Stock Market Prediction using Recurrent Neural Network's LSTM Architecture

Koushik Sutradhar
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
koushik.sutradhar@northsouth.edu

Sourav Sutradhar
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
sourav.sutradhar@northsouth.edu

Iqbal Ahmed Jhimel
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
iqbal.jhimel@northsouth.edu

Suneet Kumar Gupta
Bennett University
Gr. Nodia, India
suneet.bandha@gmail.com

Mohammad Monirujjaman Khan
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
monirujjaman.khan@northsouth.edu

***Abstract*—Stock market price prediction is a difficult undertaking that generally requires a lot of human-computer interaction. The stock market process is fraught with risk and is influenced by a variety of factors. Of all the market sectors, it is one of the most volatile and active. When buying and selling stocks from various corporations and businesses, more caution is required. As a result, stock market forecasting is an important endeavor in business and finance. This study analyzes one of the explicit forecasting tactics based on Machine Learning architectures and predictive algorithms and gives an independent model-based strategy for predicting stock prices. The predictor model is based on the Recurrent Neural Networks' LSTM (Long Short-Term Memory) architecture, which specializes in time series data classification and prediction. This model does rigorous mathematical analysis and estimates RMSE to improve forecast accuracy (Root Mean Square Error).All calculations and performance checks are done in Python 3. A number of machine learning libraries are used for prediction and visualization. This study demonstrates that stock performance, sentiment, and social data are all closely related to recent historical data, and it establishes a framework and predicts trading pattern linkages that are suited for High Frequency Stock Trading based on preset parameters using Machine Learning.**

Keywords: Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Root Mean Square Error (RMSE), Python 3.

I.INTRODUCTION

Short-term forecasting, medium-term forecasting, and long-term forecasting are the three types of stock price forecasting. Forecasting for a few seconds, minutes, days, weeks, or months is referred to as short-term forecasting. Forecasting for one or two years is referred to as medium-term predicting, while forecasting for more than two years is referred to as long-term forecasting. Because various factors based on world events influence stock market fluctuation, predicting the short-term value of a stock is significantly easier than predicting the long-term value of a stock. Fundamental analysis, technical analysis, and time series analysis are the three types of analysis that can be used to make predictions [1]. Fundamental analysis is a sort of investment study in which a company's share value is calculated by examining its sales, earnings, profitability, and other economic aspects. This strategy is best for forecasting over a long period of time. Sentiment analysis processes are used to do fundamental analysis on social media data. For technical analysis, the past price of stocks is used to predict the future price. A moving average is a popular technical analysis algorithm. It is the unweighted mean of the previous 'n' number of data points. This approach works well for making short-term predictions. Machine learning algorithms are used to do technical analysis on historical stock price data. Technical analysis includes gathering news, social media data, and extracting individual sentiments. The relationship between sentiments and stock prices is next examined. Linear and non-linear models are the two types of methods used in time series analysis. AR, ARMA, ARIMA, and their variants are linear models [2] [3] [4]. ARCH, GARCH, TAR, Neural networks, and Deep Learning Algorithms are examples of non-linear models [5]. Time series analysis is used in many forecasting methodologies. Time series data analysis aids in the discovery of patterns, trends, and periods or cycles in the data. A time series of observations for a given variable can be defined as a chronological succession of observations. The variable in this example is the stock price. Technical analysis and time series analysis are the emphasis of this research. Despite its widespread use, stock market forecasting remains a mysterious and empirical science. Few, if any, people are eager to share their effective tactics. One of the main goals of this paper is to put academic knowledge and understanding of stock market prediction into practice. Investors will be better able to avert another financial disaster if they have a better understanding of how the market works. The paper will conduct a rigorous scientific examination of certain existing tactics as well as a quantitative evaluation of novel strategies. Economists, policymakers, researchers, and market makers have become increasingly interested in forecasting markets in recent decades. The goal of this proposed project is to

2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) | 978-1-6654-0690-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/UEMCON53757.2021.9666562

978-1-6654-0690-1/21/$31.00 ©2021 IEEE

investigate and develop supervised learning systems for stock price prediction. It will assist in making a more accurate prediction based on the examination of the datasets and reports presented. The AI is trained to examine datasets of a certain stock market provided by an organization's reports using the Recurrence Neural Networking architecture's LSTM technique.

