IMPLEMENTATION OF MACHINE LEARNING TECHNIQUE IN STOCK MARKET PREDICTION

[1] N.SANTOSH BHAMI REDDY, [2] Dr. G. LAVANYA DEVI
[2]Department of Computer Science and System Engineering,
Andhra University College of Engineering (A), Visakhapatnam, AP, India.

ABSTRACT: Stock price prediction is an important and classic problem in the world of stock trading. The Stock value prediction is associate degree act of making an attempt to work out the long run worth of a stock or economic instrument that area unit listed on a monetary exchange. With successful prediction we can gain insight about market behavior over time. In earlier the traditional techniques such as the technical and important or the time series analysis were used for prediction of the stocks. The recent trend in stock market prediction is using machine learning techniques as its model makes prediction easier and authentic. We prepare a robust dataset with various features on which the model is trained and tested. So, in this paper we propose a Deep Learning (DL) approach using Long Short-Term Memory (LSTM) that will be trained from the available stocks data and gain intelligence and then uses the non-inheritable data for associate degree for correct prediction. Python programming language is used to predict the stock market using Deep learning.

Keywords- Deep Learning, Financial exchange, LSTM Model, Machine Learning, Stock Price Prediction, and Stock trading.

I. INTRODUCTION

Basically, In Stock Markets the traders usually buy stocks derivatives and equities at a cheap price and later on selling them at high price. Before investing in a stock, the investor generally analyses it in two ways first is the fundamental analysis and other is the technical analysis. In fundamental analysis investors look at the intrinsic value of stocks, performance of the company, industry, economy, country and etc. to decide whether to invest or not. On the other hand, the technical analysis it is an evaluation of various types of stock charts by the means of studying the statistics generated by market activity, such as past prices and volumes traded.

The Financial theory (Efficient-Market Hypothesis) [12] says that the market price of a stock is essentially random. The hypothesis implies that any attempt to predict the stock market will inevitably fail and the best prediction one can have about tomorrow’s value is today’s value. As the Stock price indices generates enormous data which is highly volatile, non-linear and fluctuating it needs a model which forecasts accurately.

In the recent years, with the increase in computational power there is an increase in prominence of machine learning in various industries have produced quite promising results. However, introducing Deep learning to the area of stock prediction is an efficient way that it can analyse the hidden patterns and underlying dynamics. Even the models are capable of identifying and exploiting the interactions and patterns existing in a data through a self-learning process. Unlike other algorithms, deep learning models can effectively model the time series data and can make a good prediction by analysing the interactions and hidden patterns within the data. In this paper we are using a Long Short-Term Memory (LSTM) technique which is a variant of Recurrent Neural Network in order to predict the stock market and we are using Python language for programming. However, If the model makes successful prediction in stock's future price then it could yield significant profits.

This paper will develop a financial data predictor program in which it uses a dataset which contains the calculated volatility and momentum of all historical stock prices and data which will be treated for both training and test sets for the program. The main purpose of the prediction is to predict increase or decrease in stock price for next day and to reduce uncertainty associated to investment decision making.

II RELATED WORK

From the literature survey, it is distinguished that a variety of methods have been used to predict stock prices using machine learning techniques by various researchers throughout the world. However, the accuracy and speed of various traditional prediction techniques is very low when compared to the Deep Learning Techniques. The work of various researchers is observed as follows. Significant work of Ishita Parmar and Others [1] primary contribution of the research is the application of regression model and the novel LSTM Model as a means of determining the stock prices. Both the techniques have shown an improvement in the accuracy of predictions, thereby yielding positive results with the LSTM model proving to be more efficient. Another research of Akhter,

Arun, and Sastry (2015) proposed a method of CNN and a hybrid model for prediction of stock returns [2]. All these studies provide the better result of stock price prediction. The development of CNN is more effective, to solve the problem of vanishing gradient was proposed as LSTM. Adebiyi, Charles, and Marion apply an Artificial Neural Network to predict close price. A combination of technical and fundamental analysis variable is used as the inputs. In this paper is compared the model by testing with technical variables, fundamental analysis variables, and hybrid variables. The result of the hybrid model gives the best result [3].

