Predicting Stock Prices Using Dynamic LSTM Models

Duc Huu Dat Nguyen(✉), Loc Phuoc Tran, and Vu Nguyen

Faculty of Information Technology, University of Science,
Vietnam National University, Ho Chi Minh, Vietnam
{1512125,1512302,nvu}@fit.hcmus.edu.vn

Abstract. Predicting stock prices accurately is a key goal of investors in the stock market. Unfortunately, stock prices are constantly changing and affected by many factors, making the process of predicting them a challenging task. This paper describes a method to build models for predicting stock prices using long short-term memory network (LSTM). The LSTM-based model, which we call dynamic LSTM, is initially built and continuously retrained using newly augmented data to predict future stock prices. We evaluate the proposed method using data sets of four stocks. The results show that the proposed method outperforms others in predicting stock prices based on different performance metrics.

Keywords: LSTM · Stock price prediction · Dynamic models

1 Introduction

Predicting stock prices is one of the most complex financial problems because there are many surrounding factors that directly affect the price fluctuation of the stock market. On the one hand, several authors argue that future stock prices are impossible to predict. Malkiel and Fama show that all new information is reflected on the stock price without delay, and thus, future stock price movement is not dependent on past and present information [1]. On the other hand, technical analysts argue that it is possible to capture important information about stock growth or decline according to information gathered from the historical stock data. Hence, if moving trends of a stock for a period can be captured, its prices are predictable.

In addition, Kai et al. have shown that the evolution of the stock market is directly affected by many factors such as: general economic conditions, political events, corporate policies, commodity price index, bank rate, exchange rate, investor expectations, other stock market movements, and investor sentiment [2].

Different models have been explored to use past stock values including Moving Average (MA), Exponential Smoothing, Exponential Moving Average (EMA), Autoregressive Integrated Moving Average (ARIMA) [3], and Vector Autoregression (VA). These models are used to find signals for future values of the target stock [4].

Recently with the development of large data processing capabilities based on upgraded hardware, scientists have developed several stock prediction models using approaches such as Artificial Neural Networks (ANN), fuzzy logic, and Genetic

H. Florez et al. (Eds.): ICAI 2019, CCIS 1051, pp. 199–212, 2019.
https://doi.org/10.1007/978-3-030-32475-9_15

Algorithms (GA) [5]. One application of deep learning for stock prediction is the time-series prediction, which predicts the future value of a stock at a certain time. Prediction can be mainly classified as short-term (prediction for stock prices in seconds, minutes, and days ahead) and long-term (prediction for more than one year or beyond) [6].

In previous studies on ANN, indicators for stock prices are computed to capture temporal information or patterns and then used as input features for ANN [7–11]. Chen et al. investigated an LSTM-based model to predict stock prices on the Chinese stock market [12], suggesting that this model has the potential to predicting stock prices as it leads to improvements in stock prediction accuracy. Nelson et al. developed an LSTM-based method to predict stock prices and evaluated it with baseline methods such as random forest, multi-layer perceptron, and pseudo-random models [13]. They show that the LSTM-based model generates comparatively favorable predictions. Li et al. investigated the use of investor sentiment extracted forum posts as an input for a network along with historical market data to predict CSI300 (China Securities Index 300) and sentiment [14]. They show that the other model trained with numerical data and textual representations produced higher profits than did the model trained with only numerical data [15].

Several the recent development in the analysis of time-series involved the use of deep neural networks such as Convolution Neural Network (CNN), Recurrent Neural Network (RNN), and LSTM networks [16, 17]. Previous studies used LSTM-based models to predict stock prices [15, 18]. However, this prediction model, which we call *static LSTM*, is built using a fixed training data to predict test data, and the models are not retrained when new data is available. Thus, such models may not capture the most recent information existing in the new data to predict a stock's prices.

In this study, we proposed a method, which is called *dynamic LSTM*, to predict stock prices using LSTM by continuously augmenting the most recent data to the LSTM network to predict new prices. By this, the LSTM network is continuously updated with new actual stock prices to predict the future ones. We evaluate the method on the data set collected from the Apple (AAPL) stock prices during a 10-year period, General Electric (GA), China Petroleum & Chemical Corporation (SNP), and Facebook (FB) during a 5-year period. The results show that the proposed method outperforms others similar methods including the static LSTM method in predicting closing prices of the four stocks based on different performance metrics.

