Stock Market Prediction using Bi-Directional LSTM

Jaiwin Shah
Computer Engineering Department
Sardar Patel Institute of Technology
Mumbai, India
jaiwin.shah@spit.ac.in

Rishabh Jain
Computer Engineering Department
Sardar Patel Institute of Technology
Mumbai, India
rishabh.jain2@spit.ac.in

Vedant Jolly
Computer Engineering Department
Sardar Patel Institute of Technology
Mumbai, India
vedant.jolly@spit.ac.in

Anand Godbole
Computer Engineering Department
Sardar Patel Institute of Technology
Mumbai, India
anand_godbole@spit.ac.in

2021 International Conference on Communication information and Computing Technology (ICCICT) | 978-1-6654-0430-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCICT50803.2021.9510147

***Abstract*—Stock market prediction is quite challenging as the market is volatile and its direction is stochastic. The stock market gets driven by several factors like investor sentiment, economic strength, market rumors, inflation. All these aspects together make the stock market quite turbulent and hence very difficult to predict with accuracy. In this paper, we analyzed traditional Machine Learning prediction models and figured out the drawbacks associated with them. Hence we scrutinized a range of stock prediction models and finally singled out the Bi-directional Long Short-Term Memory (Bi-LSTM) neural network. It intends to find out the title role of time series by analyzing historical data of different stocks and predict stock price trends. They form a unified framework for depth and time calculation learning faster than the one-directional approach. It can capture the temporal evolution of information which allows this model to attain the best performance.**

***Index Terms*—Stock Market, MACD, MFI, RNN, LSTM, Bi-LSTM**

I. Introduction

A stock market is a place where the stocks get exchanged. It is considered an intuitive reflection of the fiscal development of a country. But due to the variability and unpredictability, its mechanism presents the problem of perplexity and uncertainty. Predicting share prices remain one of the most complex tasks involving human-computer interaction. This dynamic nature makes it difficult to predict stock prices to the T. Stocks never vary alone. The movement of one stock tends to have an avalanche effect on several others. This factor can help in predicting the prices of many shares at once.

A hybrid end-to-end neural network can learn multiple time scale features to predict stock prices. Deep neural networks are non-linear function approximators that can map non-linear functions. They can identify hidden patterns, underlying dynamics in the data through a self-learning process and hold past information. In the Stock market, the data engendered is humongous and is highly non-uniform. To model this dynamic data, we need a model that can identify patterns and the lurking dynamics. We intend to study the applicability of Bi-directional LSTM networks on the stock price prediction. Subsequently, a comparison concerning accuracy against the existing methods is computed.

II. Literature Review

Linear regression algorithm analyzes two separate variables that define a single relationship. The simplest form of the regression equation is y = mx + c. It is quite easy to implement. Dimensionality reduction along with cross-validation helps it to overcome overfitting. The SVM is a training algorithm used for classification and regression. Support Vector Machines (SVM) can solve time-series domain problems with the help of regression techniques. Two main elements of SVM include the kernel functions and the techniques of mathematical programming [1]. K-Nearest Neighbour is a supervised machine learning algorithm that can store all possible available cases and classify new case situations based on the previous ones. For each datum, the algorithm discovers the k closest observations and classifies the data point to the majority. The k-closest points are the ones with the least Euclidean distance to the data point under our consideration [2].

But all that glitters is not gold. Linear Regression predicts the stock prices to increase linearly but in reality, the actual values jump up and down. Forecasting a trend reversal is impossible for the linear prediction model. Also, the stock prices depend on various factors that the algorithm doesn't take into consideration. SVM tries to identify patterns in stock price variation but fails to capture the correlation between stock prices in the form of long-term context-specific temporal dependencies [3]. Due to the stock market being quite volatile, the multivariate time series dataset contains many outliers. It creates noise. SVM under-performs when the data has more noise as it makes the target class overlap [3]. Also, hyper-parameters for the SVM are convoluted to gauge. It causes the decision boundary to become curvier and variance to increase and thus increase the chances of over-fitting [4]. It makes the nearer ones more influential in determining which class does the datum belongs. In

K-Nearest Neighbours, when the value of k is even, there might be ties. So to avoid this, weights are assigned to each observation. It makes the nearer ones more influential in determining which class does the datum belongs. Although K-Nearest Neighbour is a stable and robust algorithm, it is sensitive to the scale of the data and irrelevant features that can change the results drastically if there is a sudden bump in the price of a given stock. Although these algorithms work well, a model with more accuracy of a few percentage points having low variance and speedy prediction can increase profits by a great deal.

