Analysis & Stock Price Prediction and Forecasting Using Different LSTM Models

Supriya Raut
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India

Prof. Avinash Shrivas
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India

---***---

Abstract - The objective of this research is to develop a Deep Learning model to forecast the stock price, by using the variant of Long Short-Term Memory. This model predicts the close price of the stock for the future selected date, choosing as inputs the following data: open, high, low, adj close and close prices. This model shows a comparative analysis between three different LSTM networks: Long Short-Term Memory (LSTM), Stacked Long Short-Term Memory (Stacked LSTM), and Stacked Bi-directional Long Short-Term Memory (Stacked Bidirectional LSTM) concluding which one is the best and implementing the model using that variant. We have used the historical stock prices data from yahoo's financial website over 5 years, by choosing multiple datasets: Apple, Amazon, Google, Meta, Microsoft and Tesla (daily values). In order to get effective in the forecasting model, we have tested the network with different iterations and epochs. The model represents Multiple Graphs for data visualization in different comparisons.

We have estimated the effectiveness of our proposed model by using the following performance indicators: the Mean Square Error (MSE), the Root Mean Square Error (RMSE), and the R-Squared of the model. The experimental results clearly show that our Stacked Bi-LSTM model has the highest accuracy values when comparing with the LSTM and Stacked LSTM Models. Hence, we can conclude that our Stacked Bi-LSTM Model is suitable for accurate prediction of the stock market time series.

Key Words: ***stock price prediction, Machine Learning, stacked LSTM, Bi-directional LSTM, Deep Learning, Data pre-processing techniques, Data normalization, Data Visualization, Training and Testing Set, Financial Time Series, Future prediction***

I. INTRODUCTION

The prediction of stock prices and the direction of the price movement is a widely studied topic in many fields including trading, finance, statistics, and computer science. Typically, the professional stock market traders use fundamental and/or technical analysis of stocks to analyze the market elements to make investment decisions [1]. Fundamental analysis is the conventional method involving the study of company fundamentals such as sales, revenue and expenses, net profit, debts, annual growth rates, and so on, whereas technical analysis is exclusively based on the study of historical price movements. Practitioners of technical analysis refer to the price charts to study price patterns and use price data in different calculations to forecast future price movements. The technical analysis paradigm assumes there exists an inherent correlation between the performance of the company and its stock price and therefore, it is used to determine the timing of buying and selling the shares of the company.

Time-series prediction problems like the prediction of stock prices are a difficult type of predictive modeling problem. Such problems usually involve the presence of sequence dependence among the input variables [2]. Machine learning has the potential to predict the stock market by training and testing the models with the historical datasets, social media data, crawled financial news or trends. A robust type of neural network equipped to handle the sequence dependence is the Recurrent Neural Networks (RNNs). The Long Short-Term Memory network or LSTM network is a type of RNN and is used extensively in time-series problems because of its capability to handle dependency between both long and short distance sequences [3].

The objective of this research is to develop the predictive model by using the variant of LSTM to forecast the stock prices in the current and future market. We have used the stock prices market from yahoo's financial website by choosing Multiple datasets of different companies on a daily basis. To measure the effectiveness of our proposed model, we have used the best prediction values for our model compared with the experimental results of LSTM, Stacked LSTM, and Stacked Bi-directional LSTM.

Our contributions in this paper are as follows: 1. Implemented three models for daily frequency stock price prediction- LSTM, Stacked LSTM, and Stacked Bi-directional LSTM. 2. Performed rolling segmentation on the training set and testing set of the raw data to investigate the effect of the model parameter update cycle on stock forecast performance. 3. Evaluate the result based on the performance matrix and find the most effective model.

We found that Stacked Bi-directional LSTM outperformed the other two significantly with an error of 0.89 compared to 0.67 obtained by LSTM and 0.83 obtained by Stacked LSTM.

The main structure of this research is as follows: Section II presents the details of the literature reviews. Section III describes the methodology and explains the proposed model. In Section V, the details of the stock prices dataset and the forecasting performance measures are presented. In Section VI, we present the experimental results. The final conclusions are given in Section VII.

II. LITERATURE REVIEW

Stock market prediction seems a complex problem because there are many factors that have yet to be addressed and it doesn't seem statistical at first. But by proper use of machine learning techniques, one can relate previous data to the current data and train the machine to learn from it and make appropriate assumptions.

