who keep an eye on gaining the profit by any means . From traditional times uptill now , various analytical tools as well analytical models have been designed to study the trends in stocks of various companies . The fluctuating and volatile stock market always tends to challenge the analysts and forecastors . Fluctuations are based on various factors such as national crisis , natural disasters and many more . In such cases , investors are in serious need of some source that can help them predict the predicament . Analysis of data basically requires a study of ongoing trends of stock . Earlier , analytical models were trained using algorithms such CNN (Convolutional Neural Network ) , ARIMA , RNN ( Recurrent Neural Network ) etc whereas some models were just based on statistical recognition of data . The models that were based on CNN gave grid topology structures whereas RNN makes use of time series and short term memory, which studies the timewise characteristics for further enhanced performance .

Aim of preparing a model for stock analysis and prediction is to ease the functioning of human brain i.e to prepare a computational model that can itself perform these operations as human intelligence does . The process of preparing such a model to behave as human brain is known deep learning . In our case , deep learning is implemented to train the proposed model that has been trained using LSTM algorithm . LSTM (Long Short-Term Memory ) is an enhanced version of RNN that offers feedback mechanism thus making efficient use of memory space and ensures authenticity of data . LSTM is also an algorithm that makes use of time series mechanism for analysing the characteristics , such that it can train the model into an efficient one for further predictions and performance . GRU ( Gated Recurrent Unit ) is a mechanism that is better than LSTM for smaller set of data whereas LSTM is found to be better than GRU because GRU is basically a LSTM with forget gate but still lacks an output gate . In our case, we have used LSTM as the mechanism to train our model using time series . This paper consists of the terminologies in section 2 , proposed approach in section 3 , and further proceeded by experimentation and results in section 4 and 5 respectively.

1. ***Terminologies :***

***2.1 LSTM :***

LSTM (Long Short-Term Memory) systems are a sort of intermittent neural system equipped for learning request reliance in arrangement expectation issues. This is a conduct required in complex issue areas like machine interpretation, discourse acknowledgment, and then some. LSTMs are a perplexing region of profound learning. It very well may be difficult to get your hands around what LSTMs are, and how terms like bidirectional and succession to-arrangement identify with the field. LSTM is an improved version of RNN with an output gate that is capable of sensing which data to keep and which to discard at the hidden layer. This feature of LSTM helps to maintain an efficient utilization of memory . LSTM also solves the vanishing gradient problem .

A LSTM unit basically consists of an input gate , output gate , forget gate and a cell . All the three gates are responsible for the in and out of data whereas the cell is responsible for storing values at random instances . A LSTM unit is as shown in Fig.1 :

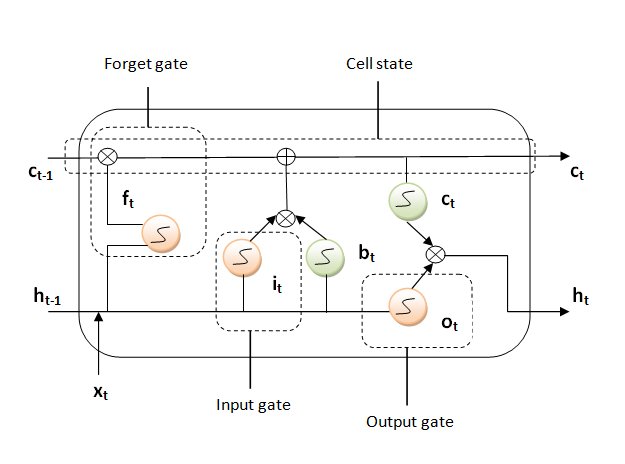


Fig .1 LSTM unit

where :

= Sigmoid function = Tanh function

= Pointwise multiplication = Pointwise addition

At the point when vectors are coursing through a neural system, it experiences numerous changes because of different math tasks . During this if a value keeps on multiplying , at some point it will become that large such that it will be insignificant . Thus a sigmoid function ensures that this value stays between 0 and 1 . Whereas the tanh function ensures that the value stays between -1 and 1 .

