Future Stock Price Prediction using Recurrent Neural Network, LSTM and Machine Learning

Shriram.S[1], Dr. K. Anuradha[2], Dr.K.P.Uma[3]
[1]Second year UG Student,
[1]Department of Computer Science and Engineering,
[1]Hindusthan College of Engineering and Technology,Coimbatore, India
[2]Associate Professor, Department of Computer Applications, Karpagam Academy of Higher Education, Coimbatore, India
[3]Professor and Head, Hindusthan College of Engineering and Technology, Coimbatore, India

***Abstract* - A stock market, equity market or share market is the aggregation of buyers and sellers ofstocks (also called shares), which represent ownership claims on businesses. The task of predicting stock prices is one of the difficult tasks for many analysts and in fact for investors. For a successful investment, many investors are very keen in predicting the future ups and down of share in the market. Good and effective prediction models help investors andanalysts to predict the future of the stock market. In this project, I had proposed Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) model by using Machine andDeep Learning models to predict stock market prediction. In present, there are several models to predict the stock market but they are less accurate. I had proposed a model that uses RNNand LSTM to predict the trend in stock prices that would be more accurate. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hiddenlayer of the network. In this work by increasing the Epochs and batch size, the accuracy of prediction is more. In proposed method, I am using a test data that is used to predict whichgives results that are more accurate with the test data. The proposed method is capable oftracing and prediction of stock market and the prediction will produce higher and accurateresults.**

Keywords: Stock Market Prediction, Recurrent Neural Network (RNN),Long Short Term Memory (LSTM), Epochs, batch size, Stock Price.

1. INTRODUCTION

Our project is recurrent neural network based Stock price prediction using machine learning.For a successful investment, many investors are very keen in predicting the future ups anddown of share in the market. Good and effective prediction models help investors andanalysts to predict the future of the stock market. In this project, I had proposed a RecurrentNeural Network (RNN) and Long Short-Term Memory (LSTM) model by using Machine and Deep Learning models to predict stock market prediction. In present, there are several modelsto predict the stock market but they are less accurate. I had proposed a model that uses RNNand LSTM to predict the trend in stock prices that would be more accurate. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in thehidden layer of the network. In this work by increasing the Epochs and batch size, theaccuracy of prediction is more. In proposed method, I am using a test data that is used topredict which gives results that are more accurate with the test data. The proposed method iscapable of tracing and prediction of stock market and the prediction will produce higher and accurate results.

2. METHODOLOGY:

(i) RNN (RECURRENT NEURALNETWORK)
(ii) LSTM (LONG SHORT TERMMEMORY)

(i) RNN(RECURRENT NEURAL NETWORK)

RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input. Unlike feed forward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other.

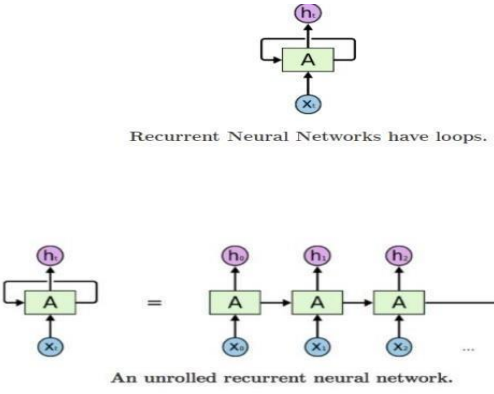


Figure 1 Recurrent Neural Network

(ii) LSTM (LONG SHORT TERMMEMORY)

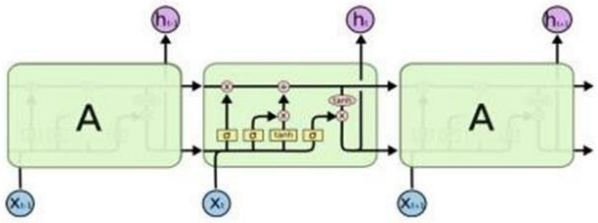


Figure 2 Long Short Term Memory

LSTM's have a Nature of Remembering information for a long periods of time is their Default behavior. look at the below figure that says Every LSTM module will have 3 gates named as Forget gate, Input gate, Output gate.