II. RELATED WORK

The use of market data analysis to anticipate stock prices is a common and interesting study area. As the amount of information on the Internet expands, some recent studies have attempted to investigate financial news in order to enhance prediction [6]. In the prediction processes, both stock prices and news articles were used. However, how to intelligently mix technical indicators from stock prices and news sentiment from textual news stories, and make the prediction model learn sequential information within time series, remains unanswered [7]. Many renowned researchers have contributed to this field. In 1988, the first attempt to use a neural network to model a financial time series was published [8]. For IBM, this project aimed to create a neural network model for deciphering nonlinear regularities in asset price movements. Despite the work's narrow scope, it aided in the establishment of evidence against EMH [9]. For the modeling of multivariate financial time series, deep learning architectures have been used. Back propagation and RNN models were utilized for stock index prediction in five distinct stock markets in 1996 [10]. For the prediction of the Shanghai stock market, a neural network model based on technical analysis factors has been constructed [11]. For daily stock prediction, time delay, recurrent, and probabilistic neural network models were presented in 1998 [12]. For the forecast of the S&P 500 stock market, machine learning techniques such as PSO and LS-SVM have been utilized [13]. The use of genetic algorithms in conjunction with neural network models has been described [14]. [15] described the use of the wavelet transform for prediction. The wavelet transform was utilized to characterize short-term patterns in stock trends in this study. According to a paper, a large number of startups and IT businesses are working on and building AI-powered trading bots [16]. Trading bots are being researched and developed by firms such as Sentient Technologies, Clone Algo Inc, Binatix, Alpaca, Aidyia, Walnut Algorithms, and many others. Some corporations have already constructed billions of artificial intelligence traders and are now attempting to divide each trader into its own company. The majority of the funding for these AI-based apps comes from AI companies, totaling around $135 million USD. To anticipate the stock market, Chinese researchers used two different types of models: a deep long short-term memory neural network (LSTM) with embedded layers and a long short-term memory neural network with automatic encoder. They employed the embedded layer and automatic encoder, respectively, to vectorize the data in these two models in order to forecast the stock using a long short-term memory neural network. Their findings suggest that the deep LSTM with embedded layers outperforms the shallow LSTM. For the Shanghai A-shares composite index, the accuracy of two models is 57.2 and 56.9 percent, respectively. Furthermore, for individual equities, they are 52.4 and 52.5 percent, respectively [17]. They also displayed research contributions in IMMT for financial analysis using neural networks. Many researchers have employed various deep learning architectures, such as Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN), to construct models that have demonstrated promising results and accuracy in predicting stock prices [18] [19]. The CNN was also used to forecast stock prices from the limit order book [20]. For stock market prediction, researchers have used the Generative Adversarial Network (GAN) architecture [21]. GAN is a new framework that uses a zero-sum game to train two models [22]. For image processing, GAN is employed, and for text processing, RNN is used. In addition, there are other examples of hybrid techniques. The use of a wavelet transformation combined with an Artificial Neural Network (ANN) showed that useful features should be extracted for ANN training [23]. Another strategy combines a long short-term memory (LSTM) network with a genetic algorithm (GA). The temporal window size and architectural variables of the LSTM network are frequently estimated through trial and error based on heuristics [24]. To achieve stock prediction, news and events in the financial sector were extracted and represented as dense vectors [25]. A 2D array was created by converting the number of orders and the price of 10 bid/ask orders by a few researchers [26]. [27] A study of the interdependence between stock price and stock volume for 29 selected NIFTY 50 firms was conducted. It focuses on the use of deep learning algorithms to predict stock prices [28] [29].

