Optimizing Long Short -Term Memory (LSTM) Model Hyperparameters for Enhanced Stock Price Forecasting and Portfolio Allocation

Siqi Li

Department of Economics, Haverford College, Haverford, 19041, USA

kli2@haverford.edu

Abstract. Due to the increasing application of Long Short-Term Memory (LSTM) models in stock price forecasting and portfolio allocation, it is crucial to tune the models for better accuracy. However, there is a limited study on how the hyperparameters of LSTM models affect the model performances. Therefore, this paper investigated the relationships between hyperparameters, particularly the number of neurons and LSTM layers, on the Mean Squared Error (MSE) of LSTM models. To shed light on practical significance, this research was conducted in the setting of portfolio optimization with a combination of LSTM stock price forecasting and Monte-Carlo Portfolio Simulation. More specifically, the LSTM model was first trained to forecast the weekly prices of five selected stocks, during which the MSE resulted from different number of neurons in the first LSTM layer as well as the total number of layers were compared. Following that, the combination of hyperparameters reaching the smallest MSE was selected for each stock, and the corresponding forecasted returns (calculated from the forecasted prices) were treated as input of the Monte-Carlo Simulation. Finally, the Monte-Carlo Simulation was employed for generating the desired portfolios. As the results demonstrated, the increase in number of layers in general leads to rising MSE, yet the increase in number of neurons in the first LSTM layer has either blurred effect or improving effects on model performances, depending on the original volatility of historical values.

Keywords: Long Short-Term Memory Model, Monte-Carlo Simulation, Stock Price Forecasting

1 Introduction

Stock price predictions have always been an intensely explored subject of study due to its significance in building a successful portfolio. Only with accurately forecasted stock movements, the resulting portfolio could earn the expected amount of return while minimizing volatilities at the same time. Traditionally, portfolio managers have widely applied time-series models such as Autoregressive Integrated Moving Average (ARIMA) model and Generalized Autoregressive Conditional Heteroskedasticity

J. Lu (ed.), *Proceedings of the International Workshop on Navigating the Digital Business Frontier for Sustainable Financial Innovation (ICDEBA 2024)*, Advances in Economics, Business and Management Research 315,
https://doi.org/10.2991/978-94-6463-652-9_64

(GARCH) to make price predictions, yet the machine learning models, such as the neural networks and random forests, are becoming the preferred tools for their improved ability in handling non-linearities and reaching more accurate forecasts.

Some traditional machine learning models such as decision trees and support vector machines (SVMs). Respectively Introduced by Morgan and Sonquist and Vapnik et al. [1], the two models were involved in a variety of classification and forecasting tasks, including applications in real-estate price predictions based on multiple features, stock market movement detections and cancer classifications. Later in 1997, the Long Short-Term Memory (LSTM) was first introduced by computer scientists for their improved structures in addressing long-term dependencies of data [2]. Compared to traditional models, the LSTM also demonstrates better performances in avoiding overfitting issues, handling noisy data, and capturing non-linear trends in timeseries. Hence, researchers have widely applied the LSTM models to classification tasks and natural language processing, including handwriting recognition as well as translation and language modelling [3]. For example, Sutskever and Vinyals found that the multilayered LSTM models demonstrated better performances in overall translation tasks, and especially long and complex sentences [4].

In addition to performing classification tasks, the LSTM models demonstrate strong abilities to accomplish forecasting tasks compared to traditional time-series models. Some of the important fields undergoing intense LSTM applications include weather forecasting, traffic flow prediction, and energy load forecasting. Zhao and Chen tested on the accuracy of LSTM in forecasting short-term traffic and proved that the LSTM model outperformed other conventional models like ARIMA and Support Vector Machines (SVM) [5]. Similarly, in the context of short-term residential energy load forecasting, Kong et al. indicated that the LSTM model provided a more accurate prediction than many of the alternative algorithms [6].

Another broadly investigated subject is stock price prediction. As mentioned before, improved accuracy to predict stock prices and financial market behaviors can greatly benefit portfolio building, which can in turn generate desirable amount of return at acceptable level of risks. For instance, Adil and Mhamed compared the performance of traditional RNN and LSTM on predicting two stocks from the New York Stock Exchange (NYSE) and found that LSTM showed higher accuracy [7].

