Received: 26 December 2020 | Revised: 25 April 2021 | Accepted: 7 May 2021

DOI: 10.1002/cpe.6437

SPECIAL ISSUE PAPER

WILEY

An effective way for Taiwanese stock price prediction: Boosting the performance with machine learning techniques

Hai T. Nguyen[1] | Toan B. Tran[2,3] | Phuong H. D. Bui[1]

[1]College of Information and Communication Technology, Can Tho University, Can Tho city, 900000, Vietnam

[2]Center of Software Engineering, Duy Tan University, Da Nang, 550000, Vietnam

[3]Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam

Correspondence
Hai T. Nguyen, College of Information and Communication Technology, Can Tho University, Campus II, 3/2 Street, Ninh Kieu District, Can Tho City, Vietnam.
Email: nthai.cit@ctu.edu.vn

Funding information
Vingroup Big Data Institute (VINBIGDATA), code VINIF.2020.ThS63; Vingroup Joint Stock Company and supported by the Domestic Master/Ph.D. Scholarship Programme of Vingroup Innovation Foundation (VINIF)

Abstract

The stock forecast is one of the most challenging tasks that have attracted numerous economists and scientists worldwide. Stock prices can be affected by many reasons, such as physiological, rational, and irrational behavior. Such factors can combine to make the prices volatile and very challenge to predict with great accuracy for a long time in numerous cases. In this study, we have deployed a long short-term memory architecture with various time-steps and classic machine learning methods such as random forests, support vector machines, and autoregression on the Taiwanese stock market collected in 14–15 years. As shown from the visual results, the predicted values have followed the patterns as the actual prices with low error rates in various metrics, including root mean square error and mean absolute error. This work is expected to provide a valuable tool for investigating stock price patterns of stock markets in the future.

KEYWORDS

long short-term memory, machine learning, stock forecast, Taiwanese stock, time-step

1 | INTRODUCTION

Stock markets hold an important and crucial role for the economies. Any changes in stock markets can significantly affect the economies and investors. Moreover, stock markets are an essential channel of the economy and a potential investment channel for the public. The stock market is the place to create instruments with high liquidity, accumulate, focus on capital distribution, and change the term of capital according to the requirements of economic development. Thanks to the stock market, the Government can mobilize financial resources without being under pressure from inflation, especially when investment capital from the state sector is still limited.

In [*], the authors stated that the stock market is considered as a "barometer" of an economy, it can accurately reflect the prospects for the coming period. According to experts, looking at the stock market will reveal the perspectives ahead of the changes in the economy in half a year. In particular, a rise in stock prices indicates a developing economy, and, conversely, a decline in stock prices is a pessimistic forecast of an economy's prospects shortly.

For the investors, the stock market provides them with a wealth of investment products, helping to diversify portfolios and minimize risks. These types of securities differ in their nature, maturity, and risk. Therefore, they allow investors to choose commodities that suit their capabilities, goals, and preferences. For example, if investors prefer facing risks, expect high-interest rates, choose to buy stocks, and vice versa, investors looking for safety, accepting low-interest rates will tend to choose Government bonds. The stock market helps businesses diversify forms of capital mobilization by issuing shares or bonds, helping businesses avoid bank loans with high-interest rates. Firms whose securities are listed on the stock market are certainly more reputable to the public, and securities are also more liquid. As a result, businesses can mobilize capital resources cheaper, use capital

[*]Why is stock exchange called a barometer of the economic and business conditions in a country? https://www.owlgen.in/why-is-stock-exchange-called-a-barometer-of-the-economic-and-business-conditions-in-a-country/, accessed on 20 November 2020

economically, flexibly, and effectively. Clearly, the stock market increasingly shows its role in the development of the economy in each country. It is a place to help the Government and businesses attract large and long-term capital flows to the national economy, and at the same time, help the public have more investment opportunities.

The most well-known analytical and forecasting methods for the stock market are technical analysis and fundamental analysis. In addition, the method of analyzing and forecasting quantitatively through mathematical models is gradually interested. Quantitative forecasts are used in a reasonably popular way around the world. Many hedge funds have set up automated trading systems using quantitative trading. The effectiveness of this approach has been proven in many markets. The advantage of the quantitative forecasting method is that the signals are pretty objective, based on the criteria of the statistical indicators from the model. Trading signals are given based on objective analysis to minimize errors caused by human factors. However, if you overuse this method, it will also create harmful consequences. The quantitative methods used for forecasting are based on mathematical models assuming that relationships between factors established in the past will repeat in the future. In other words, the quantitative method is based on past data to detect future movement trends of the elements according to a particular law. To predict the future evolution, one can use a time series model or use causal variables. Time series-based prediction models forecast the future value of a particular variable by analyzing past and present data of those variables. Time series prediction assumes that the future of the predictor variable will keep the movement trend in the past and the present. Thus, only stable chains can provide reliable forecasts.