III. Recurrent Neural Network(RNN)

RNN works as a feed-forward neural network. Besides, its connections point to the rear. A single neuron present in the RNN network layer consists of two sets of weights: One for the input x(t) and the second one is for the output of the preceding step, y(t-1), where t is the time step as depicted in Fig 1. The recurrent layers turnout can be computed directly for a mini-batch by placing all the inputs at time step t in an input matrix X(t) [5]. A function of all the input data from previous time steps is collected. A memory cell is a component of the neural network that preserves some state across time steps. RNN is a type of sequence-to-sequence networking that is beneficial for predicting stock prices [6]. When the data transforms while traversing an RNN, a lot of information gets lost at each step. It makes the RNN forget about the original inputs leaving no traces behind. It creates the problem of short-term memory.

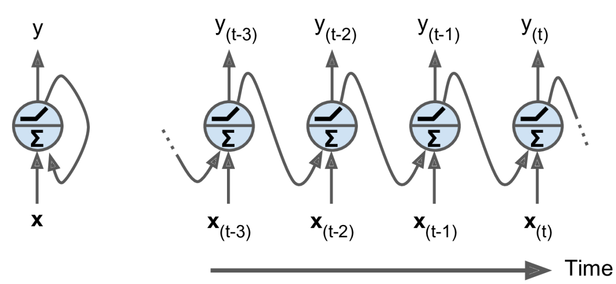

Fig. 1. Recurrent Neural Network [5]

IV. Bi-Directional Long Short Term Memory (bi-lstm)

To overcome the short-term memory problem, using the Long Short-Term Memory (LSTM) cell facilitates the detection of the long-term dependencies in the data. The key idea of the LSTM cell is that the network should be able to discern what to reserve in the long-term state, what to discard, and what to interpret from it. The LSTM cell contains a regulatory structure called gates that can be employed when the model needs to discard or append information to the cell state [7]. Hence, they optionally allow data through it. The gates get fetched from a sigmoid neural net layer with a point-wise multiplication operation. Bi-directional LSTM is a better version of LSTM [8]. It runs the input in two ways. One is from the past towards the future and the other is vice versa. Hence, the information from the future state as well as the past state gets preserved.

V. Proposed System

We propose an algorithm consisting of a neural network for forecasting the end-of-day price of a stock employing Bi-Directional Long Short-Term Memory (LSTM). For this, we use the dataset obtained from Yahoo Finance. This dataset includes historical stock market data of Tesla and Citi Bank under a period of 6 years.

A. Data Normalization

Data normalization is a technique applied as a part of data pre-processing that scales down the data set to an amount that the normalized data falls in the range between 0 and 1. The data frame of our Bi-LSTM model has different dimensions, having varying properties and order of magnitude. The stock trading volume has a much higher enormity than the stock price. Suppose we use a dataset such that the enormity of a particular vector is much higher than the others. Thus the preference of the model towards these vectors becomes considerably higher. Additionally, we may also lose some prime information which might affect the output.
Normalization deters these issues by devising novel values that preserve the general distribution in the dataset while safeguarding values within a scale applied across all arithmetic vectors used. Therefore, to ensure the authenticity of the results, seven eigenvectors need to be normalized. The maximum and minimum method selected for the normalization of input data and its formula is as follows:

$$X_{normalized} = \frac{X - X_{minimum}}{X_{maximum} - X_{minimum}}$$

The inverse operation of normalization is needed to ensure uniformity of the business meaning of input and output data. These parts are solidified in the program and a min-max scaler from the scikit-learn library is used for it.

B. Input Layer

- The input matrix consists of six columns. They are as follows: Open, High, Low, Volume, Moving average convergence divergence (MACD), and Money Flow Index (MFI). These columns together form a time series data.
- The output matrix contains a single column called Adjusting Close. It decides the market opening price for the following day.
- The data split up into a training set and a test set in the ratio of 4:1.
- After performing normalization, we have converted each constituent of input data into an interval between 0 and 1.
- Each entry of our dataset has the previous hundred days as its input. It makes our input dataset a 3-D matrix of shape (N, 100, 6).