However, though it is of interest to locate the most accurate stock forecasting model, few researchers have investigated the relationship between the hyperparameters of the LSTM and resulting model performance. Therefore, this study aims to explore how adjustments of hyperparameters, specifically the number of neurons and LSTM layers, affect the performance of the LSTM forecast on future stock prices as measured by the Mean Squared Error (MSE). In particular, the LSTM model is employed to forecast the weekly stock price from historical weekly prices between (Jul 29th, 2019 ~ Jul 20th, 2021) of five selected stocks including Apple Inc., the Boeing Co., Telsa Inc., Walt Disney Co., and the Meta Platforms Inc.

2 Methodology

2.1 Data Preparation

All the data used in this research is accessed and downloaded from Yahoo Finance. In particular, the weekly closing prices of the five stocks (i.e. Apple Inc., Boeing Co., Tesla Inc., Walt Disney Co., and Meta Platforms Inc.) from Jul 29th, 2019, to Jul 29th, 2021, are employed. Before model establishment, the time series is first normalized by the Min-Max Scaler to be distributed between 0 and 1. Also, to ensure that the models can be validated effectively, the datasets are split into training sets consisting of 80% length and testing sets of 20% length.

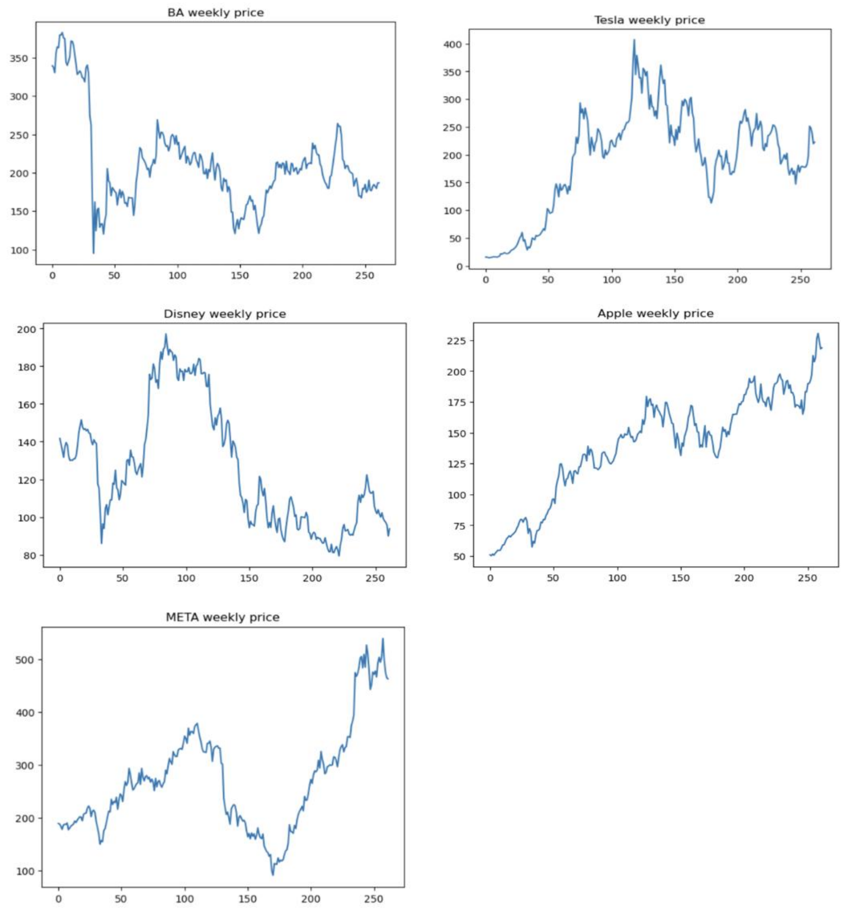

Fig. 1. Historical Weekly Prices of the 5 stocks over 262 weeks (from Jul 29th, 2019 to Jul 29th, 2021) (Photo/Picture credit: Original)

