Stock Market Feature Analysis using Deep Learning

Mridha Md. Nafis Fuad, Ishita Sur Apan, Laila Afrose Labonno, B M Mainul Hossain
Institute of Information Technology
University of Dhaka
Dhaka, 1000, Bangladesh
Email: bsse0920@iit.du.ac.bd, bsse0922@iit.du.ac.bd, bsse0911@iit.du.ac.bd, mainul@iit.du.ac.bd

***Abstract*—The stock market groups and movements of third world countries are affected by several economic factors which include political events, general economic conditions, commodity price index, investors' expectations, movements of other stock markets and, the psychology of investors. The abnormality of daily stock data is more frequent in underdeveloped and developing countries. In this study, we focus on the selection of a generic set of simple and derived features from stock data in order to better understand the trend of ups and downs in the companies listed in the Dhaka Stock Exchange(DSE). The analysis of the features includes the selection of appropriate indicators to best describe the market and also simplicity in generalizing the behavior of the markets from historical data. The comparison of the results, generated by combinations of features to understand the sequential data are described in this study. OHLC, which is the average of the trade values of each day outputs the best result in predicting the market and also represents the change of market based on historic data. Among the models under study, GRU recurrent model gives the best result in terms of RMSE metric and also through manual verification of results obtained from the test data set and up-to-date data scraped from the DSE website. The final model generated fits about 96% of the companies among which 79% of the companies show a very good fit in the trend pattern. We analyze the models based on manual verification and simplicity of the features to fit the market trend.**

***Keywords*—** Dhaka Stock Exchange(DSE), feature analysis, deep learning, stock trend

I. Introduction

The stock market is a place for investors to connect for buying and selling investments — most commonly, stocks, which are shares of ownership in a public company. It is one of the most important financial institutions where people invest and contribute to the economy of a country. A total of 750 listed securities is present on the Dhaka Stock Exchange. Investment options for an investor in this market are ordinary shares, Debenture, Bond, and Mutual funds. According to the website statistics, the current Market capitalization of DSE is US$ 46.2 billion (2020).[1] But the Dhaka stock market is witnessing a colossal shock from its dawn. Keeping in mind that in the last two decades, Bangladesh faced two major shocks in the stock market in 1996 and in 2010. This has lead to a state of mistrust among the public [1]. The Dhaka stock exchange experiences a random trend in the market patterns at regular intervals of times [2, 3].

Currently, DSE has three indices — i) DSE Broad Index of the Exchange (Benchmark Index) reflects around 97% of the total equity market capitalization, ii) DS30 is constructed with 30 leading companies which can be said as the investable Index of the Exchange."DS30" reflects around 51% of the total equity market capitalization, iii) DSE Shariah Index (DES) serves as the Shariah-compliant free float-adjusted broad market benchmark. The index is a subset of the DSE Broad Market Index (DSEX) and includes all stocks that pass rules-based screens for Shariah compliance. Analysis of the stock market enables investors to identify the intrinsic worth of security even before investing in it. Appropriate stock market predictions of each of the listed companies can help the investors to understand the market trend and conduct the research before making an investment decision. In our work, we propose a deep learning-based model to predict the market change for each of the listed companies. The models would help the general user to make decisions to trade stocks of a selected company [4]. They will predict the prices of stocks on the next day and compare them with the previous day. It will indicate if a user needs to invest more in this stock or sell the stock. Thus, this prediction will support stock market ups and downs. The validation of the models is quantified by comparing the predicted ups and downs with the actual ups and downs in the market. For this, it will store data of all companies, so that it is comparable to the prediction and make a decision about how much they can trust the system in real-life scenarios, that is, actual live testing stock data. The prediction will proceed sequentially for everyday stock data.

In recent studies regarding stock data, using recurrent deep learning models has been the gold standard [5–8]. Stock price daily data is a good fit to apply sequential time series manipulation [9], thus the popularity of recurrent neural networks. Regardless of the model used, the domain of variability in the stock market is very volatile and changes due to a lot of unpredictable parameters [10]. In literature, researchers try to estimate the future behavior of a specific company based on their previous trends [11–13]. To minimize the effect of such volatility, researchers try to tune the set of features and analytical variables to form the prediction model focusing on a single company. On the other hand, another popular approach is to test combinations of features for a generic algorithm for all the companies listed in a market [14, 15]. In our approach, we focus on the latter and explore the set of input features to best understand the trend in the Dhaka stock market. Our efforts are focused solely on fitting the market trend with a set of derived and basic stock data features for all the listed companies in DSE.

