Exploring Stock Market Forecasting with LSTMs

Jeremy Dale Coronia
College of Computer Studies
De La Salle University
Taft Ave., Malate, Manila
jeremy_coronia@dlsu.edu,ph

***Abstract*—The Long Short-Term Memory (LSTM) is a variant of a recurrent network and is one of the most popular models used in time series data such as in stock market prediction because of their capability in addressing long-term dependencies. As such, this work explores the use of vanilla and stacked Long Short-Term Memory (LSTM) models in stock market prediction. Various parameters are explored, such as batch size, number of hidden layers and dropout value, to optimize the use of LSTMs in a financial time-series forecasting task and yield the best results. The historical Closing Price from publicly available stock market data is used as the input feature to test the LSTMs. The resultant LSTM model was also tested on a more volatile, and different, dataset to test its robustness.**

I. Introduction

Time-series data is a widely used type of data in a variety of disciplines which includes signal analysis, speech recognition, and more. Time series can be defined as a sequence of observations made at regular intervals and arranged chronologically over a predetermined time period. The forecasting or prediction of the stock market is one form of time-series application. Stock market prediction is the method through which the future value of a traded value, such as company stock, is decided with the goal of profit. For reference, stocks represent company ownership and value. Unfortunately, previous approaches for predicting future stock prices have proven challenging and unreliable due to the extremely volatile and noisy character of stock market data, particularly because of the nonlinearity of stock market data.

With that, and the growing interest in deep learning models to perform operations on non-linear and complex data, neural networks are being explored to perform time-series forecasting of the stock market. Neural networks have also been shown to perform financial forecasting more accurately than linear regression models [1]. The level of complexity in financial time-series prediction can also test a deep neural network's robustness when handling very noisy data.

Within the field of financial time-series forecasting in deep learning, one of the dominant models used is the Long Short-Term Memory (LSTM) because of its capability to remember preceding data inputs or states along with the current input [2]. The LSTM in particular was developed [3] to handle the limitations of recurrent networks in handling long-term dependencies, or handling memory states extending several time steps back.

As such, this work is an exploration of LSTMs into stock price forecasting or prediction. Various parameters will also be investigated to help determine which would yield better performance when predicting the trend of the stock market. Finally, after experiments, a new more volatile dataset will be tested on the resultant LSTM model. All data used for training and testing is publicly available online.

This paper is hence divided into the following sections. The next section discusses related works in the field, such as neural networks in financial economics and the prevalence of LSTMs. This is followed by a discussion of the dataset used and the specifics of the LSTM models for experiments. Next is a discussion of the results which is then finished off by conclusions and recommendations for future work.

II. Related Work

The connection between financial economics and deep learning (DL), interchanged often with [artificial] neural networks, is extensive. Neural networks have been used in a variety of sub-disciplines in the field [1], such as bankruptcy prediction, option pricing, and short-term foreign exchange rates where ANNs were compared against traditional forecasting methods like random walk and exponential smoothing [4]. Yao [5] also compared neural networks against a baseline method called Autoregressive Moving Average (ARMA) on the exchange rates between Swiss Franc and US Dollar. concluding that ANN models yield higher returns.

Similarly, stock market prediction is an important facet of financial economics, examining several variables affecting its performance such as earnings yield, cash flow yield and more. ANNs have emerged as a promising alternative even in early years because of its capability to address the non-linear and volatile nature of stock market data, such as predicting the Japanese stock price [6].

In terms of DL model used, LSTMs have consistently been a popular choice used in many studies using time series data, including stock market prediction, particularly because of its capability to learn the temporal characteristics of time series signals [2], fitting the data better than RNNs [7]. LSTMs are a variant of the recurrent neural network designed specifically to address long-term dependencies and mitigate the vanishing gradient problem [3], though its weight updates and optimization methods stay largely the same as the RNN.