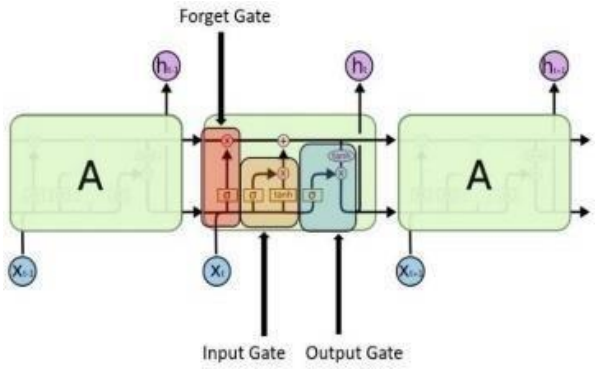
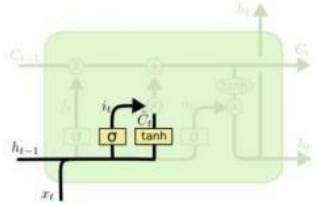


Figure 3 Long Short Term Memory Gates

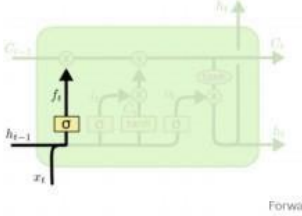
1) Input gate — discover which value from input should be used to modify the memory. Sigmoid function decides which values to let through 0,1. andtanh function gives weightage to the values which are passed deciding their level of importance ranging from-1 to1.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 4 Input gate

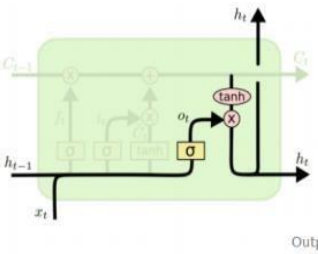
2)Forget gate — discover what details to be discarded from the block. It is decided by the sigmoid function. it looks at the previous state(ht-1) and the content input(Xt)andoutputs a number between 0(omit this)and 1(keep this)for each number in the cell state Ct−1.



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

Figure 5 Forget gate

3)Output gate — the input and the memory of the block is used to decide the output. Sigmoid function decides which values to let through 0,1. and tanh function gives weightage to the values which are passed deciding their level of importance ranging from-1 to 1 and multiplied with output ofSigmoid.



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 6 Output gate

3. DATA THAT IS USED IN THEMODEL:

In the Model Two types of data are being used. They are
(i) Train Data
(ii) TestData

(i) TrainData:
In the model Train Data of 4 Years of Google Stock Prices is used. This data will be used to train our model. We use epochs of about 200 to get more accuracy.

(ii) TestData:
In the Model Test data of 1 year of Google stock price is used to test our data. This data is used to test our model foraccuracy.

4. PROJECT IN PYTHONENVIRONMENT:

```
#Creating a GUI for our Stock price prediction

#importing tkinter module for creating GUI in python

from tkinter import *
from tkinter import ttk
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
from PIL import ImageTk, Image
import tkinter as tk

#=============================================================================================

#creating GUI Window

window=Tk()
window.geometry('1200x600')
window.title("Stock Price Prediction")
window.configure(bg="#247ba0")

#printing lines for uploading train data,test data

label_1 = Label(window, text = "Upload your train data\n\n\nUpload your test data\n\n\nClick
label_1.pack()


#Function for the button to upload train data
def trainfileopen():
    global filename1
    filename1 = askopenfilename(initialdir='C:\\',title = "Select file")

    import pandas as pd
    global file1

    file1 = pd.read_csv(filename1,index_col="Date",parse_dates=True)

#Function for the button to upload test data

def testfileopen():
    global filename2
    filename2 = askopenfilename(initialdir='C:\\',title = "Select file")
```

