Stock Price Prediction using Multi Layered Sequential LSTM

Jyoti Prakash Behura[a] , Sagar Dhanraj Pande[a]

[a] SCOPE, VIT-AP University
{jyotiprakash.20bce7355@vitap.ac.in}
{sagar.pande@vitap.ac.in}

Abstract. Stock markets are frequently among the most volatile locations to invest in. The choice to buy or sell stocks is heavily influenced by statistical analysis of prior stock performance and external circumstances. All of these variables are employed to maximize profitability. Stock value prediction is a hard undertaking that necessitates a solid computational foundation in order to compute longer-term share values. Stock prices are connected inside the market, making it harder to forecast expenses. Financial data is a category that includes past data from time series that provides a lot of knowledge and is frequently employed in data analysis tasks. This research provides a unique optimisation strategy for stock price prediction based on a Multi-Layer Sequential Long Short Term Memory (MLS LSTM) model and the adam optimizer in this context. Furthermore, in order to make reliable predictions, the MLS LSTM algorithm uses normalised time series data separated into time steps to assess the relationship between past and future values. Furthermore, it solves the vanishing gradient problem that plagues basic recurrent neural networks.

Keywords. Forecasting; deep learning; stock prices; RNN; CNN; MLS-LSTM; Adam optimizer; NSE; ARIMA; SVM

1 Introduction

The tendency of stock price fluctuation has long been one of the biggest problems in the financial industry. Several internal and external variables, such as the national and international economic climate, current affairs, company prospects, monetary data from openly traded firms, and the market's activity, all have an impact on stock prices. The behavior of the stock market is examined using the conventional analysis method. Fundamental analysis and technical analysis are the two approaches on which traditional analysis is based. The primary evaluation technique gives a significant amount of weight to external variables such as rates of interest, currency rates, inflation tendencies, policies pertaining to industry, ties to other nations, economics, and matters of politics. The price of the stock at any given time is highly influenced by both supply and demand. The value rises with increasing demand and vice versa. The stock market is where equities are bought and sold, allowing the interaction between buyers and sellers and conduct business with one another. To effectively predict the future patterns and value of stocks, one must measure and assess them., many experts and analysts have been working on this problem.

This model uses the RNN approach known as Long Short-Term Memory and takes into account the previous stock value of a corporation. The suggested method makes predictions on a certain attribute based on the share's historical information that is currently accessible.

The recommended approach adopts utilizing historical series analysis to anticipate a share value for the chosen time frame. The National Stock Exchange of India Limited (NSE), an Indian stock exchange company, will be taken into consideration in the proposal. The NSE was the first exchange in India to offer sophisticated, cutting-edge facilities to investors dispersed throughout the whole nation. It is entirely contemporary and furnished with the most current conveniences, enabling investors to conduct business from any location in India. This is crucial in changing the Indian equities market to boost the capital market's transparency, convergence, and efficiency. In order to forecast future movements of stock prices, technical analysis primarily relies on an analysis of historical and present stock prices.

Machine learning models have long been used in the financial sector, particularly in the stock market. With various degrees of accuracy, several techniques such as ordinary least squares, ARIMA, SVM and were used. Stock values may be extremely unpredictable or represented as sophisticated time series frameworks, which makes using simple ML concepts like regression to estimate stock prices difficult.

A deep learning approach will assist overcome this problem., such as the LSTM approach, which combines the stock's history price into the forecast of future value.

With the help of LSTM, long-term relationships in data may be learned, allowing for incredibly precise prediction. The research promotes the use of the MLS-LSTM method for forecasting shares. A Recurrent Neural Network can extrapolate knowledge from past data and use it to forecast future events., and the LSTM technique is one such RNN. Long-term data dependencies cannot be managed by traditional RNN. Additionally, there is no precise control over how much of the previous context the network must forget and how much information in data is transferred forwards. Another significant disadvantage of traditional RNN is the issue of vanishing gradient.

LSTM networks overcome all of these problems. LSTM networks can recall patterns selectively for lengthy sequences. This enables it to understand the data's dependence on previous values and make more accurate predictions. LSTM algorithm has tight control over selective forgetting of the data's previous context and totally eliminate problem of vanishing gradients.

2 Related Work

Identifying stock returns has become a prominent subject of study in the last two decades. While conducting the literature review, facts about stock market prediction systems that are currently in use are taken into account.