Billah, Waheed, and Hanifa proposed the Artificial Neural Network and Adaptive Neuro-Fuzzy Inference System (ANFIS) in MATLAB to predict close price [4]. As a result, ANFIS gives a better prediction result. Tsang and Kwok investigated buy and sell signal of stock price prediction. An Artificial Neural Network is proposed in this research. The model successfully predicts the stock price with 70% accuracy [5].

K. A. Althelaya, E.M. El-Alfy and S. Mohammed [6] further contributed to the field by staging experiments and simulations to assess the feasibility of applying deep learning techniques to prediction of stock prices. From the previous study, ANNs are an effectiveness and efficiency technique for stock price prediction. But ANNs is not applied on a time series prediction. As stock price change through the time period. A technique of Recurrent Neural Network is proposed as a time series prediction.

Zhuge, Xu, and Zhang proposed two models; emotional analysis model and long short-term memory (LSTM) time series learning model for stock price prediction [7]. Hengjian [8] investigated the effectiveness of LSTM networks for stock price prediction. By applying the Google daily stock data include the open, high, low, close and volume to predict the next day of daily stock price. The result shows that LSTM predicts google stock price with less than 1% error of RMSE.

EXISTING MODEL

While most of the previous research in this field were focused on traditional techniques to forecast stock price based on the historical data such as past stock trends, news and media. It has huge influence on human beings and the decisions we take. Also, the trading activities of human beings results in the fluctuations of the stock market. Therefore, identifying the trend and extracting information from past data may yield better results in predicting the stock prices. The main drawback of most of the previous researches is that they either used to predict the stock market index as a whole or it is confined to individual company. So, the problem arises when it considers to work with a group of companies that belong to one sector.

PROPOSED MODEL

As there is no application available today that can provide us a guarantee on their predictions or at least ensure the percent accuracy that their system can provide.

In this research we propose a model based on LSTM to predict price movements using an input that is based on numerical analysis and prepare a dataset which was separately built with various features, which as mentioned. The main advantage in this project is we define a new feature called "**sector momentum**", **"sector volatility"** which is specially used to represent a particular set of companies. We plan to use a different type of technical indicators to do so, and the intention is to assess the usage of such method that is something commonly used on investment strategies. In this model additionally, we want to test the hypothesis that the long short-term memory capability can present better results compared to traditional feed forward networks.

III PRELIMINARY ANALYSIS

In this a classification model was designed to perform predictions of price movements for a number of stocks based on LSTM networks. In other words, it attempts to predict increase or decrease in stock price for next day. The model is generated and trained each trading day on top of historic price data and it is used for performing predictions.
The general setup used is as follows:

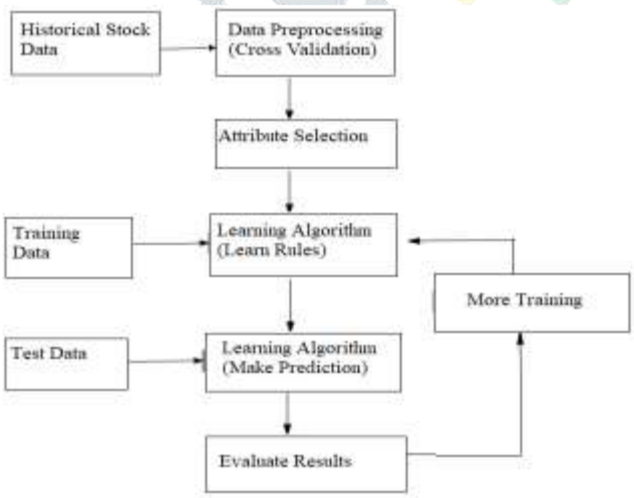


Fig -1: The Learning Environment

B. DATA PROCESSING

The Yahoo Finance acts as one of the sources for financial data. The data set consists of minute wise stock price for 9 companies for the period of 2010 January to December 2018. Historic price data for these stocks are gathered in the format of a csv files. It contains information like (date, open, close, high, low, adj volume and volume). The companies were (Apple, Amazon, Google, HP, IBM, Intel, Lenovo, Microsoft and Oracle) as all belonging to a particular sector, we classified them as "**Software Technology**" sector. The data from these companies were extracted and was subjected to pre-processing to obtain the stock price. It is taken as that the daily closing values of each of the stock as the stock value for a day. For all attributes we kept 5 days as to capture market behaviour. Initially we calculate the 'change' and 'momentum' for all the company's datasets

The following features were calculated to all the above sector companies.