2.2 LSTM-based Stock Price Prediction

$$f_t = \sigma(V_{fx}x_t + V_{fh}h_{t-1} + b_f) \quad (1)$$
$$i_t = \sigma(V_{ix}x_t + V_{ih}h_{t-1} + b_i) \quad (2)$$
$$g_t = \phi\left(V_{gx}x_t + V_{gh}h_{t-1} + b_g\right) \quad (3)$$
$$o_t = \sigma(V_{ox}x_t + V_{oh}h_{t-1} + b_o) \quad (4)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (5)$$
$$h_t = o_t \odot \phi(c_t) \quad (6)$$

The Long Short-Term Memory (LSTM) model is a type of neural network model that enhances the ability to capture time series that present long-term dependencies [8-10]. The memory cell, which is the fundamental component of the LSTM model, is further comprised of four major parts: gates(forget gate(f_t), input gate(i_t), and output gate(o_t), the weight matrix associated with each gate, the cell state(c_t), and the hidden state(h_t). In each of the gates, the weight matrix is responsible for determining what proportion of the cell state and hidden state from the previous state should be retained and carried forward.

The forget gate, as expressed in equation (1), controls for the percentage of the previous cell state. (c_{t-1}) to be retained in the current memory cell by varying the weight matrix associated with it. The input gates, as demonstrated by equations (2) and (3) together, determine to which degree the new information (x_t) will be incorporated into the current cell state. As indicated by equation (5), the forget gate (f_t)and the input gate have combined effects by weighting the current cell state (c_t)between the previous cell state (c_{t-1}) and the information absorbed currently (g_t). Regarding the output gate (o_t) in equation (6), it is calculated to determine how much of the current cell state can be exported as the hidden state (h_t). The major significance of the LSTM model is to repetitively train the weight matrices associated with each gate to find the best-performing ones that make accurate decisions on the proportion of information being passed from one cell to another.

This study explores how the number of neurons and LSTM layers affect the performance of the model in stock price forecasting as measured by the magnitude of the Mean-squared Error (MSE). A time-step of 10 days is selected. In particular, the hyperparameters in this study are set up as follows: For all the five stocks, the number of neurons ranges from 2 to 64 for the first LSTM layer, and decreases by half for every additional LSTM layer; one more LSTM layer is added until the current LSTM layer only has two neurons; a dense layer with one neuron is added after all LSTM layers; the learning rate, batch size, and the number of epochs are respectively fixed at 0.0001, 1, and 100; For all the four stocks expect for Meta Platforms, a drop-out rate fixed at 0.2 is added after the first LSTM layer. For Meta Platforms, the drop-out rate is set as 0.1 for better prediction results.

2.3 Portfolio Optimization

$$Cov(r_i, r_j) = \frac{1}{T-1} {}_{t=1}^{T} \left(r_{i,t} - \mu_i\right)\left(r_{j,t} - \mu_j\right) \quad (7)$$

$$\Sigma = \begin{pmatrix} \mathrm{Var}(r_1) & \mathrm{Cov}(r_1, r_2) & \mathrm{Cov}(r_1, r_3) & \mathrm{Cov}(r_1, r_4) & \mathrm{Cov}(r_1, r_5) \\ \mathrm{Cov}(r_2, r_1) & \mathrm{Var}(r_2) & \mathrm{Cov}(r_2, r_3) & \mathrm{Cov}(r_2, r_4) & \mathrm{Cov}(r_2, r_5) \\ \mathrm{Cov}(r_3, r_1) & \mathrm{Cov}(r_3, r_2) & \mathrm{Var}(r_3) & \mathrm{Cov}(r_3, r_4) & \mathrm{Cov}(r_3, r_5) \\ \mathrm{Cov}(r_4, r_1) & \mathrm{Cov}(r_4, r_2) & \mathrm{Cov}(r_4, r_3) & \mathrm{Var}(r_4) & \mathrm{Cov}(r_4, r_5) \\ \mathrm{Cov}(r_5, r_1) & \mathrm{Cov}(r_5, r_2) & \mathrm{Cov}(r_5, r_3) & \mathrm{Cov}(r_5, r_4) & \mathrm{Var}(r_5) \end{pmatrix} \tag{8}$$