II. Background Study

A. Feature Selection

The basic features are obtained from the DSE website. The features are date, close (closing price on a trading day), yesterday's closing Price (opening price of a trading day, YCP), high (highest trading price on a trading day), low (lowest trading price on a trading day), The derived features are stated by the given equations:

$$OHLC = (Open + High + Low + Close)/4. \quad (1)$$

$$HLC = (High + Low + Close)/3. \quad (2)$$

$$\%Change = YCP - CP/YCP. \quad (3)$$

Investors use long-term data patterns to understand the rise and fall of stocks. These time-series features are also used intensively in literature extracted from the stock data [16]. The extracted features are listed as follows:

[1] www.dsebd.org

1) **Simple Moving Average:** A simple moving average (SMA) is an arithmetic moving average calculated by adding recent prices and then dividing that figure by the number of time periods in the calculation average.
2) **Exponential Moving Avg:** The EMA is a moving average that places a greater weight and significance on the most recent data points.

$$EMA(n) = (CP * k) + (EMA(n-1) * (1-k)) \quad (4)$$

$$k = smoothing/(1+n) \quad (5)$$

In eq. (4) the value of k is the adjustment factor of weight to tune the priority of the data of recent days, where k is calculated commonly using *smoothing = 2*.

3) **Moving Avg with Convergence Divergence:** The MACD interprets the current market change based on the EMA obtained from a larger historic window.

$$MACD = EMA(short) - EMA(long) \quad (6)$$

In literature, the above mentioned simple and derived features work well with the deep learning model for those stock markets. After obtaining the list of features we now move on to the deep learning model architecture formulation to see which model and set of features work best to understand the trend in DSE. Based on our background studies, predicting some value for time series data recurrent neural networks for deep learning algorithms should be used. But the vanilla recurrent neural network fails to train over a long sequence of data which is known as the vanishing gradient problem [17]. The proposed solution to handle this problem is to use variants of the RNN. Most of the existing deep learning literature on stock prediction is based on using classical LSTM models. As stock data is absolutely time-sensitive and highly dependent on previous days' value, the LSTM algorithm is chosen.

B. Long Short Term Memory

Long Short Term Memory networks are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997) [18] and were refined and popularized by many people. They work very well where the data in sequential that is, a series is observed in the nature of the data set. They are now widely used in time series data. The primary goal of the LSTM architecture is to solve the problems faced by RNN where it fails to learn over longer sequence of data. Remembering information for long periods of time is practically their default behavior. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The output of one cell passes forward to the next through a belt like structure. The back-passing algorithms working by adding the gradients rather than shrinking as it moves further down to the earlier weights. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.The hidden state h_t for an LSTM cell can be calculated as follows:

$$i_t = \sigma(x_t U^i + h_(t-1) W^i) \quad (7)$$

$$f_t = \sigma(x_t U^f + h_(t-1) W^f) \quad (8)$$

$$o_t = \sigma(x_t U^o + h_(t-1) W^o) \quad (9)$$

$$\tilde{C_t} = \tanh(x_t U^g + h_(t-1) W^g) \quad (10)$$

$$C_t = \sigma(f_t * C_(t-1g + i_t * \tilde{C_t}) \quad (11)$$

$$h_t = \tanh(C_t) * o_t \quad (12)$$

Here, *i*, *f*, *o* are called the input, forget and output gates, respectively.The equations of the gates are almost similar in nature, just containing different sets of parameters (*W* is the recurrent connection at the previously hidden layer and current hidden layer, *U* is the weight matrix connecting the inputs to the current hidden layer).