According to RautSushrut et al. [1], supervised learning classifiers might be used to assess the capabilities of market gauge information as well as estimate stock price movement. Portfolio modeling has been used in the financial market as a computational analytical method. The paper addresses an analytical AI strategy and shows the application of tactical ways to forecast shares along with SVM technique.

The CNN-LSTM method was used by Lu et al. [2] to predict the Shanghai Index's close price for the following day (000001). The LSTM technique was used to predict the close stock price, and CNN's primary goal was to get the best qualities from the data. CNN was having trouble getting the most effective feature out of the raw data.

The authors of [3] conducted an extensive analysis on the necessity of data set algorithm in predicting shares. The National Stock Exchange (NSE) of India's official web page provided the data for the study. and included information on 56 companies from seven different industries. In order to make a prediction, an ARIMA model was applied. The findings obtained demonstrated that across all industries, the algorithm had a share estimation accuracy of more than 85%.

Jordan Prosky et al. [4] recommended using sentiment analysis to stock prediction using CNN approaches and their use in predicting stock prices.

The authors in [5] used Support Vector Machine (SVM) and Linear Regression models [6] to perform a stock price forecasting study. R was used for the analysis, and RStudio was used as the setting for development. Dataset was obtained from the beginning of 2017 to the end of 2018 and spanned exactly one year.

According to X. Shao and D. Ma [7], it is a more adaptable variant of the controlled repetitive system. LSTMs are thought to be less harmful than other deep learning approaches like RNNs or traditional feed forwards neural networks since they solve the evanescent gradient issue that RNNs have. The use of LSTM in combination with K-means algorithms for brief forecasting of stock systems is also covered.

Li et al.'s team used a hybrid deep learning approach to forecast the closure price of two separate sets of data, JQData and Pingan Bank.CNN's main objective was to extract the most useful characteristics from the provided data. The CNN-LSTM model's output is used as an input by the LSTM model, This follows with computations to predict the price of shares and modifies the focus method to make it more scalable. SVM, LSTM, and GRU findings were used to analyse the results of the experiment using the CNN-LSTM model. The CNN-LSTM model outperformed the other models, according to the findings.

3 LSTM

A special type of recurrent neural network called long short term memory (LSTM) employs cells with memories to store crucial knowledge over time. This property makes LSTMs a superior technique for learning linked sequences.

Vanishing Gradient issue plagues all deep artificial neural networks. Beginning at the output layer, back propagation moves back towards the input layer while sequentially modifying each layer's weight based on the gradient of the cost function. Simply expressed, the weights are changed to ensure that the output layer error is as small as possible.

If the weight values are less than one, the gradient will continue to decrease in each subsequent layer, resulting in no change in the weights for distant layers. Because it is closest to the output layer, the weights of the last hidden layer are adjusted the most; the weights of the second-to-last hidden layer are adjusted a little less because it is farther from the output layer; and the further we move from the output layer, the less weights are adjusted due to the error's diminishing gradient. The problem of Vanishing gradient is the name given to this topic in general.On the other hand, if the weight values are bigger than one, the gradient will continue to rise in the subsequent layers of backpropagation, causing significant changes in the weights.

A typical LSTM network has many memory blocks, also known as cells. Each cell transfers the concealed state and the cell state to the following cell.

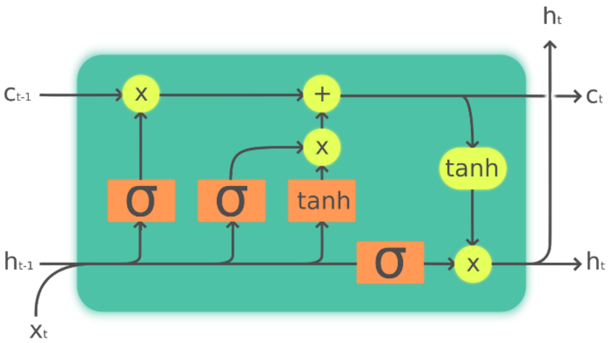


Fig 1. Memory cell (Source: https://miro.medium.com/max/3200/0*nukrZzIzKICTAfST)

The forget gate is used to delete extraneous information while storing critical information in the cell state. Equation(1) illustrates how it employs the sigmoid function to calculate two values— the latest value generated of the prior state and the present value provided —to provide the forget gate result value, which ranges from 0 to 1.

$$F_t = \sigma((W_F * \ P_{t-1}) + (W_F * i_t) + \ y_F)(1)$$

At that particular time step, the data being provided is i_t, and P_{t-1} is the value of the most recent previous state. W_F stands for weight, and y_F stands for bias. F_t is a representation of the forget gate activation function output. Equation (4) shows this process, The output of the forget gate activation function gets multiplied by the previous state of the cell to decide if previous information should be retained within the cell.