Figure 7 Project in python environment

Figure 8 Project GUI Interface

In the model the project in python environment the GUI Interface consists of three buttons (i) Open Training file,(ii) Open Test file, (iii) Click to get Graph of result of model, graph of predicted future 30 days and theclosevalueof predicted 30 days. Each button is written under a function definition in python environment. Where the first button is used to open the train data, the second media is used to open test data, the third button is used to run the RNN and LSTM and fit the model to the RNN. And the code is also written for the prediction of next future 30 days. There is also code for plotting the graph of result of the model, graph for predicted close prices for next future 30 days and there is also code written for displaying the values of predicted 30 days Close values.

```
#Creating a GUI for our Stock price prediction

#importing tkinter module for creating GUI in python

from tkinter import *
from tkinter import ttk
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
from PIL import ImageTk, Image
import tkinter as tk

#============================================================================================

#creating GUI Window

window=Tk()
window.geometry('1200x600')
window.title("Stock Price Prediction")
window.configure(bg="#247ba0")

#printing lines for uploading train data,test data

label_1 = Label(window, text = "Upload your train data\n\n\nUpload your test data\n\n\nClick
label_1.pack()


#Function for the button to upload train data
def trainfileopen():
    global filename1
    filename1 = askopenfilename(initialdir='C:\\',title = "Select file")

    import pandas as pd
    global file1

    file1 = pd.read_csv(filename1,index_col="Date",parse_dates=True)

#Function for the button to upload test data

def testfileopen():
    global filename2
    filename2 = askopenfilename(initialdir='C:\\',title = "Select file")
```

Figure 9 Functions for the buttons

```
# Initialising the RNN
regressor = Sequential()

#LSTM

regressor.add(LSTM(units = 60, return_sequences = True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 60, return_sequences = True))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 80, return_sequences = True))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 120 ))
regressor.add(Dropout(0.2))

# Adding the output layer
regressor.add(Dense(units = 1))


# Compiling the RNN
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')

# Fitting the RNN to the Training set
regressor.fit(X_train, y_train, epochs = 200, batch_size = 32)
```

Figure 10 Initializing the RNN

```
# demonstrate prediction for next 30 days
from numpy import array

lst_output=[]
n_steps=100
i=0
while(i<30):

    if(len(temp_input)>100):
        x_input=np.array(temp_input[1:])

        x_input=x_input.reshape(1,-1)
        x_input = x_input.reshape((1, n_steps, 1))
        yhat = regressor.predict(x_input, verbose=0)
        temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:]
        lst_output.extend(yhat.tolist())
        i=i+1
    else:
        x_input = x_input.reshape((1, n_steps,1))
        yhat = regressor.predict(x_input, verbose=0)
        print(yhat[0])
        temp_input.extend(yhat[0].tolist())
        print(len(temp_input))
        lst_output.extend(yhat.tolist())
        i=i+1

day_pred=np.arange(0,30)
```

Figure 11 Prediction of stock prices for future 30 days.

```
# Visualization
    plt.figure(figsize=(50,50))
    plt.subplot(231)
    plt.plot(y_test, color = 'red', label = 'Real Google Stock Price')
    plt.plot(y_pred, color = 'blue', label = 'Predicted Google Stock Price')
    plt.title('Result of the model of Prediction')
    plt.xlabel('Time')
    plt.ylabel('Google Stock Price')
    plt.legend()


    plt.subplot(233)
    lst_output=scaler.inverse_transform(lst_output)
    plt.plot(day_pred,lst_output)
    plt.title('Graph of predicted Stock Price of future 30 days')
    plt.xlabel('Time')
    plt.ylabel('Google Stock Price')


    df3=inputs.values.tolist()
    df3=scaler.inverse_transform(df3).tolist()
    df3.extend(lst_output)
    plt.subplot(235)
    plt.title('Graph of predicted Stock Price')
    plt.xlabel('Time')
    plt.ylabel('Google Stock Price')
    plt.plot(df3[1:])
    plt.show()

    for i in range (0,30):
        print("The Close Price of day",i+1,"is:",lst_output[i])
```