C FEATURE SELECTION:

In this project we use six features to predict stock price direction More details are provided in Table 1, styled in the form used by Kim [9].

Feature Name	Description	Formulae
Index Momentum	This is the average of the given Index momentum over the past 'n' days. Each day is labelled as '1' if closing price of that day is higher than the day before and '0' if the price is lower than the day before.	$\frac{\sum_{i=t-n+1}^{t} d_i}{n}$
Index Volatility	Index Volatility is defined as the It is the average over the past 'n' days of percent change in the given stocks price per day.	$\frac{\sum_{i=t-n+1}^{t} \frac{I_i - I_{i-1}}{I_{i-1}}}{n}$
Sector Momentum	This is the average of the given stock's momentum of all the sector companies over the past 'n' days. Each day is labelled as '1' if closing price of that day is higher than the day before and '0' if the price is lower than the day before.	$\frac{\sum_{i=t-n+1}^{t} S}{n}$
Sector Volatility	Sector Volatility is defined as the it is the average of percent change in the given stocks prices per day of all the sector companies over the past 'n' days.	$\frac{\sum_{i=t-n+1}^{t} V}{n}$
Stock Price Momentum	This is the average of the given stock's momentum over the past 'n' days. Each day is labelled as '1' if closing price of that day is higher than the day before and '0' if the price is lower than the day before.	$\frac{\sum_{i=t-n+1}^{t} y}{n}$
Stock Price Volatility	It is defined as the It is the average over the past 'n' days of percent change in the given stocks price per day.	$\frac{\sum_{i=t-n+1}^{t} \frac{C_i - C_{i-1}}{C_{i-1}}}{n}$

Table 1: Features used in LSTM

IV ALGORITHM

When it comes to Machine Learning, Artificial Neural Networks perform really well. Artificial Neural Networks are used in various classification task like image, audio, words. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use Convolution Neural Network. In this blog, we are going to build basic building block for CNN.

Before diving into the Convolution Neural Network, let us first revisit some concepts of Neural Network. In a regular Neural Network there are three types of layers:

1. **Input Layers:** It's the layer in which we give input to our model. The number of neurons in this layer is equal to total number of features in our data (number of pixels incase of an image).
2. **Hidden Layer:** The input from Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layers can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by addition of learnable biases followed by activation function which makes the network nonlinear.
3. **Output Layer:** The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into probability score of each class.

The data is then fed into the model and output from each layer is obtained this step is called feedforward, we then calculate the error using an error function, some common error functions are cross entropy, square loss error etc. After that, we backpropagate into the model by calculating the derivatives. This step is called Backpropagation which basically is used to minimize the loss.

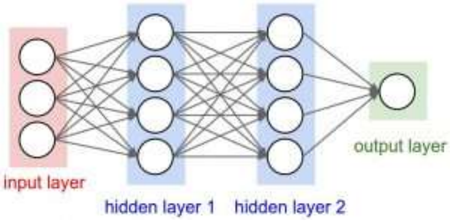
Here's the basic python code for a neural network with random inputs and two hidden layers.

filter_none

brightness_4

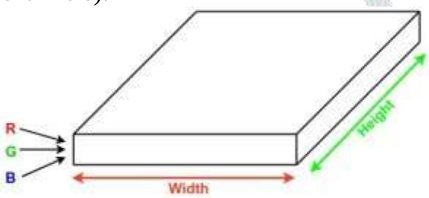
```
activation = lambda x: 1.0/(1.0 + np.exp(-x)) # sigmoid function
input = np.random.randn(3, 1)
hidden_1 = activation(np.dot(W1, input) + b1)
hidden_2 = activation(np.dot(W2, hidden_1) + b2)
output = np.dot(W3, hidden_2) + b3
```

W1,W2,W3,b1,b2,b3 are learnable parameter of the model.



Convolution Neural Network

Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image) and height (as image generally have red, green, and blue channels).



Now imagine taking a small patch of this image and running a small neural network on it, with say, k outputs and represent them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different width, height, and depth. Instead of just R, G and B channels now we have more channels but lesser width and height. his operation is called Convolution. If patch size is same as that of the image it will be a regular neural network. Because of this small patch, we have fewer weights.