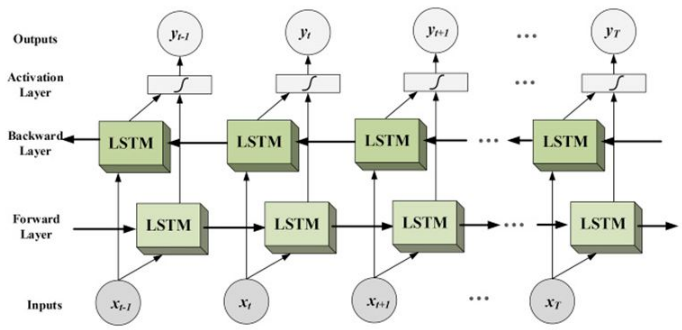

Fig. 2. Bi-Directional LSTM Cells [9]

VI. Implementation of the Bi-LSTM Model

- The primary function of the Bi-LSTM is to identify what information needs to be conserved and removed from the Bi-LSTM cell as depicted in Fig 2. The sigmoid layer makes this conclusion. If the output is '0', it discards the information and if it is '1', it retains the information.
- The subsequent step is to decide the information that we are going to store in our Bi-LSTM cell. The cell performs these operations with the help of 2 layers present in our Bi-LSTM cell. The first layer consists of a sigmoid layer. This layer determines the values which we will need to update for getting the desired output.
- The next layer is a tanh. This layer is responsible for creating the new candidate values. These values can be employed in our Bi-LSTM cell in the future. In the next step, we combine these two layers to create a new Bi-LSTM cell state that contains only the necessary information.
- The output of the Bi-LSTM will be a cleaner version of our cell state. We put through a tanh function so that the values range came in between -1 and 1. After the conversion, we combined the output with the sigmoid layer. It assured us that the throughput got trimmed according to our requirements. Hence, only the relevant information showed up as output.

A. Evaluation Metrics

The stock market analysis is time series forecasting. The price of the next day depends on the stock price of the previous day. So we cannot use the traditional means of calculating errors like standard deviation and variance that gives us an insight into where our model goes wrong. So to predict the accuracy of the stock market model we calculate the error percentage that provides an insight into how the model is performing.

To calculate the error percentage we have considered 3 measures that are as follows:

1) Mean Absolute Deviation (MAD): In the above formula, we have n data points and X_i represents the actual stock price and m(X) represents our predicted stock price.

$$MAD = (\frac{1}{n})\sum_{i=1}^{n}|X_i - m(X)|$$

2) Mean Square Error (MSE): In the above formula n is the number of data points, Y_i is the actual share price and $\hat{Y}_i$ is our predicted stock price.

$$MSE = (\frac{1}{n})\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

3) Mean Absolute Percentage Error (MAPE): In the above formula n is the no of data points, A_t is the actual stock price and F_t is the predicted stock price.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

B. Result Analysis

For our model, we have tried to fine-tune the parameters such that we do not over-fit our training set and simultaneously getting accurate results on our test set. As mentioned above, we have applied an 80-20 split in our dataset. The hyper-parameters considered are the number of epochs and the number of hidden units in the Bi-Directional LSTM Layer. We found out that the optimal values for both of them are 200 epochs and 100 units.

TABLE I
ERROR METRICS TABLE FOR THE TRAINING DATASET

Error Metrics	Tesla Inc	Citigroup Inc
Mean Absolute Deviation	1.124%	0.313%
Mean Square Error	2.142%	0.17%
Mean Absolute Percentage Error	1.669%	0.445%

TABLE II
ERROR METRICS TABLE FOR THE TEST DATASET

Error Metrics	Tesla Inc	Citigroup Inc
Mean Absolute Deviation	1.82%	0.997%
Mean Square Error	5.772%	1.527%
Mean Absolute Percentage Error	3.379%	1.412%

From the above tables, we observe that our model does not over-fit or under-fit the dataset. There is not much difference between the values of the training dataset and the test dataset.

A good measure to see if our model predicts the values accurately or not is the Mean Absolute Percentage Error, whose value must be less than 10% for accurate prediction [10]. But in our model, it is way less than 10%, which means that our model will provide the correct prediction most of the times.

VII. Experimental Analysis

The stock market is full of volatility and unpredictability. The stop-loss could get hit well before the target does. So, predicting the stock prices with accuracy is of immense importance. We aim to increase the accuracy of the prediction model by a few points. We largely succeeded in the same. The table below illustrates the gain in accuracy.

