A Hybrid Deep Learning Approach For Stock Price Prediction

Abhishek Dutta[1], Gopu Pooja[2], Neeraj Jain[3], Rama Ranjan Panda[4], and Naresh Kumar Nagwani[5]

Department of Computer Science and Engineering.
National Institute of Technology Raipur, India
abhi97.dutta@gmail.com, poojagopu98@gmail.com, neerajjain311@gmail.com, rrpanda.phd2018.cs@nitrr.ac.in, nknagwani.cs@nitrr.ac

Abstract. Prediction of stock prices has been the primary objective of an investor. Any future decision taken by the investor directly depends on the stock prices associated with a company. This work presents a hybrid approach for the prediction of intra-day stock prices by considering both time series and sentiment analysis. Furthermore, it focuses on Long Short-Term Memory (LSTM) architecture for the time series analysis of stock prices and Valence Aware Dictionary and sEntiment Reasoner (VADER) for sentiment analysis. LSTM is a modified Recurrent Neural Network (RNN) architecture. It is efficient at extracting patterns over sequential time-series data, where the data spans over long sequences and also overcomes the gradient vanishing problem of RNN. VADER is a lexicon and rule-based sentiment analysis tool attuned to sentiments expressed in social media and news articles. The results of both techniques are combined to forecast the intra-day stock movement and hence the model named as LSTM-VDR. The model is first of its kind, a combination of LSTM and VADER to predict stock prices. The dataset contains closing prices of the stock and recent news articles combined from various online sources. This approach, when applied on the stock prices of Bombay Stock Exchange (BSE) listed companies has shown improvements in comparison to prior studies.

Keywords: Time-series analysis, Sentiment analysis, Deep Learning, LSTM, CNN, RNN, VADER, Web Scraping, NLP.

1 Introduction

The stock market is volatile and dependent on various factors, such as socio-economic scenarios, political situations, and technological advancements. Also, the uncertainty involved in the prediction of intra-day stock prices can take random path and are unpredictable. It leads to every attempt of predicting intra-day stock prices futile over a larger period as discussed in random walk theory [1]. However, improvements in Artificial Intelligence methods and availability of powerful processors along with the growth of available data has significantly improved the accuracy of prediction.

Stock price is one such parameter, which can be either closing price or opening price. The reason being, its the most crucial stock market indicator on which investors rely. This paper considers closing price as stock prices and is used alternatively on various sources.

The majority of the earlier work in this domain involved prediction based on the time-series analysis over various technical indexes such as closing price, opening price, and volume of the Standard & Poor's (S & P) 500 indexes [2,3]. However, such Statistical Analysis or Artificial Intelligence models solely depend upon dataset involving only the previously collected stock prices known as financial records [4,5].

Growth of online newspapers, social media, and blogs has provided financial articles related to stocks of companies that were either scattered or unavailable earlier. The Internet has brought scattered data and information accessible easily and faster than ever before. It has geared new stock prediction techniques over those financial articles such as news impact on stock prices using sentiment analysis or event-based stock movement prediction [6,7]. The studies show that recent news can have a substantial effect on the market trend analysis and should be considered to capture subtle changes [8,10].

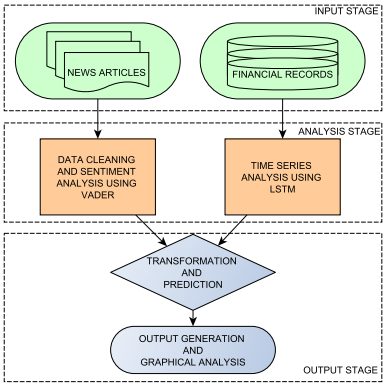

Fig. 1: LSTM-VDR Model Design

Prior research includes the use of techniques like, noun phrases, bags-of-words, or custom corpus that captures and performs better for certain stocks only [14,15]. Advancement in NLP has helped in the determination of stock movement with better accuracy for sentiment analysis as discussed in [11–13].

This work considers the best of both the analysis using the hybrid LSTM-VDR model. The model considers LSTM architecture for the time series analysis of intra-day stock prices [9]. VADER on the other hand, applies sentiment analysis over the news articles for generating sentiment scores. It considers a combination of lexicons, which are collections of features such as phrases. It is usually labeled according to its orientation of semantics as either negative or positive [16]. The predicted time series values, together with sentiment scores of recent news articles collected from various online sources, gives the final stock price predictions for certain number of days.