Stock price forecasting attracts researchers for a long time due to its significant financial benefits. Among different types of techniques applied thus far, the most ordinarily utilized system is Artificial Neural Network (ANN) proposed by Verma etal. [12]. ANNs are mainly affected by over-fitting problems. Additionally, Support Vector Machines (SVMs) can be utilized as an option to avoid such an overfitting issue [13]. Usmani etal. foresee the trend of Karachi Stock Exchange (KSE) by proposing the primary target of this examination on day closing utilizing diverse machine learning algorithms [14]. They utilized the old statistical models including ARIMA, and SMA to predict stock prices. Furthermore, other machine learning models such as SLP (Single Layer Perceptron), MLP (Multi-Layer Perceptron), RBF (Radial Basis Function), and SVM (Support Vector Machine) are also used. The MLP algorithm performed best when contrasted with different methods [14].

The LSTM model was also used in different time series forecasting applications. Shao et al. [15] introduced a framework that can forecast available parking spaces in multi-steps ahead using the LSTM model. Seong et al. [16] used an encoder decoder LSTM model that utilized current vehicle trajectory to forecast the future trajectory of surrounding vehicles. Rui et al. [17] predicted traffic flow using LSTM and GRU neural network methods. Salman et al. built [18] a flexible but robust statistical model to forecast weather conditions in the surrounding area of airports in Indonesia. For financial time series forecasting, Persio et al. [20] investigated the adequacy and proficiency of introducing LSTM. Akita et al. [21] consolidated data by the information of paper articles to display the effect of previous incidents on the opening price of the stock market. Their presented formula took care of numerical and printed information to the LSTM system to execute precise forecasting.

From Literature review, we came to know that the SVM and other models failed to consider factors that are used for testing the prediction model, this includes macro and micro factors. Additionally, implementing SVM methodologies can be very tedious due to the number of algorithms involved. Lastly, SVM models based on a small amount of training and testing sample, it is not suitable for financial time-series forecasting on a large scale. To address the limitation of a SVM Model and other models referred so far.

However, to our knowledge, none of the works presented thus far showed the comparison between LSTM and Stacked LSTM and Stacked BI-LSTM in terms of the performance improvement of stock price prediction.

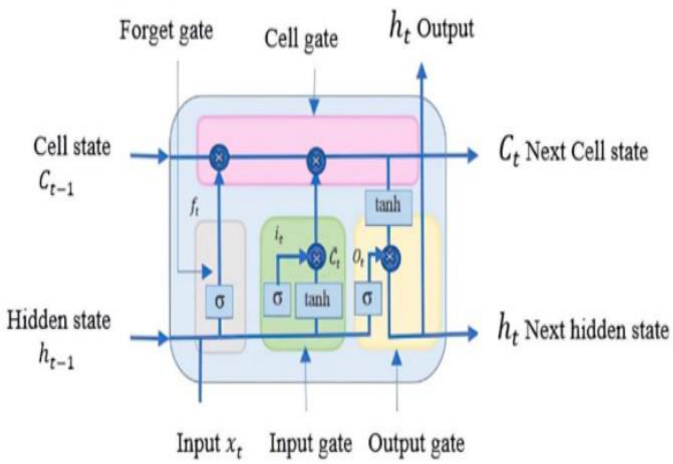
We propose to develop a Deep Learning Model. A Model which shows the comparative analysis between different LSTM Networks such as LSTM, and stacked LSTM and Stacked BI-LSTM models. It is a deep learning model whose architecture can be defined as an LSTM model composed of multiple LSTM layers. Despite the potential of the existing SVM Model, it was time consuming and is not suitable for financial time- series forecasting on a large scale which is much needed for stock price prediction. Here to know why we proposed this Model and to understand the Stacked LSTM we need to first know how the Basic LSTM Model works. Lately, the LSTM network, which is suitable for learning temporal patterns, is extensively utilized for various tasks of time-series analyses. LSTM is preferred over the conventional RNN as it overcomes the problem of vanishing (or exploding) gradients and as it can effectively learn long term dependencies through memory cells and gates. Thus, many studies on financial time-series modeling are conducted using LSTMs.