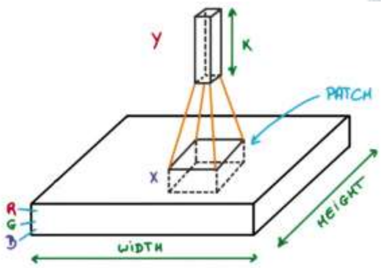


Image source: Deep Learning

Now let's talk about a bit of mathematics which is involved in the whole convolution process.

- Convolution layers consist of a set of learnable filters (patch in the above image). Every filter has small width and height and the same depth as that of input volume (3 if the input layer is image input).
- For example, if we have to run convolution on an image with dimension 34x34x3. Possible size of filters can be axax3, where 'a' can be 3, 5, 7, etc but small as compared to image dimension.
- During forward pass, we slide each filter across the whole input volume step by step where each step is called stride (which can have value 2 or 3 or even 4 for high dimensional images) and compute the dot product between the weights of filters and patch from input volume.
- As we slide our filters we'll get a 2-D output for each filter and we'll stack them together and as a result, we'll get output volume having a depth equal to the number of filters. The network will learn all the filters.

CNN is a class of neural network where connections between the computational units form a directed circle in other words in these neural networks the output of a block is fed as input to the next iteration. They work perfectly well on a large variety of problems, and are now widely used. Unlike feed forward networks, to process arbitrary sequence of inputs CNN can use their internal memory. Each CNN computing unit has a time varying real valued activation and modifiable weight. By applying the same set of weights recursively over a graph-like structure one can create CNNs

In the case of CNN, the learned model always has the same input size, because it is specified in terms of transition from one state to another. Also, the CNN architecture uses the same transition function with the same parameters at every time step. There are different variants of CNN out of which Long Short-Term Memory networks (LSTM) is a special kind. It was introduced in 1997 by Hochreiter and Schmidhuber [11]. In the LSTM architecture, it contains LSTM cells, which replaces the usual hidden layers. The cells are composed of various gates regulate the values for long term propagation while the cell remembers the values. An LSTM cell consists of cell state, input gate, forget gate, and output gate. It also consists of tanh layer, sigmoid layer, and point wise multiplication operation. LSTMs are explicitly designed to avoid the long-term dependency problem by remembering information for long time the functions of the various gates are as follows.

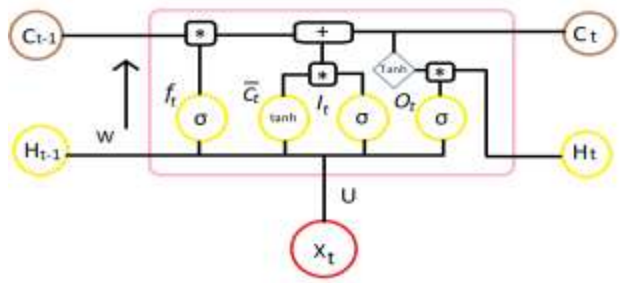


Fig. 2. Long short-term memory neural network [10]

- Input gate: Input gate allows the input into the network.
- Cell State: It runs through the entire network and has the ability to add or remove information with the help of gates.
- Forget gate layer: This gate decides the fraction of the information to be allowed.
- Output gate: It contains the output generated by the LSTM.
- Sigmoid layer: It generates numbers between zero and one, which describes how much of each component should be let through.

V. EVALUATION

The work is based on a sliding window approach for stock training and testing results. The final Data set consisted of these 6 dimensions for each of the stocks. I.E. as each company generates one record a total of 9 records will be generated at the end of each day. The total number of rows for our input data set are around 20300.