III. Methodology

A. Theoretical Overview

RNNs (Recurrent Neural Network) have proven to be one of the most powerful models for processing sequential data and LSTM (Long Short-Term Memory) is one of the successful RNN architectures. In the buried layer of the network, LSTM substitutes typical artificial neurons with memory cells as a unit of computation. Networks may successfully connect memories and input remotely in time using these memory cells. As a result, dynamically grasping data throughout time with strong prediction capacity is a good fit. RNNs process arbitrary sequences of input data using their internal memory. Each computing unit in an RNN has a real-valued application that changes over time and a weight that may be changed. RNNs are created by applying the same set of weights recursively over a graph-like structure. Many of the RNNs use equation (1) to define the values of their hidden units.

$$h^t = f(h^{t-1}, x^t; \theta) \quad (1)$$

The learned model in RNN always has the same input size as it is specified in terms of transition from one state to another. RNN also uses the same transition function with the same parameters at every time step.

LSTM was introduced in 1997 by Hochreiter and Schmidhuber [30]. The hidden layers in RNN are replaced with LSTM cells in the case of LSTM architecture. LSTM cells are composed of various gates controlling the input flow. A single LSTM cell consists of input gate, cell state, forget gate, output gate, sigmoid layer, tanh layer and point wise multiplication operation. Input gate consists of the input, cell state runs through the entire network and it has the ability to add or remove information with the help of gates. Forget gate decides the fraction of the information to be allowed and the output gate consists of the output generated by the LSTM. The Sigmoid layer generates numbers between zero and one, describing how much of each component should be let through and the tanh layer generates a new vector, which will be added to the state. The cell state is updated based on the outputs from the gates. Mathematically we can represent it using the following equations [31]:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$c_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) (4)$$

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o) (5)$$

$$h_t = o_t * tanh(c_t) \quad (6)$$

Here, x_t is input vector, h_t is output vector, c_t is cell state vector, f_t is forget gate vector, i_t is input gate vector, o_t is output gate vector and W,b are the parameter matrix and vector.

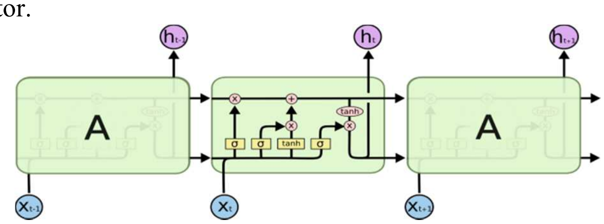

Figure 1: Architecture of LSTM
Image Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long-term dependency problems are specifically avoided with LSTMs. It is basically their default behavior to remember information for a longer amount of time. Unlike the more basic model of an RNN, it is not difficult for them to grasp.

Regression analysis is a process to find correlations and model the relation between one or more dependent or independent variables. Out of three different forms of regression, the simplest form is Linear Regression [32]. Linear regression states that there is a linear relationship between the variables, meaning that the relationship can be summarized in a formula:

$$y = \alpha + \beta x \quad (7)$$

To represent more complex relations, a simple regression model:

$$y_i = \alpha + \beta x_i + \epsilon_i (8)$$

Where α and β represents the parameters, xi the independent variable and ϵ_i the error term.