2 Long Short-Term Memory Network

LSTM was first introduced in 1997 to address several problems in previous networks such as the absence of notion of order of time in Feed Forward Neural Networks (FFNN) [19] and the *vanishing gradient* problem in RNN. This problem occurs when the gradient becomes smaller with each layer and turns out to be too small to have any effect in the deepest layers. The memory cell in LSTM allows to have a continuous gradient flow that helps address this vanishing gradient problem. A LSTM model can have one or many LSTM hidden layers. An LSTM network can be considered an enhanced version of RNN. RNN allows information to persist in the network by making use of feedback loop.

As shown in Fig. 1, in an LSTM network, current inputs and previously learned inputs are taken into consideration. An LSTM network consists of units called Memory cell unit or memory cell in the place of hidden layers. These cells have three gates including input gate, forget gate, output gate. These gates in an LSTM cell regulates the cell ability to add or remove information from cell state. Through memory cells and gates, an LSTM network can learn long-term dependencies. Recent studies investigated the use of LSTM networks to learn and capture temporal patterns for time-series analyses [20–22].

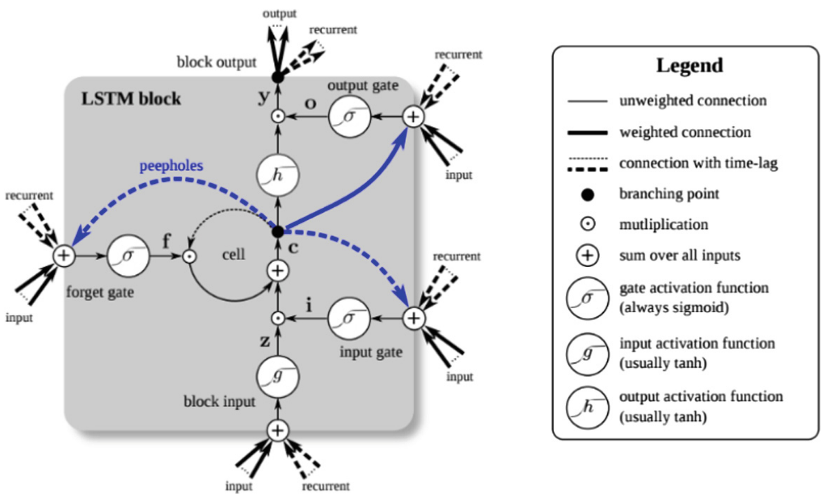


Fig. 1. An LSTM unit [23]

3 LSTM-Based Stock Price Prediction Methods

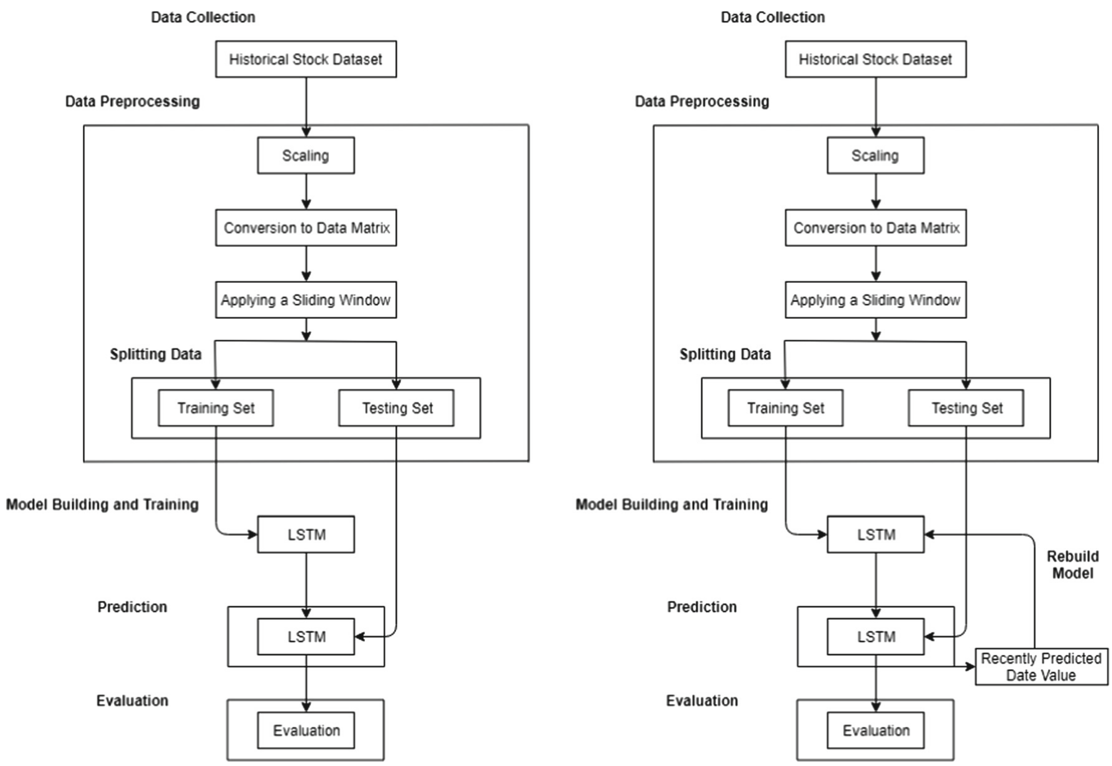
In this section, we describe the methods for constructing LSTM networks to predict stock prices. The methods consist of basic steps such as data collection and preprocessing, model building and training, and evaluation of prediction from the LSTM models.