The input gate is used to feed the cell state with new data. Both the most recent state value and the current input must be provided. Equation (2) illustrates how the sigmoid operation is employed as a filter to provide a result that ranges from 0 to 1 after the calculation is done on the input data.

$$I_t = \sigma((W_I * P_{t-1}) + (W_I * i_t) + y_I) \text{ (2)}$$

where I_t is the output of the input gate, W_I is the weight, $P_{t-1}$is the value of the most recent previous state, i_t, at this particular time step, the input is y_I is the bias.

To get access to every potential cell state value, a new cell function is developed. The latest end state value as well as the present input value are both arguments to this function. Equation (3) demonstrates how the computation is performed using the input values and how a filter called the tanh activation action is utilised to produce an output in the range of -1 and 1.

$$\tilde{S}_t = tanh((W_I * P_{t-1}) + (W_I * i_t) + y_I) \text{ (3)}$$

P_{t-1} is the value of the most recent final state, i_t at that particular time step, is the information being provided, y_I is the bias, and $\tilde{S}_t$ is the value of the cell state function at that time step, where W_I is the weight.

Equation (4) illustrates the multiplication and addition of the cell state with the results of the input gate return and cell operation.

$$S_t = S_{t-1} * F_t + I_t + \tilde{S}_t \text{ (4)}$$

The output gate's finest characteristic is provided as an output is the primary goal. The output gate does the calculation utilizing the most recent state value at that point as well as the current input value. Equation 5 shows how to filter The output value is calculated using the sigmoid activation function that operates on a value between 0 and 1.

$$O_t = \sigma((W_{og} * P_{t-1}) + (W_{og} * i_t) + y_{og}) \text{ (5)}$$

where W_O is the weight, P_{t-1} is the output value at that time step, y_O is the bias, and O_t is the weighted output value.

The tanh activation feature on the component state is used to construct a new component state function. to filter values between -1 and 1, as indicated in Equation (6).

$$\widetilde{OS}_t = \tanh\,(S_t)\ (6)$$

where the output value of the function cell state is $\widetilde{OS}_t$ and S_t is designated as the cell state.

Equation (7) illustrates how to multiply the output value by the value of the cell state function and transfer the result to the next hidden layer.

$$O_{og} = O_t * \widetilde{OS}_t \quad (7)$$

4 Proposed Methodology

The research design used for this article is a time series analysis to predict the close price for the following day or days. The dataset used is TATA Consumer and it was obtained from the National Stock Exchange website. We will only use the date and the closing price columns. The last 5 days were used as the testing data and the remaining were used for training the model. We will first use basic LSTM model and then follow it with deep learning technique. The steps are discussed in details below.

4.1 Importing Data

The TATA Consumer dataset was collected from the NSE website.

4.2 Visualizing the stock prices movement

The following dataset was visualized using line graph with the closing price as Y-axis and the date as X-axis.

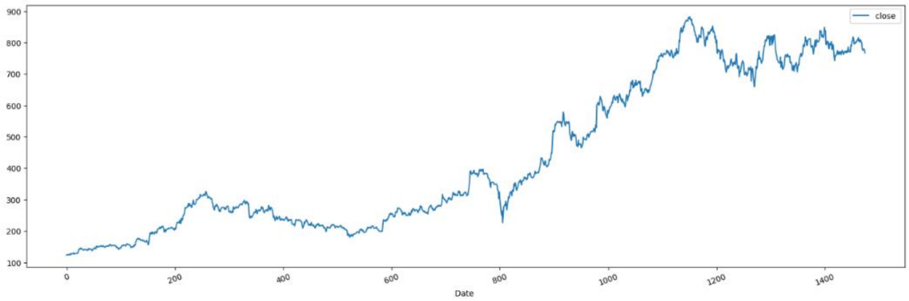
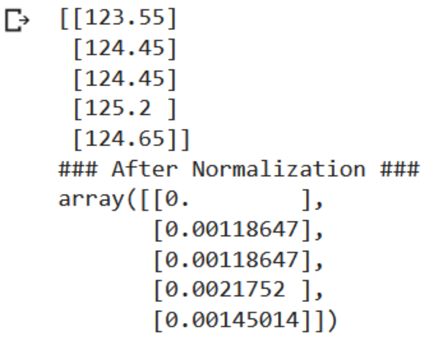


Fig 2. Stock Price Movement of TATA consumer from 1st Jan 2018 to 31st Dec 2022

4.3 Preparing the data

Data input for the LSTM model will be in the form of X Vs y. where X represents the pricing from the previous ten days and Y the price from the following day. The LSTM is expected to be capable to understand price fluctuations by examining many examples like these from the previous two years. As a result, once the price has passed the previous 10 days. It should be possible to predict the closing price of the company's shares the following day. Since the LSTM approach is based on neural networks, for a quicker and more accurate fit, the data must be normalized or standardized.