If the simple linear model does not suit for finding the correlations then the linear regression model can be expanded further by adding additional variables to the model by assuming a parabola function where $\beta_0 - \beta_2$ represents the parameters:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i (9)$$

The error is calculated by calculating the difference between the true value and the predicted value, where y_i is the actual value and $\widehat{y_\iota}$ is the predicted value.

$$\epsilon_i = y_i - \widehat{y_\iota} \ (10)$$

B. Data Preprocessing

Real-world data generally contains noises, missing values, unusable formats which cannot be directly used for machine learning models. Data preprocessing is required for cleaning the data and making it suitable for a machine learning model thus increasing the accuracy and efficiency of a machine learning model. The data preprocessing stage involves data discretization, data transformation, data cleaning, data integration. After the dataset is transformed into a clean dataset, the dataset is divided into training and testing sets to evaluate. Feature scaling is conducted by importing MinMaxScaler from the Scikit-Learn library. MinMaxScaler transforms features by scaling each of them to set range and here the range is 0 to 1. A data structure is created after feature scaling with 60 timestamps and 1 output meaning the datas from day 1 to day 60 is taken and made predictions on the 61st day followed up by taking data from day 2 to day 61 and predicting on day 62. The datasets which are being used consist of date wise stock prices for NASDAQ listed companies for various time periods. This historic data is retrieved from Yahoo Finance via yfinance python package. It contains prices for up to 01 of April 2020. These datasets have trading date, opening price, closing price, maximum-minimum prices and adjusted closing price. For this paper, two different companies from the IT Sector were selected for study. The companies are Google and Facebook. The data for these two companies were extracted from the available data and were subjected to preprocessing to obtain the stock price. The datasets were splitted for training and testing purposes. The train data consists of the stock price of Google for the time period of August-19-2004 to December-31-2019 and May-05-2012 to December-31-2019 of Facebook. The test data consists of the stock price of both companies for the time period of January-02-2020 to April-01-2020.

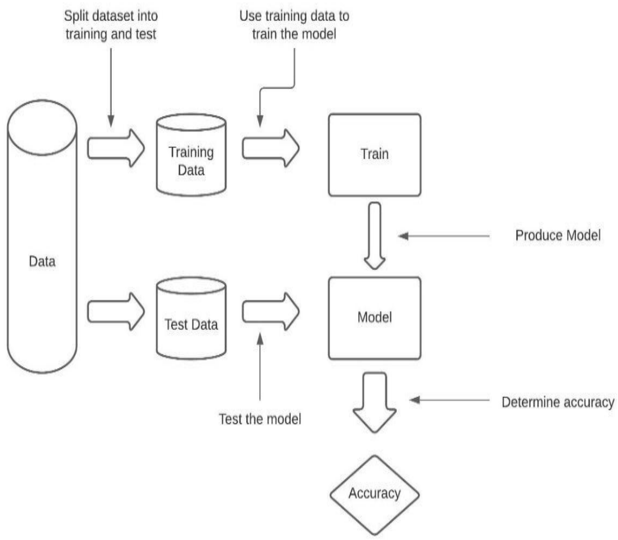

Fig.2.Train-Test Diagram

C. Building and Training the RNN

Keras is a high-level API for constructing and training deep learning models in TensorFlow. The first of the Keras libraries is Sequential, which is essentially a linear stack of layers through which a sequential model is constructed by passing a list through it. The dense layer is a typical deeply connected neural network layer that is used to adjust the dimensions of the output vector. The dense layer represents a matrix vector multiplication, thus the values in the matrices that are trainable parameters are updated during back propagation. The LSTM layer is the third, while the dropout layer is the fourth. Three sequential input layers, three LSTM layers, a dense layer with activation, and a dense output layer with the linear activation function make up the LSTM layer. Dropout is a regularization approach for decreasing overfitting in neural networks, essentially dropping out units. Dropouts are employed to strengthen the neurons, allowing them to forecast trends without focusing on a single neuron. The regression model is commonly used to solve time series problems. The initial stage in creating a regression deep learning model is to read in sequential data and allocate it to the model. Date, High, Low, Close, and Volume features are extracted and supplied into the neural network, which is then trained for prediction with random biases and weights.

