Stock Price Prediction using Deep Learning Model

Riddhesh Bonde[1] Meghana Mule[2] Dr. Savita Sangam[3]

[1, 2, 3]*Department of Information Technology, Shivajirao S. Jondhale College of Engineering, Dombivli.*

[1]riddheshsj@gmail.com

[2]meghanamule0@gmail.com

[3]savita.sangam@gmail.com

Abstract: *A stock market, equity market or share market is the aggregation of buyers and sellers (a loose network of economic transactions, not a physical facility or discrete entity) of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange, as well as stock that is only traded privately. Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Deep Learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. We are going to design an automated model for predicting future stock prices. We are going to use LSTM network, an recurrent neural network to design a model. We are going to use Keras and TensorFlow, python libraries to make the process easier and more customizable.*

Keywords: *Stock; Deep Learning; LSTM; Prediction; Neural Network*

1. INTRODUCTION

A stock trader or equity trader or share trader is a person or company involved in trading equity securities. Stock traders may be an agent, hedger, arbitrageur, speculator, stockbroker. Such equity trading in large publicly traded companies may be through one of the major stock exchanges, such as the New York Stock Exchange or the London Stock Exchange, which serve as managed auctions for stock trades. Stock shares in smaller public companies are bought and sold in over the counter (OTC) markets. Equity trading can be performed by the owner of the shares, or by an agent authorized to buy and sell on behalf of the share's owner. [6]

Factors affecting the Stock Market :

1. **Economic growth:** Higher economic growth or better prospects for growth will help firms be more profitable because there will be more demand for goods and services. This will help boost company dividends and therefore share prices.
2. **Interest rates:** Lower interest rates can make shares more attractive for two reasons. Lower interest rates help boost economic growth making firms more profitable. Also, lower interest rates make shares relatively more attractive than saving money in a bank or holding bonds. If bond yields fall, it may encourage investors to switch into shares which give a relatively better dividend.

3. **Stability:** Stock markets dislike shocks that could threaten economic stability and future growth. Therefore, they will tend to fall on news of terrorist attacks or spikes in the price of oil. They will also dislike political instability which may make it difficult to pursue strong economic policies.
4. **Price to earnings ratios:** Some investors and economists, such as Robert Shiller feel the best guide to the long-term performance of shares is their price to earnings ratios. If share prices rise significantly above historical averages, then this is a sign that shares are becoming overvalued and are due a correction at some point in the future.

In the period of the 1920s, there was a rise in the ratio of share prices to earnings. But this rapid growth ended abruptly in 1929 – with Wall Street Crash. There was a remarkable bull run in the 1980s to 2000.[6] Over the past 100 years, the average 'Price to earnings' ratio has steadily crept up, so it is not an exact science to say what the long-term average of price to earnings ratios is.

Figure 1. Price to Earnings Ratio

Problem Definition

The price of the stock is affected by numerous factors and there can be a huge loss to stock trader, if proper prediction is not made. There are many models available which can predict stock price, but they are not much efficient like LSTM. So, we are going to design a model which will use the deep learning method of stock price prediction, and we are going to design a simple stock price prediction which will be efficient and also simple compared to other machine learning models. We are also going to compare our results with other researchers.

2. Literature Survey

We have referred research of some authors who has worked on predicting stock prices by using different methods. Dev Shah and his co-authors has done a comparative study of two very promising artificial neural network models namely a Long Short-Term Memory (LSTM) recurrent neural network (RNN) and a deep neural network (DNN) in forecasting the daily and weekly movements of the Indian BSE Sensex index. With both networks, measures were taken to reduce overfitting. Daily predictions of the Tech Mahindra (NSE: TECHM) stock price were made to test the generalizability of the models[1], in some studies we have also encountered that sentiments can also be a factor that affects the stock market[2].

Bi-directional LSTM network also improves the algorithm by 9% rate upon single pipeline deep learning model[3], Wang and his co-authors has done the prediction for the 5 days of stock opening and closing prices. As we know the sentiment analysis can affect the stock market, the social media sentiment in china does not work as same as other countries, since china has a totally different social media land-scape[7]. The combination of Random Forest using Lboost also outperforms the SVM method[8] and can be used applied successfully for building predictive models for stock prices prediction. Tweets mining from twitter also gives some input upon the predicting of stock prices[9] and its done using support vector machine and not by LSTM network since it has to perform the classification and association.