In terms of stock trend prediction using LSTMs, [8] used historical data from the China stock market as input to an LSTM model, which was found to improve over a random prediction method from 14.3% to 27.2%. Similarly, [9] explored optimization strategies in LSTMs to predict future trends of the Romanian stock market, using the historical Opening, Highest, Lowest, and Closing Prices from January 2001 to November 2016. Attention mechanisms may also be added to an LSTM model for a smaller error [10]. Stock market prediction can use different input parameters, such as the daily closing or opening prices as mentioned above. Other studies have supplemented the data with generated technical indicatoyrs [11] or emotion analysis [12]. Textual data like news articles have also been used, either to supplement the quantitative data [13] or used alone [14].

Studies have also compared the performance of LSTMs with other neural networks such as RNNs, Convolutional Neural Networks (CNN) and Multilayer Perceptrons (MLP) alongside classical linear models [15], [16]. Other studies explore variants of the LSTM, such as stacking layers upon one another or using a bi-directional LSTM [17], [18]. Alongside those is an investigation into hyperparameter optimizations of LSTMs, such as one study [19] which observed that in terms of hidden layers, one hidden layer was sufficient in predicting future trends in the Indian stock market in terms of the mean Root Mean Square Error (RMSE) and validated by ANOVA testing.

III. Methodology

A. The Dataset

The data used for this is the historical stock data of Google, which is freely available on Yahoo finance.[1] The stock market data ranges from July 6, 2005 until July 6, 2022. Figures 1 and 2 shows the plot of the Closing Price and Sales Volume of Google from July 6, 2005 until July 6, 2022, respectively. The plot of the Closing Price in particular would be used to compare against the performance of the LSTM models later on. It can be seen from the plot an overall increasing trend. The closing value (of the dollar) was used as the input to be fed into the LSTM models.

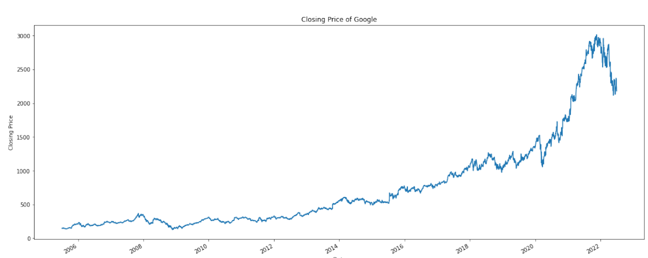

Fig. 1. The Closing Price of Google from July 2005 until July 2022.

Feature transformation was also performed by normalizing the closing price values within the (0, 1) range. A 90-10 split on the dataset was performed before training the model. To be specific since the data is time series, the training data ranges only from July 2005 until October 21, 2020.

For testing, the LSTM models will perform prediction starting October 22, 2022 until July 6, 2022.

[1] https://finance.yahoo.com/

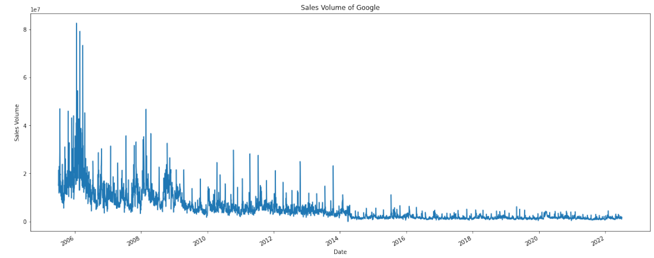

Fig. 2. The Sales Volume of Google from July 2005 until July 2022.