D. Optimization and Regularization

Compiling the RNN from the scarce model requires Optimization. The type of optimizer used can greatly affect how fast the algorithm converges to the minimum value and it is also very important that there is some notion of randomness to avoid getting stuck in a local minimum and not reaching the global minimum Here, Adam (Adaptive Movement Estimation) optimizer is being used. The Adam optimizer combines the perks of two other optimizers: ADAgrad (Adaptive Gradient Algorithm) and RMSprop (Root Mean Squared Propagation). The ADAgrad optimizer essentially uses a different learning rate for every parameter and for every step. The reason behind it is that the parameters that are infrequent must have larger learning rates while the parameters that are frequent must have smaller learning rates. The learning rate is calculated based on the past gradients that have been computed for each parameter. The RMSprop considers fixing the diminishing learning rate by only using a certain number of previous gradients. Adam computes the adaptive learning rates for each parameter based on its past gradients by considering the exponentially decaying average of the past square gradients and exponential decaying average of past gradients. When it was first used researchers observed that there was an inherent bias towards zero and they countered it by using two estimation process which leads to the gradient rule:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \qquad (11)$$

Here, the $\hat{v}$and the $\hat{m}$ are considered as the estimates of the first and second moment of the gradients respectively hence getting the name Adaptive Movement Estimation (Adam) and because of it, the learning rate is different for every parameter and every iteration and does not diminish with the ADAgrad. The gradient update uses moments of the distribution of weights allowing for more statistically sound descent. Another important aspect of training the model is making sure the weights do not get too large and start focusing on one data point, hence overfitting. That is why regularization which can also be thought of as minimization. The fact that the function space is in the reproducing Kernel Hilbert space and shows that the notion of normalization exists. It allows encoding the notion of the normalization into the regularizations.

E. Output Generation

Compiling and fitting the RNN into training sets with epoch= 100 and batch size = 64 and 256 respectively for Model-01 and Model-02. Epoch indicates the number of passes through the entire dataset the machine learning algorithm has completed and batch size refers to the number of training examples utilized in a single iteration. The output value generated by the output layer of the RNN is compared with the target value. The error or the difference between the target and the obtained output value is minimized by using a back propagation algorithm which adjusts the weights and the biases of the network.

F. Evaluation

The RMSE (Root Mean Square Error) is determined when the projected output is obtained. The root mean square error (RMSE) is a typical means of calculating a model's error in predicting quantitative data. Because RMSE is scale-dependent, it should only be used to evaluate prediction errors of different models or model configurations for a single variable, not between variables. It refers to the degree to which a regression line matches the data points. A straight line that passes through all the points which fits the data points the best way. This line contains the predicted points. The general formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \quad (Predicted_i - Actual_i)^{\ 2}}{N}} \quad (12)$$

Here,

$Predicted_i =$ Predicted value for the ith observation.

$Actual_i =$ Observed value for the ith observation.

N = Total number of observations.

G. Prediction and Visualization

When determining the long-term stability of a time series model, a rolling analysis is commonly used. One key assumption when using a statistical model to evaluate financial time series data is that the model's parameters remain constant throughout time. To get the true stock price of 2020, use the iloc function to select columns and rows by number in the order they occur in the dataframe. Iloc allows for selection based on row numbers ranging from 0 to the whole number of rows. The training and test sets must be combined on the zeroth axis to obtain the expected stock price for 2020.

The timestamp must be reset to 60 after rearranging the data. To extract the expected stock price from the matrix form, it must be inversely converted and assigned to a new dataframe. Finally, the Matplotlib software is used to show the expected and actual stock prices.

IV. RESULTS AND DISCUSSION

All the results are obtained from the models with loss function of mean squared error and the Adam optimizer. There are several LSTM architectures which can be built using the LSTM structure or the combination of LSTM or RNN [33]. In this paper, several LSTM architectures are tested to find the model with the lowest loss value. Losses observed for the best LSTM model after 100 epochs are 0.0011 (Model-01) and 3.3428e-04 (Model-02). The RMSE for Model-01 is 4.77 (Stock price range is 150 to 220 Dollar) and 11.695 for Model-02 (Stock price range is 1100 to 1500 Dollar), which indicates the models' good performance and accuracy compared to the price unit ranges.