Figure 12 Plotting the graph

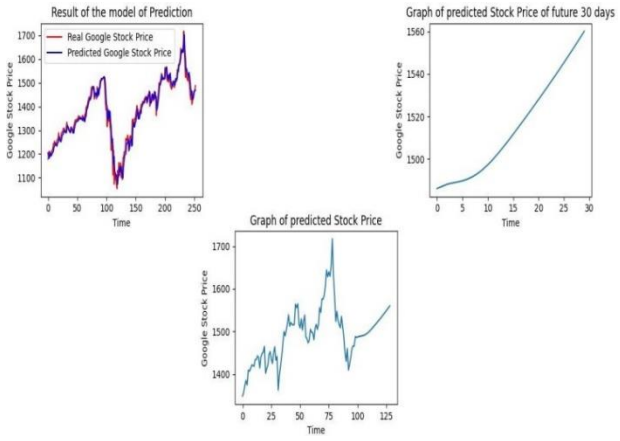


Figure 13 Resultant graph

5. RESULTS:

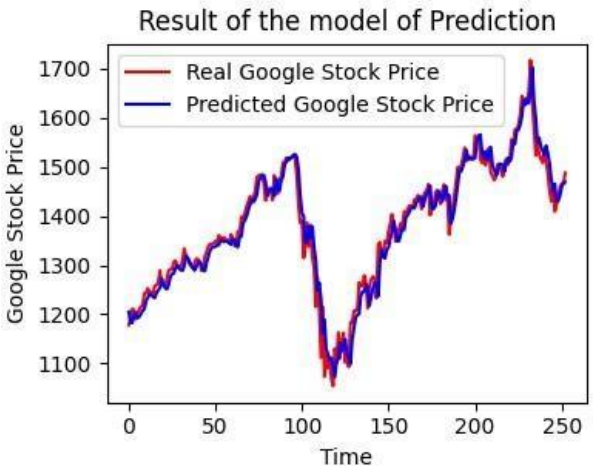


Figure 14 Resultant graph of model of prediction

From the above Graph we can see that the model has predicted the stock prices more accurately. The model has predicted the results more accurately. This graph compares the actual stock price with the predicted stock price and we can see that the model has predicted the stock prices more accurately.