B. The LSTM

The significance of the LSTM, and its difference to the standard RNN model, lies with its unit, wherein three (3) gates, mathematically expressed by a combination of a sigmoid function and point-wise operations, modulates the flow of information as it passes from layer to layer across several time steps. The sigmoid activation function ($\sigma(x)$ outputs a value between 0 and 1. In this context, 0 means that past information is completely dropped while 1 means to retain past information.

$$f_t = \sigma(X_t * U_f + H_{t-1} * W_f) \tag{1}$$

$$I_t = \sigma(X_t * U_i + H_{t-1} * W_i) \tag{2}$$

$$\bar{C}_t = tanh(X_t * U_c + H_{t-1} * W_c) \tag{3}$$

$$C_t = f_t * C_{t-1} + I_t * \bar{C}_t \tag{4}$$

$$O_t = \sigma(X_t * U_o + H_{t-1} * W_o) \tag{5}$$

$$H_t = O_t * tanh(C_t)) \tag{6}$$

Equations 1 until 6 is the transition of equations as information is passed from unit to unit. f_t, I_t, and O_t represent the forget gate, input gate, and output gate, respectively. H_t represents the resultant hidden state to be passed, and C_t represents the memory cell containing past information.

In this study, the default configurations of the LSTM model is an input layer followed by four (4) LSTM layers, with the first hidden layer having 128 units, and subsequent hidden layers having 64 units. This is followed by two (2) Dense layers of units 25 and 1, respectively, both with linear activations. It must be noted that the unit numbers for the hidden layers are arbitrary. Between each hidden layer is a 20% Dropout layer to help prevent overfitting [20]. A default batch size of 64 is used. Adam is the optimizer because of its computational efficiency [21], and the learning rate is set to 0.001. All models are trained for a fixed 50 epochs. Some parameters will be tweaked later on.

Afterward, the model is also applied on completely new stock market data with more unpredictability to test its capabilities.

IV. Results and Discussion

In this section, the results after experimentation will be discussed. An ablation study is also done and discussed in this section.

Firstly, Figure 3 shows the results of predicting the 2019 - 2022 given historical data from 2018 extending back to 2005. Compared to Figure 4 where a batch size of 32 is used, it is seen that using a larger batch size is more beneficial to perform stock market prediction.

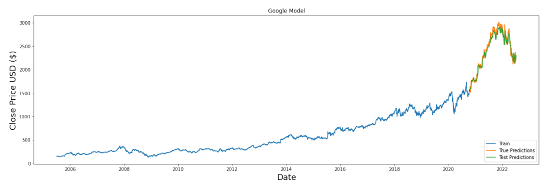

Fig. 3. **DEFAULT:** Stock Market Prediction of LSTM using a batch size of 64.

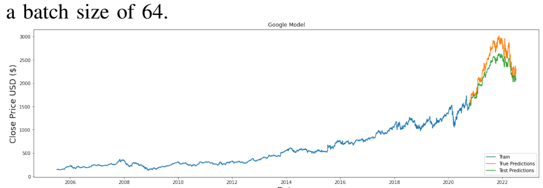

Fig. 4. Stock Market Prediction of LSTM using a batch size of 32.

A quantitative evaluation of the root mean square error (RMSE), mean absolute error (MAE) and F-Score is also shown on Table I. The results all show that using a higher batch size leads to significantly lower errors even if the F-scores are marginally close to each other. This shows that using a higher batch size helps the model better fit the data.

Batch Size	RMSE	MAE	F-score
32	229.2036	204.3936	0.7061
64 **(Default)**	83.1897	39.9417	0.7100

TABLE I
Comparison of Batch Sizes.

Next, the dropout value is explored. The dropout value, as mentioned earlier, is added after every layer to help prevent overfitting to the training data. The default value used is set to 0.2, but the value was changed up until 0.6 to see the model's change in performance on the test data. This dropout unit specifies the probability of layer outputs being dropped during training time. The results are shown below. Batch size was reverted back to 64 for this, considering the better results.

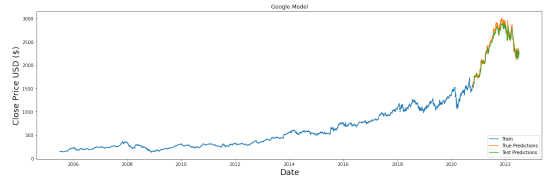

Fig. 5. **DEFAULT:** Stock Market Prediction of LSTM when dropout is 0.2.