Fig.3.Plot for Real Value vs Predicted Value using LSTM architecture with the lowest loss value (Model-01).

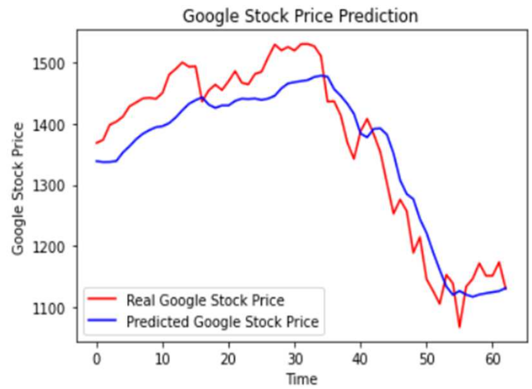

Fig.4. Plot for Real value vs Predicted value using LSTM architecture with the highest loss value (Model-02).

In order to obtain the best result, the behavior of the LSTM architecture is also checked by varying the inner architecture including the number of cells, number of layers, keeping constant activation of the hidden layer as hyperbolic tangent function and the output layer as the rectified linear unit function.

After predicting and understanding the behavior of the stock market, the next problem arises on the trading strategy. One should have the prior knowledge about how much proportion of one's share could be distributed in different stocks in one's portfolio. It is a very important factor to understand the market before investing and adjusting the proportion of wealth in different stocks to gain profit from the investments. In this scenario, Markowitz's portfolio optimization [34] provides satisfactory results to distribute shares in different markets. By performing Markowitz Portfolio Optimization technique, one can understand how much one should invest in a market. Return on investment can be calculated using the following formula:

r = (adjClose(i) - adjClose(i-1)) / adjClose(i) (13)

Annual return can be calculated by multiplying 12 to return. The variance of return, the covariance matrix of return, weights and sum should be 1.

*Expected return = transpose of weight * return* (14)

$$Volatility = (transpose\ of\ weights\ * \ covariance\ of\ return\ * weights)^{1/2} (15)$$

Sharpe ratio = expected return / volatility (16)

Adjusting the weight to gain profit and selecting the portfolio with max Sharpe ratio and volatility.

Table 1: Comparison with other papers

No	Name	Architecture/Algorithm	Trading and Investment Strategy
1	This Paper	LSTM, RNN	Yes
2	Ref [1]	MLP	No
3	Ref [6]	Heterogeneous Information Fusion	No
4	Ref [10]	Backpropagation Network	No
5	Ref [11]	Conjugate Gradual Learning	No
6	Ref [13]	PSO and LS-SVM	No
7	Ref [17]	DLSTM	No
8	Ref [18]	CNN	No
9	Ref [33]	DLSTM, LSTMP	Yes
10	Ref [31]	LSTM, CNN, RNN	No

Table 1 shows the comparison between other papers and this paper. This paper mainly focuses on LSTM architecture. Other papers have shown various approaches for stock market prediction using different predictive algorithms and techniques but very few of them have used LSTM architecture. Besides, this paper briefly discusses the trading strategy using Markowithz's Portfolio Optimization, where most of the papers only focused on prediction strategy, which makes this paper unique.