III. METHODOLOGY & PROPOSED MODEL

A. Long Short-Term Memory (LSTM):

LSTM was first proposed by Hochreiter and Schmidhuber in 1997, and later became very popular especially to address time series prediction problems [9]. Being a modified RNN method, LSTM works well on a large variety of problems, and is widely used now. LSTM handles the issue of figuring out how to recollect data over a period of time, by presenting gate units and memory cells in the neural network design. The memory cells have cell states that store recently experienced data. Each moment the information reaches a memory cell, the outcome is controlled through the combination of cell state, and then, the cell state is refreshed. Now, if any other information is received by the memory cell, the output is processed utilizing both this new information, and the refreshed cell state. LSTM is intended to maintain a problem having long term dependency. Their default conduct is to remember information for a long period of time, not something they learn through struggle. It beats the vanishing gradients problem faced by a general RNN by substituting the ordinary neuron by a complex architecture called the LSTM unit or block [6]. Here the LSTM, comprises 3 LSTM layers.

In Fig. 1, the three available gates in LSTM are the input gate(it), the forget gate (ft) and the output gate (ot). Writing a certain neural network part in the memory begins by first inputting it to the memory cell via the input gate.



The working of LSTM can be summarized by the following set of equations:

$$
\begin{aligned}
z_t &= \tanh(W^z x_t + R^z h_{t-1} + b^z) \\
i_t &= \sigma(W^i x_t + R^i h_{t-1} + b^i) \\
f_t &= \sigma(W^f x_t + R^f h_{t-1} + b^f) \\
o_t &= \sigma(W^o x_t + R^o h_{t-1} + b^o) \\
s_t &= z_t \cdot i_t + s_{t-1} \cdot f_t \\
h_t &= \tanh(s_t) \cdot o_t
\end{aligned} \quad (1)
$$

(Ref: LSTM Related Work [6].)

where 'it' denotes the input gate and 'ot' denotes the output gate. The forget gate, memory cell, and hidden state are denoted by 'ft', 'st', and 'ht', respectively. The 'σ' and 'tanh' functions are defined respectively. The σ and tanh functions are defined in (10) respectively.

$$
\sigma(z) = \frac{1}{1+e^{-z}}
$$

$$
\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2)
$$

(Ref: LSTM Function Equation [6].)

B. Stacked Long Short-Term Memory (Stacked LSTM):

Now, Stacked LSTMs are a stable technique for challenging sequence prediction problems. A Stacked LSTM architecture can be defined as an LSTM model composed of multiple LSTM layers. An LSTM layer above provides a sequence output rather than a single value output to the LSTM layer below. Specifically, one output per input time step, rather than one output time step for all input time steps. The Input is only required for the first layer.

Stacked LSTM networks provide a deeper model for learning and are composed of multiple hidden layers of LSTMs. These multiple hidden layers act as a Deep Recurrent Neural Network (DRNN). In Fig 2. We can see the Architecture of a Stacked LSTM.

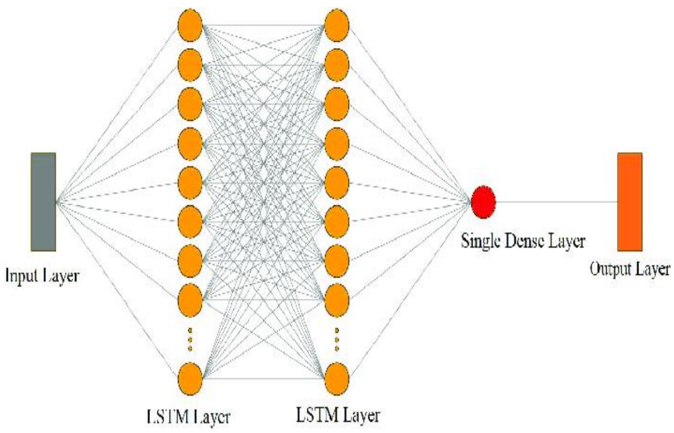


Fig. 2: Stacked LSTM layer [13]

The main reason that we decided to use a stacked LSTM is to allow for greater model complexity. At every time step an LSTM, besides the recurrent input. If the input is already the result from an LSTM layer (or a feedforward layer) then the current LSTM can create a more complex feature representation of the current input. Having a stacked LSTM representation, more complex input patterns can be described at every layer. Stacked LSTM hidden layers makes the model deeper, more accurately earning the description as a deep learning technique [18]. The additional hidden layers are understood to recombine the learned representation from prior layers and create new representations at high levels of abstraction. For example, some complex problems like stock prediction may sometimes require several stacked hidden LSTM layers.