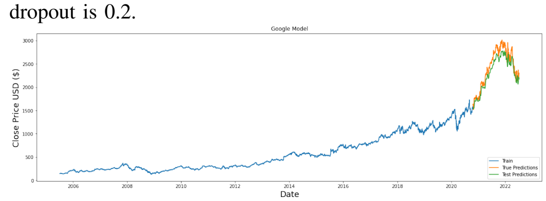

Fig. 6. Stock Market Prediction of LSTM when dropout is set to 0.3.

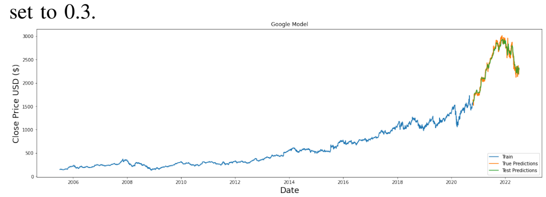

Fig. 7. Stock Market Prediction of LSTM when dropout is set to 0.4.

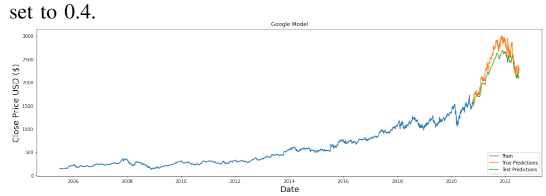

Fig. 8. Stock Market Prediction of LSTM when dropout is set to 0.5.

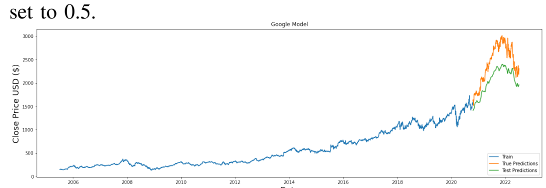

Fig. 9. Stock Market Prediction of LSTM when dropout is set to 0.6.

Dropout	RMSE	MAE	F-score
0.2 **(Default)**	83.1897	39.9417	0.7100
0.3	151.3694	127.1575	0.7100
0.4	76.8386	4.8946	0.7100
0.5	200.1323	174.7358	0.7100
0.6	406.8974	382.4356	0.7100

TABLE II
Comparison of Dropout Value.

Moreover, Table II shows the RMSE, MAE and F-score per dropout percent. The 40% Dropout is shown to perform the best with the lowest RMSE and MAE result. Figures 5 and 7 when dropouts are 0.2 or 0.4, respectively, show the best results when plotting the data, with the model being able to predict the stock price trend the most accurately. It is also interesting to note, however, that the F-score has remained unchanged throughout.

Finally, the number of LSTM layers were explored. The default number of layers was set to four (4), and the number of layers was changed from one (1) until six (6) layers. The results after experimentation are shown below in Figures 10 to 15. For further comparison, the RMSE, MAE and F-Score of the following settings is shown on Table III. Dropout has been changed to 0.4 to reflect the better results from before.

Interestingly, it can be seen here that the best performing models are with a single-layered LSTM model or when two layers are used as opposed to using the default four (4) layers. These findings are somewhat similar and further validate an earlier discussion over a simple one-layer LSTM model performing better than a stacked LSTM [19].

Moreover, as more layers are used, it can be seen that the RMSE and MAE grow larger, and the plotting of values reflect the difficulty of these LSTM models to predict the trend of the stock market.

No. of Layers	RMSE	MAE	F-score
1	61.5139	28.3134	0.7041
2	66.5001	21.6711	0.7100
3	78.6617	34.5100	0.7100
4 **(Default)**	83.1897	39.9417	0.7100
5	241.7486	185.5751	0.7100
6	382.2452	330.2196	0.7099

TABLE III
COMPARISON OF NUMBER OF LAYERS.