C. Gated Recurrent Unit

Gated recurrent units (GRUs) are another popular recurrent architecture to solve the problems of a vanilla RNN, introduced in 2014 by Kyunghyun [19] There is a close similarity between the architecture of a GRU cell and an LSTM cell. The GRU also has a forget gate and seems to be more complex than LSTM looking at its diagram. But it also lacks an output gate and has lesser parameters than an LSTM cell. GRU does not lack in performance to much extent to that of an LSTM but the training speed is considerably faster due to its fewer parameters [20, 21]. GRU's exhibit better performance on certain smaller and less frequent data sets [21, 22]. For such behaviour of GRU architecture, this model was also considered alongside LSTM for comparison to fit the trend with minimal set of features. For the GRU the hidden state h_t can be calculated as follows:

$$z_t = \sigma(x_t U^z + h_(t-1) W^z) \quad (13)$$

$$r_t = \sigma(x_t U^r + h_(t-1) W^r) \quad (14)$$

$$\tilde{h_t} = \tanh(x_t U^h + (r_t * h_(t-1)) * W^h) \quad (15)$$

$$h_t = (1-z_t) * h_(t-1) + z_t * \tilde{h_t} \quad (16)$$

Here in case of GRU r is a reset gate, and z is an update gate.

III. Methodology

This section presents the process of extracting features and the selecting models. It also contains the process of manual verification of stock trend prediction and how to find out the best approach. Figure 1 shows the overall architectural working methodology for our system.

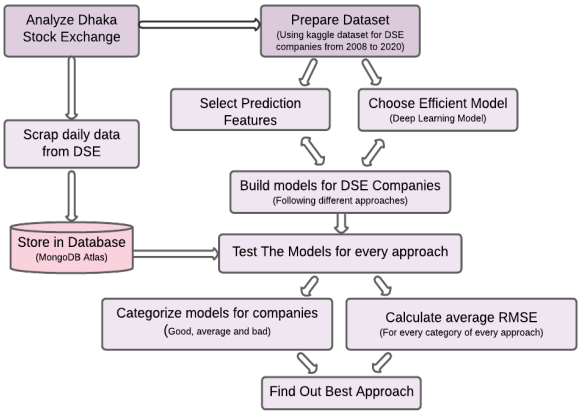

Figure 1. Overall Architectural Process

The data set of the daily stock trades obtained from the DSE website was used to recognize important features and build models. We examined the existing literature [23] to find out the usage of derived features in predicting the stock market with deep learning algorithms. The literature study mainly consisted of the Chinese stock market data set and the New York stock exchange [16].

A. Selecting Approach

From all the features in the data set, we needed to select the important features for building our prediction model. After the preliminary analysis, we ended up with the following features: date, trading_code, last_traded_price, high, low, opening_price, close, yesterdays_closing_price, trade value_mn, and volume. Then we discarded some of the features and derived some other features

according to the consultancy with our feature selection team. The derived features are OHLC, EMA(12), EMA(26), SMA(sample moving average), and MACD. After that, we built different prediction models for several feature sets.

1) **Approach 1:** Predicting the Close price of the next day with High, Low, Open, and Close prices of the previous day.
2) **Approach 2:** Predicting Close price of the next day with the OHLC(average of open, high, low, and close prices) value of the previous day.
3) **Approach 3:** Predicting Close price of the next day with the derived features EMA, SMA, and MACD prices of the previous day.
4) **Approach 4:** Predicting OHLC of next day with High, Low, Open, and Close prices of the previous day.
5) **Approach 5:** Predicting OHLC price of next day with OHLC price of the previous day.
6) **Approach 6:** Predicting OHLC price of next day with the derived features EMA, SMA, and MACD prices of the previous day.

From these six approaches, we built six models and analyzed those results via testing. First, we discarded three approaches(1, 2 and 6) as they were not showing expected results. So, we take three approaches(2, 3 and 5) for further analysis. Finally, the model with the feature selection approach 5, which predicted the OHLC price of the next day using the OHLC price of the previous day. The analysis process is discussed in detail in the later sections.

B. Model Verification

The generated data were preprocessed to drop companies having data of less than one month. We also dropped the companies that contain dummy data. The derived and time series features were computed for the corresponding days in the stock data. The data of each company was split into 75% training and 25% testing. Since the goal is to fit the same model for all the companies, we filter out data of each company and build the two LSTM and GRU models for each company using Keras(v2.3.1)[2]. Then for each of the model, we compute the model and use Matplotlib[3] to generate the graphs to verify the degree to which the model fits the market trend. Plotting the prediction of each day for both training set and testing set reveals the level of fit. The RMSE if the models were saved as a metric of the accuracy of fit. Trial and error revealed that scaling the feature between [0, 1] gave slightly better results. The graphs were manually grouped based on human verification of fitness of prediction with respect to the actual value. They were grouped as good, average and bag based on the gap between the actual value and predicted value and also the trend line of the graph. For each set of approaches, we finally select the set of features that produce the best result among the six approaches in respect to both fitness and simplicity.