The first part of our method is like the LSTM model for predicting stock prices proposed in [15]. The second part involves training the LSTM model with new actual stock prices. This part enables the model to capture the trend of the data in the closest time to the predicted time. Each of these steps is explained in these sections.

For the experiment purpose, closing prices of daily trading data are predicted. Flowcharts of the steps involved in predicting stock prices using LSTM models are shown in Fig. 2.

3.1 Model Flowcharts

Two methods are investigated in this study, the static and the dynamic models. The static model is not rebuilt when the stock price of the recently predicted date is available while the dynamic model is rebuilt using the stock price of the recently predicted date. The flowcharts of two models shown in Fig. 2 consist of three phases (data collection, data processing, and evaluation) with the dynamic model having an additional phase (rebuilding model).



(a) Static LSTM method (b) Dynamic LSTM method

Fig. 2. Flowcharts of the static and dynamic LSTM methods

- **Data Collection**: this phase is focused on retrieving and achieving data from sources such as Yahoo! Finance.
- **Data Preprocessing**: a linear transformation is applied to normalize the closing price and obtain the values between 0 and 1 for faster computation. The data is then converted into a matrix to which the sliding window approach is applied. The sliding window approach uses a window of *n* trading days of which the first *n-1* days are used as input, and the last day (the *n*th day in the window) is used as output for the LSTM network. The window is then moved forward one day, and the input and output for the network are determined accordingly.
- **Evaluation**: when the training step is completed, the resulted models are used to generate predictions for the test set. The models are then evaluated using four performance measures which are defined in Sect. 4.4.
- **Rebuilding Model**: the recently predicted date's stock price will be added to the training set to rebuild the model.

4 Experimental Design

4.1 Dataset Preparation and Preprocessing

The stock data is collected automatically from Yahoo! Finance [24]. For a given stock collected, each data point consists of the date collected, trade volume, opening, closing, high, low, and adjusted closing prices. In this study, the closing price is used for prediction, and the sliding window approach is used for training the neural networks [18].

We collect and use data from four stocks listed in the NASDAQ stock market [25], including Apple (stock symbol AAPL), General Electric (GE), China Petroleum & Chemical Corporation (SNP), and Facebook (FB). For AAPL, the dates collected range from 05/21/2009 to 05/20/2019 (about 10 years) or 2511 data points each presenting a working day. The data of Apple stock prices is represented in Fig. 3 with the x-axis showing the number of trading days and the y-axis showing the closing price. For GE, SNP, and FB, the stock data is collected from 05/21/2014 to 05/20/2019 (5 years).