Lastly, we vary the learning rate to determine how strong its effect it would be on stock market prediction. For reference, the default learning rate is 0.001. It must be noted that the learning rates chosen are mostly arbitrary and approximated so that it would not be too little or too large. The RMSE and MAE are shown below on Table IV. Due to the results above and to consume less resources, only a single LSTM layer is used for this set of experiments.

Learning Rate	RMSE	MAE
0.0003	112.9354	87.0816
0.0005	119.3865	96.8160
0.001 **(Default)**	83.1897	39.9417
0.005	251.4531	233.2539

TABLE IV
COMPARISON OF LEARNING RATE.

After experimentation (some values not shown on the table), the learning rate which yielded the best result is the default 0.001. We plot them on Figures 17 until 18 reflecting the results.

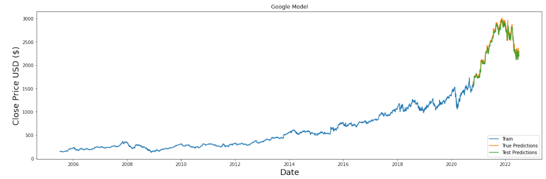

Fig. 10. Stock Market Prediction of a single-layer LSTM model.

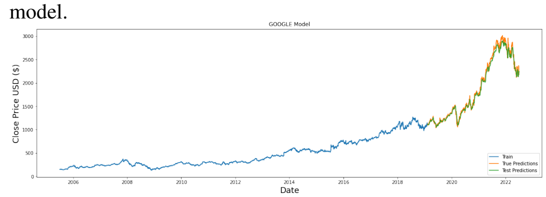

Fig. 11. Stock Market Prediction of a model with two (2) LSTM layers.

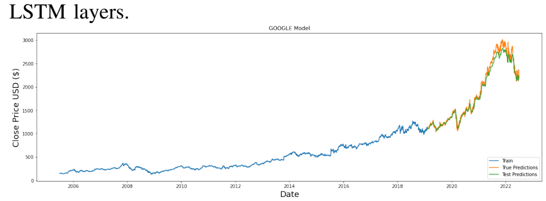

Fig. 12. Stock Market Prediction of a model with three (3) LSTM layers.

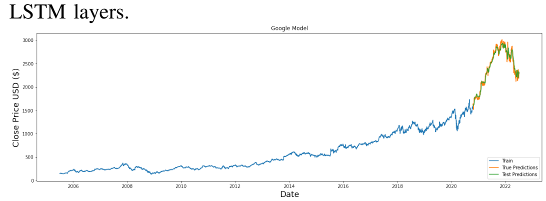

Fig. 13. **DEFAULT:** Stock Market Prediction of a model with four (4) LSTM layers.

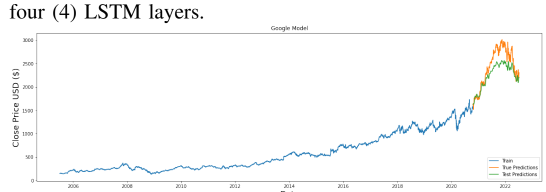

Fig. 14. Stock Market Prediction of a model with five (5) LSTM layers.

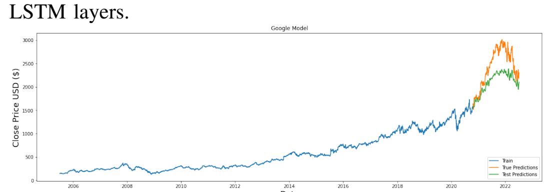

Fig. 15. Stock Market Prediction of a model with six (6) LSTM layers.

A. *Further Testing*

After singling out the best performing settings, the new LSTM model is also tested on a new stock market dataset. This new dataset, from HP, ranges from January 2, 1970 until November 10, 2017. As shown in Figure 20, the data is significantly more volatile than the previously used dataset. This dataset is also publicly available online.

