ISSN 2063-5346

ANALYTICAL STUDY OF EFFECTIVENESS OF LONG SHORT TERM MEMORY MODEL FOR STOCK TIME SERIES DATA

Satyakam Behar[*1], Dr. K.P.Yadav [2], Dr.Anurag Sharma [3]
[*1] Research Scholar, Dept. of CSE, MATS School of Engineering and Information Technology, Aarang
[2] Vice-Chancellor, MATS University, Aarang
[3] Principal, RSR - Rungta College of Engineering & Technology, Bhilai

Article History: Received: 02.04.2023 **Revised:** 20.05.2023 **Accepted:** 22.06.2023

Abstract
Stock market prediction is always a challenge for investors to make a profit in the financial markets. A variety of models have been proposed for better prediction but none of them are capable of producing a highly accurate prediction. However, Deep Learning (DL) models are providing good results for stock market analysis. Long Short Term Memory (LSTM) is one such DL model which performs well for stock data. Therefore, this paper offers an LSTM-based stock market prediction model. The LSTM model is tested on the stock data of Apple (AAPL), Google (GOOG), Microsoft (MSFT), and Amazon (AMZN). The predicted outcomes are evaluated with standard error metric. The results show the effectiveness of the LSTM model for stock time series analysis by providing lower error rates.
Key words: Stock Market, Deep Learning (DL), Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM).

1. Introduction

In nearly every nation, there exists at least one stock exchange, serving as a platform for the buying and selling of shares from listed companies. These exchanges function as secondary marketplaces. When a company decides to go public and lists itself on a stock exchange, the promoter group adheres to governmental regulations by selling a significant portion of shares to the general public. In the primary market, during the company's incorporation, shares are typically acquired by promoter groups or institutional investors. Subsequently, once the promoter group sells a substantial portion of shares to retail investors, those shares can be traded in the secondary market, specifically within the stock exchanges. Investors are always interested to know the future value of the stock. Therefore, stock market prediction is of keen interest to researchers. Historical stock data (present in the form of time series) is used for this purpose [1].

The process of extracting useful statistics from data available is the crux of time series prediction and analysis. It is related to the field of data mining and so it is implemented for predictive analysis of behavioural patterns in trends [2]. This time series prediction and analysis is actually the prediction of each character or element in a manner that is the additional rare stage of granularity [3]. The running of predictive analytics focuses on observing the connection between a

number of the predicted factors and the explanatory factors that have taken place in the past. It is then used for the prediction of the unknown final results and these figures are inspected for the future prediction. These models are much more advanced for the prediction of patterns and the occasions of time series in the future. It has been observed that most of the trends in this prediction may produce the progress of these results thereby constituting a progress probability about these events. Some of the notable applications of this field are time series change control, retention of clients, fraud detection, advertising and marketing, clinical choice, help structures, and analytical patron dating control [4]. Machine learning, a form of artificial intelligence comprises different progressive statistical approaches of category and regression which are proven to be beneficial within the predictive evaluation. In this process, four Neural networks (NNs) are used in the evaluation, prediction, forecasting, and deep neural networks (DNN). It is a variation of the multiple layered perceptron that is used by different examiners in extrapolative analytics which further acts as a particularly disturbing location inside the subject of commercial enterprise intelligence, and it's essential within the basic boom of the employer [5].

Predicting stock prices is a complex task, as it is influenced by various factors such as market trends, political events, and financial indicators. The fluctuations in stock prices are driven by the forces of supply and demand, which can be unpredictable at times [1]. To identify patterns and trends in stock prices, DL techniques can be used for machine learning. Long short-term memory (LSTM) is a type of Recurrent Neural Network (RNN) that is specifically designed for sequence modeling and prediction [6]. LSTM is capable of retaining information over an extended period of time, making it an ideal approach for predicting stock prices. As a result, RNNs are well-suited to time series data, where they process data step-by-step, maintaining an internal state where they store the information they have seen so far in a compressed form. Accurate prediction of a stock's future price can provide significant financial gain to investors.