This model was designed to work on a sliding window fashion. At the end of each trading day a new neural network is generated meaning that a new set of weights is defined using a new set of training and validation data. Daily stock price is forecasted using features designed by using close price. Moreover, the sliding window technique is applied for training the model. As we are working with a time series problem, the supervised learning algorithm that chosen was the LSTM neural network (Long Short Term Memory), which is a recurrent neural network capable of classifying input data taking into account the previous instances? The input data of yesterday is used to predict the output data of today. All stocks are tested with a different length of training data. The dataset is divided for both train and test purpose. We train the model on 14000 rows and the remaining dataset of 6324 is used for test purpose. The entire model was backed by Google's TensorFlow. Further another LSTM network is trained for 30 epochs. This Keras library is used to design the sequential model as the data is sequential data along with that it uses 'Mean Absolute Error' (MAE), The mean absolute error of a model with respect to a test set is the mean of the absolute values of the individual prediction errors on over all instances in the test set.
The functions of MAE are shown as follows:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |X_i - X|}{n}$$

For evaluating the network performance, metrics around the algorithm performance and the financial results were collected and the metrics were accuracy considered.
The experiment was done for LSTM model. The accuracy metrics along with Standard Deviation and Variation obtained for the model.

	LSTM
Epoch	30
Mean Accuracy	66.65190
Standard Deviation	0.45758
Variation	1.8615

Table 1: **T-Test Results of LSTM**

The results of the experiments are reported and discussed in the following figures show the plots that compare the actual values and the predicted values of all technological companies.

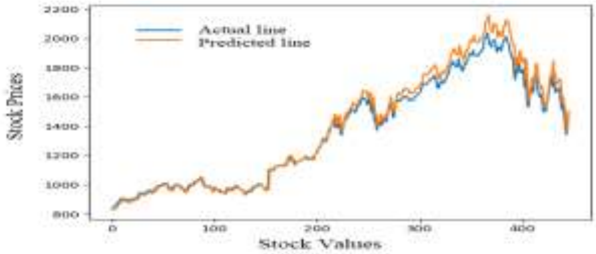


Fig. 3: Plot for Real value vs Predicted value for Amazon using LSTM

From Fig(3) of Amazon, it is clear that the predicted values of LSTM matches with the pattern of original data exactly from 0 to 300 and there is a mismatch from 300 to 400 and again the LSTM tried its best in capture the trends and dynamics of predicted line when compared with the actual line towards the end i.e, there is a change in the behavior of the stock pattern for that window when compared to the earlier prediction.

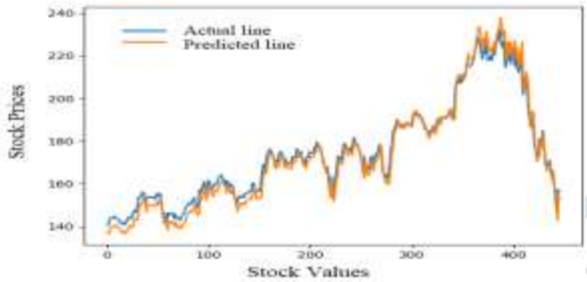


Fig. 4: Plot for Real value vs Predicted value for Apple using LSTM

From Fig (4) of Apple it is observed, that the LSTM predicted line initially fails to identify the actual line but follows the path with minimal difference until 150 and later till 400 the model perfectly identified actual line and the predicted line followed the exact path with it.

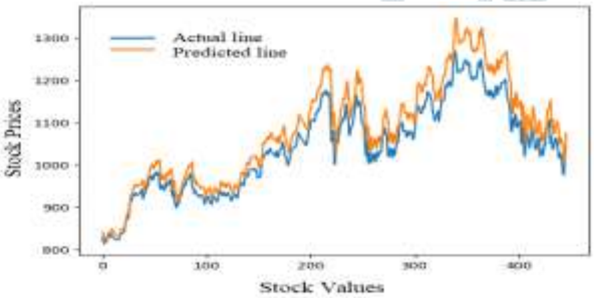


Fig. 5: Plot for Real value vs Predicted value for Google using LSTM

It is evident from the Fig (5) of Google, LSTM networks are identifying the pattern in the beginning of the window exactly. There is a change in the trend followed by Google line failed to predict exactly during next period. This makes the predictions a bit less accurate, whereas at the end the difference between the actual and predicted line decreased gradually.