The discussion of paper follows section-2 which describes the hybrid LSTM-VDR model. Section-3 explains the working through an example. Following it, section-4 evaluates the result and perform comparison between different models.

2 Model Design and Problem Evaluation

2.1 Input Stage

The architecture considers two categories of input, the first one is financial records, and the second one is the sequence of recent news articles. In order to differentiate them, the input layer is restated as a quantitative layer and qualitative layer respectively in following paragraphs.

The quantitative layer takes, chronologically ordered seven technical indicators as input from Yahoo Finance for S & P 500 index [17,18]. For this work closing prices is selected as the technical indicator and is arranged chronologically according to its corresponding date. The qualitative layer takes news articles from web scrapping of some online newspapers such as economics times *et al.* [19,20]. Web scrapped data from various sources is combined and prepared in a paragraph as the master article. Then, sentiment scores are evaluated and arranged in chronological order according to date.

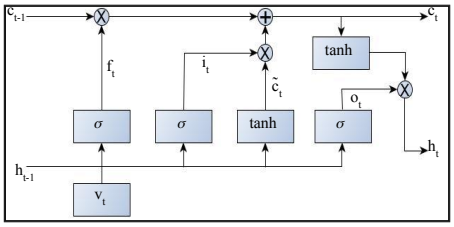

Fig. 2: LSTM Architecture [22]

2.2 Analysis Stage

Time series analysis of financial records performed using LSTM have internal mechanisms known as gates that regulates the flow of information [21]. The architecture as in Fig. 2 and equations (1) to (9) explains the work in detail. Algorithm 1 explains the cell state calculations that selects the next cell state which in turn predicts the final value.

Cell state is the most important part of LSTM architecture, denoted as c_t, where t represents the timestamp. Three different gates, namely, the forget-gate, the input-gate, and the output-gate evaluates the cell state c_t and output h_t. The forget-gate, f_t, decides which value from previous data to forget or remember. The input-gate, i_t selects the input signal that updates the values of current cell state. The output-gate, o_t allows the cell state to determine whether it has effect on other neurons or not. It generates the output considering the dependencies through activation function at gates. It also prevents the vanishing gradient problem of RNN.

$$f_t = \sigma(W_{vf} * v_t + W_{hf} * h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma(b_i + W - vi * v_t + W_{hi} * h_{t-1}) \tag{2}$$

$$\tilde{c_t} = \tanh(b_c + W_{vc} * v_t + w_{hc} * h_{t-1}) \tag{3}$$

$$c_t = i_t * \tilde{c_t} + f_t * c_{t-1} \tag{4}$$

$$o_t = \sigma(b_o + W_{vo} * v_t + W_{ho} * h_{t-1}) \tag{5}$$

$$h_t = \tanh(c_t) * o_t \tag{6}$$

The equations (1) to (6) represent the LSTM equations, where v_t is the recurrent layer input, h_t is recurrent unit output, and W is the weight matrices as in [22].

Algorithm 1: Calculation of cell state values.

Output: Determination of cell state c_t and value h_t, the next state should have from previous values.
Input: Previous c_t and h_t values along with current input V
c_t = [0, 0, ..., 0];
h_t = [0, 0, ..., 0];
while v_t *in* V **do**
  combine = h_t + v_t;
  f_t = forget_layer(*combine*);
  $\tilde{c_t}$ = candidate_layer(*combine*);
  i_t = input_layer(*combine*);
  c_t = c_t * f_t + $\tilde{c_t}$ * i_t;
  o_t = output_layer(*combine*);
  h_t = o_t + tanh(c_t);
  if h_t >t **then**
    select that hidden state as the next state;
  else
    Save h_t and c_t values for next iteration;
    continue to loop;
  end
end

[1]**Note:** i_t is input-gate, f_t is forget-gate, and o_t is output-gate respectively and $*$ determines element wise multiplication.

The values v_t and h_t concatenated as *combine* and fed into the $forget\_layer$ which in turn removes any unnecessary data. A *candidate_layer* is created using *combine*. The *candidate_layer* holds possible values for combining with the cell state. Moreover, *combine* is further supplied to the *input_layer*. This layer selects the data from the *candidate_layer* that should be added to the next cell state. After computing the $forget\_layer$, *candidate_layer*, and the *input_layer*, calculations using these newly generated values and the previous cell evaluates the next cell value.