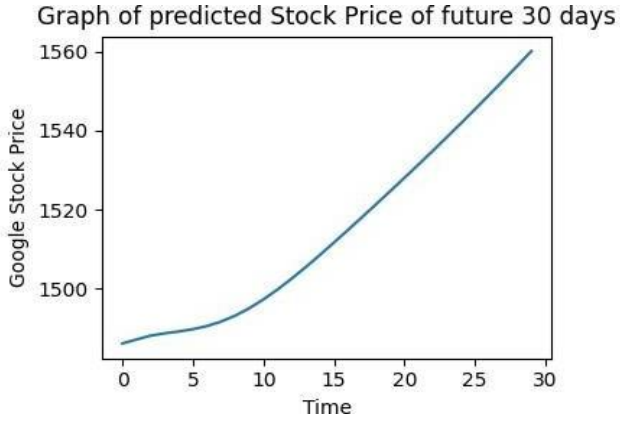


Figure 15 Graph of Predicted Stock Price of future 30 day

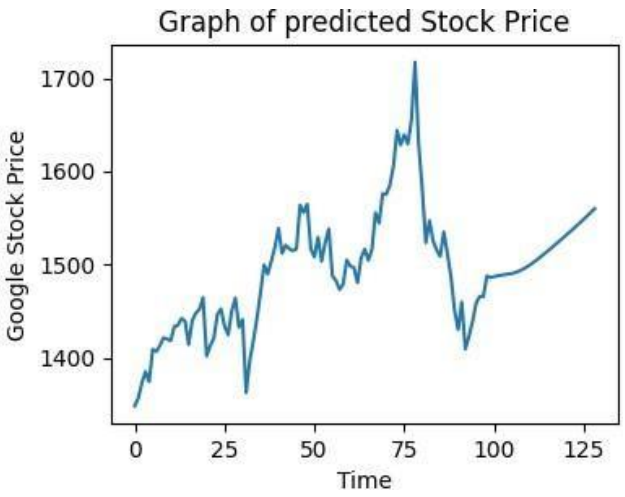


Figure 16 Graph of Predicted Stock Price

```
The Close Price of day 1 is: [1486.10257588]
The Close Price of day 2 is: [1487.11432863]
The Close Price of day 3 is: [1488.09404443]
The Close Price of day 4 is: [1488.67297476]
The Close Price of day 5 is: [1489.14612652]
The Close Price of day 6 is: [1489.71373065]
The Close Price of day 7 is: [1490.51890171]
The Close Price of day 8 is: [1491.65305826]
The Close Price of day 9 is: [1493.15871398]
The Close Price of day 10 is: [1495.03631383]
The Close Price of day 11 is: [1497.25551978]
The Close Price of day 12 is: [1499.76722471]
The Close Price of day 13 is: [1502.51423131]
The Close Price of day 14 is: [1505.44015134]
The Close Price of day 15 is: [1508.49543268]
The Close Price of day 16 is: [1511.64043366]
The Close Price of day 17 is: [1514.84615114]
The Close Price of day 18 is: [1518.09438229]
The Close Price of day 19 is: [1521.3751358]
The Close Price of day 20 is: [1524.68566102]
The Close Price of day 21 is: [1528.02741417]
The Close Price of day 22 is: [1531.40460214]
The Close Price of day 23 is: [1534.82199809]
The Close Price of day 24 is: [1538.28453702]
The Close Price of day 25 is: [1541.79581903]
The Close Price of day 26 is: [1545.35774531]
The Close Price of day 27 is: [1548.97072037]
The Close Price of day 28 is: [1552.63344978]
The Close Price of day 29 is: [1556.34294019]
The Close Price of day 30 is: [1560.09518699]
```

Figure 17 The Close prices of predicted 30 days

6. NOVELTY OF THEPROPOSED MODEL:

The quality of being new in my project is that in my Stock Price prediction model the prediction is more accurate than other existing models and my project is also different in a way that I had created a Graphical User Interface (GUI) where we can upload the Train data, Test Data and we can get the result of the model and the future 30 days predicted graph with the stock prices. In my Project I had created to give a result graph which consists of the Future 30 days

Predicted Close Stock Prices. And also in the model there is a special feature where it can display the close values of the predicted future 30 days.

7. BACKGROUND OF THEPROPOSED MODEL:

The Field of the proposed model is ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, DEEP LEARNING and my model is FUTURE STOCK PRICE PREDICTION USING RECURRENT NEURAL NETWORK, LSTM AND MACHINE LEARNING. TheStock Price prediction model can predict more accurate than other existing models and my project is also different in a way that I had created a Graphical User Interface (GUI) where we can upload the Train data, Test Data and we can get the result of the model and the future 30 days predicted graph with the stock prices. In my Project I had created to give a result graph which consists of the Future 30 days Predicted Close Stock Prices. In my project the user can get the future 30 days predicted Close prices of Stock prices. And also in the model there is a special feature where it can display the close values of the predicted future 30days.