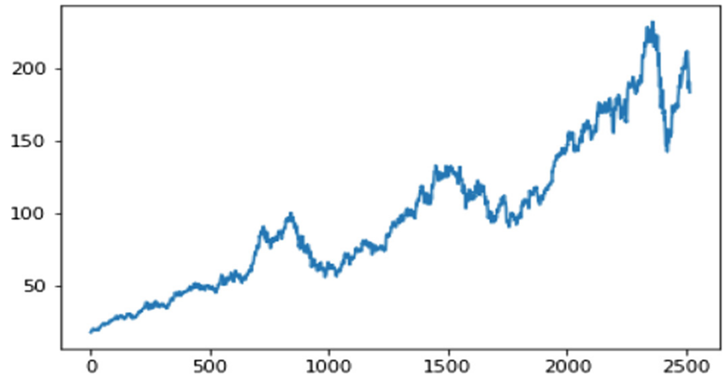


Fig. 3. Historical AAPL dataset

A linear transformation is applied to the stock price to normalize the value to the range between 0 and 1. The data is then converted into matrix for faster computation, and the sliding window approach is applied to the matrix.

Each stock dataset is split into 80% for training and 20% for testing. The training set is further divided into the training set (80%) and validation set (20%). After the preprocessing step, the sequential deep neural network models are developed and trained using the training data.

For the static LSTM method, the model is built once and is not updated with recently available stock data (see Fig. 2(a)). This model is used to predict stock prices for all days in the testing set.

For the dynamic LSTM method, the actual closing price of the recently predicted date is added to the training dataset to rebuild the model to make predictions for data in the testing set.

4.2 Framework and Hardware

In our experiments, we use Kensas and TensorFlow for implementing the LSTM network. Kensa is a high-level neural network API. Tensorflow is an open-source machine learning framework on which Keras is based. We use Python version 3.6.4, TensorFlow version 1.12.0, and Keras version 2.2.4 [26]. The PC platform used for training is Intel Core i7 8700 with RAM 16.00 GB, GPU NVIDIA Quadro P2000 with 5 GB VRAM. The models are trained for 7 days, about 158 h and 43 min, in normal conditions without interruption.

4.3 Training

For finding the best results in predicting stock prices, we decided to conduct training with different conditions and adjustments:

- Epoch ranging from 10–180 for the static LSTM model and 10–30 for the dynamic LSTM model.
- Experimenting 4 time periods to build training and testing models with the Apple stock: 10-year (05/21/2009 to 05/21/2019), 5-year (05/21/2014 to 05/20/2019), 2-year (05/21/2017 to 05/20/2019), 1-year (05/21/2018 to 05/20/2019) (all cases are divided into 80% training set and 20% testing set).
- Sliding window ranging from 5–30.

In building the LSTM model, we use hyper-parameters for layers of LSTM network as follows:

- Size of the cell state: 256
- Dense: 1
- Optimizer: 'ADAM' [27]

ADAM is an algorithm which is used to update the network weights during training which for different parameters an adaptive learning rates are computed. To prevent over-fitting in the neural network, a regularization technique known as dropout is used with the dropout rate representing the percentage of nodes dropped for each iteration.

4.4 Performance Measures

In this paper, we use the following measures including Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and relative Root Mean Squared Error (rRMSE) to evaluate the performance of prediction models. These measures are often used in the evaluation of stock price prediction [10]. These performance measures are computed as:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_t - x_t|}{|y_t|} \times 100 \tag{1}$$

$$\mathrm{MAE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}\frac{|\mathrm{y_t} - \mathrm{x_t}|}{|\mathrm{y_t}|} \tag{2}$$

$$\mathrm{rRMSE} = \sqrt{\frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}\left(\frac{\mathrm{y_t} - \mathrm{x_t}}{\mathrm{y_t}}\right)^2} \tag{3}$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - x_t)^2 \tag{4}$$

Where y_t is the actual value at time t, x_t is the predicted value at time t, n is the number of days predicted.

5 Experimental Results

This section represents the implementation details, observations, results obtained from the experiment.

5.1 Performance of the Dynamic Model with Different Time Periods

This provides the results from our experiment using the dynamic model with different periods. We report performance measures in MAE, MAPE, rRMSE, and MSE. We train and test the model on four periods of data, 10-year (05/21/2009 to 05/21/2019), 5-year (05/21/2014 to 05/20/2019), 2-year (05/21/2017 to 05/20/2019), 1-year (05/21/2018 to 05/20/2019). The model is trained with the number of epochs increasing from 10 to 30.

Table 1 describes the prediction accuracy based on MAE, MAPE, rRMSE, and MSE obtained from the dynamic model for four periods. The first column shows the statistics, and the second to the fifth column show the values for four performance measures.