C. Stacked Bi-directional Long Short-Term Memory (Stacked BI-LSTM):

Stacked Bidirectional Long Short-Term Memory (Bi-LSTM) is a type of artificial neural network architecture commonly used in natural language processing tasks. It consists of multiple layers of Bi-LSTMs, where each layer processes the input sequence and passes its output to the next layer. Here the Stacked BI-LSTM, comprises 3 BI-LSTM layers.

BI-LSTM presented in Fig. 3 is a modified augmentation of the LSTM model. BI-LSTM improves the execution of the model for sequence classification types of problems. BI-LSTM incorporates two LSTMs in the training process of the sequence of inputs rather than using one LSTM. The notion behind Bidirectional Recurrent Neural Networks (BRNNs) can be easily understood.

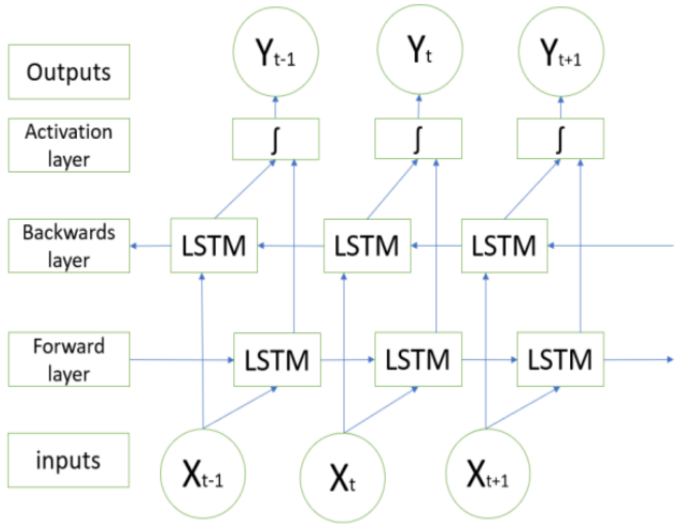


Fig. 3: Bidirectional LSTM layer [13]

This overcomes the constraints of a conventional RNN. The deep learning model known as BRNN can be utilized to access all the previous information, and predicted future information simultaneously. The state neurons of regular RNN are splitted into two types, where one can act as the forward states (positive time heading), and other as the backward states (negative time heading).

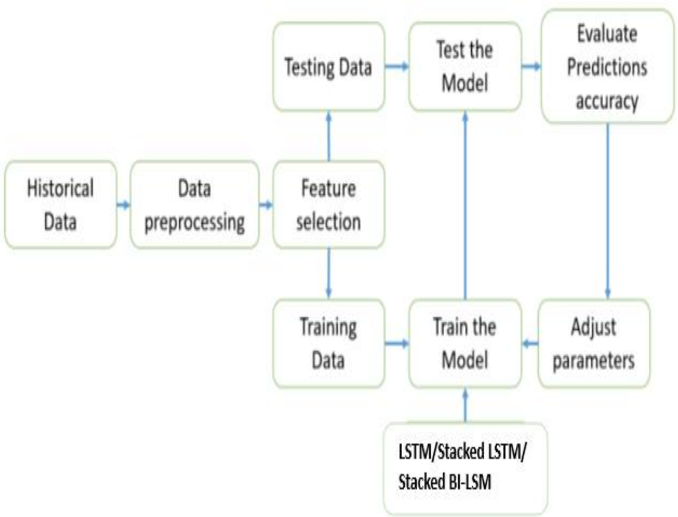


Fig. 4: The proposed architecture of stock price prediction

The process involves obtaining stock data from Yahoo Finance for 6 stocks and then preprocessing the data by cleaning, transforming, and restructuring it. This ensures the data is of high quality and ready for analysis. Common techniques such as data cleaning, normalization, transformation, and integration are used. Feature selection, including basic 5 features and date/time feature extraction, is performed. The processed dataset is divided into training and testing sets before being used to train three models with different tuning parameters to predict stock prices. The accuracy of the predictions is evaluated by comparing them with the testing datasets.