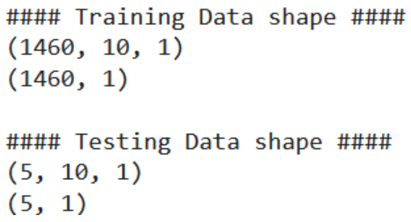


```
[[123.55]
 [124.45]
 [124.45]
 [125.2 ]
 [124.65]]
### After Normalization ###
array([[0.        ],
       [0.00118647],
       [0.00118647],
       [0.0021752 ],
       [0.00145014]])
```

Fig 3. Values of closed price after normalization

4.4 Dividing the data into test and training

Utilizing the remaining data to train the model and the most current few days of information to evaluate the model's learnings. I'm putting the last five days to the test here.



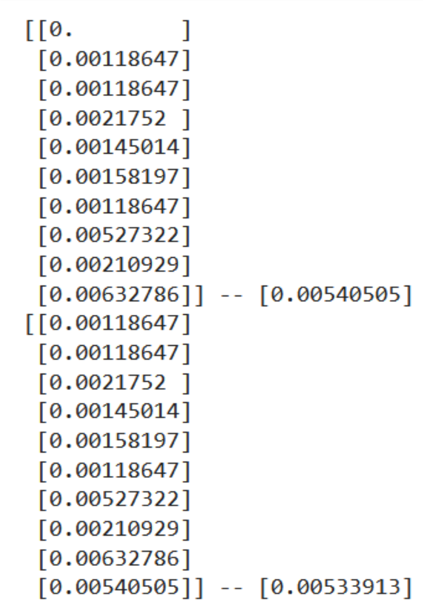
```
#### Training Data shape ####
(1460, 10, 1)
(1460, 1)

#### Testing Data shape ####
(5, 10, 1)
(5, 1)
```

Fig 4. Size of Training and Testing Data Set

4.5 LSTM input and output data visualisation

In order to better understand how the LSTM model will figure out the prices, print some sample input and output values.



```
[[0.        ]
 [0.00118647]
 [0.00118647]
 [0.0021752 ]
 [0.00145014]
 [0.00158197]
 [0.00118647]
 [0.00527322]
 [0.00210929]
 [0.00632786]] -- [0.00540505]
[[0.00118647]
 [0.00118647]
 [0.0021752 ]
 [0.00145014]
 [0.00158197]
 [0.00118647]
 [0.00527322]
 [0.00210929]
 [0.00632786]
 [0.00540505]] -- [0.00533913]
```

Fig 5. Sending 10 entries through a LSTM model to predict for the next day

4.6 Construction of the LSTM deep learning model

Consider the definition of the hidden layers using the LSTM function as opposed to Dense. We have one neuron in the output layer since we are foreseeing pricing for the following day. One output layer, three hidden LSTM layers were used. The model becomes slower as you add more neurons and layers, because there must now be many more calculations made.

There are a few hyperparameters for each layer that need to be adjusted.

units=10: indicates construction of layer that contains number of nodes. The input values will be sent to each of these five neurons.

input_shape = (TimeSteps, TotalFeatures): The 3D format of the input is what LSTM anticipates. Number of specimens, number of intervals, and number of attributes make up our training data, and has the shape (1455, 10, 1). This means that there are 1455 examples in the training data that we need to learn from. The share price from the previous day, the day before that, and on and so forth, up to the most recent 10 days, are some examples of the 10 steps in time each sample looks back. Time steps are what we call this. The number of features is indicated by the last number, "1."

kernel initializers= “uniform”: Before the Nodes can begin computing, The numerical value for every weight must be determined using an algorithm. That is made evident by this attribute. You may change it to different settings, such "normal" / "glorot_uniform."

'relu' : The calculation's function of activation carried out by every node is specified by the attribute. You can select values such as "relu," "tanh," "sigmoid," etc.