8. ADVANTAGE OF THEMODEL:

The main Advantage is that since the model uses RNN, LSTM, Machine Learning and Deep Learning models the prediction of stock prices will be more accurate. And also in the model it can predict the future 30 days Stock Prices and it can show it in a graph. Also the main feature is that the model can show an output of the Individual Predicted Close prices of the Predicted 30 days as shown in the figurebelow.

```
The Close Price of day 1 is: [1486.10257588]
The Close Price of day 2 is: [1487.11432863]
The Close Price of day 3 is: [1488.09404443]
The Close Price of day 4 is: [1488.67297476]
The Close Price of day 5 is: [1489.14612652]
The Close Price of day 6 is: [1489.71373065]
The Close Price of day 7 is: [1490.51890171]
The Close Price of day 8 is: [1491.65305826]
The Close Price of day 9 is: [1493.15871398]
The Close Price of day 10 is: [1495.03631383]
The Close Price of day 11 is: [1497.25551978]
The Close Price of day 12 is: [1499.76722471]
The Close Price of day 13 is: [1502.51423131]
The Close Price of day 14 is: [1505.44015134]
The Close Price of day 15 is: [1508.49543268]
The Close Price of day 16 is: [1511.64043366]
The Close Price of day 17 is: [1514.84615114]
The Close Price of day 18 is: [1518.09438229]
The Close Price of day 19 is: [1521.3751358]
The Close Price of day 20 is: [1524.68566102]
The Close Price of day 21 is: [1528.02741417]
The Close Price of day 22 is: [1531.40460214]
The Close Price of day 23 is: [1534.82199809]
The Close Price of day 24 is: [1538.28453702]
The Close Price of day 25 is: [1541.79581903]
The Close Price of day 26 is: [1545.35774531]
The Close Price of day 27 is: [1548.97072037]
The Close Price of day 28 is: [1552.63344978]
The Close Price of day 29 is: [1556.34294019]
The Close Price of day 30 is: [1560.09518699]
```

Figure 18 The Close prices of predicted 30 days

9. CONCLUSION:

In present, there are several models to predict the stock market but they areless accurate. We proposed a model that uses RNN and LSTM to predict the trend instock prices that would be more accurate. LSTM introduces the memory cell, a unit ofcomputation that replaces traditional artificial neurons in the hidden layer of thenetwork. In this work by increasing the Epochs and batch size, the accuracy ofprediction is more. In proposed method, we are using a test data that is used to predictwhich gives results that are more accurate with the test data. The proposed methodiscapable of tracing and prediction of stock market and the prediction will producehigher and accurate results. In our above model we are getting accurate results which willbe more useful to stock analysts, Business analysts, Stock Market Investors.

REFERENCES

[1] Batres-Estrada, B. (2015). Deep learning for multivariate financial time series.
[2] Emerson, S., Kennedy, R., O'Shea, L., & O'Brien, J. (2019, May). Trends and Applications of Machine Learning in Quantitative Finance. In 8th International Conference on Economics and Finance Research (ICEFR 2019).
[3] Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. Applied Stochastic Models in Business and Industry, 33(1), 3-12.
[4] Moritz, B., & Zimmermann, T. (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. Available at SSRN 2740751.
[5] Olah, C. (2015). Understanding lstm networks–colah's blog. Colah. github. io.
[6] Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2018). Decision-Making for Financial Trading: A Fusion Approach of Machine Learning and Portfolio Selection. Expert Systems with Applications.
[7] Patterson J., 2017. Deep Learning: A Practitioner's Approach, O'Reilly Media.
[8] Siami-Namini, S., &Namin, A. S. (2018). Forecasting economics and financial time series: Arima vs. lstm. arXiv preprint arXiv:1803.06386.
[9] Takeuchi, L., & Lee, Y. Y. A. (2013). Applying deep learning to enhance momentum trading strategies in stocks. In Technical Report. Stanford University.
[10] Wang, S., and Y. Luo. 2012. "Signal Processing: The Rise of the Machines." Deutsche Bank Quantitative Strategy (5 June).