The stock can accurately reflect the economic outlook for the coming period. According to numerous analyses, the stock market can be ahead of the changes of the economy for several months. Specifically, a rise in stock prices will indicate a developing economy, whereas a decline in stock prices is a negative forecast for future economic prospects. Besides, we are witnessing the rapid development of information technology in industries with breakthroughs from machine learning tools. However, using machine learning tools in specific cases will require much investigation to make reasonable algorithm choices. It is crucial to develop a framework to efficiently do stock forecast tasks by leveraging information technologies' advancements. This study considers long-term memory architectures for forecasting the Taiwanese stock market with configuration on time-steps. We also evaluate the efficiency of machine learning techniques on the Taiwanese stock forecast problem. With the random forest (RF) model, we focus on the number of trees and the number of max-leaf nodes, which play an essential role in building trees in the forest. As observed from the results, the number of max-leaf nodes is more important than the number of trees. Furthermore, with the support vector regression (SVR), we investigate the performance with three different kernel functions. Among the considered kernels, RBF achieves the best performance. Long short-term memory (LSTM) seems like be the most efficient in comparison with considered classic machine learning algorithms. On the proposed dataset or the external dataset, the LSTM still achieves the highest performance. Our contributions include multifold:

- We present in this work machine learning-based approaches for stock prices forecast in Taiwan. The considered classical approaches include robust algorithms such as support vector machines (SVM) and RF, autoregression algorithm. Recent well-known deep learning algorithms such as LSTM are also leveraged for the time series problem. Some parameters are modified in SVM, RF, and LSTM to improve the performance of the work in Reference 1.
- We evaluate the effects of various parameters of RF on prediction performance. As shown from the obtained results, the number of full leaf nodes can be more significant than the number of trees for the Taiwanese stock price forecast problem.
- We also investigate the time-step of LSTM to evaluation the effect of this parameter on the performance. The obtained predictions, as revealed in the experiments are followed the actual values patterns.

In the rest of this study, the related studies are introduced in Section 2. We present the information on considered datasets in Section 4.1. The learning architectures and configurations are described in Section 3. Our experimental results are presented in mean absolute error (MAE) and root mean square error (RMSE) in Section 5. Section 6 conducts some closing remarks for the work.

2 | RELATED WORK

With its significant impact on economics, stock forecast research has attracted the attention of investors and researchers with numerous forecast models proposed.

Taiwan stock exchange (TAIEX) has been investigated in numerous studies. The authors in Reference 2 used to partition the universe of discourse, and an autoregression high-order fuzzy time series (FTS) model utilizes historical data. Another work in Chen and Chang[3] introduced a method for forecasting the TAIEX by utilizing the multivariable fuzzy method based on the fuzzy clustering method and fuzzy rule interpolation techniques. A recent review paper was introduced in Reference 4 to provide exciting work related to FTS forecasting models between 1993 and 2017, focusing on developing state-of-the-art. The authors in Reference 5 also presented a FTS forecasting method to forecast the TAIEX based on fuzzy variation groups and achieved better results in comparison to their previous work. Moreover, Chen and Jian[6] presented a method for forecasting the TAIEX based on two factors, namely, second-order fuzzy-trend logical relationship groups and the probabilities of the down-trend, the probabilities of the

equal-trend and the probabilities of the up-trend of the two-factors second-order fuzzy logical relationships. Chen and Kao[7] leveraged the particle swarm optimization (PSO) techniques and SVM to forecast the TAIEX based on FTS. To obtain optimal intervals in the universe of discourse, the authors utilized the PSO techniques and classified the training dataset by the SVM. In Reference 8, Chen et al. proposed a similar method to Reference 7 by using the PSO techniques to get the optimal weighting vector of each group of the fuzzy-trend logical relationship groups. The authors also forecast the TAIEX based on the second-order fuzzy-trend logical relationship groups. Chen and Phuong[9] presented a method for forecasting the TAIEX based on optimal weighting vectors of two-factors second-order fuzzy-trend logical relationship groups and optimal partitions of intervals in the universe of discourse. The authors also applied the PSO techniques to simultaneously obtain the optimal partitions of intervals and optimal weighting vectors. Huarng et al.[10] presented a method to forecast the TAIEX by using a multivariate heuristic model integrating with univariate fuzzy time-series models. Yu[11] presented a method to forecast the TAIEX with the weighted fuzzy time-series method. In Reference 12, the authors added to the existing body of knowledge related to FTS. They developed a new FTS model based on a single-valued neutrosophic hesitant fuzzy set and evaluated it on three real datasets.