3. Conceptual Overview of the Project

In a Neural Network, each neuron receives a set of x-values (numbered from 1 to n) as an input and compute the predicted y-hat value as shown in figure 2. Vector x contains the values of the features in one of m examples from the training set. Each of units has its own set of parameters, usually referred to as w (column vector of weights) and b (bias) which changes during the learning process. In each iteration, the neuron calculates a weighted average of the values of the vector x, based on its current weight vector w and adds bias. Finally, the result of this calculation is passed through a non-linear activation function g. [5]

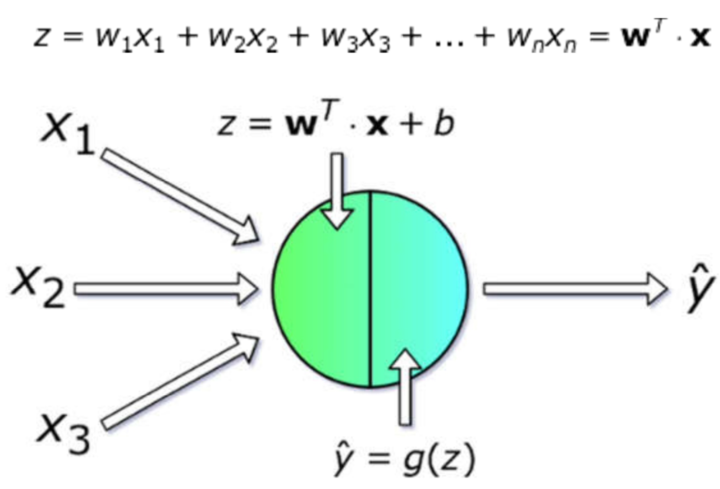

Figure 2. Single Neuron

3.1 Long Short Term Memory (LSTM)

Long short-term memory (LSTM) is an artificial recurrent neural network, (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections that make it a "general purpose computer" (that is, it can compute anything that a Turing machine can). It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).[5] For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. []

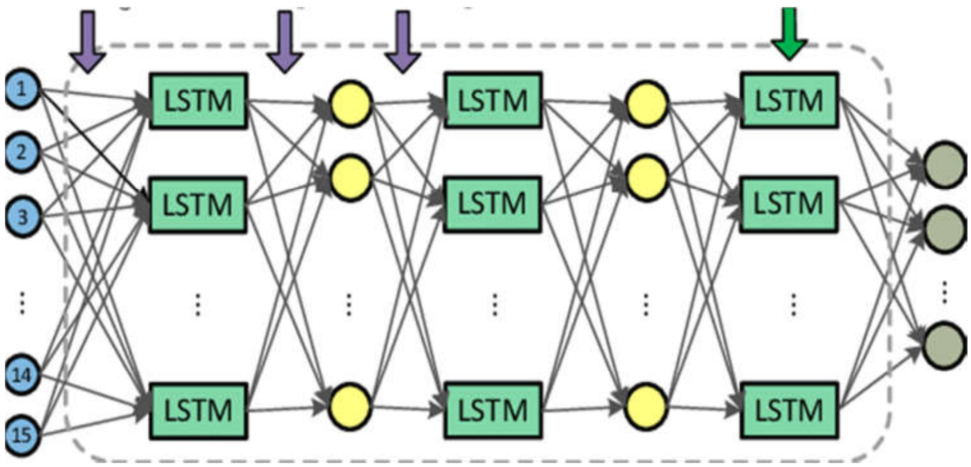

Figure 3. LSTM Network

3.2 Working of LSTM

The learning process is about changing the values of the W and b parameters so that the loss function is minimized. In order to achieve this goal, we will turn for help to calculus and use gradient descent method to find a function minimum. In each iteration we will calculate the values of the loss function partial derivatives with respect to each of the parameters of our neural network. Backpropagation is an algorithm that allows us to calculate a very complicated gradient, like the one we need. The parameters of the neural network are adjusted according to the following formulae.

$$\mathbf{W}^{[l]} = \mathbf{W}^{[l]} - \alpha \mathbf{dW}^{[l]}$$
$$\mathbf{b}^{[l]} = \mathbf{b}^{[l]} - \alpha \mathbf{db}^{[l]}$$

In the equations above, α represents learning rate - a hyperparameter which allows you to control the value of performed adjustment. Choosing a learning rate is crucial — we set it too low, our NN will be learning very slowly, we set it too high and we will not be able to hit the minimum. dW and db are calculated using the chain rule, partial derivatives of loss function with respect to W and b. The size of dW and db are the same as that of W and b respectively. Figure 9. shows the sequence of operations within the neural network. We can see clearly how forward and backward propagation work together to optimize the loss function.[5]

