International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

# Future Stock Price Prediction using LSTM Machine Learning Algorithm

**Mrs. Nivethitha[1], Pavithra.V[2], Poorneshwari. G[3], Raharitha. R[4]**

[2,3,4]*Students, Dept of Computer Science and Engineering, Panimalar Engineering College, Tamilnadu, Chennai*
[1]*Asst. Professor, Dept of Computer Science and Engineering, Panimalar Engineering College, Tamilnadu, Chennai*

---

***Abstract -*** *The process of forecasting the stock prices has been a difficult task for many of the researchers analysts and new investors. In fact, investors are highly interested in the research area of stock price prediction. For a good and successful investment, many investors are keen in knowing the future situation of the stock market. Good and effective prediction systems for stock market help traders, investors, and analyst by providing supportive information like the future direction of the stock market. Predicting stock market price is a complex task that traditionally involves extensive human-computer interaction. There are multiple prediction methodologies for share price forecasting. Time Series Forecasting is basic for share price forecasting and other financial model forecast. As share price are more nonlinear, more intelligent time series prediction systems are required. Existing systems accuracy are not efficient enough in predicting. In this paper, we propose to use LSTM Machine Learning Algorithm for efficient forecasting of stock price. This will provide more accurate results when compared to existing stock price prediction algorithms.*

***Key Words: RNN , LSTM, Stock price analysis , Future Prediction***

## 1. INTRODUCTION

The stock market is a vast array of investors and traders who buy and sell stock, pushing the price up or down. The prices of stocks are governed by the principles of demand and supply, and the ultimate goal of buying shares is to make money by buying stocks in companies whose perceived value (i.e., share price) is expected to rise. Stock markets are closely linked with the world of economics —the rise and fall of share prices can be traced back to some Key Performance Indicators (KPI's). The five most commonly used KPI's are the opening stock price (`Open'), end-of-day price (`Close'), intraday low price (`Low'), intra-day peak price (`High'), and total volume of stocks traded during the day (`Volume'). Economics and stock prices are mainly reliant upon subjective perceptions about the stock market. It is near impossible to predict stock prices to the T, owing to the volatility of factors that play a major role in the movement of prices. However, it is possible to make an educated estimate of prices. Stock prices never vary in isolation: the movement of one tends to have an avalanche effect on several other stocks as well. This aspect of stock price movement can be used as an important tool to predict the prices of many stocks at once. Due to the sheer volume of money involved and number of transactions that take place every minute, there comes a trade-off between the accuracy and the volume of predictions made; as such, most stock prediction systems are implemented in a distributed, parallelized fashion. These are some of the considerations and challenges faced in stock market analysis. Also there are a lot of complicated financial indicators and the fluctuation of the stock market is highly violent. However, as the technology is getting advanced, the opportunity to gain a steady fortune from the stock market is increased and it also helps experts to find out the most informative indicators to make a better prediction. The prediction of the market value is of great importance to help in maximizing the profit of stock option purchase while keeping the risk low. Recurrent neural networks (RNN) have proved one of the most powerful models for processing sequential data.

Long Short-Term memory is one of the most successful RNNs architectures. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With these memory cells, networks are able to effectively associate memories and input remote in time, hence suit to grasp the structure of data dynamically over time with high prediction capacity.

### 1.1 Existing System

In this existing system, sliding window algorithm has been used which won't do dropout process. Because of this, unwanted data have been processed which leads to wastage of time and memory space. The prediction of future stock price by Sliding Window Algorithm is less efficient because of processing unwanted data. The Sliding Window Algorithm which is used in the existing system is not that much effective in handling non linear data. So, in our proposed the future stock price prediction is done using LSTM(Long Short Term Memory) which is more efficient than Sliding Window Algorithm.