After obtaining the predicted weekly prices of five stocks on Aug 5th, 2024, the expected returns are calculated by taking the percentage change of the forecasted price and the latest historical price. In addition to the expected return vector (μ) containing the weekly return of five stocks, the covariance matrix (Σ) is another pre-requisite for deriving the weight vector (w) which allocates the portfolio to reach a maximized Sharpe-ratio. First, the covariance between each two of the five stocks can be calculated using equation (7), as followed by a construction of the covariance matrix as equation (8) demonstrates.

$$\text{Maximize } \frac{w^T\mu}{\sqrt{w^T\Sigma w}} \text{ subject to } {}_{i=1}^{n} w_i = 1 \tag{9}$$

$$\text{Minimize } w^T\Sigma \mathrm{w} \text{ subject to } {}_{i=1}^{n} w_i = 1 \tag{10}$$

After reaching the expected return vector as well as the covariance matrix, which are the two key components for portfolio optimization, the Monte-Carlo simulation is applied to randomly try for many weight assignments, among which two combinations of weights are chosen: one weight combination that obtains the highest Sharpe-ratio, and the other combination that reaches the lowest portfolio volatility. In the implementation, the number of trials is set to be 10,000. The mathematical expressions of searching for the maximized portfolio shape-ratio and the minimized portfolio volatility are shown as equation (8) and (9): the denominator in (8) represents the portfolio standard deviation, while the nominator calculates for the expected portfolio return. Also, the sum of weights must equal one, indicating no shorting in the desired portfolio.

3 Results and Discussion

3.1 The Performance of the Model

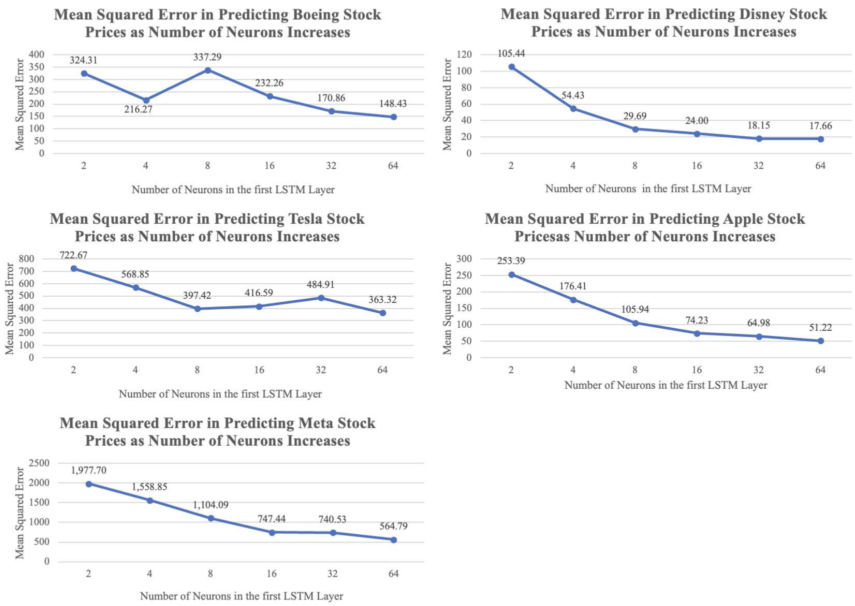

Fig. 2. Changes in MSE as the number of neurons increases from 2 to 64 in the first LSTM layer (Photo/Picture credit: Original).

Return Forecasts. From the five graphs in Fig. 2 above, it can be observed that the five stocks demonstrate two types of MSE trends as the number of neurons increases in the first LSTM layer. For Walt Disney Co., Apple Inc., and Meta Platforms Inc., the MSE all show a smooth decreasing trend as the number of neurons increases from 2 to 64. In contrast, though overall decreasing, the MSE in the prediction of Boeing Co. and Tesla Co. exhibit more fluctuations without displaying a distinct pattern between MSE and the number of neurons.