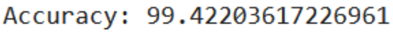
return_sequences=True : Since LSTMs backpropagate through time, they pass the values of the output from each time step to the following hidden layer. This maintains the 3D format of the anticipated input for the following hidden layer. This is False for final hidden layer.

The optimizer='adam' determines the appropriate weight figures for each of the nodes in the neural network with the help of the parameter. 'adam' is a beneficial efficiency enhancer.

Epochs=10: As indicated by this parameter, the same weight-adjusting activity is repeated 10 times. To put it simply, the LSTM adjusts its weights after examining the entire training set ten times.

4.7 Evaluating the model's precision using test data

We are currently determining whether or not the forecasted costs for the past five days closely match the real costs using the trained model. Keep an eye out for the predictions' inverse transform. Because we normalised the data prior to model training, the forecasts based on the test data will likewise be normalised. The accuracy percent is then calculated.



```
Accuracy: 99.42203617226961
```

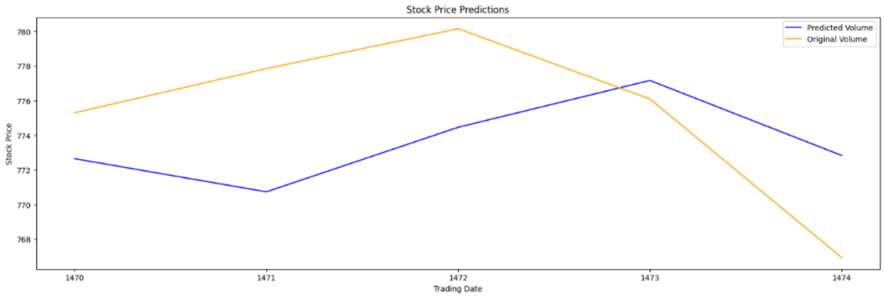


Fig 6. Testing Trained model on the testing dataset

4.8 Displaying the predictions for the entire dataset

Plotting both the training and test sets of data to evaluate how well the LSTM model fits.

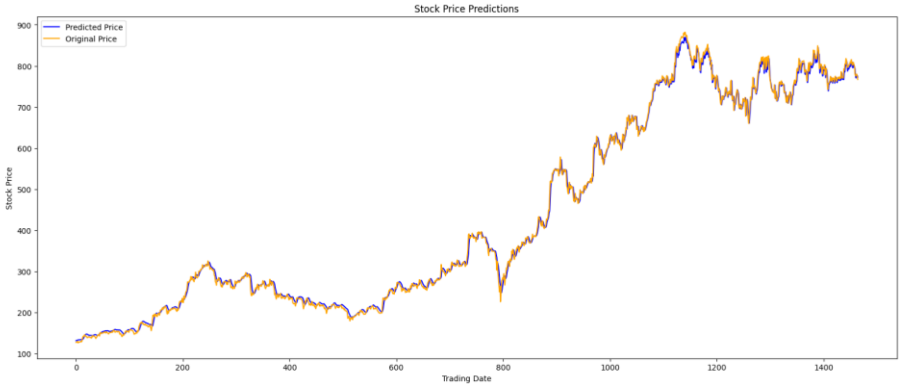
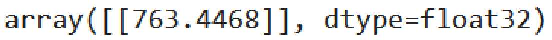


Fig 7. Prediction analysis of the model

4.9 How to forecast the price of stocks for tomorrow

Simply pass the model the prices from the previous 10 days If you wish to forecast the price for the future, do it in three-dimensional structure like it was done throughout training.

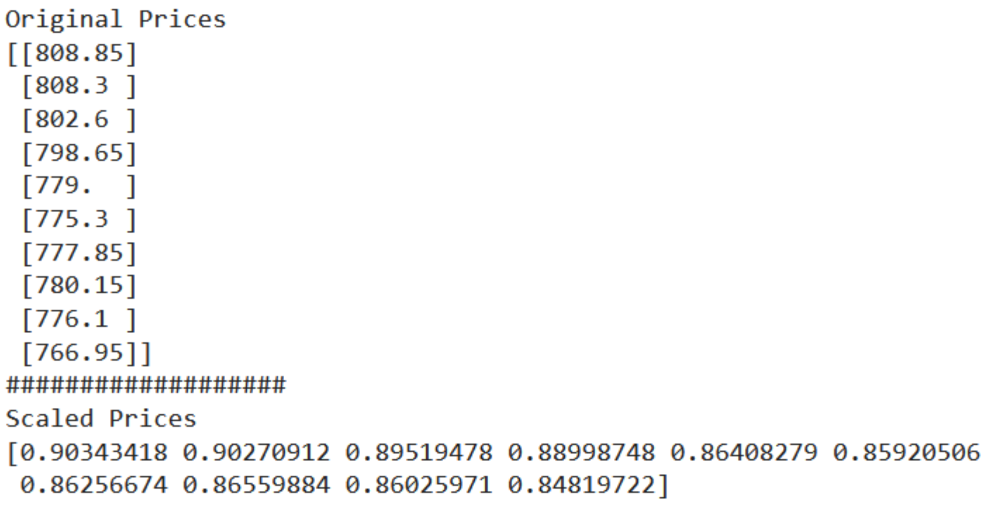


```
array([[763.4468]], dtype=float32)
```