IV. DATASETS AND FORECASTING PERFORMANCE MEASURES

A. Dataset and System Architecture

We obtained historical data of 6 stocks from Yahoo Finance(https://in.finance.yahoo.com/) that are being actively traded on India's National Stock Exchange (NSE). To maximize the available data-points, the stocks were selected such that their historical data range from January 17, 2019 to January 16, 2024. Total data of 5 years is used. There are 253 trading days in a year for the NSE, so on an average we get 5500 data-points for every stock at an interval of 1 day. The format of the input data is numeric. The historical price data contains 5 features- 'Open Price', 'Maximum Price', 'Minimum Price', 'Closing Price' and 'Trading Volume'. Google Collaboratory is used with GPU. Tensor flow is used as our deep learning framework. The data was first preprocessed using minmax feature scaling. Then, the processed dataset is divided into two segments known as training, and testing dataset.

Among the dataset, 70% of the data is utilized as training dataset, and the remaining 30% data is used as testing dataset. The training dataset was run through all the three models, with different tuning parameters to produce the predicted stock prices. Traditionally, data is split for training and testing the models using fixed partitioning methods. Nevertheless, the trading pattern of the stock market changes regularly, for example, investors at times favor stocks with high volatility whereas sometimes choose to invest in technology stocks. Accordingly, we should update the model parameters frequently to adjust to the change of market pattern. For making our experiments closer to real transactions, we carry out rolling segmentation on the training set and testing set of the experimental data. Then, this predicted dataset was compared with the testing datasets, and the prediction accuracy was evaluated.

TABLE I. THE DETAILS OF STOCK PRICES DATASETS

Stock prices Dataset	Feature	Training set 70%	Testing set 30%
Google	open, low, high, close, adj close, volume.	January 2019 – July 2022 (879 days)	July 2022 – January 2024 (378 days)
Amazon	open, low, high, close, adj close, volume.	January 2019 – July 2022 (879 days)	July 2022 – January 2024 (378 days)
Apple	open, low, high, close, adj close, volume.	January 2019 – July 2022 (879 days)	July 2022 – January 2024 (378 days)
Meta	open, low, high, close, adj close, volume.	January 2019 – July 2022 (879 days)	July 2022 – January 2024 (378 days)
Microsoft	open, low, high, close, adj close, volume.	January 2019 – July 2022 (879 days)	July 2022 – January 2024 (378 days)
Tesla	open, low, high, close, adj close, volume.	January 2019 – July 2022 (879 days)	July 2022 – January 2024 (378 days)

B. Performance Measures

The performance of our proposed Model is evaluated by using three performance measures as Mean Square Error (MSE), Root Mean Square Error (RMSE), and R-Squared Error (R^2). The described variables have the following significance: 'yt' is the original value, 'ft' is the predicted value, 'et=yt – ft' is the predict error and 'n' is the extent of the testing dataset. The details of the forecasting evaluation measures are as follows:

1. **Mean square error (MSE):** is a network performance function. MSE measures network performance based on the average error of the forecasting error [19]. This function is used in prediction and regression analysis to verify the results of the experimental. The Value of MSE should be less or nearer to zero. The formula of the MSE is given by:

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^{\,2}$$

2. **Root mean square error (RMSE**): is one of the most common measures used in regression error metrics. It equals the square root of the MSE. RMSE is an error measure that can see how spread out these residuals are. This error measurement can tell how the concentration of data is around the most suitable line [19]. By basic principles, the best value of RMSE should also be close to zero. The equation to calculate the RMSE is given by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^{\,2}}$$

3. **R-Squared**: R-squared is a statistical measure that indicates how much of the variation of a dependent variable is explained by an independent variable in a regression model. In investing, R-squared is generally interpreted as the percentage of a funds or security's price movements that can be explained by movements in a benchmark index. R-squared values range from 0 to 1 and are commonly stated as percentages from 0% to 100%.

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}}$$

V. EXPERIMENTAL RESULTS

After proper scaling, training, and testing between the real data, and predicted data, we observed different types of RMSE, and MSE for using different layers, different units in the hidden layers and dense layers, and also for different epochs for the output prediction. After analyzing different epochs for the LSTM model, Stacked LSTM and Stacked BI-LSTM model, we came to realize that the training epochs should be chosen in the best way to train the model.