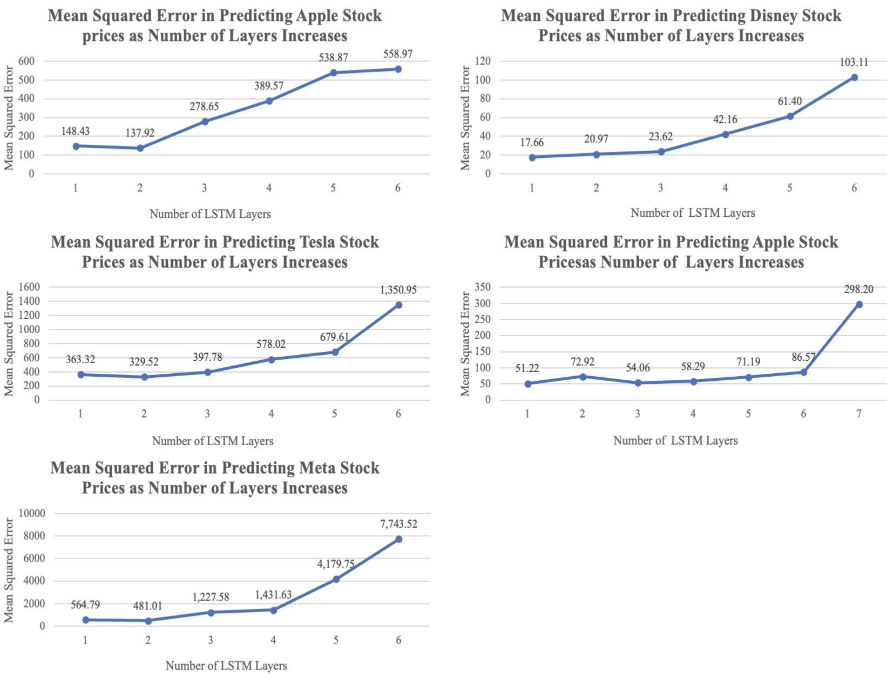

Fig. 3. Changes in MSE as the number of layers increases from 1 to 6 (Photo/Picture credit: Original).

Observing from Fig. 3, all five stocks exhibit similar trends regarding the patterns and the increased layers. Although minor fluctuations are present, the MSE for each stock consistently rises as the number of layers increases from one to six, making the fluctuations negligible in the broader context.

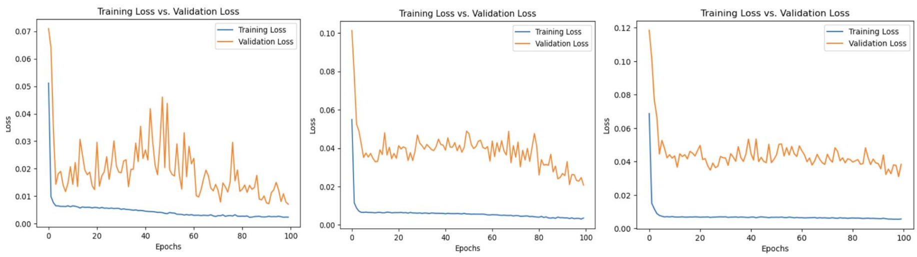

Fig. 4. Plots of Training and Validation Loss with 4, 5, and 6 LSTM layers over 100 trials for Meta (Photo/Picture credit: Original).

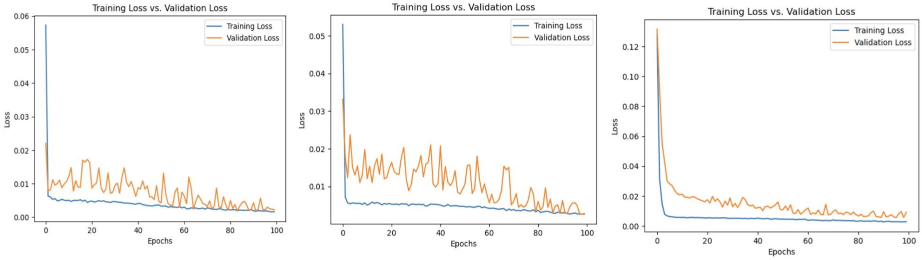

Fig. 5. Plots of Training and Validation Loss with 4, 5, and 6 LSTM layers over 100 trials for Apple (Photo/Picture credit: Original).