Table 1. Prediction performance of the static model

Statistics	MAE	MAPE	rRMSE	MSE
Max	0.0212	2.1201	0.0299	29.5886
Min	0.0144	1.4429	0.0212	14.9390
Mean	0.0169	1.6905	0.0242	19.7096
Median	0.0167	1.6642	0.0238	19.0654

Figure 4 depicts prediction accuracy based on MAPE, MAE, rRMSE, and MSE from the static model on 5-year period of the AAPL stock data for different epoch values and sliding window sizes. The x-axis represents the values of four performance measures, and y-axis represents the sliding window size (from 5 to 30).

The prediction accuracy from the model trained with the AAPL stock's 5-year period is more stable than those with other time periods using different epochs and sliding window sizes. As shown in Fig. 4, the dynamic model using the epoch of 30 generally produces better prediction performance than using other epoch values on the AAPL stock data. The window size of 5 days also results in the lowest error. Thus, we will use 5-year period, the epoch of 30, and the sliding window size of 5 days for further analysis.

Fig. 4. Performance of the dynamic model across epochs and window sizes

5.2 Comparison Between the Static and Dynamic LSTM Models

The static and dynamic LSTM models are both built using the AAPL stock data for 5-year period from 05/21/2014 to 05/20/2019 and the epoch of 30. The results from these models are presented in Table 2 with the last two columns showing prediction accuracy in terms of MAPE and MSE. The dynamic model improved prediction accuracy significantly over the static model, reducing mean MAPE by 0.82 or 31.2% and mean MSE by 19.41 or 45.9%.

Table 2. Performance of the static and dynamic models

Model	Statistics	MAPE	MSE
Static	Min	1.77	21.57
	Max	4.02	97.64
	Mean	2.63	42.25
	Median	2.51	34.86
Dynamic	Min	1.55	16.72
	Max	2.12	29.59
	Mean	1.81	22.84
	Median	1.82	22.66

Average MAPE and MSE values across window sizes obtained by the models are depicted in Fig. 5, using the 5-year AAPL stock data and the epoch of 30. MAPE and MSE values produced by the dynamic model are much lower than those by the static model across almost window sizes. The dynamic model tends to be less dependent on window sizes than does the static model with the MAPE and MSE results from the latter fluctuating significantly across window sizes. MAPE ranges from 1.55 to 2.12 and MSE from 16.73 to 29.59 for the dynamic model while MAPE ranges from 1.77 to 4.02 and MSE from 21.57 to 97.64 for the static model.

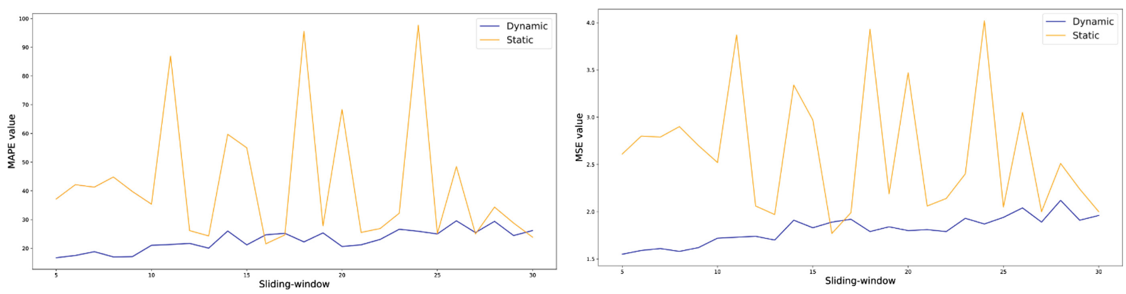


Fig. 5. MAPE and MSE from the static and dynamic models across sliding window sizes