### 1.2 Proposed System

The aim of proposed solution is to support investors in their decisions and recommend buying the assets which provide the greatest profits. In our paper, we propose to use LSTM (Long Short Term Memory) algorithm to provide efficient stock price prediction. Long short-term memory (LSTM) is an artificial neural network (RNN) architecture used in the field of deep learning. Unlike standard feed

forward neural networks, LSTM has feedback connections that make it a "general purpose computer" . It can not only process single data points (such as images), but also entire sequences of data. Because of the dropout process which takes place in LSTM algorithm, it is comparatively faster than Sliding window algorithm. LSTM algorithm is more efficient in prediction the future stock price than Sliding Window algorithm because of removing the unwanted data. The time and memory consumption is also reduced when compared to the exciting system due to dropout process .LSTM algorithm is more suitable in handling non linear data.

## 2. SYSTEM ARCHITECTURE

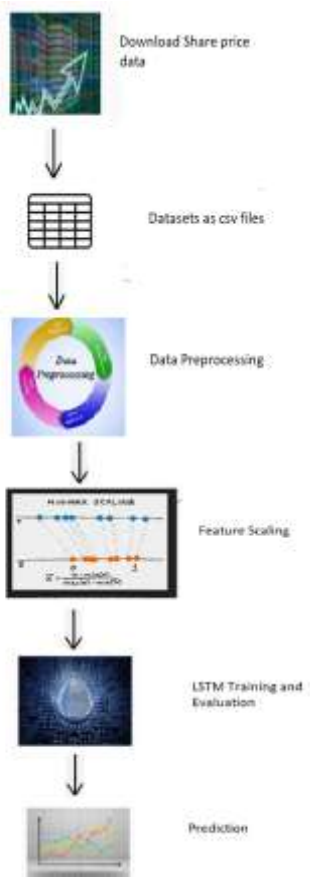


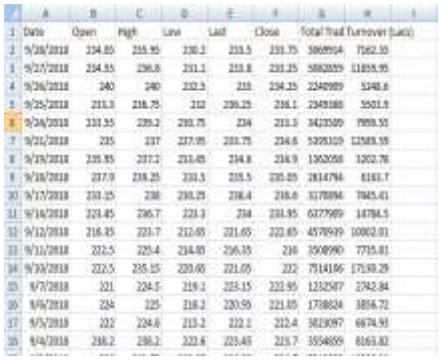

Fig 1 Architecture Overview

### 2.1 Download Share price Data

A share price is the price of a single share of a number of saleable stocks of a company, derivative or other financial asset. In layman's terms, the stock price is the highest amount someone is willing to pay for the stock, or the lowest amount that it can be bought for. In this paper, we download the share data of particular company using the Kaggle data source. The data such as date, open share price, high value, low value, last share value close share price, total trade and turn over values are used.

### 2.2 Dataset(CSV files)

A dataset is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows. Here we keep all our data in the form of csv files. In computing, a comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more field, separated by commas. The use of the comma as a field separator is the source of the name for this file format. Our dataset is kept in tabular format in excel with values such as date, open, high, low, last, low ,total trade and turnover values.



| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Date | Open | High | Low | Last | Close | Total Trad | Turnover (Lacs) | |
| 2 | 9/28/2018 | 234.05 | 235.95 | 230.2 | 233.5 | 233.75 | 3069914 | 7162.35 | |
| 3 | 9/27/2018 | 234.55 | 236.8 | 231.1 | 233.8 | 233.25 | 5082859 | 11859.95 | |
| 4 | 9/26/2018 | 240 | 240 | 232.5 | 235 | 234.25 | 2240909 | 5248.6 | |
| 5 | 9/25/2018 | 233.3 | 236.75 | 232 | 236.25 | 236.1 | 2349368 | 5503.9 | |
| 6 | 9/24/2018 | 234.55 | 239.2 | 233.75 | 234 | 233.3 | 3423509 | 7999.55 | |
| 7 | 9/21/2018 | 235 | 237 | 227.05 | 233.75 | 234.6 | 5393319 | 12589.59 | |
| 8 | 9/20/2018 | 235.95 | 237.2 | 233.45 | 234.8 | 234.9 | 1362058 | 3202.78 | |
| 9 | 9/18/2018 | 237.9 | 238.25 | 233.3 | 235.5 | 235.85 | 2814794 | 6661.7 | |
| 10 | 9/17/2018 | 233.35 | 238 | 230.25 | 236.4 | 236.6 | 5178894 | 7845.41 | |
| 11 | 9/14/2018 | 223.45 | 236.7 | 223.3 | 234 | 233.95 | 6377989 | 14784.5 | |
| 12 | 9/12/2018 | 216.35 | 223.7 | 212.65 | 221.65 | 222.65 | 4578509 | 10002.01 | |
| 13 | 9/11/2018 | 222.5 | 225.4 | 214.65 | 216.35 | 216 | 3508990 | 7715.81 | |
| 14 | 9/10/2018 | 222.5 | 235.15 | 220.65 | 221.05 | 222 | 7514186 | 17130.29 | |
| 15 | 9/7/2018 | 221 | 224.3 | 219.1 | 223.15 | 222.95 | 1232587 | 2742.84 | |
| 16 | 9/6/2018 | 224 | 225 | 218.2 | 220.55 | 221.05 | 1738824 | 3856.72 | |
| 17 | 9/5/2018 | 222 | 224.8 | 213.2 | 222.1 | 222.4 | 3823097 | 6874.93 | |
| 18 | 9/4/2018 | 218.2 | 228.2 | 222.8 | 223.45 | 223.7 | 3554859 | 8163.82 | |

Fig 2 dataset of stock price

### 2.3 Data Preprocessing

Data preprocessing is an important step in the machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis. Often, data preprocessing is the most important phase of a machine learning project, especially in computational data.

If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data preprocessing includes cleaning, instance selection, normalization, transformation, feature extraction and selection etc. The product of data preprocessing is the final training set.

### 2.4 Feature Scaling

Feature scaling is a method used to standardize the range of independent variables or features of data. In data preprocessing, it is also known as data normalization and is generally performed during the data preprocessing step. Since the range of values of raw data varies widely, objective functions will not work properly without normalization. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. Another reason why feature scaling is applied is that gradient descent converges much faster with feature scaling than without it.

Also known as min-max scaling or min-max normalization, is the simplest method and consists in rescaling the range of features to scale the range in [0, 1] or [−1, 1]. Selecting the target range depends on the nature of the data. The general formula is given as

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

where x is an original value, x' is the normalized value.

### 2.5 LSTM Training and Evaluation

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedback neural networks, LSTM has feedback connections that make it a "general purpose computer". It can not only process single data points, but also entire sequences of data.

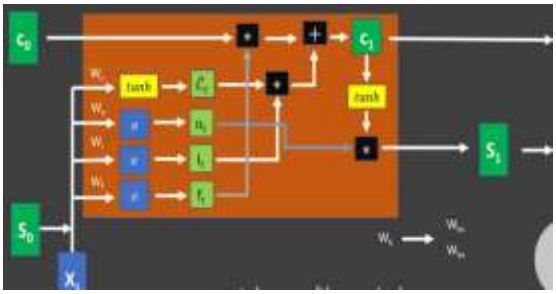


Fig 3 LSTM Overview

Long Short-Term Memory (LSTM) networks are an extension for recurrent neural networks, which basically extends their memory. Therefore it is well suited to learn from important experiences that have very long time lags in between.
The units of an LSTM are used as building units for the layers of a RNN, which is then often called an LSTM network.
LSTM's enable RNN's to remember their inputs over a long period of time. This is because LSTM's contain their information in a memory, that is much like the memory of a computer because the LSTM can read, write and delete information from its memory.

An LSTM memory cell, has the following three components, or gates:

**1. Forget gate:** the forget gate decides when specific portions of the cell state are to be replaced with more recent information. It outputs values close to 1 for parts of the cell state that should be retained, and zero for values that should be neglected.