On the other hand, VADER used for text sentiment analysis of news articles provides three different categories of polarity scores namely, positive, negative, and compound . These scores quantifies the emotion intensity of a statement. It

combines quantitative analysis and validation of empirical results using human-raters and wisdom of crowd technique.

Algorithm 2: Determination of new stock values through the processing of news articles and financial data.

Output: Comparison graphs for actual and predicted stock prices.
Input: News articles and Financial Records.
while *true* **do**
    Web scrapping of news articles;
    Applying Natural Language Processing;
    Collecting Financial records and generating sentiment scores;
    Linear transformation of sentiment scores and closing prices;
    Generation of values considering date as index;
    Predicting news values using LSTM;
    Generating graphs and accuracy reports;
end

[2]**Note:** Results are generated for companies listed in BSE.

2.3 Output Stage

The predicted output value from the time series analysis for day n validated by combining with the sentiment score of the recent news articles of $(n-1)^{th}$ days produce the final stock price. The predictions are combined with sentiment scores of news articles since market trends are susceptible to recent changes as explained by Manuel R. Vargas *et al.* [22]. The proposed hybrid model evaluates over time series data using deep learning as well as handle recent market changes using sentiment analysis. Moreover, it is robust in predicting some stock prices for S & P 500 index for the Indian market over similar hybrid models as shown Table 2. It also shows advancements with VADER, not considered in previous works. Algorithm 2 demonstrates the workflow.

3 Experimental Setup

This section illustrates a working example of the proposed model and comparison of various models.

This illustration takes sample data for stock prices of six days as the input X shown in Table 1 and predicts the stock price for the seventh day. Average of input values is $\bar{X} = 1582.85$ and normalization of data between -1 and 1 calculated as $\frac{X-\bar{X}}{\bar{X}}$. The average of normalized value turns out to be $V = 0.003$.

LSTM work in similar fashion where the average of first n days calculate price for the $(n+1)^{th}$ day and feed again along with next n values to calculate the $(n+2)^{th}$ day closing price and so on. This example considers c_t and h_t as 1 such

Table 1: Sample stock closing prices of for six days with associated dates.

Date	Closing Price	Date	Closing Price
12-12-2019	1,599.10	12-18-2019	1,566.60
12-13-2019	1,609.95	12-19-2019	1,582.90
12-16-2019	1,575.85	12-20-2019	1,568.20
12-17-2019	1,562.70		

that the cell in focus has a higher probability of selection to calculate the next value. De-normalized value for the seventh day turns to be 1577.45 which results to a difference of 5.39 from the actual price and an accuracy of $100 - 59.04 = 40.96\%$. The sentiment analysis through VADER generates the polarity scores between $[-1, 1]$, where values closer to 1 signifies greater possibilities of increased stock prices for the following day and vice-versa. The difference of predicted and actual value is 5.39 and considering sentiment score of -0.67 decreases the value as $1577.45 - 5.39 * 0.67 = 1573.84$ that increases the accuracy to 64.05%.

The accuracy of different models are compared based on stock prices. The comparison is with respect to combination of methods that include architectures such as Linear Regression (LRGS), Moving Average (MAVG), k-nearest Neighbour (KNBR), and Auto ARIMA (ARM) to evaluate the technical values and NLP methods like Naive Bayes and SVM for sentiment analysis.

4 Experimental Results

This section shows a comparative study between different models for stock prices over the same period through accuracy scores as shown in Table 2. It is for the stock prices of PC Jewellers Ltd (PCJ).

Table 2: Comparison of architectures for PCJ stocks [18]

Model	Accuracy	Model	Accuracy	Model	Accuracy
LRGS	51.231	LSTM	70.896	ARM-VDR	63.376
MAVG	53.746	LRGS-VDR	52.443	LSTM-NBY	47.865
KNBR	53.984	MAVG-VDR	55.418	LSTM-SVM	66.795
ARM	61.412	KNBR-VDR	59.785	LSTM-VDR	77.496

Stand-alone models, as well as hybrid models, are constructed using a combination of the above models, namely, LRGS-VDR, MAVG-VDR, KNBR-VDR, ARM-VDR, LSTM-NBY, LSTM-SVM, and LSTM-VDR to predict as described in the paragraph above.

Graphical comparison of the predicted and actual values using the proposed model LSTM-VDR is shown in Fig. 3 & 4 for the stocks of PCJ and Reliance

Industries Ltd (RIL), respectively. It reinstates the relevance of customized input and the novel approach with better results.