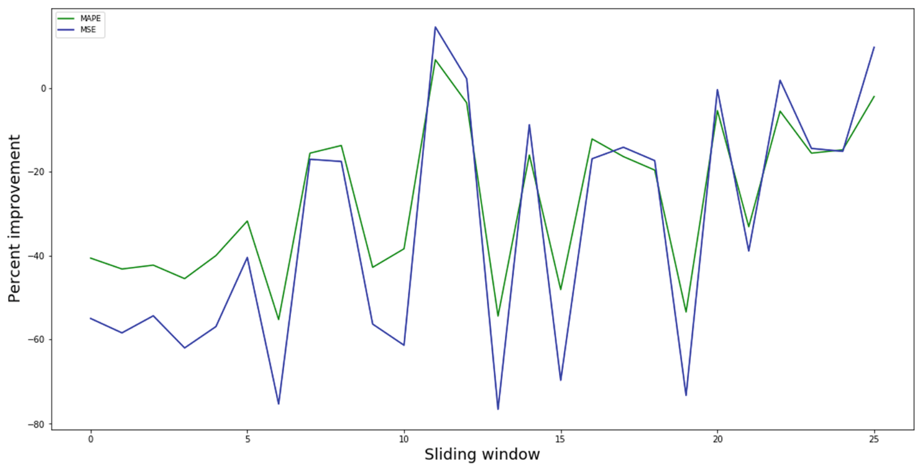
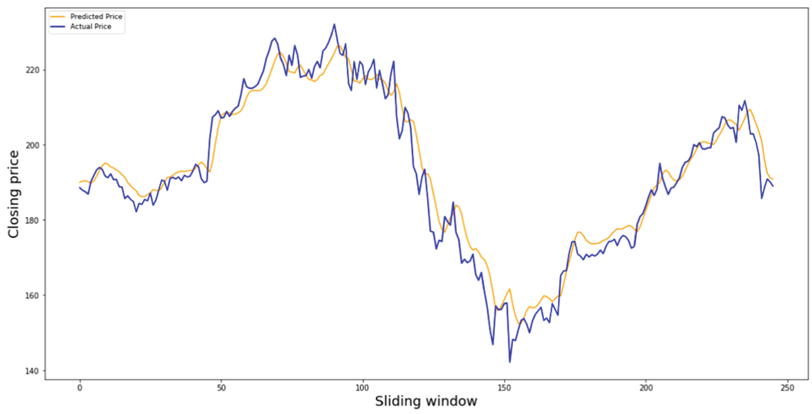


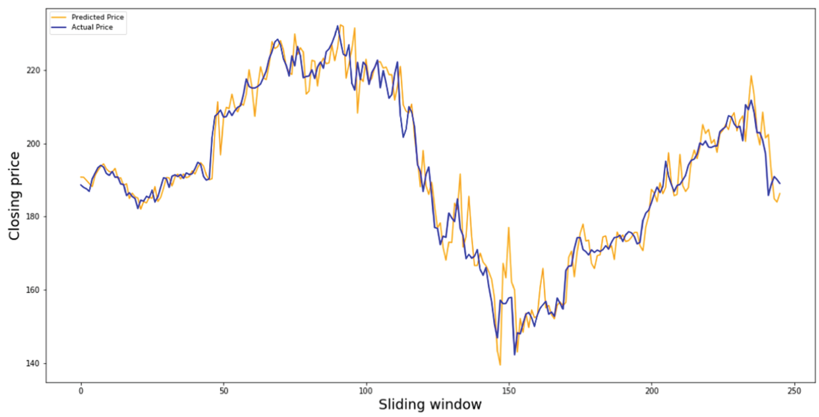
Fig. 6. Percent improvement in MAPE and MSE by the dynamic model over the static model, using the epoch of 30.

Figure 6 depicts percent improvements in MAPE and MSE across window sizes of the dynamic model over the static model using 5 year AAPL stock data and the epoch of 30. MAPE for the dynamic model decreases up to 55% while MSE for this model decreases up to 78% over the static model. These results show that the improvements in MSE are larger than those in MAPE by the dynamic model over the static model.

Actual and predicted closing prices of the AAPL stock are shown in Fig. 7 with those by the static model depicted in Fig. 7(a) and those by the dynamic model depicted in Fig. 7(b). The predicted closing prices by the dynamic model are closer to the actual prices than those by the static model. The predicted closing prices by the static model seem to fluctuate more significantly than those by the dynamic model. This observation is clearly reflected in MAPE and MSE results shown in Fig. 5.



(a) Static model



(b) Dynamic model

Fig. 7. Actual and predicted closing prices by the models for the AAPL stock (30 epochs and window size of 5 days)

5.3 Comparison Between the Dynamic Model and Linear Methods of Prediction

We choose two linear methods including Standard Averaging (SA) with the sliding window of 5 days and Exponential Moving Average (EMA) with the decay of 0.5 to compare with the dynamic model. SA and EMA are computed using the following formulas:

$$\text{SA:}\ x_n = \frac{1}{w}\sum_{n-w}^{n-1} x_i \tag{5}$$

$$\text{EMA}:\ x_n = x_{n-1} \times (1-d) + d \times EMA_{n-2} \tag{6}$$

Where x_n is the predicted value at the nth time, w is the size of sliding window in trading days, and d is the decay value.