Machine learning algorithms have been used widely for stock forecasts. As in Reference 13, authors deployed a vast amount of machine learning such as SVM and RF, and Naïve–Bayes on the Taiwan stock price movement. LSTM is among the most popular because it seems to be suitable for time series problems. It is increasingly in a vast of work for stock prediction. The work in Reference 14 proposed to merge membership and non-membership values by a minimum operator into the inputs of the LSTM that can be called a high-order Intuitionistic fuzzy time series model. The proposed method was evaluated on Giresun Temperature data and the Nikkei 225 stock exchange time series. Other interesting works using LSTM can be found at References 15-21. However, no studies evaluate the performance with the influence of time-steps in LTSM for the stock forecast. Our work investigated the performance of the LSTM on the Taiwan stock dataset with various time-steps. The results present time-step play a key role in predicting stock price with LSTM. Furthermore, the TAIEX dataset is the main stock index in Taiwan.

3 | STOCK PRICE PREDICTION USING MACHINE LEARNING

This section introduces the proposed approach based on the RF, SVR, and LSTM with time-step models for stock market prediction. We systematically present the related technologies in Sections 3.1, 3.2, and 3.4.

3.1 | The RF regression model

RF is a popular machine learning model that is commonly used for classification or regression tasks. RF is an ensemble of decision trees constructed in a specific random way form. Each tree is established and makes its prediction. Then, these predictions are averaged to construct a final result. However, the range of predictions a RF can make is limited by the highest and lowest labels in the training data due to the covariate shift and the extrapolation problems. In this study, we implement RF regression with the tree quality estimation by mean squared error (MSE) which is equal to variance reduction as the feature selection criterion. MSE is the square of RMSE. Assume that we have K trees, the prediction by the individual trees, $\hat{y}_k$, and $\hat{y}$ is the final prediction computing subsequently averaged over all the trees in the RF, as indicated in Equation (1). We also present the construction of a RF in Figure 1.

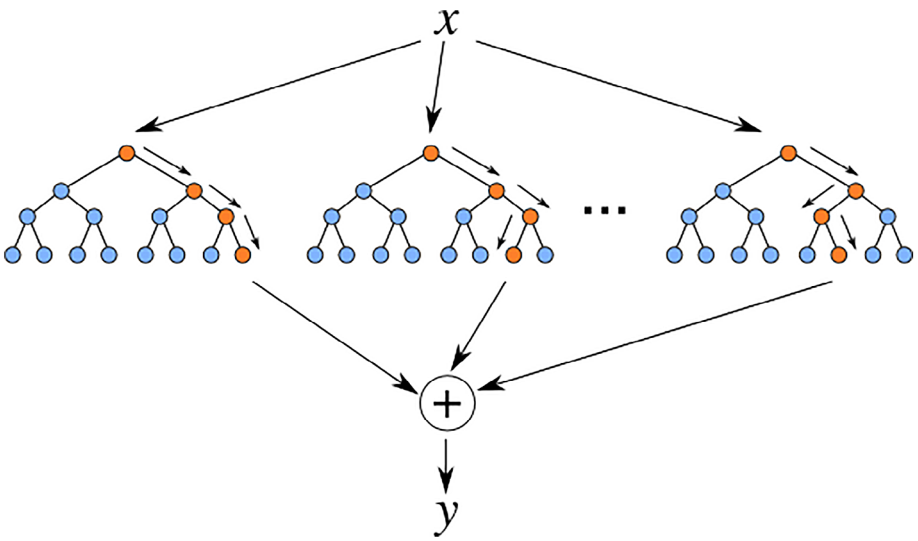

FIGURE 1 The graphical representation of the random forest[22,23]

Furthermore, we also investigated the effectiveness of the various number of *K* tree and max-leaf nodes, which contributes significantly to the performance of the RF model. The max-leaf nodes in each tree play a key role in growing the tree in the best-first fashion resulting in a relative reduction in impurity, whereas the number of *K* denotes the number of trees within a RF before making a prediction. Generally, the higher the number of *K*, the better performance, but it is more computationally expensive.

$$\hat{y} = \frac{\sum_{k=1}^{K} \hat{y}_k}{K} \tag{1}$$

3.2 | The SVR model

SVR is an extension of the well-known SVM algorithm applied in multivariate regression problems. SVR is flexible in defining the acceptable error in the model and finding an appropriate hyperplane to fit the data. The approximating function is a linear combination of kernel and nonlinear basis functions. The SVR uses the same principles as the SVM for classification tasks, with several minor differences. The approximating function is a linear combination of kernel and nonlinear basis functions. The SVR uses the same principles as the SVM for classification tasks, with several minor differences. However, the main purpose is to minimize error, individualizing the hyperplane, which maximizes the margin, keeping in mind that part of the error is tolerated. The linear, polynomial, and radial basis function kernel functions are dissimilar in making the hyperplane decision boundary. The main purpose of the kernel functions is to map the original dataset into a higher dimensional space. The radial basis function is more time-consuming but more accurate than the linear and polynomial kernels. The example of using different kernel functions is presented in Figure 2.[24] In this study, we implemented the SVR model with the default regularization parameter of 1, the gamma is 2, and the radial basis function kernel. The radial basis function kernel on two samples *x* and *z*, represented as feature vectors, is defined as Equation (2).

$$k(x, z) = \exp(-\gamma ||x - z||^2) \tag{2}$$

Where

- γ is a parameter that sets the spread of the kernel.
- $||x - z||^2$ may be identified as the squared Euclidean distance between the two feature vectors *x* and *z*.