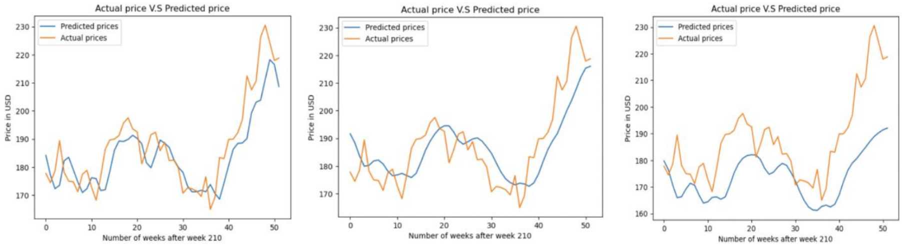

Fig. 6. Actual Prices V.S. Predicted Prices after the 210th week for Apple (Photo/Picture credit: Original).

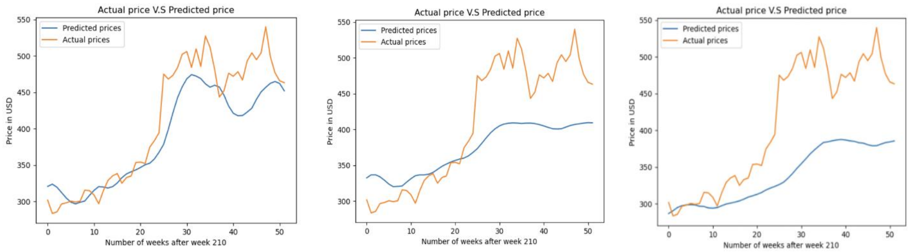

Fig. 7. Actual Prices V.S. Predicted Prices after the 210th week for Meta (Photo/Picture credit: Original).

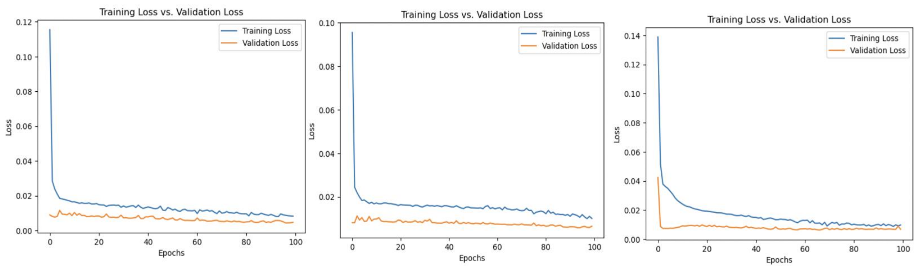

Fig. 8. Plots of Training and Validation Loss with 4, 5, and 6 LSTM layers over 100 trials for BA (Photo/Picture credit: Original).

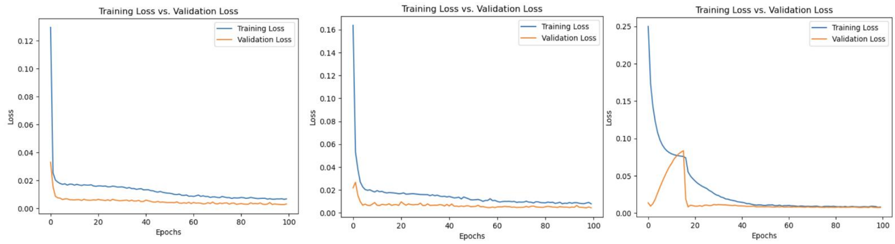

Fig. 9. Plots of Training and Validation Loss with 4, 5, and 6 LSTM layers over 100 trials for DIS (Photo/Picture credit: Original).

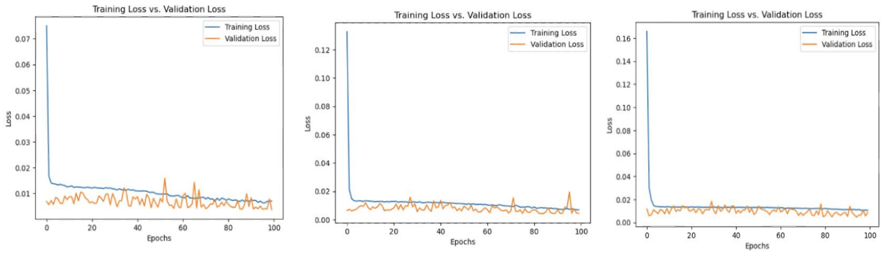

Fig. 10. Plots of Training and Validation Loss with 4, 5, and 6 LSTM layers over 100 trials for TSLA (Photo/Picture credit: Original).

