Nano World Journal

https://doi.org/10.17756/nwj.2023-s4-080

Research Article

Open Access

# Stock Price Prediction Using Long Short-term Memory

**B. Veda Vidhya*** **and Ajmeera Kiran**

*Department of Computer Science and Engineering, M.L.R. Institute of Technology, Hyderabad, Telangana, India*

***Correspondence to:**
B. Veda Vidhya
Department of Computer Science and Engineering,
M.L.R. Institute of Technology,
Hyderabad, Telangana, India.
**E-mail:** vedavidhyab@mlrinstitutions.ac.in

**Received:** September 19, 2023
**Accepted:** December 01, 2023
**Published:** December 06, 2023

**Citation:** Vidhya BV, Kiran A. 2023. Stock Price Prediction Using Long Short-term Memory. *NanoWorld J* 9(S4): S477-482.



Published by United Scientific Group

## Abstract

One of the most difficult challenges facing investors in this day and age is trying to accurately predict stock values. Since the prices of stocks are very non-linear and fluctuate often, it is exceedingly difficult to make precise forecasts regarding their future performance. Prior to the advent of artificial intelligence, it was never simple to visualize them in an exact manner. People are now able to accurately measure and predict the tends of the stock market, which in turn is producing an abundant amount of profit for the companies. This is made possible by the exponential growth in techniques and algorithms for artificial intelligence (AI), machine learning (ML), and deep learning (DL). In this area, numerous different algorithms were developed to anticipate stock values; nevertheless, when compared to LSTM (Long Short-Term Memory), these other algorithms could not produce accurate results. This study focuses primarily on the application of a Recurrent Neutral Network (RNN) model known as LSTM to the problem of forecasting stock prices.

## Keywords

Stock prediction, Long short-term memory, Online learning, Recurrent neural networks

## Introduction

The stock exchange is rapidly gaining acclaim and favor across all demographics of contemporary society. Because of the hoopla around the stock market, every person, whether deliberately or unknowingly, becomes aware of the current trends in the market. Stock markets unquestionably have an effect on all levels, ranging from the level of an individual to that of a nation. Investing in the stock market may be quite lucrative for a variety of businesses and individuals, both of which can reap the rewards. When one is able to accurately foresee the future and invest properly, this proportion of profit might climb at an exponential rate. This takes a substantial amount of historical knowledge about each stock-associated company, such as the company's tendencies over the years, the causes that have an effect on it, and how it affects other stocks and vice versa [1]. This is a massive field that everyone is keeping a close eye on, and there is a significant amount of scientific research being conducted in order to forecast the future stocks and get a hold of the profit. For this problem, a wide variety of predictive models and algorithms have been presented, each of which has achieved varying degrees of accuracy [2, 3]. The application of a ML strategy to a DL-based model, which is fundamentally an RNN-based stacked-LSTM network, is the primary emphasis of our paper. It is believed that the LSTM algorithm is the most advanced technology that can be used to analyze equities. It is believed to deliver the accurate results that any other algorithm in the same situation would produce, and it is said to give those findings accurately [4].

### Nanotechnology applications

The LSTM-based stock price prediction method can be used to a variety of elements of nanotechnology, including:

- Predicting the stock prices of nanotechnology companies: This can assist investors make investment decisions and capitalize on possibilities in the constantly increasing nanotechnology field.
- Recognizing prospective market trends: LSTMs can identify upcoming trends and patterns in the nanotechnology sector by evaluating historical data, offering useful information to firms and investors.
- Creating risk management strategies: LSTMs can be used to develop risk management models that can assist investors in mitigating potential losses connected with volatile nanotechnology stocks.
- Strategic decision-making assistance: LSTMs can help organizations make informed strategic decisions about investments, collaborations, and market expansion in nanotechnology by offering insights into future pricing changes.