V. FUTURE IMPACT

Artificial intelligence and machine learning have come a long way in the twenty-first century. The stock market prediction's future conclusion appears to be highly favorable. One of the main concerns is improving prediction accuracy. Because it must go through multiple difficult prediction algorithmic approaches and a significant amount of time in training and testing, this research does not focus on the absolute correctness of our predictor models. Because stock markets are significantly influenced by a variety of events, including government policies, politics, natural disasters, exchange and interest rates, inflation, and so on, precisely anticipating market moves is nearly difficult. As a result, studying past data and looking for trends are effective methods for improving and ensuring the accuracy of predictions. The models are currently trained and tested on a TensorFlow backend using different machine learning libraries, with the intention of implementing the K-fold cross-validation method in the future. It's a popular strategy since it's easy to grasp and produces a less skewed or optimistic estimate of model competence than other approaches, such as a simple train/test split. The evaluation of the models will be more accurate in terms of market movements and stock prices if this approach is used. Its installation will boost the algorithm's accuracy, and its analysis will hopefully yield the best outcomes based on historical data, market patterns, and market scenarios. Web-based apps and cross-platform mobile applications with capabilities like online trading, stock buying and selling, and a digital wallet can also be used to deploy developed prediction models. Users can acquire real-time information regarding market movements by connecting the application to international markets such as New York, Tokyo, and London, and studying the EOD of those markets. Artificial intelligence can be used to create a trading bot based on the predictive model. This trading bot will not only generate predictions, but will also be able to trade based on market and data analysis. The dominance of the hedge fund sector will arise from the successful construction of a trading bot and the training of its trading talent to an advanced degree.

VI.CONCLUSION

The popularity of stock market trading is rapidly increasing, prompting experts to develop new methods for forecasting utilizing new techniques. The forecasting technique is beneficial not just to researchers, but also to investors and anyone involved in the stock market. A high-accuracy forecasting algorithm is necessary to accurately anticipate stock values. This paper focuses on one of the most precise forecasting technologies, which employs RNNs and LSTM units to provide investors, analysts, and anybody interested in investing in the stock market with a good understanding of the stock market's future position. Predicting future stock price swings will set a new benchmark. Companies are already employing AI and machine learning to study human behavior for a variety of purposes. Some want to tailor advertising to the most vulnerable consumers, while others want to utilize it to hire the best personnel to help their company succeed. This similar technology is being used by many financial firms to forecast future stock patterns. Predictions made possible by AI make it easier for users and other investors, such as venture capitalists, to understand their investments on time. Not only that, but it also aids in raising one's public profile and giving clients and suppliers with assurance. This paper shows the potential of one of the most precise machine learning-based stock market prediction models.

Acknowledgment: Authors would like to thank the Department of Electrical and Computer Engineering of North South University

REFERENCES

[1] Devadoss AV, Ligori TA. Forecasting of stock prices using multi layer perceptron. International journal of computing algorithm. 2013 Dec;2(1):440-9.

[2] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," International journal of forecasting, vol. 22, no. 3, pp. 443–473,2006.

[3] V. K. Menon, N. C. Vasireddy, S. A. Jami, V. T. N. Pedamallu,V. Suresh Kumar, and K. Soman, "Bulk price forecasting using spark over nse data set," in International Conference on Data Mining and Big Data. Springer, 2016, pp. 137–146.

[4] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: forecasting and control. John Wiley & Sons, 2015.

[5] G. Batres-Estrada, "Deep learning for multivariate financial time series," ser. Technical Report, Stockholm, May 2015.

[6] X. Zhang, Y. Zhang, S. Wang, Y. Yao, B. Fang, and S. Y. Philip, "Im-proving stock market prediction via heterogeneous information fusion,"Knowledge-Based Systems, vol. 143, pp. 236–247, 2018.

[7] X. Li, P. Wu, and W. Wang, "Incorporating stock prices and news sent-timents for stock market prediction: A case of Hong Kong,"InformationProcessing& Management, vol. 57, no. 5, p. 102212, 2020.

[8] H. White, Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns, ser. Discussion paper- Department of Economics University of California San Diego. Department of Economics, University of California, 1988.

[9] B. G. Malkiel, "Efficient market hypothesis," The New Palgrave: Finance. Norton, New York, pp. 127–134, 1989.

[10] J. Roman and A. Jameel, "Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns," in System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on, vol. 2. IEEE, 1996, pp. 454–460.