Discussion. The training curves of based on various configurations are shown in Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9 and Fig. 10. One possible explanation for the decrease in MSE as the number of neurons increases is that, compared to simple models with a limited number of neurons, a more complicated model with more neurons can capture intricate patterns and non-linear dependencies implied by the historical inputs. However, as indicated by the constantly rising MSE when the number of layers increases, it can be inferred that an overly complicated LSTM model can result in worsened predictions as well. The model performances of Apple and Meta have demonstrated that complicated LSTM with too many layers is prone to overfitting. From their historical loss graphs (Fig. 5), it can be observed that the validation loss started growing larger and failing to converge to the training loss as the number of layers was added to around four, which became worse as the number increased to five and six. Meanwhile, by looking into the plot of their predicted prices and real prices (Fig. 6, Fig. 7), it can be observed that predictions gradually failed to move along with the actual trend as there were too many layers built into the model.

For the remaining three stocks, though the MSE kept surging as the number of layers increased, the validation loss was always below the training loss, indicating that the model was not overfitted. One potential explanation for this is that the increasing complexity of the LSTM model structure may bring extra difficulties in determining the optimal weight associated with each gate. This possibility is supported especially by the historical loss curves of Boeing and Tesla, where the validation loss stopped showing decreasing trends and started to display obvious fluctuations as the number of layers increased.

Regarding Boeing and Tesla, the unclear MSE patterns as neuron numbers increased can be potentially attributed to their volatile historical prices. From the graphs of historical weekly prices of the two stocks (Fig. 1), both curves exhibit numerous price spikes and are accompanied by substantial price fluctuations with a magnitude around 300, outweighing the fluctuations presented by the other three stocks.

3.2 Portfolio Optimization

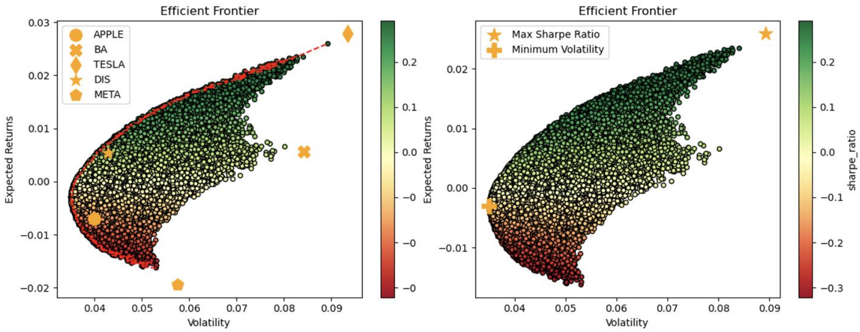

Fig. 11. Visualization of the Monte-Carlo Simulated Results (Photo/Picture credit: Original).

Table 1. Optimal Weights of Each Stock in the Two Simulated Portfolios

Stock Names	**Maximized Sharpe-Ratio**	**Minimized Volatility**
Apple Inc.	2.22%	48.87%
Boeing Inc.	1.20%	0.21%
Tesla Inc.	94.16%	0.38%
Walt Disney Co.	1.01%	40.87%
Mata Platforms Inc.	1.41%	9.66%
Expected Performance		
Expected Portfolio Return	2.60%	-0.30%
Expected Portfolio Volatility	8.93%	3.50%

Table 2. Actual and Predicted Weekly Return of 5 stocks from Jul 29th to Aug 5th

	Apple	**Boeing**	**Tesla**	**Walt Disney**	**Meta Platforms**
Predicted	-0.0071	0.0057	0.0279	0.0054	-0.0194
Actual	-0.0165	-0.0120	-0.0369	-0.0375	0.0607

Abs. Error	0.0094	0.0177	0.0648	0.0429	0.0801

Table 3. Actual Portfolio Returns under Monte-Carlo Simulated Weights

	Maximized Sharpe-Ratio	**Minimized Volatility**
Predicted	2.60%	-0.30%
Actual	-3.48%	-1.77%

Portfolio Results. Fig. 11 provides the visualization of the Monte-Carlo Simulated Results, The Table 1 displays the constructed portfolio with the corresponding weights of each stock according to the Monte-Carlo Simulation results, one for maximizing the portfolio Sharpe-ratio and the other one by simply minimizing portfolio volatility. From the results, the Sharpe-ratio maximizing portfolio suggests a heavy investment in Tesla, while the risk-minimizing portfolio suggests a relatively even distribution between Apple and Disney stocks.