### Literature survey

We have looked into a number of different research that have been done on different ways that ML can be used to predict stock market prices. There have been many different sorts of research papers written on the topic of forecasting stock prices, such as those that use linear regression, support vector machines, random basic functions, moving average convergence/divergence, and so on. One such work has achieved some progress through the use of regression and supervised learning by applying them to 14 years' worth of previous data from Google [5-8]. They trained the model with the help of a linear regression classifier, which is a form of supervised learning. Using this method, they discovered some patterns, improved their accuracy, and were able to forecast the final price of the trading day. Another research that served as a source of motivation for us was one that forecasted stock prices using support vector machines (SVM). They obtained the historical information from IBM. The problem of overfitting is present in earlier models, but it is not present in the SVM. Because of this, the SVM is one of the most efficient algorithms currently available, which is why it has been selected by a large number of people.

The LSTM model has been the subject of research from a variety of other fields, and these fields have produced a significant number of findings. However, since the introduction of the multi-layered LSTM model, it has been shown to be given much more accurate results than the rest [9]. This is due to the fact that this method considers various combinations of LSTM hyperparameters and selects the combination that is most suitable for forecasting stock prices. Some academics have tried to create a function that turns the input performance space into a performance class variable by employing Artificial Neural Networks (ANN) in conjunction with generic techniques. Some people employed the ANN technique as well as the Random Forest (RF) technique to forecast the closing prices of the following trading day. They were able to demonstrate that the ANN technique is more accurate than the RF technique. Some people made use of regression and LSTM [10-14]. The reason that particular methods were selected is due to the fact that regression reduces the error, and when LSTM is employed, it is possible to store the prior data and outcomes. In any case, the model that we are employing is a great deal more sophisticated than that; it is denoted by the letters RNN and layered LSTM. The LSTM takes the place of the neurons in the hidden layers of the network, which elevates it to the position of the network's primary computational memory cell. As a result, these networks are able to store the historical data well, which enables them to produce accurate forecast results.

## Experimentation

### Proposed system

An approach for ML, namely an enhanced kind of RNN that we call LSTM.

#### ML and RNN

The study of computer algorithms that can automatically improve themselves via experience is referred to as "ML." In reality, it is a subfield of AI that, when combined with ML, may be applied in a wide variety of contexts. These ML algorithms, in their most basic form, construct a model on the basis of certain data known as a training dataset, and then use that model to make predictions or forecasts about the future using the training dataset.

Traditionally, there are three distinct buckets into which ML algorithms are placed.

- Supervised learning: The learner is presented with a model that already has certain labeled input data and is then shown the intended output.
- Unsupervised learning is a form of ML in which the model is given data that has not been labeled and left to figure out how to recognize patterns and reach conclusions on its own.
- The system interacts with a dynamic environment in which it must complete particular goals in order to obtain rewards, and in the end, the program strives to maximize the rewards by receiving as many as possible. This type of learning is known as reinforcement learning. The greater institution of algorithms known as sequential models contains something of a subset that is referred to as recurrent neutral networks. They are exactly the same as feed forward neural networks, with the exception that the connections point in the opposite direction.

At a minimum, a simple neural network will have three layers, which are referred to as the input, hidden, and output layers. Through the use of links, every node on one layer is connected to every other node on the layer below it. The nature of the connection between the two layers is denoted by a coefficient known as weight. When it comes to making the selection, this is a significant consideration to take into account.

The activation function is typically applied in the hidden layers, and doing so would generate numbers with substantially less inaccuracy depending on the type of activation function that we select. In common practice, the activation function is applied in the hidden layers. The output layer is connected to the nodes in the network. Because the output that we get could not be accurate, we have to back propagate the errors to the hidden layer in order to achieve the best possible value for them.

After finishing up all of these steps, the model will be trained and will be ready to receive input from the testing dataset as well as outputs in their respective forms.

### LSTM overview

The category of RNN known as LSTM networks. It is primarily implemented due to the fact that there are many different instances in which RNN fails, and there is a requirement for an advanced network that can handle those scenarios. In its most basic form, a RNN evaluates current input by taking into account past outputs (feedback) and retaining the information for a noticeably shorter period of time (Figure 1).

An LSTM memory cell is comprised of three gates, which are outlined in the following manner:

### Forgate gate

The forgate is responsible for determining whether or not the previous cell output that has been obtained should be replaced. If it generates a value that is close to 1, then it should be kept; otherwise, if it generates a value that is close to 0, then it should be deleted.

### Input gate

The decision of whether the new information should be entered or not was made by the input gate.

### Output gate

The output gate is responsible for determining which portion of the cell state should be transmitted to the next node in the network.