Table V meanwhile show the comparison of the quantitative results. Looking at the RMSE, MAE, F-Score, it can be seen that there is not much difference between the F-Score results; however, the New LSTM has significantly lower RMSE and MAE than the Default LSTM settings used. Similarly, as seen on Figures 21 and 22 showing the plot of the 2 models, the

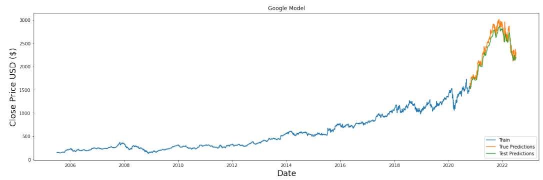

Fig. 16. Plot when using a Learning Rate of 0.0003.

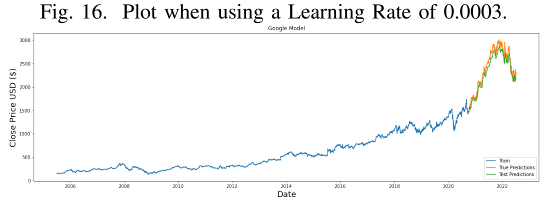

Fig. 17. Plot when using a Learning Rate of 0.0005.

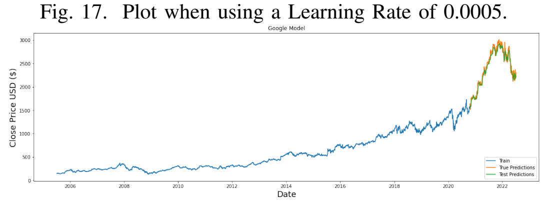

Fig. 18. **DEFAULT:** Plot when using a Learning Rate of 0.001.

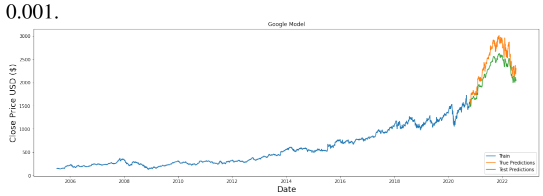

Fig. 19. Plot when using a Learning Rate of 0.005.

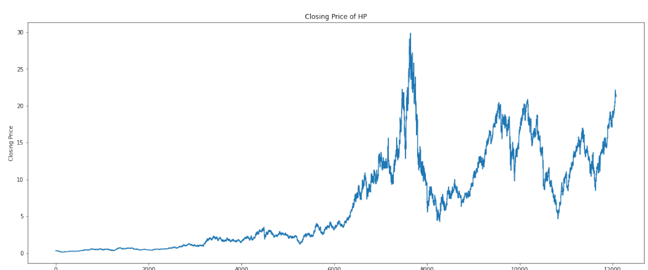

Fig. 20. The Closing Price of HP from January 1970 until November 2017.

default LSTM tends to overestimate when predicting the stock market trend. The new LSTM model is seen to commit fewer errors.

V. Conclusion

In this work, the LSTM model was applied on the stock market forecasting problem. Various settings were explored to determine which configuration would output the best results. We determine the optimal number of layers, the best possible learning rate for this specific stock market data and task, as well as other hyperparameters. The resultant LSTM model was also applied on a larger, more erratic dataset to test its performance and robustness.

	RMSE	MAE	F-score
NEW LSTM	0.2876	0.0056	0.7026
Default LSTM	0.5030	0.3901	0.7030

TABLE V
Comparison of Models.

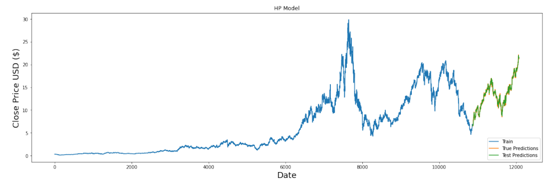

Fig. 21. Stock Market Prediction of the new LSTM model.