Based on Fig.1 , following are the 6 information processing equation of LSTM :

1. it ** Wi xt Hiht-1  bi
2. ft ** Wf xt H f ht -1  bf)
3. ot  ** Wo xt Hoht-1  bo
4. bt  tanhWb xt Hbht-1  bb)
5. ct  ft ct1  it bt
6. ht  ottanhct 

where :

it = input gate , it provides input to cell .

ft = forget gate , it determines which data to be dropped .

ot = output gate , it determines what to output .

bt = eligible value for memory cell state at time ‘t’.

ct = present memory cell state at time ‘t’ .

ht  = output value calculated as element-wise multiplication of ot and tan value of ct .

xt  = the input fed at time ‘t’ .

Wi , Wf , Wo , Wb , Hi , Hf , Ho , Hb are the weight matrices .

bi , bf , bo , bb are the biasing vectors .

***2.2 GRU :***

GRU (Gated Recurrent Unit) is also a modified version of RNN that consists of gates . A GRU unit consists of a reset gate and an update gate .It is similar to LSTM but lacks an output gate . Update gate acts combinely as an input and forget gate . It is responsible to determine which information to intake and process as well as which information to keep or throw . It was designed to solve the problems of vanishing gradient . Generally when the error gradient is substracted from the weighted value of each node during back propagation , the magnitude of gradient keeps on becoming smaller and smaller while getting synchronised using the activation functions within the early nodes. Whereas the update gate in GRU provides the forget gate’s vector within the additive structure to update the values along with cell state gradient to avoid the vanishing of gradient .

***2.3 Yahoo! Finance as API :***

Yahoo! Finance is an agency of Yahoo, that supplies economical news , updates , data , stock statements , media exclusives as well as various online tools that allow users to maintain their own financial status . It is basically a web of updates regarding economical markets . It also facilitates investing and trading via it’s application with large number of users currently active on it .

It can also be utilised as an API(Application Programming Interface) within the python programming to fetch the earlier or latest financial datasets of any company . This can be done by importing the ‘yfinance’ package .

It is an easier , flexible and efficient source that facilitates the implementation and study of stock markets as well as finance related activities in programming world .

1. ***Proposed Approach :***

**Input**

***Capture1.PNG***

Distinction

of data

Data

Preprocessing

Normalization

of data

Training the

model

Feature

extraction

Re-Combining all

the data

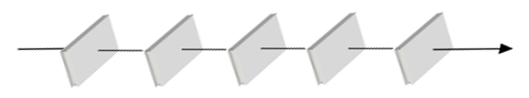
Filling vacant

spaces

ReLU

LSTM

LSTM



Output layer

Hidden layer n

Hidden layer 2 **.......**

Hidden layer 1

Input layer

Output as

prediction

**Fig .2**

***3.1 Data sample ( Input ) :***

Raw data from either the downloaded dataset or from the dataset fetched from the API is fed during the first phase . This data is used for the stock analysis and prediction .

***3.2 Data Pre-processing :***

Data pre-processing refers to the process of analysing, modifying and correcting the data if needed ; before feeding it to the next phase . It is basically composed of four steps :

***3.2.1 Data Distinction :***

Also known as the separation of data is the process of reducing the amount of data based on it’s

Importance by distinguishing different elements of data .

***3.2.2 Data Normalization :***

Normalization of data refers to the reduction in magnitude of the data so as to lessen up the complex

processing and avoid the redundancy of data .

***3.2.3 Data Filling :***

In this step , vacant spaces present in the data sample is filled in, manually .

***3.2.4 Data Recombination :***

Here the recombination of data takes place i.e the data processed in the earlier three steps is combined

into a whole so as to complete the pre-processing of data .

After these steps, the refined data is split into training and testing sets of data. The ratio that we have used is 7:3 for training : testing respectively .