Hyper parameters are tuned using trial and error methods. To reach out the optimal model parameters are tuned many many times. Setting an epoch of 50 and batch size 10, the optimum point was reached using Mean Square Error as the loss function.

IV. DATA

A publicly available data set from Kaggle[4] was used for generating the machine learning model. The data set contained data for each company from 2008 to 2020. After obtaining the available data, a web scraping tool was built based on the structured data. The data available from the source contained many companies with ambiguous data and static data to prevent blank values. The data set was cleaned labeling such companies as bad instances. Such companies were left out from building models. The cleaned data was used to train the models and for feature extraction. The web scrapper output producing a similar data structure was used in the data storage and daily scheduled to get daily data to be used for manual verification. Scrapping produces two collections of data, one for the company details and another for the daily stock market data for each company. The scraped data is stored in a cloud-based NoSQL MongoDB server, which is MongoDB Atlas[5]. NoSQL is selected for this project because of its performance in data fetching as chunks of data. The data contains daily share price information of around 600 companies.

It is very important to understand and analyze the data well before implementing any prediction model. As we were required to build a stock prediction model for the data from DSE through manual verification, three scrappers were designed to collect the daily data.

1) **CompanyList :** This spider fetches company list data (name, category, trading_code) from this url: https://www.dse.com.bd/company_listing.php
2) **Companies:** This spider fetches the specific company features like company list, category, sector, name, total_no_of_outstanding_securities from this url: https://www.dse.com.bd/displayCompany.php?name=[tradingcode]
3) **LatestShare:** This spider fetches the share data (latest share price like trading_code, last_traded_price, high, low, closing_price, yesterdays_closing_price, change, trade, value_mn, volume) of a specific date from the url: https://www.dsebd.org/day_end_archive.php?startDate=<YYYY-MM-DD>&endDate=<YYYY-MM-DD>&inst=All%20Instrument&archive=data

V. RESULT ANALYSIS

In this work, different models were built for every 405 companies of the Dhaka Stock Exchange. Every model is responsible for predicting some different values based on different features and using different algorithms. We figure out which model can find the data relationship more accurately from the time-series graph. This section aims for finding out the best model for DSE stock prediction so that the model performance is satisfactory, time duration for result generation is as least as possible and users get the appropriate insights from the results.

For analyzing the result of the models, firstly we generated company wise graphs to make the comparison between actual results and predicted results. To analyze the model performance, the output graph of every company was categorized in Good, Average and Bad for every model according to their prediction performance. After the categorization, we calculated the RMSE values for every category. Finally, the best models through analysis of RMSE values were determined.

When a graph of comparing actual result and predicted result for a company shows that the model can identify the stock trend accurately and the actual OHLC average value and predicted OHLC average value for test data is close enough, we consider the model for the corresponding company as a **Good** one. RMSE values for the good models lie below 20.

Again when we see from a graph of a company that the model can understand the stock trend for the corresponding company but the actual values and predicted values are not pretty close, we consider the model as an **Average** one. In the case of average models, the RMSE values are between 20 to 35.

Finally, we categorize a model as a **Bad** one when the graph of a company shows us that for some cases the model couldn't identify the stock trend for the corresponding company correctly. RMSE values for the Bad models are above 35.

Table I is the result for the models built with Long Short Memory and different sorts of features. The first row shows us the percentage of good, average and bad models when we predict OHLC average using the previous OHLC average. From the second row, we see

[2]https://keras.io/

[3]https://matplotlib.org

[4]https://www.kaggle.com/mahmudulhaque/dsebd

[5]https://www.mongodb.com/cloud/atlas

Table I
RESULTS FOR DIFFERENT FEATURE SETS

Prediction model: Long Short Term Memory			
Feature and Prediction	**Model Results**		
	Good	**Average**	**Bad**
OHLC -> OHLC	78.98%	16.96%	4.05%
EMA, SMA, MACD -> CP	41.65%	31.45%	26.8%
OHLC -> CP	22.59%	30.82%	46.58%

Table II
PREDICTING OHLC AVERAGE WITH PREVIOUS OHLC AVERAGE USING LSTM

Category	Average RMSE value
GOOD	14.83
AVERAGE	29.30
BAD	57.28

the model's result of predicting closing price(CP) using the derived features EMA, SMA, and MACD. Similarly, the third row of table I shows the result of predicting closing price(CP) using OHLC average as the feature.