Here, we have chosen a 50 epochs training model that has lower MSE and RMSE error which gives us the best prediction accuracy than others.

TABLE II. THE EXPERIMENTAL RESULTS (LSTM MODEL)

Historical stock price data	Epochs/ Iterations	Forecast Measures		
		MSE	RMSE	R-Squared
Google				
	50	0.0052	0.0722	0.678
Apple				
	50	0.0105	0.1029	0.243
Amazon				
	50	0.0042	0.0735	0.842
Meta				
	50	0.0065	0.0791	0.918
Microsoft				
	50	0.0251	0.1584	0.080
Tesla				
	50	0.0061	0.0785	0.546

TABLE III. THE EXPERIMENTAL RESULTS (STACKED LSTM MODEL HAS 3 MAIN LSTM LAYERS).

Historical stock price data	Epochs/ Iterations	Forecast Measures		
		MSE	RMSE	R-Squared
Google				
	50	0.0128	0.1134	0.835
Apple				
	50	0.0020	0.0451	0.852
Amazon				
	50	0.0037	0.0615	0.889
Meta				
	50	0.0021	0.0465	0.941
Microsoft				
	50	0.0024	0.0489	0.901
Tesla				
	50	0.0025	0.0509	0.809

TABLE IV. THE EXPERIMENTAL RESULTS (STACKED BI-LSTM MODEL HAS 3 MAIN BI-LSTM LAYERS).

Historical stock price data	Epochs/ Iterations	Forecast Measures		
		MSE	RMSE	R-Squared
Google				
	50	0.0031	0.0557	0.897
Apple				
	50	0.0030	0.0553	0.880
Amazon				
	50	0.0017	0.0042	0.842
Meta				
	50	0.0016	0.0407	0.958
Microsoft				
	50	0.0022	0.0469	0.919
Tesla				
	50	0.0014	0.0374	0.896

From Table II, Table III, and Table IV, we can clearly see that, the best results of the RMSE, MSE, R-Squared of our model is best when we are using the Stacked BI-LSTM which is more for all the 6 stocks such as 0.897, 0.880, 0.842, 0.958,0.919, and 0.896 respectively. On the other hand, the results for the LSTM and Stacked LSTM are lesser compared to the Stacked BI-LSTM.

The below graphs show the best results of the original and the predicted stock prices (close price) on testing data using Stacked BI-LSTM for some of the above stock price datasets.

Fig. 5: The comparison of the original and the predicted close price of Amazon Stock data with 84 % accuracy.

Fig. 6: The comparison of the original and the predicted close price of Apple Stock data with 88 % accuracy.

Fig. 7: The comparison of the original and the predicted close price of Tesla Stock data with 89 % accuracy.

Fig. 8: The comparison of the original and the predicted close price of Microsoft Stock data with 91 % accuracy.

VI. EXPERIMENTAL RESULTS

Our research presents a Deep Machine Learning technique in forecasting daily stock value price by using the technique of Stacked Bi-LSTM (Bi-directional long short-Term Memory). The daily stock data which was stored on the Yahoo Finance website for more than 5 years from January 2019 until January 2024 have been used. The experimental data have been selected from Multiple databases namely Apple, Amazon, Google, Meta, Microsoft, and Tesla. In designing a forecasting model, we have selected as input data open, high, low, adj close, close and estimate closing price of daily stock market. Test results for all stock data sets clearly shows that the Stacked Bi-LSTM model has been confirmed to be effective in predicting closing price of the stock market with the most accurate prediction values. The results of the Google stock forecast by using the Stacked Bi-directional LSTM have led to the least predictive error in MSE, RMSE measurements of 0.0031 and 0.0557 respectively with an accurate prediction of the stock market closing price as high as 0.89 percent. To assess the effectiveness of the model, we have compared our test results with the performance measurements of our other two models, that is LSTM which gives an Accuracy of 0.67 percent and Stacked LSTM which gives an Accuracy of 0.83 percent. Hence By Evaluated Results It is obvious that the model we have designed provides the best predictive results for futuristic stock market prices.