In this approach, LSTM is particularly efficient at drawing out the interdependencies across numerous stocks and deriving conclusions about such interdependencies (Figure 2). They are also capable of precisely determining the time period for which the information must be retained in order to get accurate numbers. This ability allows them to achieve accurate results.

## System design

The planning stage of any system is of critical importance, on par with its development and implementation phases. The first phase in the process of constructing a system is the designing of the system. The system design provides a high-level overview of how a system is constructed, including what characteristics, modules, and constants are incorporated into it. During the phase of a system's design known as "design," all of the requirements (both software and hardware) that have been obtained are implemented.

## Proposed algorithm

The basic flow of the algorithm is described as follows:

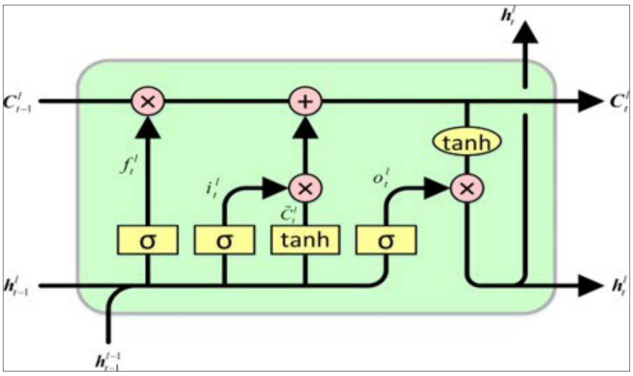



**Figure 1:** Overview of LSTM network structure.

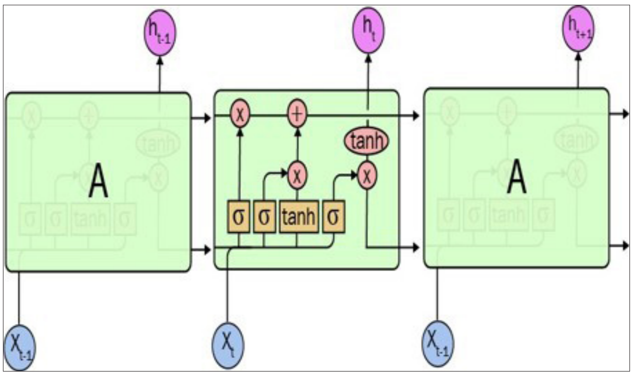



**Figure 2:** Interdependencies among various stocks and drawing.

**Step 1:** Start.

**Step 2:** The input stock data are stored in a three-dimensional NumPy array.

**Step 3:** Building a network structure with [1, a, b, 1] dimensions, where 1 represents the input layer, "a" represents number of neurons on the next layer, "b" represents number of neurons on the subsequent layer, and an activation function dedicated to a single layer.

**Step 4:** Training of the data using training dataset which constitutes approximately 80 percent of the entire dataset.

**Step 5:** The output layers are used as the input as the predicted layer for the next step.

**Step 6:** Repeat step 4 and step 5 until optimal error rate is achieved.

**Step 7:** Obtain the prediction by providing the test dataset to the model.

**Step 8:** Evaluating the accuracy by comparing predictions with the actual data.

**Step 9:** End.

## Overall system architecture

The main components of the architecture are listed and detailed in figure 3.