**2. Input gate:** based on the input (i.e., previous output o(t-1), input x(t), and previous cell state c(t-1)), this section of the network learns the conditions under which any information should be stored (or updated) in the cell state.

**3. Output gate:** depending on the input and cell state, this portion decides what information is propagated forward (i.e., output o(t) and cell state c(t)) to the next node in the network. Thus, LSTM networks are ideal for exploring how variation in one stock's price can affect the prices of several other stocks over a long period of time. They can also decide (in a dynamic fashion) for how long information about specific past trends in stock price movement needs to be retained in order to more accurately predict future trends in the variation of stock prices.

$$f_t = \sigma(W_f S_{t-1} + W_f X_t) \quad \text{- Forget Gate}$$
$$i_t = \sigma(W_i S_{t-1} + W_i X_t) \quad \text{- Input Gate}$$
$$o_t = \sigma(W_o S_{t-1} + W_o X_t) \quad \text{- Output Gate}$$
$$\tilde{C}_t = tanh(W_c S_{t-1} + W_c X_t)$$
$$c_t = (I_t * \tilde{C}_t) + (f_t * c_{t-1}) \quad \text{- Cell State}$$
$$h_t = o_t * tanh(c_t) \quad \text{– New State}$$

Fig 4 Working of LSTM

### 2.6 Visualizing the Predictions

In this paper we use matplotlib which is a plotting library to view the prediction of future stock price.
Matplotlib is a 2-D plotting library that helps in visualizing figures. Matplotlib emulates Matlab like graphs and visualizations. Matlab is not free, is difficult to scale and as a programming language is tedious. So, matplotlib in Python is used as it is a robust, free and easy library for data visualization.

StockPrice data will be visualized using matplotlib in the form of 2D graph. Current data and Predicted data will be presented for comparison. Accuracy will be presented based on prediction and existing data.

### 3. LSTM Program Explanation

LSTM are very powerful in sequence prediction problems because they're able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price.

We will first start by importing Numpy for scientific computation, Matplotlib for plotting graphs, and Pandas to aide in loading and manipulating our datasets. The next step is to load in our training dataset and select the Open and High columns that we'll use in our modeling. We check the head of our dataset to give us a glimpse into the kind of dataset we're working with.

The open column is the starting price while the Close column is the final price of a stock on a particular trading day. The High and Low columns represent the highest and lowest prices for a certain day.

From previous experience with deep learning models, we know that we have to scale our data for optimal performance. In our case, we'll use Scikit Learn's MinMaxScaler and scale our dataset to numbers between zero and one. LSTMs expect our data to be in a specific format, usually a 3D array. We start by creating data in 60 timesteps and converting it into an array using NumPy. Next, we convert the data into a 3D dimension array with X_train samples, 60 timestamps, and one feature at each step.

In order to build the LSTM, we need to import some modules from Keras

1.Sequential for initializing the neural network
2.Dense for adding a densely connected neural network layer
3.LSTM for adding the Long Short-Term Memory layer
4.Dropout for adding dropout layers that prevent overfitting

We add the LSTM layer and later add a few Dropout layers to prevent overfitting. We add the LSTM layer with the following arguments:

1.50 units which is the dimensionality of the output space
2.return_sequences=True which determines whether to return the last output in the output sequence, or the full sequence
3.input_shape as the shape of our training set.

When defining the Dropout layers, we specify 0.2, meaning that 20% of the layers will be dropped. Thereafter, we add the Dense layer that specifies the output of 1 unit. After this, we compile our model using the popular adam optimizer and set the loss as the mean_squarred_error. This will compute the mean of the squared errors. Next, we fit the model to run on 100 epochs with a batch size of 32.