From Table I, it is clear that predicting OHLC average from the previous data of OHLC average gives better graphs, hence better results for the companies of DSE.

Now to decide whether to predict OHLC average or Closing price, our study provokes to give importance to OHLC average over Closing price. Because estimating the next day price of a stock for a target company, knowing the closing price will give an insight of the last price of the stock at the end of the day. But the average of open, high, low, and close prices will give a better insight into the whole day's stock price.

Again, only one feature was used to train the model resulted in a faster run time in comparison with the others. In summary, after considering the three criteria- i) better performance in prediction, ii) least run time for result generation and iii) more useful insight for users- to predict next day, OHLC average from the previous OHLC average values delivered the best result.

Now that features are selected for the LSTM model and categorization of companies was conducted. We calculated the RMSE values for all companies and taken the average RMSE values for every category. Table II shows the average RMSE values.

Similarly, in the case of building prediction models with Gated Recurrent Unit(GRU), following the same approach and came to the conclusion that predicting OHLC average with the previous OHLC data served this work purpose the best. Table III shows the average RMSE values for the three categories.

The comparative result of the two tables shows that GRU works better than the LSTM model. the companies for which the RMSE were considered bad contained too much inconsistent data, for this in both models they show up with bad results.

Example of Good result: Company Code: EMERALDOIL

Using the LSTM model, a test RMSE of 4.36 for this company was obtained. The graph is shown in figure 2.

Table III
PREDICTING OHLC AVERAGE WITH PREVIOUS OHLC AVERAGE USING GRU

Category	Average RMSE value
GOOD	13.87
AVERAGE	26.15
BAD	55.61

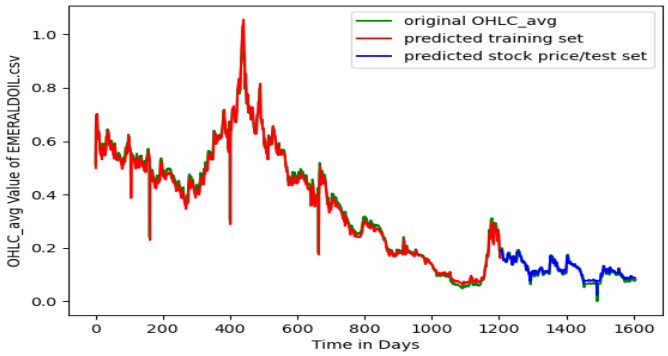

Figure 2. Good Prediction in LSTM

Using the GRU model got a testing RMSE of 4.33 for this company.The graph is shown in figure 3.

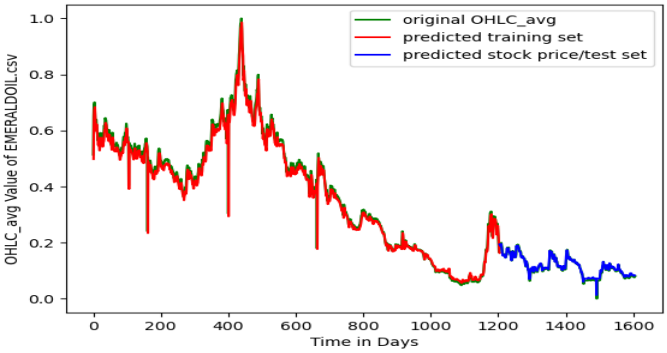

Figure 3. Good Prediction in GRU

Example of Average result: Company code: *UCBL*. Using the LSTM model got a testing RMSE of 20.77 for this company. The graph is shown in figure 4. Again the GRU model got a testing RMSE of 17.00 for this company.The graph is shown in figure 5.