[11] M.-C. Chan, C.-C. Wong, and C.-C. Lam, "Financial time series forecasting by neural network using conjugate gradient learning algorithm and multiple linear regression weight initialization," in Computing in Economics and Finance, vol. 61, 2000.

[12] E. W. Saad, D. V. Prokhorov, and D. C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," IEEE Transactions on neural networks, vol. 9, no. 6, pp. 1456–1470, 1998.

[13] O. Hegazy, O. S. Soliman, and M. A. Salam, "A machine learning model for stock market prediction," arXiv preprint arXiv:1402.7351, 2014.

[14] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," Expert systems with Applications, vol. 19, no. 2, pp. 125–132, 2000.

[15] Y. Kishikawa and S. Tokinaga, "Prediction of stock trends by using the wavelet transform and the multi-stage fuzzy inference system optimized by the ga," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 83, no. 2, pp. 357–366, 2000.

[16] A. Srivastava and I. Sengupta, "Predicting stock market: An approach with artificial intelligence,"Management Insight, vol. 13, no. 2, pp. 73–77, 2017.

[17] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction,"The Journal ofSupercomputing, vol. 76, no. 3, pp. 2098–2118, 2020.

[18] E. Hoseinzade and S. Haratizadeh, "Cenapred: Cnn-based stock market prediction using a diverse set of variables,"Expert Systems with Applications,vol. 129, pp. 273–285, 2019.

[19] M. Hiransha, E. A. Gopalakrishnan, V. K. Menon, and K. Soman, "Nse Stock market prediction using deep-learning models,"Procedia computer science, vol. 132, pp. 1351–1362, 2018.

[20] A. Tsantekidou, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, andA. Iosifidis, "Forecasting stock prices from the limit order book using con-volutional neural networks," in2017 IEEE 19th Conference on BusinessInformatics (CBI), vol. 1. IEEE, 2017, pp. 7–12.

[21] K. Zhang, G. Zhong, J. Dong, S. Wang, and Y. Wang, "Stock market prediction based on generative adversarial network,"Procedia computer science, vol. 147, pp. 400–406, 2019.

[22] M. D. Kamrul Hasan, Sakil Ahmed, Z. M. Ekram Abdullah, Mohammad Monirujjaman Khan, Mehedi Masud et al., , "Deep Learning Approaches for Detecting Pneumonia in COVID-19 Patients by Analyzing Chest X-Ray Images," *Mathematical Problems in Engineering, Hindawi, Volume 2021, Article ID 9929274, PP. 1-8, https://doi.org/10.1155/2021/9929274*, 2021

[23] Fazle Rabby Khan, Md. Muhabullah, Roksana Islam, Mohammad Monirujjaman Khan et al., , "A Cost-Efficient Autonomous Air Defense System for National Security," *Security and Communication Networks, Hindawi, Impact Factor 1.791, Scopus Indexed. vo. 2021, no. 9984453,, pp. 1-10. https://doi.org/10.1155/2021/9984453*, 2021

[24] Morshedul Bari Antor, A. H. M. Shafayet Jamil, Maliha Mamtaz, Mohammad Monirujjaman Khan et al.,, "A Comparative Analysis of Machine Learning Algorithms to Predict Alzheimer's Disease," *Journal of Healthcare Engineering,Hindawi, vo. 2021, no. 9917919, pp. 1-12. https://doi.org/10.1155/2021/9917919*, 2021

[25] Sumaiya Tarannum Noor, SyedaTasmiah Asad, Mohammad Monirujjaman Khan et al., , "Predicting the Risk of Depression Based on ECG Using RNN," *Computational Intelligence and Neuroscience, (*Article ID 1299870. vol. 2021, pp. 1-12. https://doi.org/10.1155/2021/1299870, 2021

[26] Sadman Bin Islam , Mohammad Mahabubul Hasan, Mohammad Monirujjaman Khan **,** "Prediction of Stock Market Using Recurrent Neural Network," *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEEE IEMCON*