### Obtaining dataset and data preprocessing

The "Google stock price" dataset, which is available for

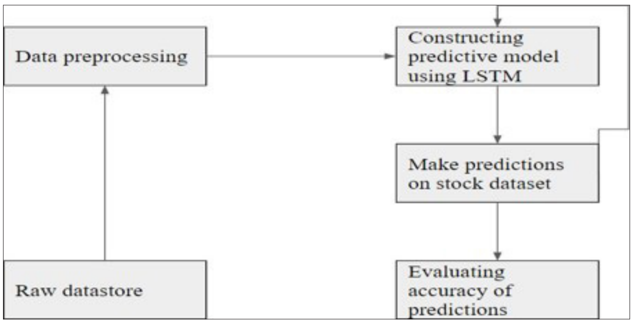
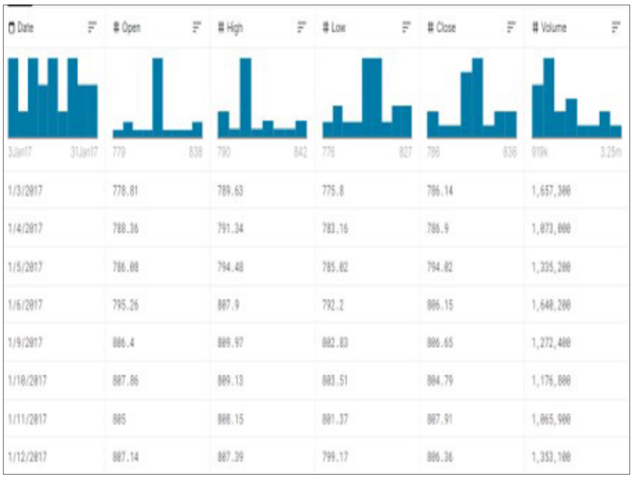



**Figure 3:** Data flow diagram of proposed technique.



| Date | # Open | # High | # Low | # Close | # Volume |
|---|---|---|---|---|---|
| 3Jan17 – 31Jan17 | 779 – 838 | 790 – 842 | 776 – 827 | 786 – 836 | 910k – 3.25m |
| 1/3/2017 | 778.81 | 789.63 | 775.8 | 786.14 | 1,657,300 |
| 1/4/2017 | 788.36 | 791.34 | 783.16 | 786.9 | 1,073,000 |
| 1/5/2017 | 786.08 | 794.48 | 785.02 | 794.02 | 1,335,200 |
| 1/6/2017 | 795.26 | 807.9 | 792.2 | 806.15 | 1,640,200 |
| 1/9/2017 | 806.4 | 809.97 | 802.83 | 806.65 | 1,272,400 |
| 1/10/2017 | 807.86 | 809.13 | 803.51 | 804.79 | 1,176,800 |
| 1/11/2017 | 805 | 808.15 | 801.37 | 807.91 | 1,065,900 |
| 1/12/2017 | 807.14 | 807.39 | 799.17 | 806.36 | 1,353,100 |

**Figure 4:** Features of google stock price dataset.

public use, was the one that was utilized for the purpose of this study (Figure 4). The CSV file can be downloaded from the internet and used to make projections about stock levels.

The following is a list of the most fundamental technical indicators and aspects of this dataset:

- Open: Indicates the price of the share when it was first traded.
- Close: Indicates the price of the share when it was last traded.
- Date: Indicates the date on which the stock price was recorded.
- Volume: This refers to the total number of shares that have been exchanged.
- High: Indicates the share price at its all-time high for the day.
- Low: Indicates the share price at its all-time low for the day.
- Turnover: Indicates the total number of shares traded for the day.

The illustration on the right illustrates a sample dataset that includes the characteristics of the Google stock price dataset, such as the date, open, close, high, and low values, as well as the volume.

Figure 5 illustrates the block flow of the numerous phases

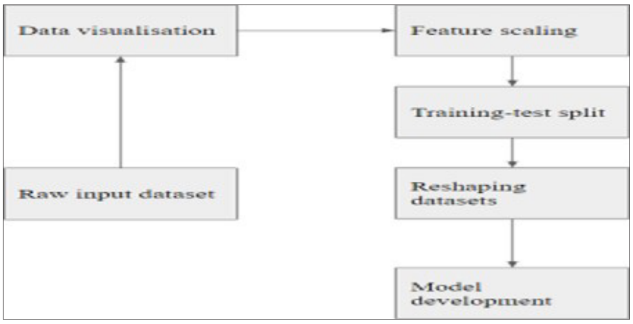



**Figure 5:** Block flow of various LSTM predictive model.

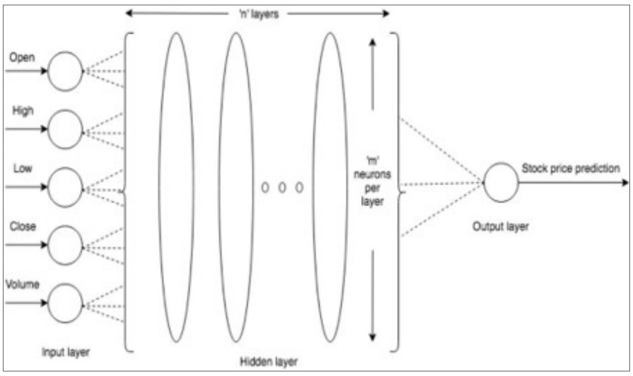



**Figure 6:** Construction overview of LSTM predictive model.

that the data go through in order to be able to input them into the LSTM prediction model.