Recently researchers paid attention to the DL model for financial time series analysis. Verma et al. [7] analysed behaviour of beauty industries during COVID19. The author used LSTM and Convolutional Neural Network (CNN) for the forecasting of stock prices and concluded that LSTM performed better than CNN. Ache et al. [8] created the Deep LSTM Neural Network, which offered improved execution in expectation for the Shanghai A-share composite list, however the development of the research-related datasets for checking the relevance of the models in another securities exchange with expanded exact-ness was unrealistic in this strategy. Singh and Srivastava [9] planned the two-directional two-dimensional head segment examination (2D)2PCA+Deep NN strategy, which upgraded the Stock Multimedia Prediction Accuracy, yet, the DL approach if there should arise an occurrence of huge window size and high measurement offered just low execution. Hiransha et al. [10] used four DL model (Multi-Layer Perceptron (MLP), RNN, CNN, and LSTM) for prediction of stock closing prices of NSE and NYSE. The author compared the DL models with the statistical models and concluded that DL performed better where CNN outperformed all the models. Nabipour et al. [11] compared performance of the tree-based models, Artificial Neural Network (ANN), RNN, and LSTM. The author's finding was that LSTM is better in predicting with reduced error rate.

Literature motivated us to utilize the DL model for stock market prediction. The objective of this paper is to predict stock market prices using the LSTM model. The subsequent section of the paper is

organized as follows. Section two provides the details of LSTM architecture, section three gives the implementation steps, section four gives the experimental results, and section five gives the conclusion of the paper.

1. LSTM Architecture

2.1 An overview of Recurrent Neural Network (RNN)

In a traditional ANN, the outputs at the final stage are typically not utilized as inputs for subsequent steps. However, when examining real-world phenomena, it becomes evident that the ultimate output depends not only on external inputs but also on previous outputs. A concrete example is the act of reading a book, where the comprehension of each sentence relies not just on the current set of words but also on understanding the preceding sentence and establishing context from past sentences. Unlike humans who maintain continuity in their thinking process, classical neural networks lack the ability to utilize context-based reasoning, thus posing a significant limitation. To address this limitation, RNN were developed, which incorporate feedback loops within the network to retain and propagate information. Figure 1 depicts a basic RNN with a feedback loop, alongside its unrolled equivalent representation.

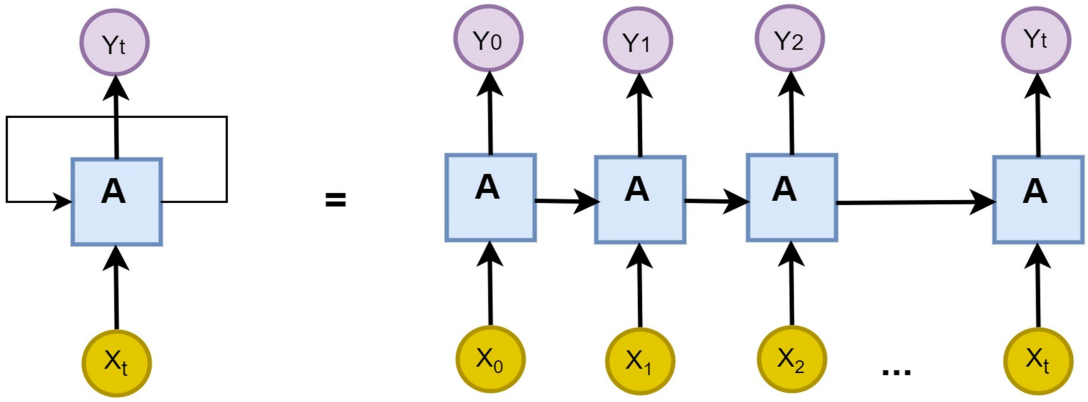

Figure 1: Structure of Recurrent Neural Network

In a classical ANN, the output at each time step (t) is generated based on the input (X_t) provided. Subsequently, at the next time step ($t+1$), the network takes inputs X_{t+1} and the previous output (h_t) to produce the subsequent output (h_{t+1}). This iterative process is facilitated by a loop that enables the flow of information across different steps of the network. However, conventional RNN exhibit limitations. While they excel in utilizing recent contextual information for accurate outputs, they struggle when it comes to relying on distant context information learned from a long time ago for generating correct predictions. This limitation of RNN has been extensively discussed by Hochreiter [12] and Bengio et al. [13], who have examined the underlying factors contributing to the challenges faced by RNNs in long-term scenarios. Fortunately, Long Short-Term Memory (LSTM) networks have been specifically developed to address and overcome this issue.

2.2 LSTM Networks

Hochreiter and Schmidhuber [14] introduced a specialized variant of RNNs designed to address the challenge of learning long-term dependencies. This pioneering work laid the foundation for subsequent advancements by various researchers [15]. Over time, LSTM networks have undergone refinement to effectively handle the issue of long-term dependencies. Detailed explanations

regarding the evolution and development of LSTMs from RNNs can be found in [16]. Unlike the repeating modules in standard RNNs, which typically consist of a simple structure like a single *tanh* layer (as illustrated in Figure 2), LSTMs exhibit a chain-like structure with a unique repeating module.