TABLE III
COMPARISON OF LSTM AND BI-LSTM

Error Metrics	For LSTM	For BI-LSTM
Training Dataset	0.78	0.98
Test Dataset	0.72	0.96

Our current model gives an accuracy of 98.331% on the training set and about 96.621% on the test set for the Tesla Inc shares. For Citigroup Inc, it gave an accuracy of about 99.555% on the training set and about 98.588% on the test set. On average, it gives around 98% for the training set and 96% for the test set. The accuracy achieved by our model has significantly increased from the accuracy achieved by [11] which was about 78%.

The reasons for achieving better accuracy was using the Bi-Directional LSTM layer as compared to just the LSTM layer used by [11]. Another reason for getting better results was fine-tuning the hyper-parameters by applying them over a range of values and then choosing the one which suits our requirements perfectly. Improving the accuracy of our model depends on the parameters we feed into our model. The additional parameters which we have fed into out Neural Network as compared to [11] are Moving average convergence divergence (MACD) and Money Flow Index (MFI). These additional parameters give a better insight into the stock market by calculating their moving average and Money Flow Index. These parameters assist's to estimate how well the stock is performing, and thereon our models output the predicted results.

VIII. Results

1) This paper tried to propose a framework for predicting the closing price of different stocks based on the Bi-Directional LSTM model. While comparing the results with the actual ones, we found them to be pretty close and accurate.
2) With the help of this paper, we proposed combining the attention mechanism that depends on the length of the input matrix. It was done keeping in mind the various advantages of deep neural nets over different machine learning techniques.
3) Deep neural networks comprise lots of hidden units that identify various underlying aspects of our model. It helps in achieving better accuracy as we find the global minimum. Thus, deep learning models work better than the traditional machine learning models.
4) The graphs below plotted between the actual and the predicted price for Tesla Inc and Citigroup Inc.
5) The X-axis represents the period (number of days) for which we predict the price of the stock and on the Y-axis, there is the price of the given stock.
6) For the training dataset, one unit on the x-axis represents 200 days while one unit on the y-axis represents 20$.
7) For the test dataset, one unit on the x-axis represents 20 days while one unit on the y-axis represents 20$.
8) Figures 3 and 4 represent the analysis between the actual and the predicted closing price for the stock of Tesla and Citibank respectively. The training dataset represents 80% of the total dataset.
9) Figures 5 and 6 represent the analysis between the actual and the predicted closing price for the stock of Tesla and Citibank respectively. The test dataset represents 20% of the total dataset.
10) The red line in the graphs depict the predicted stock price while the blue line represents the actual stock price.