### Construction of LSTM predictive model

Figure 6 presents construction overview of LSTM predictive model.

## Results and Discussion

### Building RNN

It was our responsibility to import all of the necessary libraries and packages so that RNN could be constructed. Keras includes a number of different library options, including dense layers, sequential, LSTM, and dropouts. Keras is essentially a tensor flow application programming interface that can be used to create and train DL models. The dense layers are a matrix multiplication of the input values that were obtained in the phase before this one, and this dense layer is the one that is utilized to change the proportion of the output. This model is a stack of plain layers, which, in simpler terms, means that it only has one input and one output tensor. Sequential layers are employed here to produce a sequential model, which consists of exactly one input and one output tensor.

### Stacked LSTM

The sequential input dense layer is then followed by LSTM layers, which consist of an activation function in one dense layer, linear activation in one output dense layer, and another linear activation in the output dense layer, respectively. Because there is a possibility of overfitting when LSTM is employed, dropouts have been incorporated. When an input connection is given along with some probabilities, the data on the node can be removed from the activation function and weight update of each LSTM data block.

### Prediction and accuracy

The model is trained by utilizing the dataset that is used for training (Figure 7). It has been decided that 80% of the whole dataset would be used for training purposes. The train and test split are accomplished in the proportion of 80:20. The selection of a higher ratio of the training set should be done so that one may achieve greater accuracy and precision. After the model has been "fitted" to the training dataset, it may then be utilized to make predictions regarding the end-of-day stock of a certain stock. Because the prediction changes depending on the initial conditions that are used, RMSE (Root Mean Squared Error) is utilized to analyze the data. Therefore, by iteratively performing the model building numerous times while simultaneously providing RMSE with string conditions and outputs. After that, the value average RMSE will show how well the model predicts or forecasts entirely new data that has not been observed before. It implies contrasting the forecasts with the observable patterns by inferring information from previous years. RMSE is a measure that can be used to determine how accurate a model is.

### Visualization of results

The Google stock price dataset was used to compile the information shown in the accompanying graph, which represents the company's shares as of the end of the trading day. From figure 8, it is possible to deduce that the predicted value is extremely near to that of the actual value that was derived from the historic dataset.