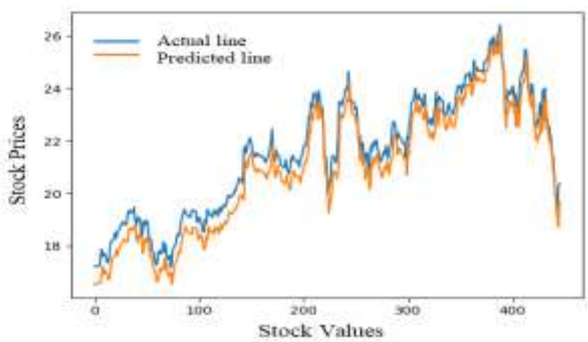


Fig. 6: Plot for Real value vs Predicted value for HPQ using LSTM

In Fig (6), of HP we can see that LSTM tried its best in capture the trends and dynamics of actual line. It is observed that the predicted line captured less price and moved below the actual line and makes the prediction very accurate in the case of HP. Even though the predicted line moves with actual line but only few instances between 100 and 400 where it fails to move on the trend where the difference between them is very negligible.

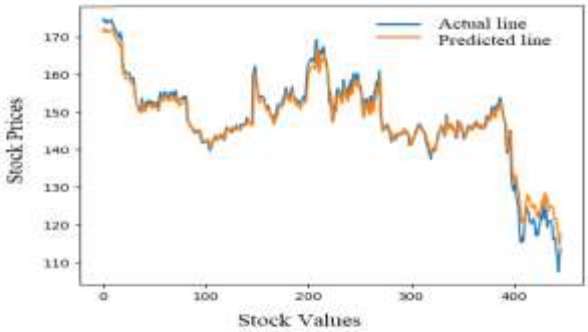


Fig. 7: Plot for Real value vs Predicted value for IBM using LSTM

From the Fig (7), of IBM it is evident that the predicted line of LSTM moves perfectly with the pattern of original data by capturing the trend of the actual line. Which makes the model prediction very accurate between 0 to 400 of the two lines.

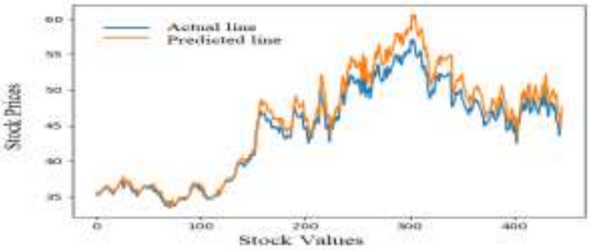


Fig. 8: Plot for Real value vs Predicted value for Intel using LSTM

In case of Intel, Fig (8), we can observe that LSTM model captured the actual line initially and followed the trend, between the 200 and 400 during this period the model works better but, in few instances, it failed to match with the actual line trend.

Fig. 9: Plot for Real value vs Predicted value for Lenovo using LSTM

In Fig (9) of Lenovo, it is clearly evident that that the LSTM predicted line completely fails to identify the actual line from the beginning of 0 to 400. As there is a change in the behavior of the stock pattern for that window the predicted line move above the actual line but followed the trend and path with it.

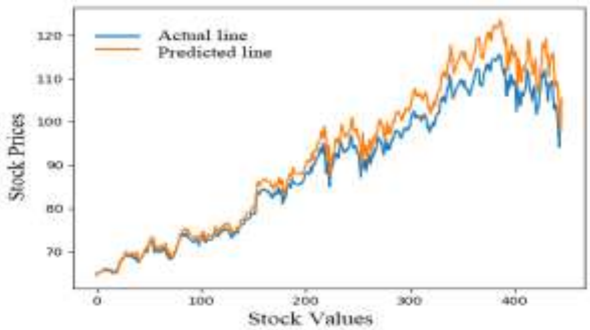


Fig. 10: Plot for Real value vs Predicted value for Microsoft using LSTM

From Fig(10), of Microsoft it is observed that model captures the actual trend line exactly and is capable of capturing the changes in the trend for the stock values in initial period till 200 there after the LSTM failed to capture the trend and the predicted line completely fails to match with the actual line in the next period but follows the path.