Discussion on Simulated Results. To assess whether the portfolio allocation calculated from the Monte-Carlo Simulation is reliable in a practical context, the real weekly returns of the five stocks from Jul 29th to Aug 5th were paired with their corresponding weights for calculating the actual portfolio return. As shown by the Table 2, only the LSTM forecast on Apple's stock return shows the correct sign in alignment with the actual return, and hence also having the smallest absolute error. The remaining four predictions on returns, manually calculated from price predictions, all show opposite signs from the actual values, which would potentially render the portfolio allocation inaccurate. Now, seeing the Table 3, both simulated portfolios failed to attain a positive return. As mentioned before, the failure in constructing a winning portfolio can be mostly attributed to inaccurate forecasts on prices done by the LSTM models. One possible drawback of the LSTM models established in this research is the limited number of features employed. In this study, the historical stock closing price is the only feature that is considered when building the model, which means it is assumed that the future price movement solely depends on their past values. However, this assumption poses a limitation on price forecasts since the price movements are volatile and often influenced by multiple factors. Hence, one potential improvement to the current result is to include more features, such as the market factors, stock volumes, stock opening prices, and financial data of the firms into the LSTM model. It is also worth investigating combining the LSTM with other models, such as the factor model and ARIMA model, to reach a more reliable prediction.

4 Conclusion

This study aimed to investigate how hyperparameters, especially the number of neurons and the number of LSTM layers, affect the performance of LSTM models in making stock price forecasts of five selected stocks: Apple Inc., BA Inc., Tesla Inc., Walt Disney Co., and Meta Platforms Co. In addition, this paper applied the Monte-Carlo Simulation method in determining the optimized portfolio consisting of the five stocks based on the stock prices predicted by the LSTM models with hyperparameters that resulted in minimal mean squared errors (MSE). Specifically, the weekly historical prices from July 29th, 2019, to July 29th, 2024 were collected from which 80% were used for training the models and 20% were used for model validations. From the ultimate results in the forecasting section, interesting trends and possible explanations were observed and derived: The increase in the number of neurons augmented the model complexity and hence improved its capability to capture intricate non-linear patterns. Meanwhile, the overly complicated models with an increasing number of layers impair the model performance through the disturbance of model stability and over-fitting issues. Another finding is that the large volatilities in price fluctuations of historical data are possible reasons for the fluctuating relation between model performance and the number of neurons. Finally, due to the undesirable portfolio outcome, the LSTM models can be further advanced by incorporating other important factors, such as the market factors, the financial situation of firms, and the opening prices of stocks.

References

1. Morgan, J. N., Sonquist, J. A.: Problems in the analysis of survey data, and a proposal. Journal of the American Statistical Association 58(302), 415-434 (1963).
2. Hochreiter, S.: Long Short-term Memory. Neural Computation. MIT Press (1997).
3. Greff, K., Van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. arXiv preprint arXiv:2012.05208 (2020).
4. Sutskever, I.: Sequence to Sequence Learning with Neural Networks. arXiv preprint arXiv:1409.3215 (2014).
5. Zhao, Z., Chen, W., Wu, X., Chen, P. C., Liu, J.: LSTM network: a deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems 11(2), 68-75 (2017).
6. Kong, W., Dong, Z., Jia, Y., Hill, D., Xu, Y., Zhang, Y.: Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Transactions on Smart Grid 10(1), 841-851 (2017).
7. Moghar, A., Hamiche, M.: Stock market prediction using LSTM recurrent neural network. Procedia Computer Science 170, 1168-1173 (2020).
8. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Computation 31(7), 1235-1270 (2019).
9. Staudemeyer, R. C., Morris, E. R.: Understanding LSTM: A tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586 (2019).
10. Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena 404, 132306 (2020).