Fig 8. Output of the model

4.10 Data Preparation for Multi Step LSTM

By setting **FutureTimeSteps=5**, The information division rules from the previous model was modified to create pairings of the inputs and outputs. This establishes that we want to forecast prices for the next five days using data from the previous ten.

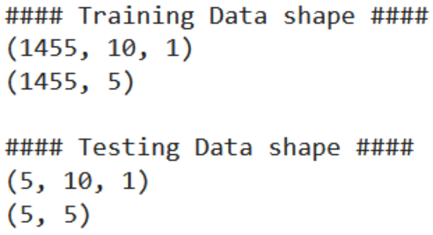


```
Original Prices
[[808.85]
 [808.3 ]
 [802.6 ]
 [798.65]
 [779.  ]
 [775.3 ]
 [777.85]
 [780.15]
 [776.1 ]
 [766.95]]
###################
Scaled Prices
[0.90343418 0.90270912 0.89519478 0.88998748 0.86408279 0.85920506
 0.86256674 0.86559884 0.86025971 0.84819722]
```

Fig 9. Sample data for LSTM multi step stock prices prediction

4.11 LSTM Multi-step model input-output visualization

A LSTM model's process is always easier to understand if some records of input and output are printed. A three-dimensional set of prices from the ten days prior serves as an input, while a set of prices from the next 5 days serves as the output.



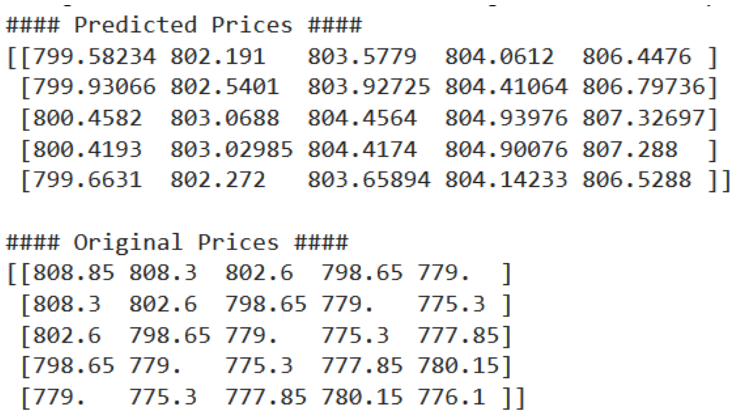
```
#### Training Data shape ####
(1455, 10, 1)
(1455, 5)

#### Testing Data shape ####
(5, 10, 1)
(5, 5)
```

Fig 10. Splitting the dataset for training and testing

4.12 The development of the DL MLS-LSTM algo

Similar set ups as the previous model are used. Dense layer is where the change is made. The dense layer now produces a value output that is equal to the FutureTimeSteps. which in this instance is 5 because we're trying to forecast the next 5 days.



```
#### Predicted Prices ####
[[799.58234 802.191   803.5779  804.0612  806.4476 ]
 [799.93066 802.5401  803.92725 804.41064 806.79736]
 [800.4582  803.0688  804.4564  804.93976 807.32697]
 [800.4193  803.02985 804.4174  804.90076 807.288  ]
 [799.6631  802.272   803.65894 804.14233 806.5288 ]]

#### Original Prices ####
[[808.85 808.3  802.6  798.65 779.  ]
 [808.3  802.6  798.65 779.   775.3 ]
 [802.6  798.65 779.   775.3  777.85]
 [798.65 779.   775.3  777.85 780.15]
 [779.   775.3  777.85 780.15 776.1 ]]
```

Fig 11. Each row represents the original prices and the predicted prices.