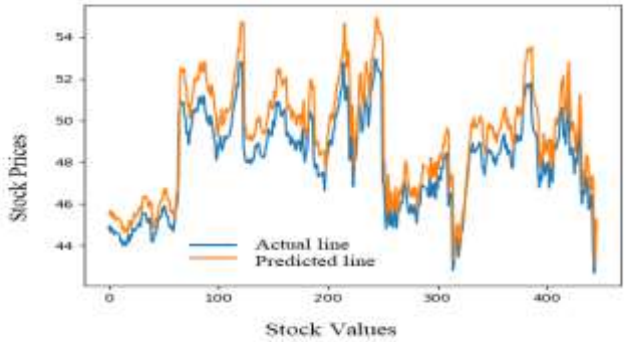


Fig. 11: Plot for Real value vs Predicted value for Oracle using LSTM

In Case of Fig (11), Oracle it is clear and evident that the LSTM model identify and capture the trend of the actual pattern line from the beginning 0 to 400 but fails in few instances with a minimum difference with actual line.

VI CONCLUSION

We propose a deep learning model-based formalization for stock price prediction. It is seen that; deep neural network architectures are capable of capturing hidden dynamics and are able to make predictions much better compared to traditional techniques. We trained the model with the processed dataset of different companies along with Nasdaq index values predict stock price movement of Apple, Amazon, Google, HPQ, IBM, Intel, Lenovo, Microsoft and Oracle. This shows that, the proposed system is capable of identifying some inter relation with in the data. Also, it is evident from the results that, LSTM architecture is capable of identifying the changes in trends exactly for few companies and near exactly to remaining all companies. When compared to traditional techniques LSTM provide better results. For the proposed methodology LSTM is identified as the better model as it uses the information given of a time series data for prediction. As the stock market is more volatile in nature there maybe the sudden changes that occurs in markets. The changes most of the time follows irregular pattern or may not always follow the same cycle with in the stock market. Based on the Indexes, sectors and companies listed, the existence of the trends and the period of their existence will differ. A small drawback of LSTM network is it uses information from previous lags to predict the future instances. Since stock market is a highly dynamical system, the patterns and dynamics existing within the system will not always be the same. This cause learning problems to LSTM architecture and hence the models fail to capture the dynamical changes accurately. So to overcome that problem we can use the better available models like CNN. With CNN we can overcome the drawback of LSTM as well as we can get more accuracy and other metrics for most of time series problems.

REFERENCES

[1] Ishita Parmer, N Agarwal, S Saxsena, R Arora, L Chouhan, " Stock Market Prediction Using Machine Learning," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), pp, 574-576

[2] A. M. Rather, A. Agarwal, and V. N. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," Expert Syst. Appl., vol. 42, no. 6, pp. 3234–3241, 2015.

[3] A. A, Adebiyi, A. K, Charles, A. O, Marion, and O. O, Sunday, "Stock Price Prediction using Neural Network with Hybridized Market Indicators," J. Emerg. Trends Comput. Inf. Sci., vol. 3, no. 1, pp. 1–9, 2012.

[4] M. Billah, S. Waheed, and A. Hanifa, "Predicting Closing Stock Price using Artificial Neural Network and Adaptive Neuro Fuzzy Inference System (ANFIS): The Case of the Dhaka Stock Exchange," Int. J. Comput. Appl., vol. 129, no. 11, pp. 975–8887, 2015.

[5] P. M. Tsang et al., "Design and implementation of NN5 for Hong Kong stock price forecasting," Eng. Appl. Artif. Intell., vol. 20, no. 4, pp. 453–461, 2007.

[6] K. A. Althelaya, E. M. El-Alfy and S. Mohammed, "Evaluation of bidirectional LSTM for short-and long-term stock marketprediction," 2018 9th International Conference on Information and Communication Systems (ICICS), Irbid, 2018, pp. 151-156.

[7] Q. Zhuge, L. Xu, and G. Zhang, "LSTM neural network with emotional analysis for prediction of stock price," Eng. Lett., vol. 25, no. 2, pp. 167–175, 2017.

[8] H. Jia, "Investigation Into The Effectiveness Of Long Short Term Memory Networks For Stock Price Prediction," arXiv[cs.NE], pp. 1– 6, 2016.

[9] K. jae Kim, "Financial time series forecasting using support vector machines," Neurocomputing, vol. 55, 2003.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] K. Greff, R. K. Srivastava, J. Koutn´ık, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," arXiv preprint arXiv:1503.04069, 2015.

[12] S.Kevin , Security analysis and portfolio management, PHI Learning; 2nd edition.

[13] Stock market, [online] Available: http://www.google.com

[14] Stock values, [datasets] Available: https://in.finance.yahoo.com/