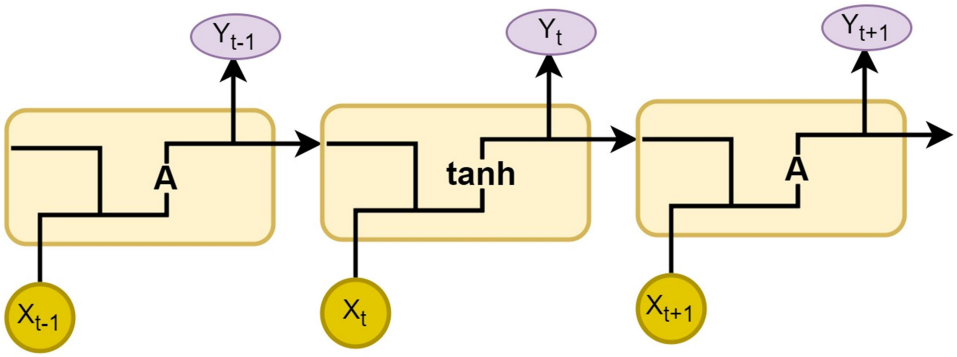

Figure 2: A standard RNN's repeating module consists of a single layer.

This module is composed of four layers that interact in a distinctive manner, as depicted in Figure 3. Each line in Figure 3 represents a complete feature vector, connections between nodes serve to link the output of one node to the inputs of others. In this visual representation, the pink circles depict pointwise operations like vector addition, while the yellow boxes symbolize neural network layers that are learned through training. Lines merging indicate concatenation, where the outputs are combined, while a line forking signifies the duplication of content, with the copies directed to different locations.

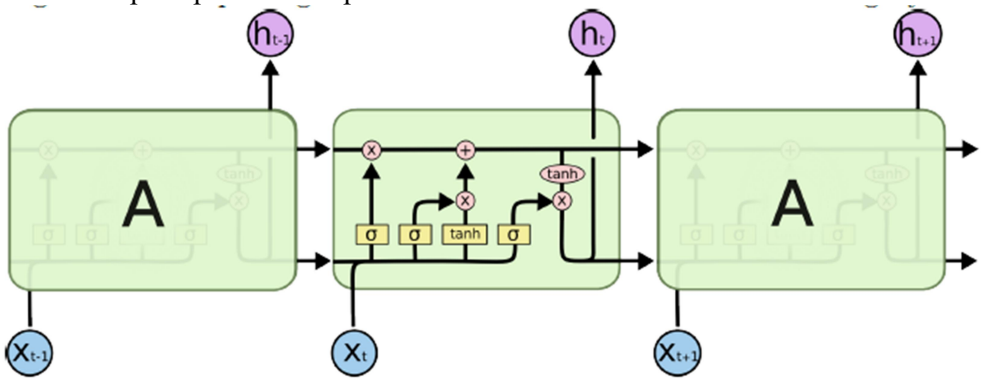

Figure 3: An LSTM's repeating module comprises four layers that interact with each other.

2.2.1 Working of LSTM

The crucial element in LSTMs is the cell state, represented by the horizontal line spanning the top of the diagram. The cell state functions akin to a conveyor belt, running vertically throughout the entire chain while undergoing minor linear interactions. What sets LSTMs apart is their capability to selectively add or remove information from the cell state, which is regulated by specialized structures known as gates. These gates