In order to predict future stock prices we need to do a couple of things after loading in the test set:

1.Merge the training set and the test set on the 0 axis.
2.Set the time step as 60 (as seen previously)
3.Use MinMaxScaler to transform the new dataset
4.Reshape the dataset as done previously

After making the predictions we use inverse_transform to get back the stock prices in normal readable format. Finally, we use Matplotlib to visualize the result of the predicted stock price and the real stock price. From the plot we can see that the real stock price went up while our model also predicted that the price of the stock will go up. This clearly shows how powerful LSTMs are for analyzing time series and sequential data.

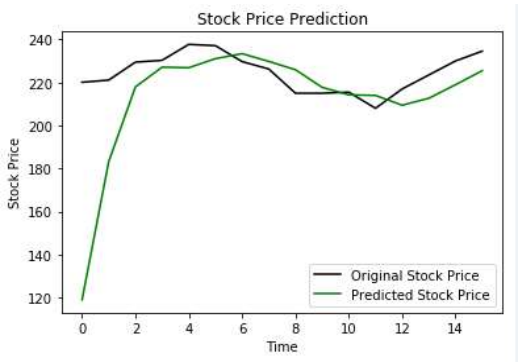



Fig 5 Output Graph

## 4. CONCLUSIONS

The popularity of stock market trading is growing rapidly, which is encouraging researchers to find out new methods for the prediction using new techniques. The forecasting technique is not only helping the researchers but it also helps investors and any person dealing with the stock market. In order to help predict the stock indices, a forecasting model with good accuracy is required. In this work, we have used one of the most precise forecasting technology using Long Short-Term Memory unit which helps investors, analysts or any person interested in investing in the stock market by providing them a good knowledge of the future situation of the stock market. When compared with ARIMA (Auto Regressive Integrated Moving Average) algorithm, it is shown that ARIMA algorithm understands the past data and does not focus on the seasonal part. Therefore accuracy is less. LSTM provides more accurate results than ARIMA algorithm.

The future enhancement includes comparing the accuracy of LSTM with other prediction algorithms. LSTM is more accurate than any other prediction algorithms.

## 5. REFERENCES

1) Suresh, Harini, et al. "Clinical Intervention Prediction and Understanding using Deep Networks." arXiv preprint arXiv:1705.08498 (2017).

2) M. Göçken, M. Özçalıcı, A. Boru, and A. T. Dosdoğru, "Integrating metaheuristics and Artificial Neural Networks for improved stock price prediction," Expert Systems with Applications, vol. 44, pp. 320-331, 2016.

3) Zhu, Maohua, et al. "Training Long Short-Term Memory WithSparsified Stochastic Gradient Descent." (2016)

4) Ding, Y., Zhao, P., Hoi, S. C., Ong, Y. S. "An Adaptive Gradient Method for Online AUC Maximization"In AAAI (pp. 2568-2574). (2015, January).

5) J. P. He, L. Cai, P. Cheng, and J. L. Fan, "Optimal Investment for Retail Company in Electricity Market," Iee Transactions on Industrial Informatics, vol. 11, no. 5, pp. 1210-1219, Oct, 2015.

6) A. A. Arafah, and I. Mukhlash, "The Application of Fuzzy Association Rule on Co-movement Analyze of Indonesian Stock Price," Procedia Computer Science, vol. 59, pp. 235-243, 2015/01/01, 2015.

7) Zhao, P., Hoi, S. C., Wang, J., Li, B. "Online transfer learning". Artificial Intelligence, 216, 76-102. (2014)

8) Pascanu, Razvan, Tomas Mikolov, and YoshuaBengio. "On the difficulty of training recurrent neural networks." International Conference on Machine Learning. 2013.

9) Recht, Benjamin, et al. "Hogwild: A lock-free approach to parallelizing stochastic gradient descent." Advances in neural information processing systems. 2011.

10) Y. Perwej, A. Perwej, and "Prediction of the Bombay Stock Exchange (BSE) market returns using artificial neural network and genetic algorithm," Journal of Intelligent Learning Systems and Applications, vol. 4, pp. 108-119, 2012.