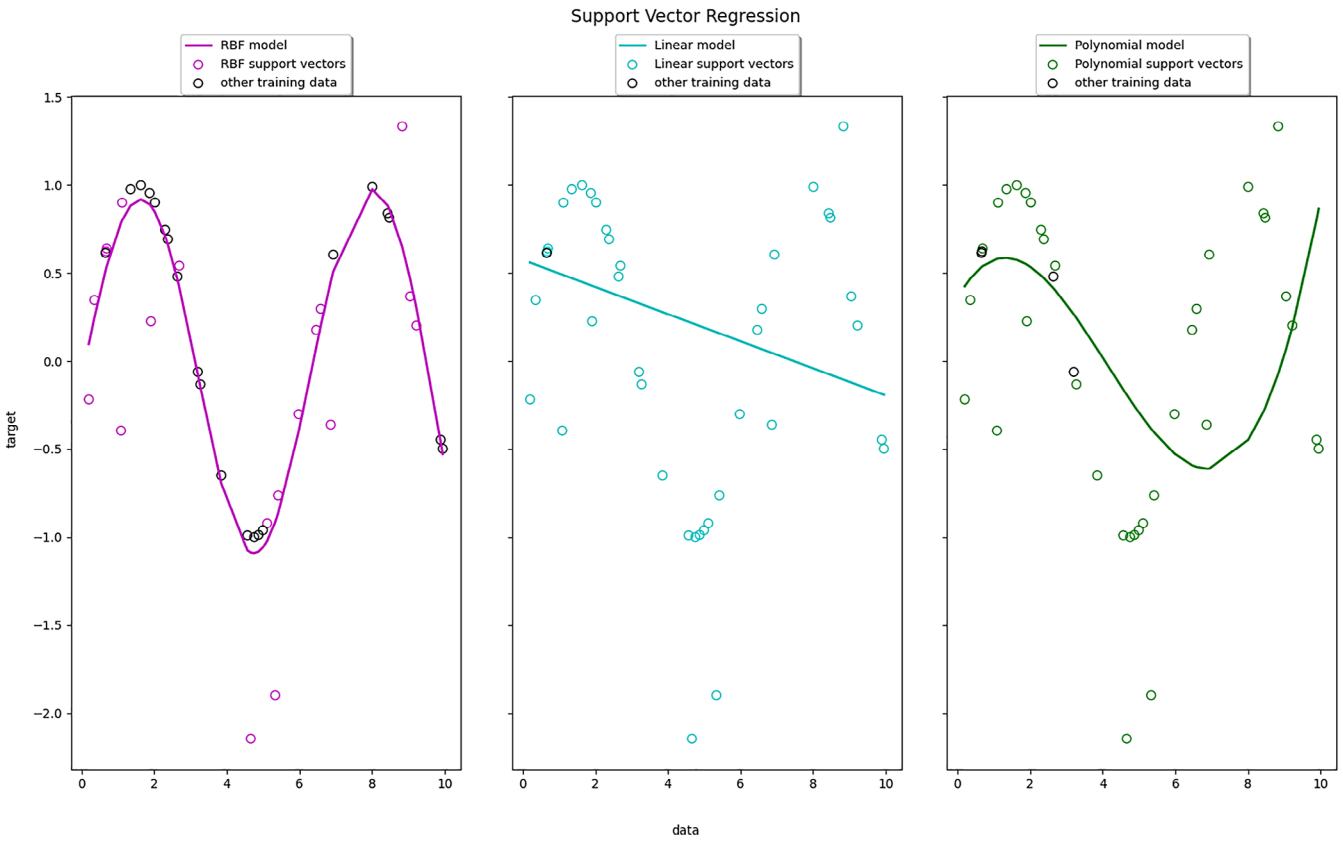

FIGURE 2 The example of support vector regression using RBF, linear, and polynomial kernels[24]

3.3 | The autoregression model

The autoregression model refers to a time series model utilizing the observations from the previous step to make the forecasts. The predicted values from the autoregression model depend on the previous values and a stochastic term. Hence the model is in the form of a stochastic differential equation. Mathematically, an autoregression model can be expressed as in Equation (3). This study leveraged the advantages of the learned coefficients retrieving from the model and manually made the stock price predictions. We utilized the history of 29 prior observations.