4.13 Evaluating the model's accuracy using test data

Given that this is a multi-step model, it can predict the next five days. Five days' worth of prices will be generated by each prediction, which we can compare to the initial prices.

Each row will be compared separately.

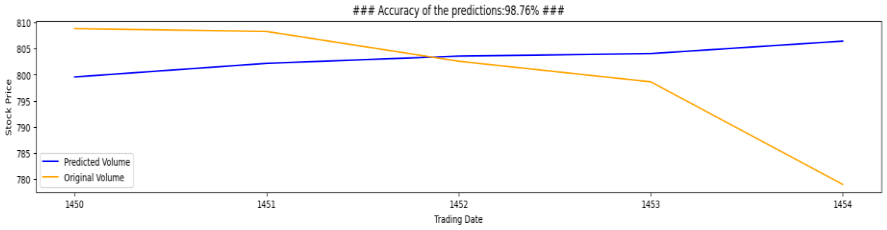


Fig 12. Prediction accuracy for the first row

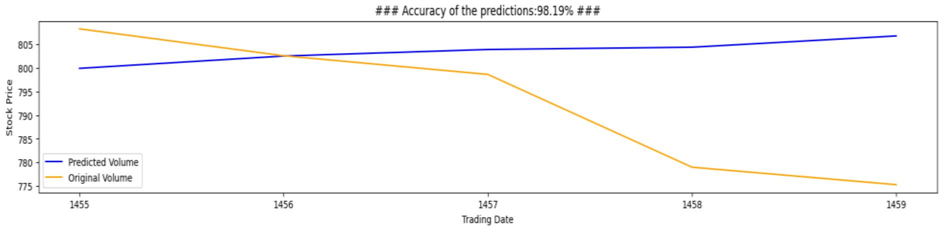


Fig 13 Prediction accuracy for the second row

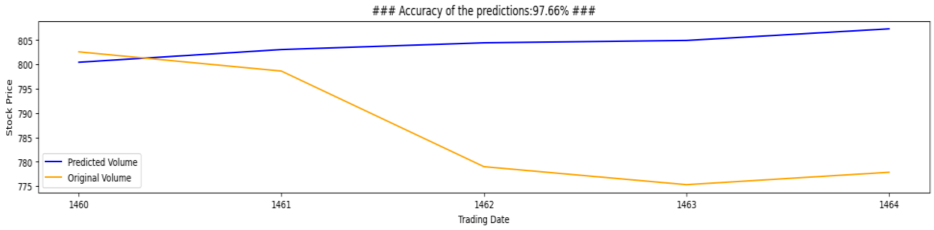


Fig 14. Prediction accuracy for the third row

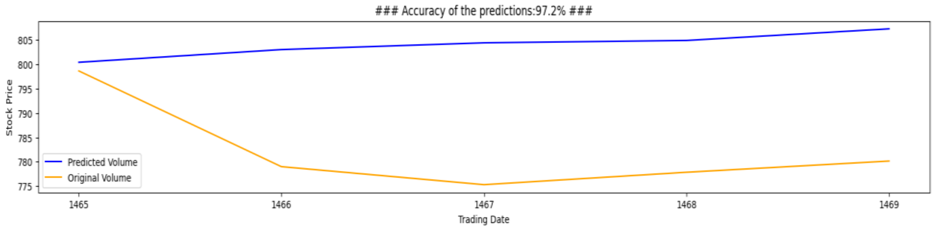


Fig 15. Prediction accuracy for the fourth row

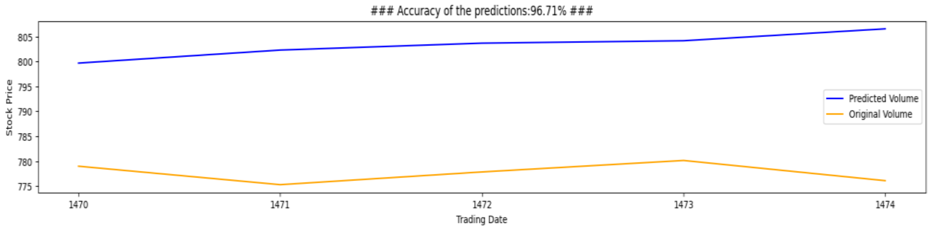


Fig 16. Prediction accuracy for the fifth row

The initial values and the forecasted prices are shown in each row.

Each row will be compared separately.