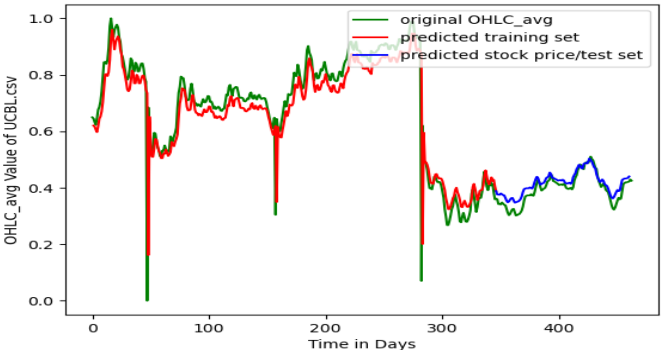

Figure 4. Average Prediction in LSTM

Example of Bad Result: Company code: *AOL*. Using the LSTM model, a testing RMSE of 56.32 for this company. The graph is shown in figure 6.

Using the GRU model, a testing RMSE of 56.20 for this company. The graph is shown in figure 7.

All the plot images of models using Gated Recurrent Unit model for companies in DSE after cleaning the data is publicly available on github.[6]

[6]https://github.com/IshitaApan/StockTrendAnalysis-DSE

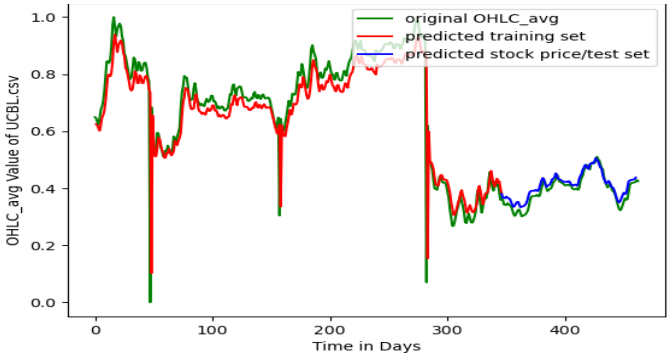

Figure 5. Average Prediction in GRU

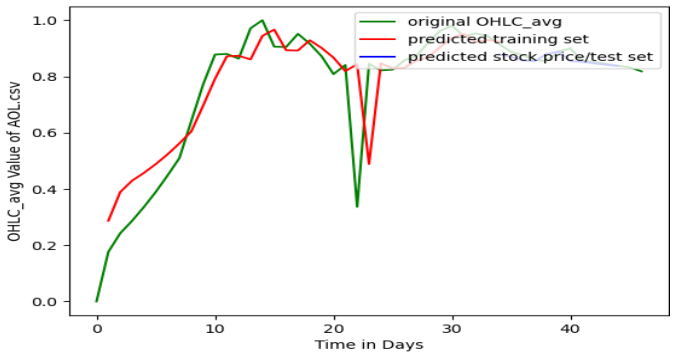

Figure 6. Bad Prediction in LSTM

VI. CONCLUSIONS

In this study, we discuss the whole process of developing a prediction system along with the results of the manual verification of the tool. In literature, most studies focused on individual companies and adjusting the derived stock variables to recognize the pattern. In our study, we analyze and justify manually that using only OHLC average can best recognize the pattern in the stock market listed companies in DSE. As OHLC indicates the central tendency of a particular stock, it can define the change in the market from day to day. Most analogous to the real world, we see that companies that are irregular and do not belong to the A or B category according to DSE (regular companies form the A and B category), fail to reproduce good decision criteria. The final remarks of our study, the OHLC average alone consist the set of input feature trained using the GRU model to predict the market change of the Dhaka stock exchange.

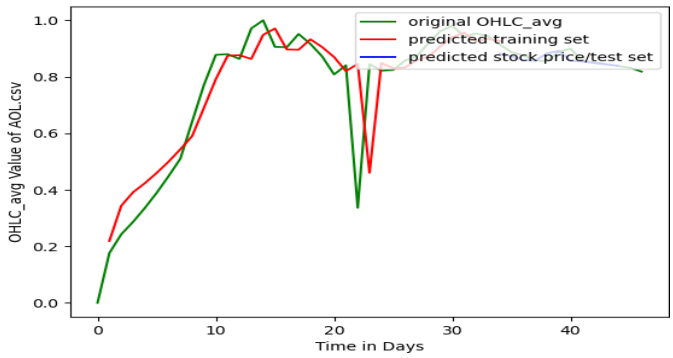

Figure 7. Bad Prediction in GRU

REFERENCES

(1) Akhter, S.; Misir, M. A. Capital markets efficiency: evidence from the emerging capital market with particular reference to Dhaka stock exchange. *South Asian Journal of Management* **2005**, *12*, 35.