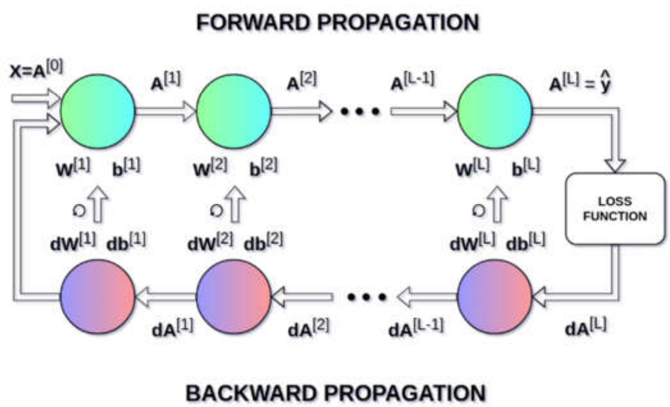

Figure 4. Forward and Backward Propagation

4. Results

We used the Apple Inc data set for the prediction, to keep things simple, we used the 80% of the dataset as training data set and used that dataset to predict the opening price of the rest 20% dataset.

We used the jupyter notebook, which is a python based ide used for executing python script line by line, which helps to explore the dataset more precisely

```
dataset_train = np.array(df[:int(df.shape[0]*0.8)])
dataset_test = np.array(df[int(df.shape[0]*0.8)-50:])
print(dataset_train.shape)
print(dataset_test.shape)
```

Implementing the LSTM alogorithm

```
tf.logging.set_verbosity(tf.logging.ERROR)
model = Sequential()
model.add(LSTM(units=96,return_sequences=True,input_shape=(x_train.shape[1],1)))
model.add(Dropout(0.2))
model.add(LSTM(units=96, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=96))
model.add(Dropout(0.2))
model.add(Dense(units=1))
```

First, we have settled the logging status of TensorFlow to Error only, so that it will prompt us only if there is error in applying the algorithm. Then we have called the LSTM function from Keras library and passed the required argument such as units, return sequence and input shape, where units is the means the dimension of the inner cells in LSTM. Because in LSTM, the dimension of inner cell (C_t and C_{t-1} in the graph), output mask (o_t in the graph) and hidden/output state (h_t in the graph) should have the SAME dimension, therefore you output's dimension should be unit-length as well. return_sequence is set to True because If you set return_sequence=True, it will return something with shape: (batch_size, timespan, unit). If false, then it just return the last output in shape (batch_size, unit) and since it is a 2 layer LSTM model we have to keep the status as true for 2 times and finally we passed the trained shaped we created after data pre-processing which is stored in a variable so that the final shape that we will get will be same as the input shape.

Then we also added the dropout function in the model, to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase.

A dense layer is just a regular layer of neurons in a neural network. Each neuron recieves input from all the neurons in the previous layer, thus densely connected. The layer has a weight matrix W, a bias vector b, and the activations of previous layer a.

The keras documentation has the docstring of class Dense:

output = activation(dot(input, kernel) + bias) where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer, and bias is a bias vector created by the layer.

```
if(notos.path.exists(r'C:\Users\riddz\Desktop\stock_prediction.data)):

    model.fit(x_train, y_train, epochs=50, batch_size=32)

    model.save(r'C:\Users\riddz\Desktop\stock_prediction.data)
```

The above condition will check if there is a saved model is there on the machine, if not it will create the model using the save function.

In figure 5, we can see the red curve is the original opening stock prices and the blue curve is the predicted opening stock prices. The figure 6, is the sub plot of the figure 5 which shows the true opening stock price and predicted opening stock prive