***3.3 Feature extraction :***

In this phase, the essential feature or attributes present amongst the data sample are extracted it i.e those features that will be fed to the neural network present in next phase . Here we are going to feed Open Price, Low Price, High Price, Volume, Close Price and the time .

***3.4 Training the model :***

Training the model basically refers to the training of the neural network to which the data from previous phase would be fed . Training proceeds by assigning weights and bias vectors to the neurons . This model is basically a layered LSTM model that consists of an input layer proceeded by two or more LSTM layers with a further dense layer that consists of ReLU activation function. Number of layers of LSTM required is dependent on how complex the operation is. Then finally comes the output layer that outputs the prediction . It has linear activation functions that are applied to the output .

***3.5 Output :***

Output layer is the last phase in this approach . It receives the output from all the earlier layers and then compares the computed value with the expected value and finds the error from the difference of both . Then, it back propagates this error through each and every neuron so as to modify the weighted vectors and further propagate further forward to obtain a less error free value . This value is known as our prediction .

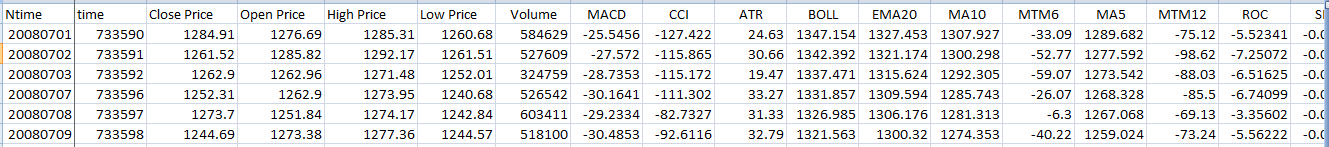
1. ***Experimentation :***

To test the analysing and prediction capacity of our model , we have performed the prediction process in three different ways . In all the three ways the prediction was nearly equivalent to the real time values i.e the error factor was negligible .

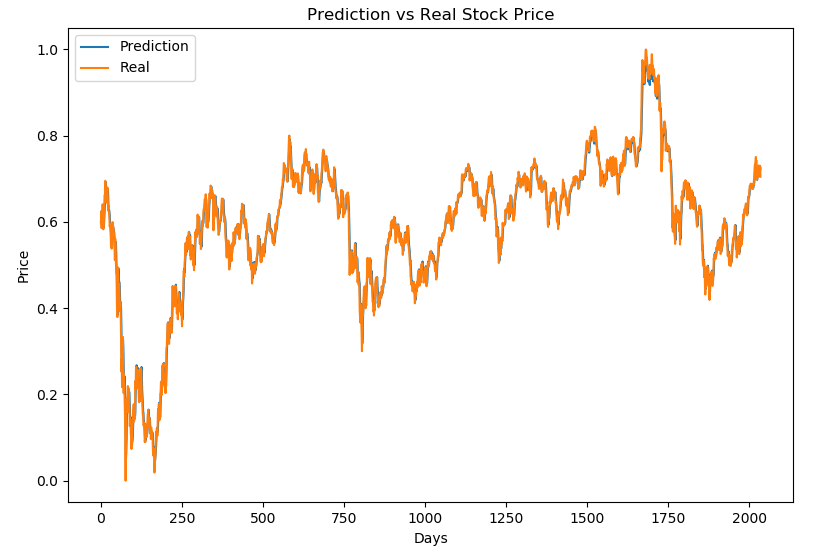
Keeping in mind the time series mechanism used to boost the LSTM , we have considered the most recent data i.e the last two days stats to predict the next day’s value . The three ways of our experimentation was as follows :

***4.1 Analysis and prediction on historic datasets :***

Here, we have downloaded the dataset of stock prices for Hang Seng Index , Hong Kong from 2008-07-01 to 2016-09-30 and then analysed those data values to feed in our model . Thus we obtained the contrast between the real time values and the predicted value . A segmental view of dataset is as follows :



Output was as shown below :