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \tag{3}$$

Where

- $\varphi_1, \varphi_2, \ldots \varphi_p$ stand for the parameters of the model.
- c denotes a constant.
- ε_t defines white noise.

3.4 | The LSTM model

LSTM is a special kind of recurrent neural networks (RNNs) and can be used to avoid the long-term dependency problem and better than the standard version. In theory, RNNs can handle such long-term dependencies, but in practice, RNNs cannot handle this circumstance. LSTM is explicitly designed to avoid the long-term dependency problem. Remembering information for long periods is the practically default manner. As with RNNs, LSTM also has a chain of repeating modules of the neural network, but the repeating module has a different structure. LSTM has four single neural network layers instead of one and interacting extraordinarily.

In LTSM cells, each line transmits an entire vector, particularly the output of one node and the inputs of others. The pink circles represent pointwise operations, such as vector addition, whereas the yellow boxes are learned neural network layers. The merged lines indicate the concatenation, while a line forking denotes the information being copied and the copies going to different locations. For more specific, a standard LSTM unit includes a cell, an input gate, an output gate, and a forget gate. The purpose of the cell is to remember the values over arbitrary time intervals. The gates can be considered as the artificial neuron in a neural network. The gates are responsible for computing the activation by utilizing the activation function. The LSTM cell details are visualized in Figure 3. The output of each cell is calculated as follows, conduct elementwise multiplication with forgetting gate f with the previous memory state C_{t-1} to determine if present memory state C_t. In the forget gate, the value equals 0, the previous memory state is unremembered. Otherwise, the cell receives the previous memory state. The gating variables, memory cell state, and cell hidden state are defined as follows, from Equations (4) to (9).

Gating variables

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_t) \tag{4}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{5}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{6}$$

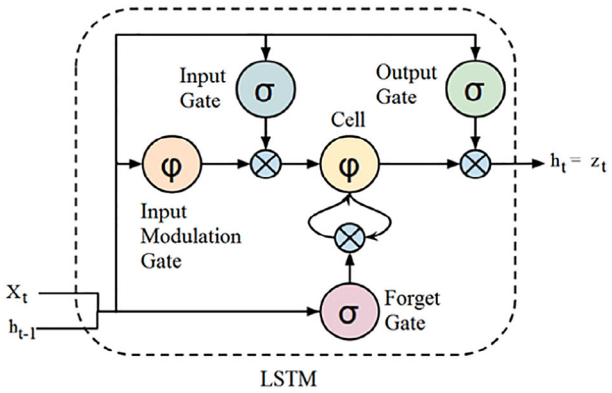

FIGURE 3 The cell of long short-term memory[26]

Memory cell state

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{7}$$

Hidden state

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{8}$$

$$h_t = o_t \circ \tanh(c_t) \tag{9}$$

Where

- f_t, i_t, and o_t denote the forget gate, input gate, and output gate, respectively. They are neural network with sigmoid σ.
- $\tilde{c}_t$ is the cell input activation vector.
- c_t is the memory state vector.
- W is the weight matrices.
- The subscript t presents the time-step.
- The operator $\circ$ denotes the Hadamard product.
- h_t denotes the hidden state or the output vector.

In this work, we implement the short-term long memory with Adam optimizer function[25] and MSE loss function. The learning rate is initiated at 0.001. We train our short-term long memory on 100 epochs with a batch size of 128. We also perform several experiments. Each experiment utilizes a different number of time-steps from 1 to 20. A representation with a 1-time-step is the default representation of the stateful LSTM. The specified time-steps determine the number of input variables used to predict the output. Specifically, the data organization observed at the previous time-step is used as an input to predict the observation at the current time step.

4 | EVOLUTION

This section presents the research problems and the materials and methods used in the evaluation, and the performance of each learning model on the considered dataset. Section 4.1 introduces the stock price dataset. Afterward, the experimental configurations and the metrics for comparison are presented in Sections 4.2 and 4.3, respectively.

4.1 | The dataset

The TAIEX dataset is used in our experiments. The TAIEX dataset consists of the stock price on each trading day over 15 straight years, from 1990 to 2004. The further information, for example, minimum, maximum, average stock prices, and the number of total samples of the dataset, are described in Table 1. We also visualize the stock prices change of the TAIEX dataset in Figure 4. The minimum price is **2560.47**, whereas the maximum and average are **12,495.34** and **6053.86**, respectively.

We split the dataset into two parts, the training, and testing dataset, with the following configuration.