serve the purpose of allowing or blocking the passage of information. Each gate is composed of a sigmoid neural network layer and a pointwise multiplication operation. The sigmoid layer produces values ranging from 0 to 1, indicating the extent to which each component should be allowed through. A value of 0 signifies "do not allow anything through," while a value of 1 signifies "allow everything through." Within an LSTM, there are three of these gates, which serve the purpose of safeguarding and regulating the cell state. The initial stage of the LSTM involves determining which pieces of information are to be discarded from the cell state. The "forget gate layer" is constructed using a sigmoid layer. It examines h_t, h_{t-1}, and $x_t x_t$ as inputs and produces a value between 0 and 1 for each element in the cell state $c_{t-1}c_{t-1}$. A value of 1 (represented as 11) indicates that the corresponding component should be entirely retained, while a value of 0 (represented as 00) signifies complete removal. The subsequent step involves determining which new information will be stored in the cell state. This process consists of two parts. Firstly, a sigmoid layer known as the "input gate layer" determines which values should be updated. Subsequently, a *tanh* layer generates a vector of new potential values, C_tC_t, that could be incorporated into the state. Finally, these two components are combined to create an update to the cell state. The previous cell state, $c_{t-1}c_{t-1}$, is updated to the new cell state, c_tc_t. This is achieved by multiplying the old state by f_tf_t, obtained from the forget gate. Then, the product is added to the new candidate values, C_t, which are scaled based on the decision of how much to update each state value. Moving forward, the output is determined. It is a filtered version of the cell state. Firstly, the cell state is passed through a sigmoid layer that determines which parts of the cell state should be outputted. Next, the cell state is processed through a *tanh* function to constrain the values between -1 and 1. This result is then multiplied by the output of the *sigmoid* gate, ensuring that only the selected parts of the cell state are ultimately outputted.

2. Implementation Steps

This section discusses the implementation steps involved in the prediction of stock market prices.

i. Data Collection: The historical stock prices of selected companies are collected on a daily basis from the official website of Yahoo Finance.

ii. Data Pre-processing:
a) Data discretization: This step involves reducing data by converting numerical data into discrete categories, which is particularly important for certain applications.
b) Data transformation: One of the transformations commonly applied is normalization, which scales the data to a standard range for better analysis.
c) Data cleaning: Involves handling missing values by filling them in using appropriate techniques to ensure data completeness and accuracy.
d) Data integration: Combining multiple data files or sources into a unified dataset to gain a comprehensive view of the data.
Once the dataset has undergone transformation and cleaning, it is divided into training and testing sets to evaluate the model's performance. For instance, in the context of a time-series problem, a data structure is created with 60 time steps as input and 1 output.

iii. Feature Selection: In this particular step, specific data attributes are selected to be utilized as inputs for the neural network. For this study, the chosen features are Date and Close Price.

iv. Train the RNN model: The neural network (NN) model is trained by providing the training dataset as input. Initially, the model is initialized with random weights and biases. The proposed LSTM model is structured with a sequential input layer followed by three LSTM layers. Subsequently, a dense layer with activation function is added. The output layer comprises a dense layer with a linear activation function.

v. Output Generation: The output generated by the RNN is compared to the target values, and the difference in error is calculated. The Backpropagation algorithm is employed to minimize this error by adjusting the biases and weights of the neural network.

vi. Updating of Test set: Repeat step 2 for updating test data

vii. Error and companies' net growth calculation: By calculating the deviation, we assess the percentage of error in our prediction compared to the actual price and visualize the data.

viii. Investigate different time interval: We performed this process repeatedly to predict the price at various time intervals. In our case, we used a 2-month dataset as training data to forecast the close price of the share for time spans of 3 months, 6 months, 1 year, and 3 years. For each time span, we calculated the percentage of error in the future prediction. It's important to note that this error percentage may vary across different sectors. Therefore, this approach can aid in establishing a framework specific to a particular sector, enabling the prediction of future net growth for companies within that sector.

These steps are implemented on the Python 3.7. We have used sequential LSTM, adam optimizer, Mean Square Error (MSE) as loss function,1000 epochs, and 100 as batch size.

3. Experimental Results

For implementation purposes, we have used the dataset collected from the website of Yahoo Finance. Historical stock data of 'AAPL', 'GOOG', 'MSFT', and 'AMZN' are collected from 27-June-2022 to 23-06-2023. The dataset contains 250 observations for each selected company. The dataset is spilt into 80:20 train-test ratio. We have predicted next day's stock closing price. Performance of LSTM is evaluated using MSE, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Table 1 provides the performance metric. Figure 1-4 provide the prediction graph for AAPL, GOOG, MSFT, and AMZN. It can be observed from the table and figures that LSTM is good in predicting a day ahead stock market prices

.