As shown , the blue lines represent the predicted value by our model followed by the real time values

represented by the orange lines . An average error of 0.01874 was obtained which was enough to give

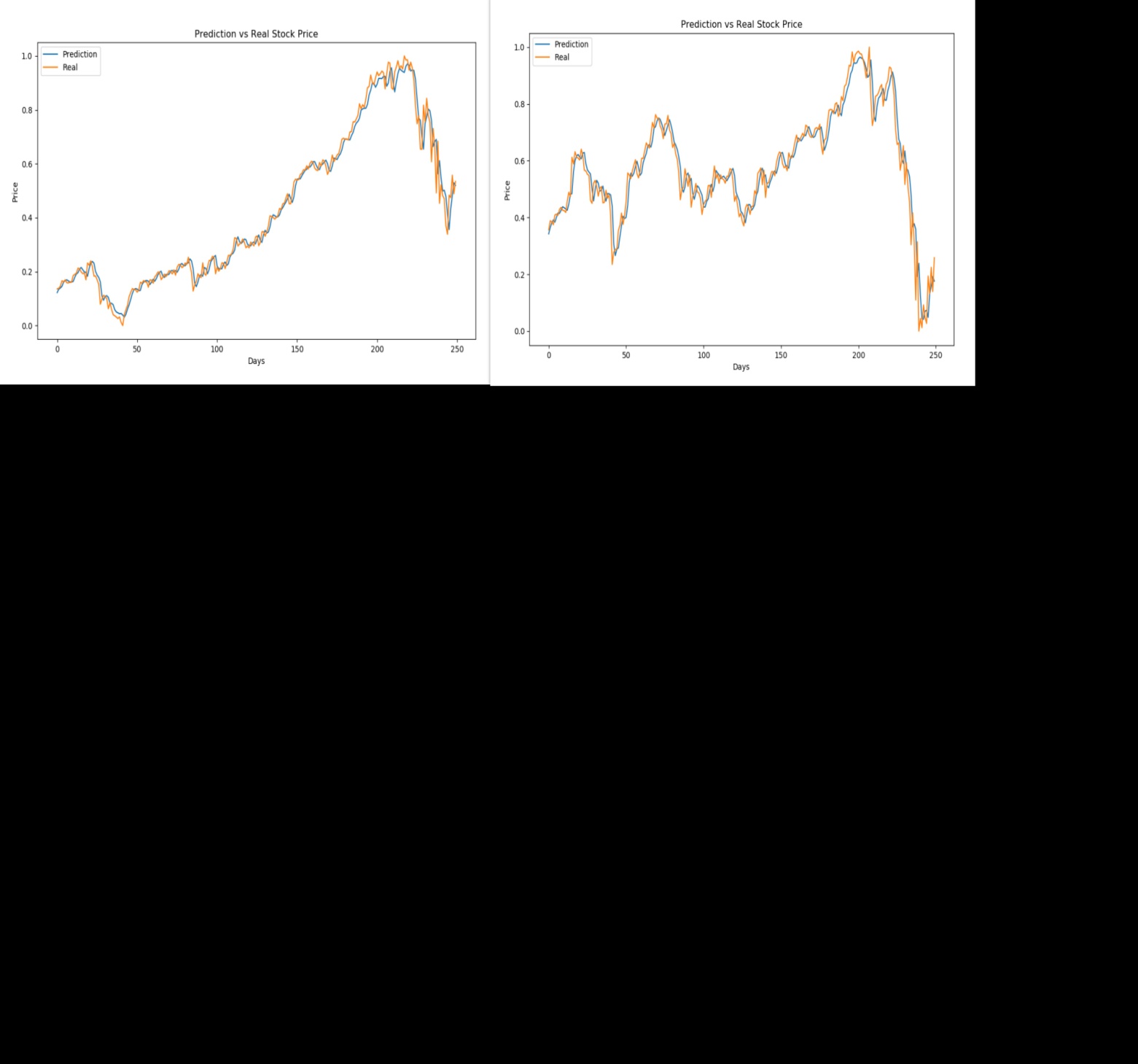
us great accuracy of prediction .