TABLE 1 The further information of Taiwan stock exchange (TAIEX) dataset

Information	Description
Dataset	TAIEX
Min price	2560.47
Max price	12,495.34
Avg price	6053.86
Total number of samples	4086

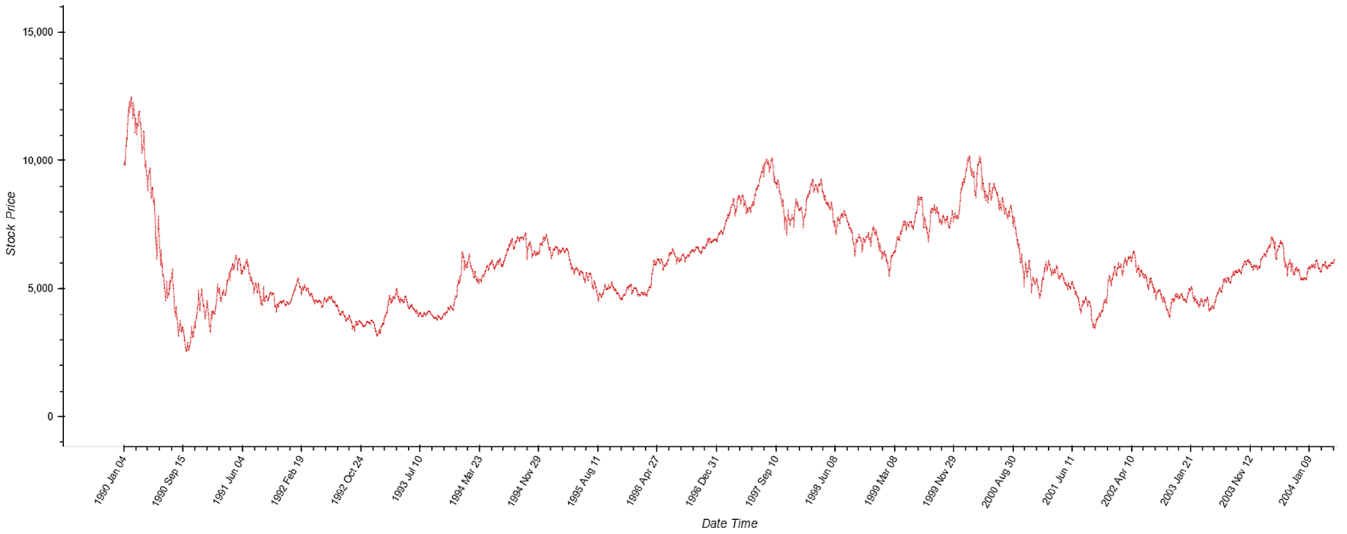

FIGURE 4 The visualization of Taiwan stock exchange dataset from 1990 to 2004

- We took the first 9 years in the dataset for training, the rest for testing.
- We took the first 10 continuous years for training, the last 5 years for testing.
- We took the first 11 years for training, 4 years for testing.

We also normalized the inputs by rescaling the stock price values within the range of 0 and 1 due to the differences in the scales across the stock prices. In practice, the inputs and outputs with a significant value may negatively influence the models' learning process. The rescaling process is conducted as Equation (10).

$$m = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{10}$$

Where

- m denotes the new value.
- x is the original value.
- $x_{\min}, x_{\max}$ are the minimum and maximum value, respectively.

In addition, due to the limited computational capability, the data normalization enhances the numerical stability of the learning models and reduces the training time, and the learning models can converge rapidly. Besides, the accuracy is unaffected by normalizing the inputs.

4.2 | Settings

LSTM model requires a powerful computational performance. We considered using the computer with the configurations listed in Table 2. Afterward experiment, the LSTM is more effective than SVR and RF. Thus, we evaluated the LSTM time-steps for stock price prediction and observed the impact of utilizing various lagged observations as input time-steps for LSTM architecture. In this study, we considered using one to five time-steps for evaluating the performance of LSTM on the stock price prediction.

4.3 | Metrics for comparison

By running the regression models on the testing dataset, we obtained the predicted stock prices over several years. To evaluate the performance of the learning models, we considered using two metrics, namely, MAE and the RMSE. MAE and RMSE are the most popular metrics in regression tasks and determine the prediction errors in the range of 0 to ∞. MAE and RMSE are negatively oriented scores, which means lower values are better and defined as the following Equations (11) and (12).

TABLE 2 Hardware and software configurations

Name	Description
RAM	16 GB
CPU	Intel® i7-8700 CPU @ 3.20 GHz
GPU	NVIDIA GeForce GTX 1070
OS	Windows 10 Pro 64-bit
Python	3.6.8
Tensorflow[27]	1.15.0
Scikit-learn[24]	0.23.1

$$\text{MAE} = \frac{\sum_{n}^{i=1} |y_i - \hat{y}_i|}{n} \tag{11}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{n}^{i=1} (y_i - \hat{y}_i)^2}{n}} \tag{12}$$

Where

- n denotes the number of trading days of the historical testing data.
- y_i denotes the forecasted value of the historical testing datum of the TAIEX on trading day i.
- $\hat{y}_i$ denotes the actual value of the historical testing datum of the TAIEX on trading day i.