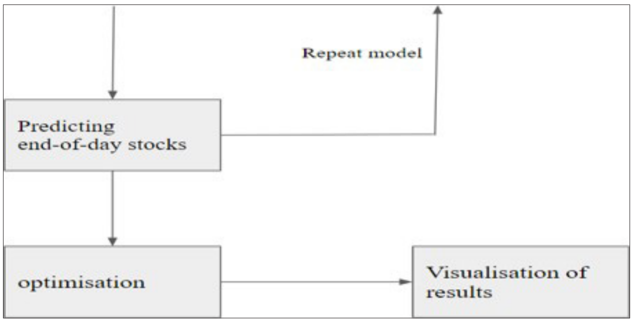



**Figure 7:** Analysis of predicted accuracy.

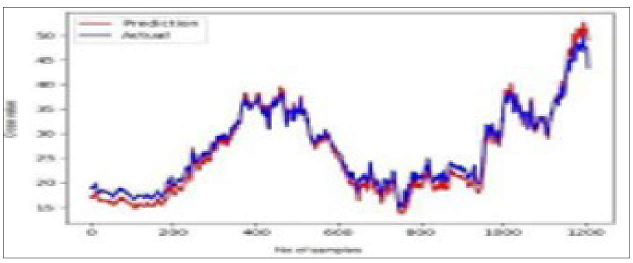


**Figure 8:** Virtualized outputs of the proposed method.

## Conclusion and Future Work

Since the beginning of time, people have been fascinated with the stock market, and ever since then, that fascination has only grown. This is pushing a lot of researchers to create new algorithms that are efficient and new techniques of predict stock prices. We have utilized RNN with layered LSTM layers in this paper because it has been demonstrated to be more accurate than ANN. In ANN, a node in a hidden layer is nothing more than an activation function; however, in LSTM, each node has the capacity to store some data and use it in the processing of other input. Additionally, it is believed that the accuracy improves as the size of the dataset grows, which immediately refers to the importance of expanding the training set. Because with more data, more patterns can be discovered, and weights may be modified in a manner that is more effective. The incorporation of sentiment analysis into the process of stock market forecasting may prove to be quite beneficial in the long run. A lot of information regarding people's perspectives on stocks may be found on social media platforms like Twitter and Facebook. The amount of interest that people have in a particular share as well as the fluctuations in that share's price over the course of time can have an effect on investments in that share.

## Acknowledgements

None.

## Conflict of Interest

None.

### References

1. Pathak A, Shetty NP. 2019. Indian Stock Market Prediction Using Machine Learning and Sentiment Analysis. In Behera H, Nayak J, Naik B, Abraham A (eds) Computational Intelligence in Data Mining. Advances in Intelligent Systems and Computing. Springer, Singapore, pp 595-603.
2. Reddy VKS. 2018. Stock market prediction using machine learning. *Int Res J Eng Technol* 5(10): 1033-1035.
3. Moghar A, Hamiche M. 2020. Stock market prediction using LSTM recurrent neural network. *Procedia Comput Sci* 170: 1168-1173. https://doi.org/10.1016/j.procs.2020.03.049
4. Chen K, Zhou Y, Dai F. 2015. A LSTM-based method for stock returns prediction: a case study of China stock market. In IEEE International Conference on Big Data, Santa Clara, CA, USA.
5. Abbasimehr H, Shabani M, Yousefi M. 2020. An optimized model using LSTM network for demand forecasting. *Comput Ind Eng* 143: 106435. https://doi.org/10.1016/j.cie.2020.106435
6. Nazar NB, Senthilkumar R. 2017. An online approach for feature selection for classification in big data. *Turkish J Electr Eng Comput Sci* 25(1): 163-171. https://doi.org/10.3906/elk-1501-98
7. Chowdary MK, Turaka R, Alabduallah B, Khan M, Babu JC, et al. 2023. Low-power very-large-scale integration implementation of fault-tolerant parallel real fast fourier transform architectures using error correction codes and algorithm-based fault-tolerant techniques. *Processes* 11(8): 2389. https://doi.org/10.3390/pr11082389
8. Arulananth TS, Chinnasamy P, Babu JC, Kiran A, Hemalatha J, et al. 2023. Edge detection using fast pixel based matching and contours mapping algorithms. *PLoS One* 18(8): e0289823. https://doi.org/10.1371/journal.pone.0289823

9. Ali L, Azim MI, Ojha NB, Peters J, Bhandari V, et al. 2023. Integrating forecasting service and Gen2 blockchain into a local energy trading platform to promote sustainability goals. *IEEE Access* 12: 2941 - 2964. https://doi.org/10.1109/ACCESS.2023.3347432

10. Kiran A, Mathivanan P, Mahdal M, Sairam K, Chauhan D, et al. 2023. Enhancing data security in IoT networks with blockchain-based management and adaptive clustering techniques. *Mathematics* 11(9): 2073. https://doi.org/10.3390/math11092073

11. Costea A. 2014. Applying fuzzy logic and machine learning techniques in financial performance predictions. *Procedia Econ Financ* 10: 4-9. https://doi.org/10.1016/S2212-5671(14)00271-8

12. Vijh M, Chandola D, Tikkiwal VA, Kumar A. 2020. Stock closing price prediction using machine learning techniques. *Procedia Comput Sci* 167: 599-606. https://doi.org/10.1016/j.procs.2020.03.326

13. Parmar I, Agarwal N, Saxena S, Arora R, Gupta S, et al. 2018. Stock market prediction using machine learning. In First International Conference on Secure Cyber Computing and Communication, Jalandhar, Punjab, India.

14. Pati S, Aga S, Jayasena N, Sinclair MD. 2022. Demystifying bert: system design implications. In IEEE International Symposium on Workload Characterization, Austin, TX, USA.