***4.2 Prediction using Yahoo! finance :***

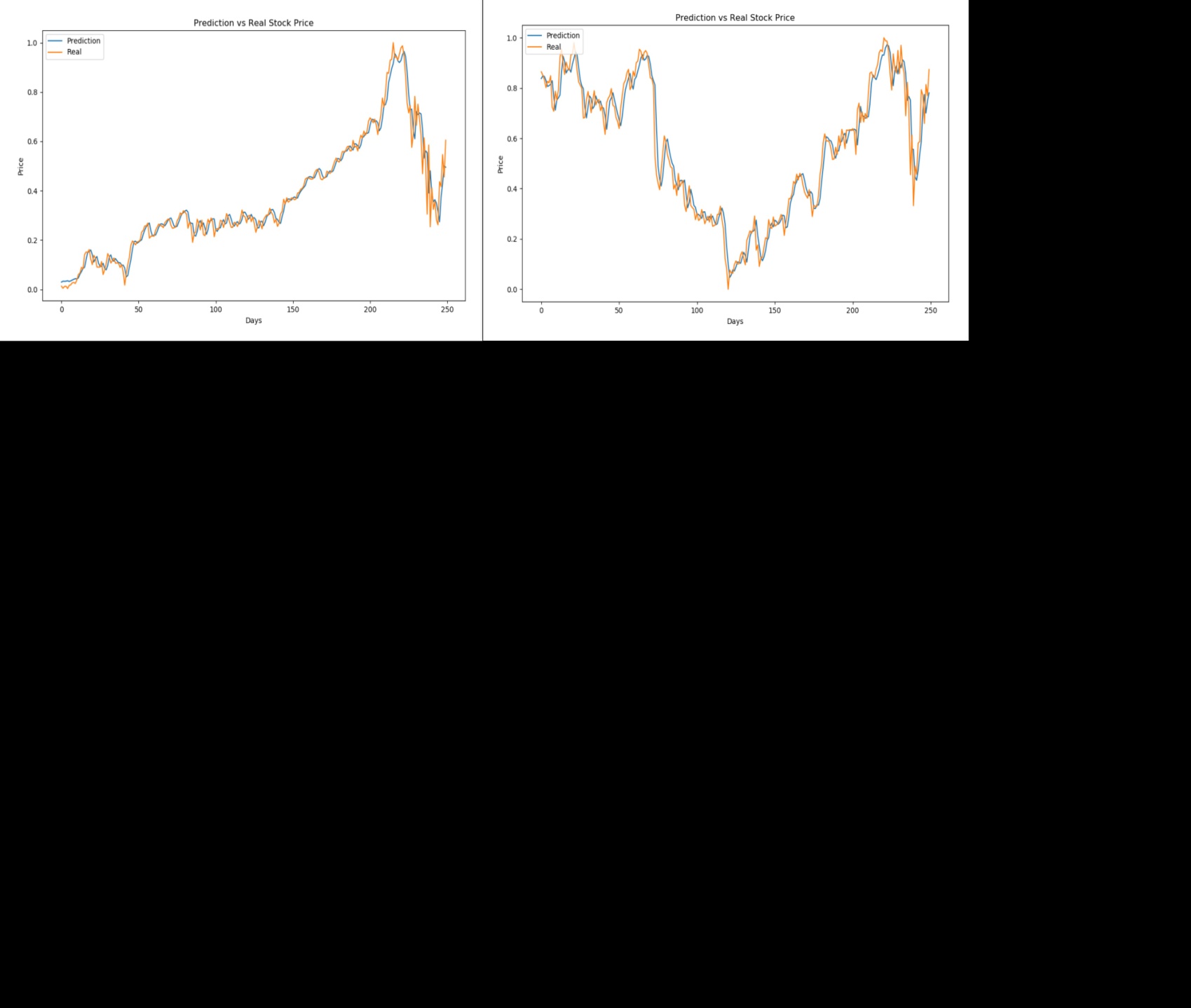
In this practical, we have fetched stock datasets of various companies for previous one year from the

current date and then we have made a contrast between the real time and the predicted values .