Table 3. Performance measures obtained by three methods

Method	MAPE	MAE	rRMSE	MSE
SA-5	1.7984	0.0240	0.0395	21.3490
EMA-0.5	1.6424	0.0274	0.0305	17.7174
Dynamic LSTM model (30 epochs, sliding window size of 5)	1.5519	0.0155	0.0222	16.7154

Table 3 shows the prediction performance based on MAPE, MAE, rRMSE, and MSE of three methods using four-year AAPL stock prices for training and the following year for testing. The results for the dynamic LSTM model are the mean values obtained from using 30 epochs and the sliding window of 5 days.

As shown in Table 3, the dynamic model produces better predictions based on four performance measures than SA-5 and EMA-0.5. This model reduces MAPE, MAE, rRMSE, and MSE by 13.7%, 35.4%, 43.8%, and 21.7% over SA-5, respectively. Similarly, the model also reduces these performance measures by between 5.5% and 43.4% over EMA-05.

5.4 Evaluating the Methods Using GE, SNP, and FB Stock Prices

In this analysis, we run four models on GE, SNP, and FB stock data instead of APPL. The closing prices of these stocks are for 5-year period from 05/21/2014 to 05/20/2019.

The results from this analysis are shown in Table 4. The first column shows three stocks, the second columns shows for models, and the remain columns show the values for four performance measures.

Table 4. Prediction performance measures of the models on three stocks

Stock	Model	MAPE	MAE	rRMSE	MSE
GE	Dynamic model	**2.5722**	**0.0257**	**0.0359**	**0.2051**
	Static model	3.6577	0.0366	0.0474	0.2114
	SA-5	2.6805	0.0288	0.0415	0.2078
	EMA-0.5	2.6107	0.0268	0.0392	0.2062
SNP	Dynamic model	**1.6186**	**0.0162**	**0.0215**	**3.4050**
	Static model	1.6872	0.0169	0.0219	3.5745
	SA-5	1.7904	0.0182	0.0331	5.6721
	EMA-0.5	1.6748	0.0177	0.0308	4.2368
FB	Dynamic model	**1.9031**	**0.0190**	**0.0275**	**20.9702**
	Static model	1.9471	0.0195	0.0302	25.0147
	SA-5	2.9013	0.0278	0.0358	28.2258
	EMA-0.5	2.7281	0.0215	0.0311	25.3692

Across the stocks, the dynamic model consistently outperforms the other models based on all four performance measures. This model improves between 2% to 30% of MAPE, MAE, rRMSE, and MSE over the static model on three stocks. The reductions in prediction error by the dynamic are even higher when comparing to SA-5 and EMA-0.5. The static model outperforms SA-5 and EMA-0.5 when predicting SNP and FB, but it performs worse than these model on the GE stock. This result shows that this model is not consistent across stocks.

6 Conclusions

The proposed work involves the use of the dynamic LSTM model by retraining the model using newly added data for short-term prediction of stock prices. We comparatively evaluated this approach with the static LSTM model that is not retrained throughout the prediction process. The dataset of daily closing prices of four stocks including AAPL, GE, SNP, and FB was used for analysis.

The results show that stock prediction accuracy based on MAE, MAPE, rRMSE, and MSE obtained by the dynamic LSTM model is much better than that by the static LSTM model across four stocks investigated. The dynamic model also consistently outperforms the linear models SA-5 and EMA-0.5 when predicting four stocks. This model improves prediction accuracy by 45.9% on average based on MSE and 31.2% based on MAPE over the static model when predicting the AAPL stock.

Unlike the static model which is not designed to take advantage of the temporal information, the dynamic LSTM model takes into account both spatial and temporal information of a stock to predict its prices. This is a possible explanation for the advantage of this model in terms of prediction accuracy over the other models investigated in this study.

This study offers evidence that the LSTM network designed to incorporate temporal information has the potential for stock price prediction. Updating the LSTM network continuously with recently available data is a relevant approach to incorporating temporal information for the network.

As a future research direction, we plan to improve the dynamic LSTM model by using more stock-related factors and indicators such as simple moving average, momentum, relative strength index, and volume [28]). We are also interested in making predictions for longer time such one to ten days ahead of time.