Table 1. Performance Metric

Dataset	MSE	RMSE	MAE	MAPE
AAPL	143.3647	11.9735	10.8471	14.7472
GOOG	26.4504	5.1430	11.4895	34.7352
MSFT	26.6462	5.1620	12.6379	28.6992
AMZN	7.9914	2.8269	10.3527	28.1896

Figure 1. AAPL

Figure 2. GOOG

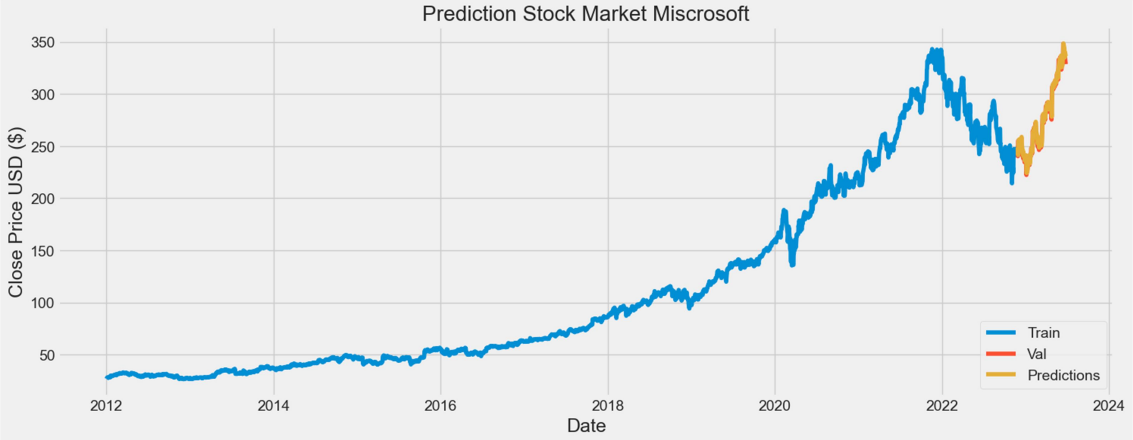

Figure 3. MSFT

Figure 4. AMZN

4. Conclusion

This paper presents a RNN model based on LSTM specifically designed for forecasting future values of the AAPL, GOOG, MSFT, and AMZN. The experimental results demonstrate the promising performance of our model despite of nonlinear pattern of stock market data. The testing phase confirms that our model effectively captures the trends in closing prices for all four assets. In our future work, we plan to explore optimal combinations of data length and training epochs to further enhance the accuracy of our predictions and better suit the characteristics of the assets under consideration.

References

[1] S. Verma, S. P. Sahu, and T. P. Sahu, "Stock Market Forecasting with Different Input Indicators using Machine Learning and Deep Learning Techniques: A Review.," *Eng. Lett.*, vol. 31, no. 1, 2023.

[2] R. Buettner, "Predicting user behavior in electronic markets based on personality-mining in large online social networks: A personality-based product recommender framework," *Electron. Mark.*, vol. 27, pp. 247–265, 2017.

[3] C. Nyce and A. Cpcu, "Predictive analytics white paper," *Am. Inst. CPCU. Insur. Inst. Am.*, pp. 9–10, 2007.

[4] A. Mosavi and A. Vaezipour, "Developing effective tools for predictive analytics and informed decisions," *Univ. Tallinn, Tech. Rep.*, 2013.

[5] A. Thakkar and K. Chaudhari, "Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions," *Inf. Fusion*, vol. 65, no. July 2020, pp. 95–107, 2021, doi: 10.1016/j.inffus.2020.08.019.

[6] H. Rezaei, H. Faaljou, and G. Mansourfar, "Stock price prediction using deep learning and frequency decomposition," *Expert Syst. Appl.*, vol. 169, no. September 2020, p. 114332, 2021, doi: 10.1016/j.eswa.2020.114332.

[7] A. Jain, T. Sukhdeve, H. Gadia, S. P. Sahu, and S. Verma, "COVID19 Prediction using Time Series Analysis," *Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021*, pp. 1599–1606, 2021, doi: 10.1109/ICAIS50930.2021.9395877.

[8] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," *J. Supercomput.*, vol. 76, no. 3, pp. 2098–2118, 2020, doi:

10.1007/s11227-017-2228-y.

[9] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimed. Tools Appl.*, vol. 76, no. 18, pp. 18569–18584, 2017, doi: 10.1007/s11042-016-4159-7.

[10] M. Hiransha, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "NSE Stock Market Prediction Using Deep-Learning Models," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1351–1362, 2018, doi: 10.1016/j.procs.2018.05.050.

[11] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. Shahab, "Deep learning for stock market prediction," *Entropy*, vol. 22, no. 8, 2020, doi: 10.3390/E22080840.

[12] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," *Diploma, Tech. Univ. München*, vol. 91, no. 1, 1991.

[13] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[15] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif. Intell. Rev.*, vol. 53, pp. 5929–5955, 2020.

[16] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D Nonlinear Phenom.*, vol. 404, p. 132306, 2020.