REFERENCES

[1] V. Atanasov, C. Pirinsky, and Q. Wang, "The efficient market hypothesis and investor behavior," 2018.

[2] S. Rehman, I. U. Chhapra, M. Kashif, and R. Rehan, "Are stock prices a random walk? an empirical evidence of Asian stock markets," ETIKONOMI, vol. 17, no. 2, pp. 237–252, 2018.

[3] S. Agrawal, D. Thakkar, D. Soni, K. Bhimani, and C. Patel, "Stock market prediction using machine learning techniques", IJSRCS, pp. 1099–1103, 04 2019.

[4] M. Afeef, A. Ihsan, and H. Zada, "Forecasting stock prices through univariate Arima modeling," 2018.

[5] P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," Omega, vol. 33, no. 6, pp. 497–505, 2005.

[6] D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, "Comparison of predictive algorithms: Backpropagation, svm, lstm and kalman filter for stock market," in 2019 Amity International Conference on Artificial Intelligence (AICAI). IEEE, 2019, pp. 228–234.

[7] E. Ahmadi, M. Jasemi, L. Monplaisir, M. A. Nabavi, A. Mahmoodi, and P. A. Jam, "New efficient hybrid candlestick technical analysis model for stock market timing on the basis of the support vector machine and heuristic algorithms of imperialist competition and genetics," Expert Systems with Applications, vol. 94, pp. 21–31, 2018.

[8] S. Sharma and B. Kaushik, "Quantitative analysis of stock market prediction for accurate investment decisions in future," Journal of Artificial Intelligence, vol. 11, pp. 48–54, 2018.

[9] S. Jain, M. Kain, and N. Singh, "Prediction for stock marketing using machine learning," Bhagwan Parshuram Institute of Technology, p. 1, 2018.

[10] A. A. Heidari, H. Faris, I. Aljarah, and S. Mirjalili, "An efficient hybrid multilayer perceptron neural network with grasshopper optimization," Soft Computing, pp. 1–18, 2018.

[11] W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," Computer Speech & Language, vol. 30, no. 1, pp. 61–98, 2015.

[12] S. A. Verma, G. Thampi, and M. Rao, "Inter-comparison of artificial neural network algorithms for time series forecasting: Predicting Indian financial markets," International Journal of Computer Applications, vol.162, no. 2, 2017.

[13] M. S. Babu, N. Geethanjali, and B. Satyanarayana, "Clustering approach to stock market prediction," International Journal of Advanced Networking and Applications, vol. 3, no. 4, p. 1281, 2012.

[14] M. Usmani, S. H. Adil, K. Raza, and S. S. A. Ali, "Stock market prediction using machine learning techniques," in 2016 3rd International Conference on Computer and Information Sciences (ICCOINS). IEEE,2016, pp. 322–327.

[15] W. Shao, Y. Zhang, B. Guo, K. Qin, J. Chan, and F. D. Salim, "Parking availability prediction with long short term memory model," in International Conference on Green, Pervasive, and Cloud Computing. Springer, 2018, pp. 124–137.

[16] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder- decoder architecture," in 2018 IEEE Intelligent Vehicles Symposium (IV).

[17] IEEE, 2018, pp. 1672–1678. Q. Xie, G. Cheng, X. Xu, and Z. Zhao, "Research based on stock predicting models of neural networks ensemble learning," in MATEC Web of Conferences, vol. 232. EDP Sciences, 2018, p. 02029.

[18] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

[19] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in 2017 (ICACCI). IEEE, 2017, pp. 1643–1647.

[20] L. Di Persio and O. Honchar, "Artificial neural networks architectures for stock price prediction: Comparisons and applications," International journal of circuits, systems and signal processing, vol. 10, no. 2016, pp. 403–413, 2016.

[21] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). IEEE, 2016, pp. 1–6.

[22] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai et al., "Recent advances in convolutional neural networks," Pattern Recognition, vol. 77, pp. 354–377, 2018.

[23] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data," PloS one, vol. 14, no. 2, p. e0212320, 2019.

[24] R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua, "Using deep learning neural networks and candlestick chart representation to predict stock market," arXiv preprint arXiv:1903.12258, 2019.

[25] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," Expert Systems with Applications, vol. 38, no. 8, pp. 10 389–10 397, 2011

------------------------------***-------------------------