References

1. Malkiel, B.G., Fama, E.F.: Efficient capital markets: a review of theory and empirical work. J. Finan. **25**(2), 383–417 (1970)
2. Miao, K., Chen, F., Zhao, Z.-G.: Stock price forecast based on bacterial colony RBF neural network. J. Qingdao Univ. (Nat. Sci. Ed.) **2**, 011 (2007)
3. Bollerslev, T.: Generalized autoregressive conditional heteroskedasticity. J. Econ. **31**, 307–327 (1986)
4. Box, G.E., Jenkins, G., Reinsel, G., Ljung, G.: Time Series Analysis: Forecasting and Control. John Wiley & Sons, Hoboken (2015)
5. Hadavandi, E., Ghanbari, A., Abbasian-Naghneh, S.: Developing an evolutionary neural network model for stock index forecasting. In: Huang, D.-S., McGinnity, M., Heutte, L., Zhang, X.-P. (eds.) ICIC 2010. CCIS, vol. 93, pp. 407–415. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14831-6_54
6. Jain, S., Dr. Roopam, G., Dr. Asmita A.M., Stock price prediction on daily stock data. Int. J. Neural Netw. Adv. Appl., 5 (2018)
7. Göçken, M., Özçalıcı, M., Boru, A., Dosdoğru, A.T.: Integrating metaheuristics and artificial neural networks for improved stock price prediction. Expert Syst. Appl. **44**, 320–331 (2016)
8. Kim, K.-J., Han, I.: Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Syst. Appl. **19**, 125–132 (2000)
9. Lendasse, A., de Bodt, E., Wertz, V., Verleysen, M.: Non-linear financial time series forecasting - application to the Bel 20 stock market index. Eur. J. Econ. Soc. Syst. **14**(1), 81–91 (2000)
10. Patel, J., Shah, S., Thakkar, P., Kotecha, K.: Predicting stock market index using fusion of machine learning techniques. Expert Syst. Appl. **42**, 2162–2172 (2014)
11. Ticknor, J.L.: A bayesian regularized artificial neural network for stock market forecasting. Expert Syst. Appl. **40**, 5501–5506 (2013)
12. Chen, K., Zhou, Y., Dai, F.: 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)
13. Nelson, D.M.Q., Pereira, A.C.M., Oliveira, R.A.d.: Stock market's price movement prediction with LSTM neural networks. In: 2017 International Joint Conference on Neural Networks (IJCNN) (2017)
14. Chiang, W.-C., Enke, D., Wu, T., Wang, R.: An adaptive stock index trading decision support system. Expert Syst. Appl. **59**, 195–207 (2016)
15. Baek, Y., Kim, H.Y.: ModAugNet: a new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. Expert Syst. Appl. **113**, 457–480 (2018)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)

17. LeCunn, Y., Bengio, Y.: Convolutional networks for images, speech, and time-series, The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge (1995)
18. Selvin, S.: Stock price prediction using LSTM, RNN and CNN-sliding window model, In: International Conference on Advances in Computing, Communications and Informatics (ICACCI) (2017)
19. Svozil, D., Kvasnicka, V., Pospichal, J.: Introduction to multi-layer feed-forward neural networks. Chemom. Intell. Lab. Syst. **39**, 43–62 (1997)
20. Lipton, Z.C., Kale, D.C., Elkan, C., Wetzel, R.: Learning to diagnose with LSTM recurrent neural networks, In: ICLR 2016 (2016)
21. Sak, H., Senior, A., Beaufays, F.: Long short-term memory recurrent neural network architectures for large scale acoustic modeling, In: INTERSPEECH 2014 (2014)
22. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling, In: INTERSPEECH 2012 (2012)
23. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: a search space odyssey. IEEE Trans. Neural Netw. Learn. Syst. **28**, 2222–2232 (2017)
24. Finance, Y.: Yahoo! Finance. https://finance.yahoo.com/
25. Nasdaq, “Nasdaq”. https://www.nasdaq.com/
26. Keras, “Keras”. https://pypi.org/project/Keras/
27. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: The 3rd International Conference for Learning Representations, San Diego (2014)
28. Kara, Y., Acar Boyacioglu, M., Baykan, Ö.K.: Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange. Expert syst. Appl. **38**, 5311–5319 (2011)