Some of those outputs is as shown below :



**Apple with an average error of 0.03796** **Facebook with an average error of 0.05231**



**Netflix with an average error of 0.06291 Microsoft with an average error of 0.05231**

***4.3 Real time future prediction :***

Using Yahoo! Finance we were able to predict the next day’s stock value along with the judgement

whether the value next day will increase or decrease as compared to current day’s value . Some of the

predictions along with their present day values is listed below :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Company’s Name | Value as on  2020/03/31 | Value predicted for 2020/04/01 | Judgement | Actual | Value as on  2020/04/01 |
| APPLE | 254.81 | 251.66 | Decrease | Decrease | 251.89 |
| MICROSOFT | 160.23 | 155.53 | Decrease | Increase | 161.33 |
| FACEBOOK | 165.95 | 161.86 | Decrease | Decrease | 162.10 |
| NETFLIX | 370.96 | 364.76 | Decrease | Decrease | 364.12 |

1. ***Conclusion :*** Stocks are volatile. Prices change according to supply and demand.so by to analysis and measure the accuracy of prediction of stock so that investors get profit by investing.so we are going to check various algorithms that which algorithm gives more accuracy. The algorithm will be a great asset for brokers and investors for investing money in the stock market since it is trained on a huge collection of data . has been chosen after being tested on a sample data. The project demonstrates the machine learning model to predict the stock value with more accuracy as a compared to previously implemented machine learning models. We consisted of one LSTM layers and Attention layer, and was used for extracting the fluctuation characteristics in stock price data through neural network and recurrent neural network. The LSTM recurrent neural networks to extract feature value and analyze the stock data. The experimental results show our model can play a better forecasting effect. In addition, LSTM architecture was used for addressing the gradient disappearance and expansion issues of time series data and then efficiency was improved by using Attention***.***
2. ***References :***

1.Wang Bo, Zhang Fengling. "Comparison of Artificial Neural Network and Time Series Model for Forecasting Stock Prices." Journal of Wuhan University of Technology (Information and Management Engineering) 27.6(2005):69-73.

2. Lin Jie, Guo Yaohuang. "Short Term Prediction of Stock Prices Based on Neural Networks." Journal of Southwest Jiaotong University 33.3(1998):299-304.

3. Gododfellow I, Bengio Y, Courville A. Deep learning [M]. The MIT Press, 2016 .

4. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986c). Learning representations by back-propagating errors. Nature, 323, pp. 533-536.

5. Gododfellow I, Bengio Y, Courville A. Deep learning [M]. The MIT Press, 2016 [5] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986c). Learning representations by back-propagating errors. Nature, 323, pp. 533-534

6. M. Owayjan, R. Achkar and M. Iskandar, "Face Detection with Expression Recognition using Artificial Neural Networks," Beirut, Lebanon, 3rd Middle East Conference on Biomedical Engineering (MECBME), October 2016, pp. 116-120.

7. H. C. Yin, C. Y. Zhao, “Research on stock forecasting based on neural network”, Natural Science Journal of Harbin Normal University, vol. 23, no. 3, pp. 47-49, 2007.

8. S. Hochreiter, J. Schmidhuber, “Long short-term memory.”, Neural computation, vol. 9, pp. 1735–1780, 1997.

9. Khaled A. Althelaya, El-Sayed M. El-Alfy, Salahadin Mohammed, “Evaluation of Bidirectional LSTM for Short- and Long-Term Stock Market Prediction,” in 9th International Conference on Information and Communication Systems (ICICS), 2018 IEEE, pp. 151-156.

10. Sun Jihong. "Long time series clustering method and its application in stock price.". Diss. WuHan University, 2011.