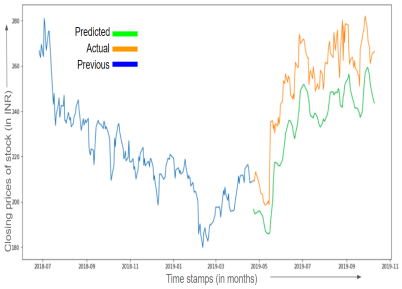

Fig. 3: PCJ

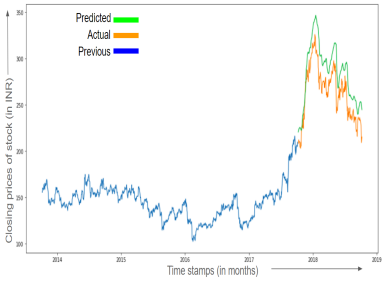

Fig. 4: RIL

5 Conclusion

The work demonstrates that the hybrid model that combines financial records and news articles, can have better performance for certain market conditions. It, also captures the temporal features satisfactorily even though the model considers news only from recent days. This results reinforce the fact that the information of news articles has a short temporal effect for analysis of the stock market.

References

1. Malkiel, B. G. (1999). A random walk down Wall Street: including a life-cycle guide to personal investing. WW Norton & Company.
2. Mizuno, H., Kosaka, M., Yajima, H., & Komoda, N. (1998). Application of neural network to technical analysis of stock market prediction. Studies in Informatic and control, 7(3), 111-120.
3. Leigh, W., Purvis, R., & Ragusa, J. M. (2002). Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. Decision support systems, 32(4), 361-377.
4. Kim, K. J. (2003). Financial time series forecasting using support vector machines. Neurocomputing, 55(1-2), 307-319.
5. Kim, K. J., & Lee, W. B. (2004). Stock market prediction using artificial neural networks with optimal feature transformation. Neural computing & applications, 13(3), 255-260.
6. Maqsood, H., Mehmood, I., Maqsood, M., Yasir, M., Afzal, S., Aadil, F., ... & Muhammad, K. (2020). A local and global event sentiment based efficient stock exchange forecasting using deep learning. International Journal of Information Management, 50, 432-451.

7. Lim, S. L. O., Lim, H. M., Tan, E. K., & Tan, T. P. (2020). Examining Machine Learning Techniques in Business News Headline Sentiment Analysis. In Computational Science and Technology (pp. 363-372). Springer, Singapore.
8. Schumaker, R. P., & Chen, H. (2009). A quantitative stock prediction system based on financial news. Information Processing & Management, 45(5), 571-583.
9. Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE.
10. Nofsinger, J. R. (2001). The impact of public information on investors. Journal of Banking & Finance, 25(7), 1339-1366.
11. Wang, W., Li, W., Zhang, N., & Liu, K. (2020). Portfolio formation with preselection using deep learning from long-term financial data. Expert Systems with Applications, 143, 113042.
12. Hamori, S. (2020). Empirical Finance.
13. Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Transactions on Information Systems (TOIS), 27(2), 1-19.
14. Yoo, P. D., Kim, M. H., & Jan, T. (2005, November). Machine learning techniques and use of event information for stock market prediction: A survey and evaluation. In International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06) (Vol. 2, pp. 835-841). IEEE.
15. Wüthrich, B., Permunetilleke, D., Leung, S., Lam, W., Cho, V., & Zhang, J. (1998). Daily prediction of major stock indices from textual www data. Hkie transactions, 5(3), 151-156.
16. Hutto, C. J., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eighth international AAAI conference on weblogs and social media.
17. Yahoo Finance data for Reliance Industries Limited- https://in.finance.yahoo.com/quote/RELIANCE.NS. (Last accessed on: 23-12-2019)
18. Yahoo Finance data for PC Jeweller Limited- www.in.finance.yahoo.com/quote/PCJEWELLER.NS. (Last accessed on: 23-12-2019)
19. Economic Times articles for PC Jeweller Limited- www.economictimes.indiatimes.com/pc-jeweller-ltd/stocks/companyid-42269.cms. (Last accessed on: 23-12-2019)
20. Economic Times articles for Reliance Industries Limited- www.economictimes.indiatimes.com/reliance-industries-ltd/stocks/companyid-13215.cms. (Last accessed on: 23-12-2019)
21. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
22. Vargas, M. R., De Lima, B. S., & Evsukoff, A. G. (2017, June). Deep learning for stock market prediction from financial news articles. In 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) (pp. 60-65). IEEE.