5 | EXPERIMENTAL RESULTS

In this section, we present the detail of our experimental results. The results were obtained by conducting the learning models on the dataset. We present the performance of RF, SVR, autoregression, and LSTM models by calculating the RMSE, MAE in Sections 5.1–5.4, respectively. Finally, Section 5.5 reveals the comparison between the performance of the learning models.

5.1 | The performance of the RF model

The RMSE and MAE scores obtained by utilizing the RF on the dataset with several training years are reported in Table 3. From the table, the best performance has been achieved in 11 training years with the obtained validation RMSE and MAE values of 86.2171 and 63.8813, respectively. The performance with 10 training years is also close with the error rates of 89.5733 and 67.5931 for RMSE and MAE. The actual and predicted stock prices performed by the RF model by utilizing 11 training years are visualized in Figure 5. The x-axis and y-axis are described as the stock prices and the trading day, respectively. The blue dots represent the actual stock prices, whereas the red dots denote the predicted stock prices. As we mentioned in Section 3.1, we also investigated the performance of the RF model based on the building tree. The number of trees, K, and the max-leaf nodes are the most critical parameters of the RF model. We examined the performance with the following settings.

- The values of K include 10, 100, 200, 300, and 500.
- We also consider various numbers of max-leaf nodes N which consist of 10, 30, 50, 70, and 90.

TABLE 3 The performance of the random forest model over several training years

No. of training years	Training RMSE	Val RMSE	Training MAE	Val MAE
9 years	105.4135	109.9811	73.0181	79.9001
10 years	111.0101	89.5733	89.4739	67.5931
11 years	110.0662	86.2171	76.8682	63.8813

Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

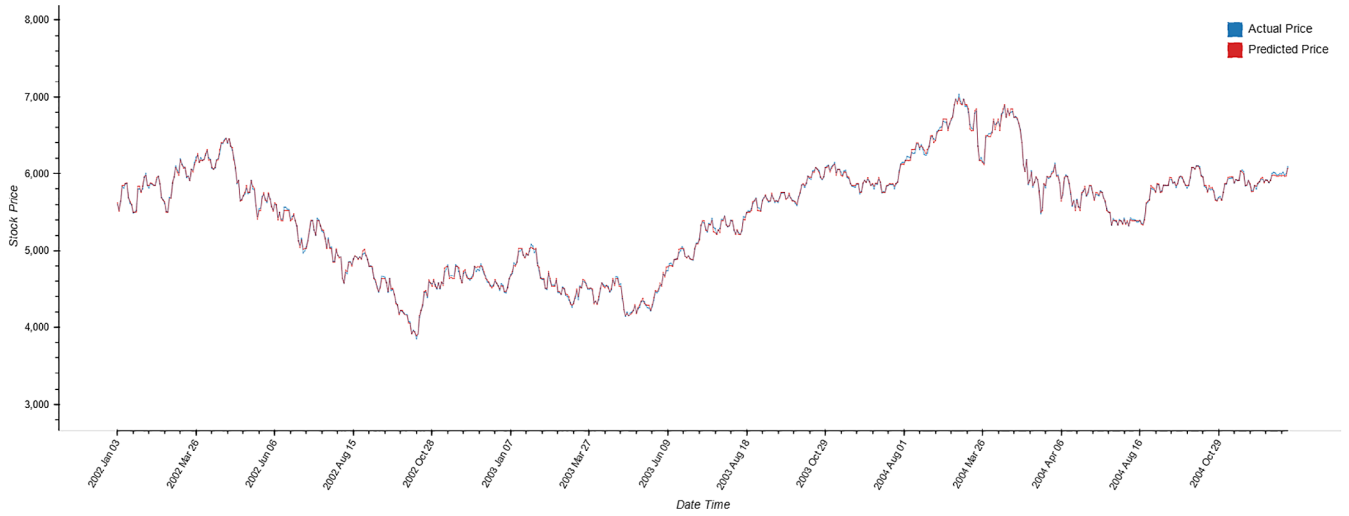

FIGURE 5 The visualization of the actual and the predicted stock prices performed by the random forest model with the number of training years is 11

We present the error rate of each setting for RF model in Figures 6 and 7. Figure 6 illustrates the validation RMSE and MAE with 10, 100, 200, 300, and 300 trees in the RF whereas Figure 7 visualizes the error rate of 10, 30, 50, 70, and 90 implemented max-leaf nodes. The results indicate the error rates depend a lot on the max-leaf nodes rather than the number of trees. With the small number of max-leaf nodes, that is, 10, the achieved RMSE and MAE will be high, and the model will make poor predictions.