5 Results

The algorithm had an accuracy of 99.42% for measuring the model and the accuracy of the testing data varied from 98.76 to 96.71%. the results were better while using ‘adam’ optimizer and 3 hidden layers for LSTM.

The process was repeated for two and four hidden layers. The outputs are given in the below table:

Table 1: Results for different number of hidden layers

Number of Hidden Layers	Accuracy for measuring model	Accuracy of the testing data
2	99.12%	98.99% - 97.57%
3	99.42%	98.76% - 96.71%
4	98.91%	98.44% - 97.75%

We can use more or less hidden layers as the accuracy are almost similar although there is a trend of less accuracy once the hidden layers increase.

6 Conclusion

This forecast only covers the near future. This fails miserably when you attempt to forecast for periods of time longer than one day, such as the upcoming 30 or 60 days. Stock markets are extremely volatile, not because our LSTM model is flawed. So don't put all of your money on this model. Before adopting this model as a supplementary tool for analysis, do some study.

There will be basic evaluation and sentiment assessment incorporated into the forecasting process in an effort to create a future prediction system that is more effective and trustworthy. By doing this, we will be able to determine how the public generally feels about the company, allowing us to adjust our prediction method accordingly and produce future predictions that are even more precise. Training accuracy can also be improved by altering how Unavailable info is managed, i.e., by omitting empty fields with the median or rolling mean of a given set of data as input. It is necessary to make an effort to cut down on the amount of time and resources needed for LSTM training.

7 References

[1] Raut Sushrut Deepak, Shinde Isha Uday, Dr. D. Malathi, “Machine Learning Approach In Stock Market Prediction”, IJPAM 2017

[2] Lu, W.; Li, J.; Li, Y.; Sun, A.; Wang, J. A CNN-LSTM-Based Model to Forecast Stock Prices. Complexity 2020, 2020, 6622927.

[3] P. Mondal, L. Shit, S. Goswami, Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices, Int. J. Comput. Sci. Eng. Appl. 4 (2) (2014) 13.

[4] Jordan Prosky, Andrew Tan, Xingyou Song, Micael Zhao. "Sentiment Predictability for Stocks" - arXiv:1712.05785v2 [cs.CL], 18 Jan 2018.

[5] V. Gururaj, V.R. Shriya, K. Ashwini, Stock market prediction using linear regression and support vector machines, Int. J. Appl. Eng. Res. 14 (8) (2019) 1931–1934.

[6] N.I. Sapankevych, R. Sankar, Time series prediction using support vector machines: a survey, IEEE Comput. Intell. Mag. 4 (2) (2009) 24–38.

[7] X. Shao, D. Ma, Y. Liu, Q. Yin. "Short-term forecast of stock price of multi-branch LSTM based on K-means", 2017 4th International Conference on Systems and Informatics (ICSAI).

[8] Li, C.; Zhang, X.; Qaosar, M.; Ahmed, S.; Alam, K.M.R.; Morimoto, Y. Multi-factor Based Stock Price Prediction Using Hybrid Neural Networks with Attention Mechanism. In Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Fukuoka, Japan, 5–8 August 2019; pp.1–6.

[9] H. Jia, Investigation into the effectiveness of long short term memory networks for stock price prediction, 2016, arXiv preprint arXiv:1603.07893.

[10] M. Roondiwala, H. Patel and S. Varma, "Predicting stock prices using LSTM," International Journal of Science and Research(IJSR), vol. 6, no. 4, pp. 1754-1756, 2017.

[11] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan,V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in International Conference on Advances in Computing, Communications and Informatics, 2017.

[12] Ojo, S.O.; Owolawi, P.A.; Mphahlele, M.; Adisa, J.A. Stock Market Behaviour Prediction using Stacked LSTM Networks.In Proceedings of the 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Vanderbijlpark, South Africa, 1–22 November 2019; pp. 1–5. https://doi.org/10.1109/IMITEC45504.2019.9015840.

[13] Lounnapha, S.; Zhongdong, W.; Sookasame, C. Research on Stock Price Prediction Method Based on Convolutional Neural Network. In Proceedings of the 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Jishou, China, 14–15 September 2019; pp. 1–4. https://doi.org/10.1109/ICVRIS.2019.00050.

[14] K. Pawar, R.S. Jalem, V. Tiwari, Stock market price prediction using LSTM RNN, in: Emerging Trends in Expert Applications and Security, Springer, Singapore, 2019, pp. 493–503.

[15] A.S. Saud, S. Shakya, Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE, Procedia Comput. Sci. 167 (2020) 788–798.