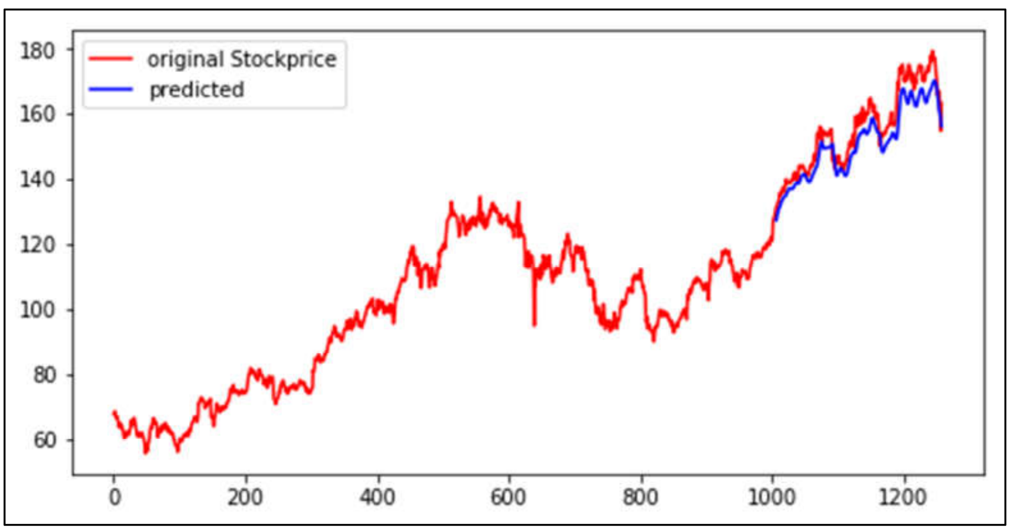

Figure 5. Result 1

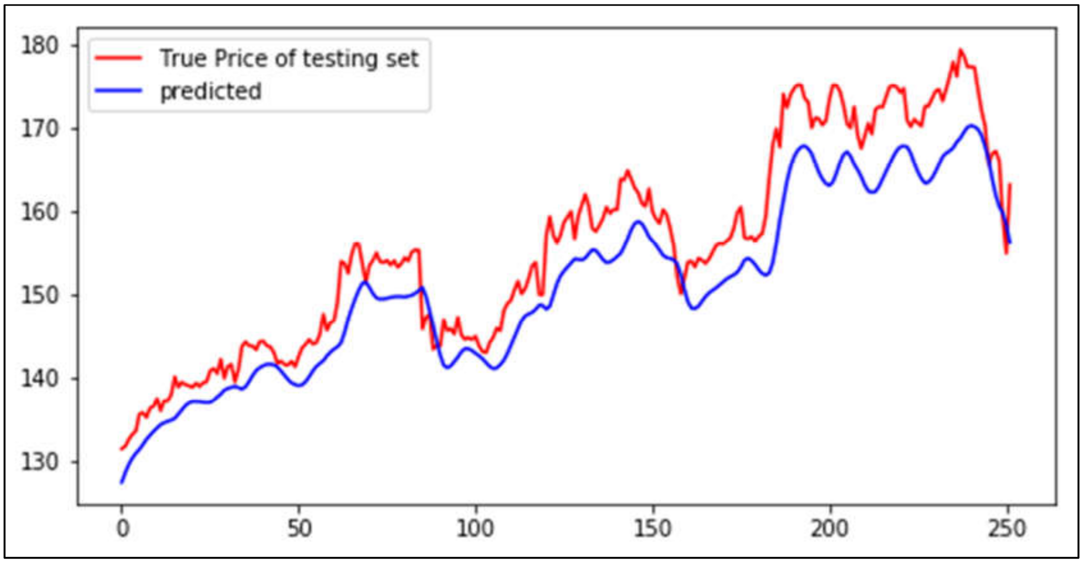

Figure 6. Result 2

5. Conclusion

After studying different research papers published nationally and internationally, we have seen many types of algorithms, techniques. We implemented LSTM using the Keras and TensorFlow which reduced the lines of codes which are actually required, if we implemented everything from scratch but we also faced some difficulties also such as customization of the model as we want which result in moderate results and we are going to tackle that problem by exploring the Keras and TensorFlow library, and bring more customization to our project. Although we learned a lot about the implementation of deep learning algorithms using the Keras and TensorFlow libraries. The efficiency of the algorithm can be further improved by working on the dropout coefficient and other factors that affects the stock prices such as sentiment analysis using SVM.

REFERENCES

[1] *Dev Shah, Wesley Campbell & Farhana H. Zulkernine, A Comparative Study of LSTM and DNN for Stock Market Forecasting, IEEE, 2018, 4148,4149 & 4155.*

[2] *Dev Shah, Wesley Campbell & Farhana H. Zulkernine, Predicting the Effects of News Sentiments on the Stock Market, IEEE, 2018.*

[3] *Jithin Eapen, Abhishek Verma & Doina Bein, Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction, IEEE, 2019, 264-266,268 & 269.*

[4] *Yang Longguang & Wang Qing, Predicting the stock price based on BP neural network and big transaction, IEEE, 2012.*

[5] *https://towardsdatascience.com/https-medium-com-piotr-skalski92-deep-dive-into-deep-networks-math-17660bc376ba*

[6] *https://www.economicshelp.org/blog/2841/economics/factors-affecting-the-stock-market/*

[7] *Tong Sun, Jia Wang, Pengfei Zhang, Yu Cao, Benyuan Liu & Degang Wang, Predicting Stock Price Returns using Microblog Sentiment for Chinese Stock Market, IEEE, 2017.*

[8] *Nonita Sharma & Akanksha Juneja, Combining of Random Forest Estimates using LSboost for Stock Market Index Prediction, IEEE, 2017.*

[9] *Siddhaling Urolagin, Text Mining of Tweet for Sentiment Classification and Association with Stock Prices, IEEE, 2017, 384, 386 & 387.*