(2) Faruqui, F.; Rahman, M. H. Factors Influencing the Crash in the Share Market in Dhaka Stock Exchange. *Research Journal of Finance and Accounting* **2013**, *4*, 139–147.

(3) Islam, M. A.; Islam, M. R.; Siddiqui, M. H. Stock market Volatility: comparison between Dhaka stock exchange and Chittagong stock exchange. *International Journal of Economics, Finance and Management Sciences* **2014**, *2*, 43–52.

(4) Milosevic, N. Equity forecast: Predicting long term stock price movement using machine learning. *arXiv preprint arXiv:1603.00751* **2016**.

(5) Dey, P.; Nahar, N.; Hossain, B. News Impact on Stock Trend. *International Journal of Education and Management Engineering* **2019**, *9*, 40–49.

(6) Shah, D.; Isah, H.; Zulkernine, F. Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies* **2019**, *7*, 26.

(7) Nabipour, M.; Nayyeri, P.; Jabani, H.; Mosavi, A.; Salwana, E., et al. Deep learning for stock market prediction. *Entropy* **2020**, *22*, 840.

(8) Hu, Z.; Zhao, Y.; Khushi, M. A Survey of Forex and Stock Price Prediction Using Deep Learning. *Applied System Innovation* **2021**, *4*, 9.

(9) Shin, D.-H.; Choi, K.-H.; Kim, C.-B. Deep learning model for prediction rate improvement of stock price using RNN and LSTM. *The Journal of Korean Institute of Information Technology* **2017**, *15*, 9–16.

(10) King, B. F. Market and industry factors in stock price behavior. *the Journal of Business* **1966**, *39*, 139–190.

(11) Asadi, S.; Hadavandi, E.; Mehmanpazir, F.; Nakhostin, M. M. Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction. *Knowledge-Based Systems* **2012**, *35*, 245–258.

(12) Miao, K.; Chen, F.; Zhao, Z. Stock price forecast based on bacterial colony RBF neural network. *Journal of Qingdao University (Natural Science Edition)* **2007**, *2*.

(13) Naeini, M. P.; Taremian, H.; Hashemi, H. B. In *2010 international conference on computer information systems and industrial management applications (CISIM)*, 2010, pp 132–136.

(14) Mohan, S.; Mullapudi, S.; Sammeta, S.; Vijayvergia, P.; Anastasiu, D. C. In *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2019, pp 205–208.

(15) Patel, M. B.; Yalamalle, S. R. Stock price prediction using artificial neural network. *International Journal of Innovative Research in Science, Engineering and Technology* **2014**, *3*, 13755–13762.

(16) Chong, E.; Han, C.; Park, F. C. Deep learning networks for stock market analysis and prediction: Methodology,

data representations, and case studies. *Expert Systems with Applications* **2017**, *83*, 187–205.

(17) Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **1998**, *6*, 107–116.

(18) Sundermeyer, M.; Schlüter, R.; Ney, H. In *Thirteenth annual conference of the international speech communication association*, 2012.

(19) Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* **2014**.

(20) Ravanelli, M.; Brakel, P.; Omologo, M.; Bengio, Y. Light Gated Recurrent Units for Speech Recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2018**, *2*, 92–102.

(21) Su, Y.; Kuo, C.-C. J. On extended long short-term memory and dependent bidirectional recurrent neural network. *Neurocomputing* **2019**, *356*, 151–161.

(22) Gruber, N.; Jockisch, A. Are GRU cells more specific and LSTM cells more sensitive in motive classification of text? *Frontiers in Artificial Intelligence* **2020**, *3*, 1–6.

(23) Dey, P. P.; Nahar, N.; Hossain, B. Forecasting Stock Market Trend using Machine Learning Algorithms with Technical Indicators. *International Journal of Information Technology and Computer Science* **2020**, *12*, 32–38.