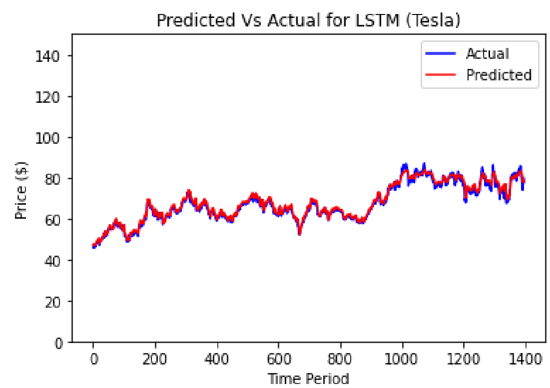

Fig. 3. Predicted Price vs Actual Price for training dataset of Tesla

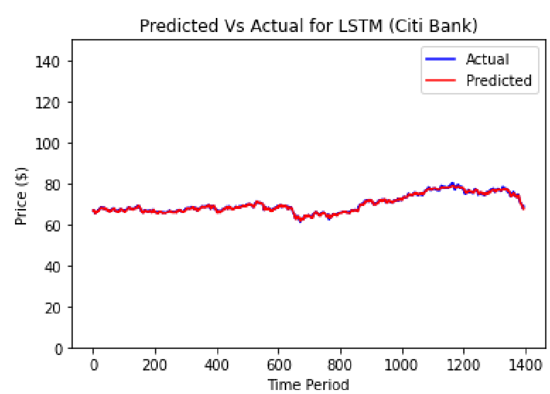

Fig. 4. Predicted Price vs Actual Price for training dataset of Citi Bank

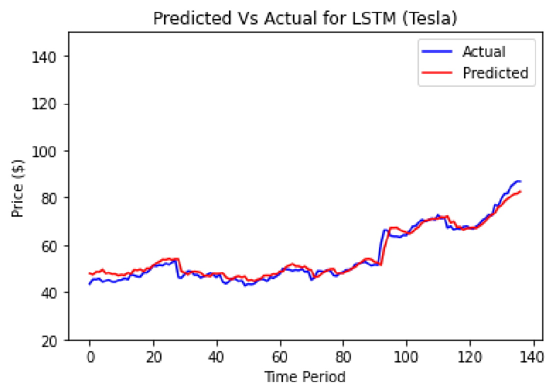

Fig. 5. Predicted Price vs Actual Price for test dataset of Tesla

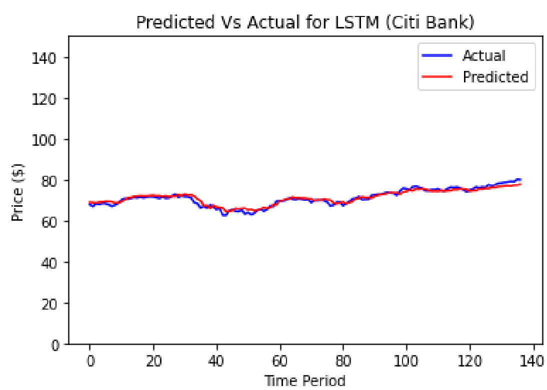

Fig. 6. Predicted Price vs Actual Price for test dataset of Citi Bank

IX. Conclusion

We can observe that the model proposed in this paper outran the standard models with very few exceptions. It is very promising as compared to the approaches employed before. Large datasets help our model in fleshing out more patterns and hence, the weight of the layers can be better adjusted. Bi-directional LSTMs perform better as they can preserve the information from the past state and the future state. It helps to keep track of the context-specific short-term dependencies between stock prices for longer duration. When we compare our model to other stock prediction models, it demonstrated an appreciable rise concerning accuracy. On the other hand, we destine to keep exploring ways to revamp our model drastically and its predictions by suggesting changes in the model's architecture by adding some new features. Our model may have compromised on the issue of efficiency. Further modifications in the model could help decrease the long dependency chain of the gradient. We also hope to evaluate and improve the model using different technical indicators.

Yet, in its essence, the stock market is a pure reflection of human emotions and feelings. Number crunching and price analysis have their barriers. A possible addition to this model could be an augmentation with news feed analysis from the financial world around us. It could link with sentiment analysis to better understand the emotions associated with the price.

References

[1] Y. Xia, Y. Liu and Z. Chen, "Support Vector Regression for prediction of stock trend," 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering, Xi'an, China, 2013, pp. 123-126, doi: 10.1109/ICIII.2013.6703098.

[2] Teja Kodali, "Using kNN Classifier to Predict Whether the Price of Stock Will Increase". [online]. Available: https://datascienceplus.com/knn-classifier-to-predict-price-of-stock/

[3] Sai Krishna Lakshminarayanan and John McCrae, "A Comparative Study of SVM and LSTM Deep Learning Algorithms for Stock Market Prediction". [online]. Available: http://ceur-ws.org/Vol-2563/aics41.pdf

[4] D. Karmiani, R. Kazi, A. Nambisan, A. Shah and V. Kamble, "Comparison of Predictive Algorithms: Backpropagation, SVM, LSTM and Kalman Filter for Stock Market," 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 228-234, doi: 10.1109/AICAI.2019.8701258.

[5] Aurelien Geron, "Chapter 4. Recurrent Neural Networks". [online]. Available: https://www.oreilly.com/library/view/neural-networks-and/9781492037354/ch04.html

[6] A. J. P. Samarawickrama and T. G. I. Fernando, "A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market," 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 2017, pp. 1-6, doi: 10.1109/ICIINFS.2017.8300345.

[7] Raghav NandaKumar, Uttamraj K R, Vishal R and Y V Lokeswari, "Stock Price Prediction Using Long Short Term Memory", International Research Journal of Engineering and Technology (IRJET), Mar-2018, Volume 5, Issue 3, pp.3342-3348

[8] M. A. Istiake Sunny, M. M. S. Maswood and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model," 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2020, pp. 87-92, doi: 10.1109/NILES50944.2020.9257950.

[9] i2tutorials, "Deep Dive into Bidirectional LSTM". [online]. Available: https://www.i2tutorials.com/deep-dive-into-bidirectional-lstm/

[10] Michael Gilliland, "Forecasting FAQs". [online]. Available:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119199885.app1

[11] S. Liu, G. Liao and Y. Ding, "Stock transaction prediction modeling and analysis based on LSTM," 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, China, 2018, pp. 2787-2790, doi: 10.1109/ICIEA.2018.8398183.