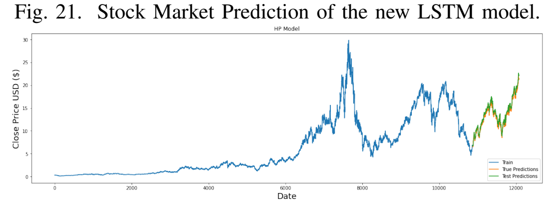

Fig. 22. Stock Market Prediction of LSTM using the **original configurations**.

For future work, tinkering more with the LSTM and its hyperparameters, or experimenting with other LSTM variants such as the Gated Recurrent Unit (GRU) may be explored. Supplementing the input with other forms of data and performing hyperparameter tuning may also be undertaken, as well as a comparison with other DL architectures. Due to limited computational resources, future work may also consider a higher epoch or more complex DL architectures with more data. Comparisons against traditional techniques of forecasting may also be explored.

References

[1] Y. Li and W. Ma, “Applications of artificial neural networks in financial economics: A survey,” in *2010 International Symposium on Computational Intelligence and Design*, vol. 1, 2010, pp. 211–214.

[2] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, “Financial time series forecasting with deep learning : A systematic literature review: 2005–2019,” *Applied Soft Computing*, vol. 90, p. 106181, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620301216

[3] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] C. Bellgard and P. Goldschmidt, “Forecasting foreign exchange rates: Random walk hypothesis, linearity and data frequency,” in *12th Annual Australasian Finance & Banking Conference*. Citeseer, 1999, pp. 1–18.

[5] J. Yao, N. Jingtao, Y. Li, C. L. Tan, and L. Tan, “Forecasting the exchange rates of chf vs usd using neural networks,” 1997.

[6] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, “Stock market prediction system with modular neural networks,” in *1990 IJCNN International Joint Conference on Neural Networks*, 1990, pp. 1–6 vol.1.

[7] Q. Jiang, C. Tang, C. Chen, X. Wang, and Q. Huang, “Stock price forecast based on lstm neural network,” in *Proceedings of the Twelfth International Conference on Management Science and Engineering Management*, J. Xu, F. L. Cooke, M. Gen, and S. E. Ahmed, Eds. Cham: Springer International Publishing, 2019, pp. 393–408.

[8] K. Chen, Y. Zhou, and F. Dai, “A lstm-based method for stock returns prediction: A case study of china stock market,” in *2015 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 2823–2824.

[9] E. Dezsi and I. A. Nistor, "Can deep machine learning outsmart the market? a comparison between econometric modelling and long-short term memory," *Romanian Economic and Business Review*, 2016.

[10] D. Wei, "Prediction of stock price based on lstm neural network," in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, 2019, pp. 544–547.

[11] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1419–1426.

[12] Q. Zhuge, L. Xu, and G. Zhang, "Lstm neural network with emotional analysis for prediction of stock price." *Engineering letters*, vol. 25, no. 2, 2017.

[13] 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)*, 2016, pp. 1–6.

[14] L. dos Santos Pinheiro and M. Dras, "Stock market prediction with deep learning: A character-based neural language model for event-based trading," in *Proceedings of the Australasian Language Technology Association Workshop 2017*, Brisbane, Australia, Dec. 2017, pp. 6–15. [Online]. Available: https://aclanthology.org/U17-1001

[15] H. M, G. E.A., V. K. Menon, and S. K.P., "Nse stock market prediction using deep-learning models," *Procedia Computer Science*, vol. 132, pp. 1351–1362, 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918307828

[16] 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.

[17] K. A. Althelaya, E.-S. M. El-Alfy, and S. Mohammed, "Evaluation of bidirectional lstm for short-and long-term stock market prediction," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, 2018, pp. 151–156.

[18] M. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi, "Deep learning-based stock price prediction using lstm and bi-directional lstm model," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 2020, pp. 87–92.

[19] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing lstm for time series prediction in indian stock market," *Procedia Computer Science*, vol. 167, pp. 2091–2100, 2020, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920307237

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.