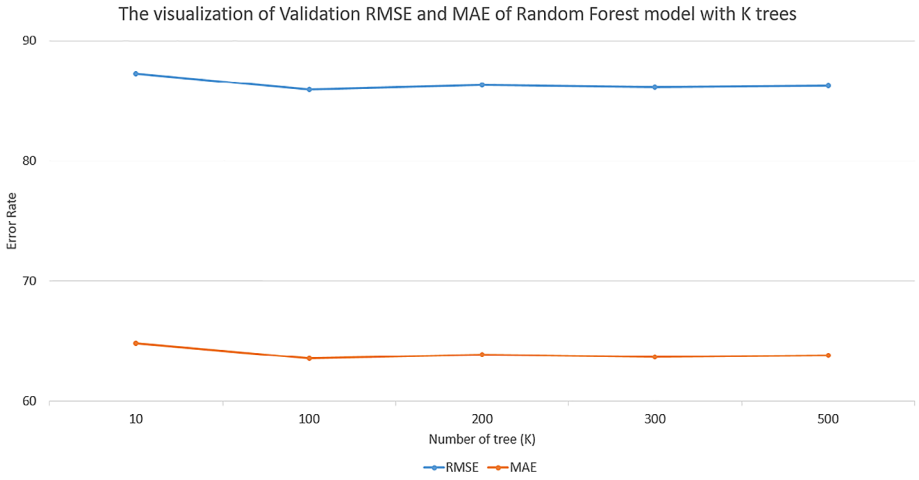

FIGURE 6 The visualization of validation RMSE and MAE of random forests with *K* trees. The connected blue dot denotes for RMSE, the orange stands for MAE. MAE, mean absolute error; RMSE, root mean square error

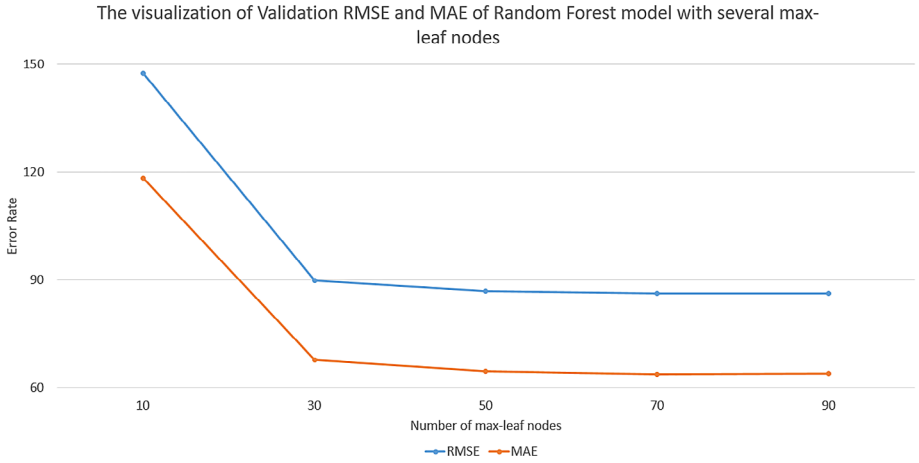

FIGURE 7 The visualization of validation RMSE and MAE of random forest model with several various considered numbers of max-leaf nodes. The connected blue dot denotes for RMSE, the orange stands for MAE. MAE, mean absolute error; RMSE, root mean square error

We are interested in fine-tuning the random forest model with the proposed parameters. Table 4 presents the performance of the model with different parameters. The best estimator is combined by 100 numbers of *K* and 70 numbers of *N*. As observed from the results, the number of *N* or the number of max-leaf nodes plays a vital role in predicting stock price problems; with $N = 10$, the model achieved poor results compared with the others. The optimal parameters obtained RMSE and MAE of 86.0326 and 63.7075, respectively. Furthermore, the number of trees, *K*, remains a question of how it affects the final results.

5.2 | The performance of the SVR model

Table 5 presents the performance comparison between 9, 10, and 11 training years based on the SVR model. We also calculated the RMSE and MAE over several training years. To be concrete, we observed that the SVR model performs the best results on 11 training years with the obtained RMSE and MAE values of 108.5686 and 81.7351. It is very close to the RF results on 11 training years, but the SVR does not occur the overfitting problem. Similar to Figure 5, we also visualize the historical stock prices and the expected prices from the SVR model in Figure 8. In terms of stock price prediction, the RBF is the optimal kernel function for the SVR model due to the distribution of data shown in Figure 2. Table 6 presents the performance comparison between the three kernel functions of the SVR model. The results demonstrate that the RBF is the best kernel function and achieved RMSE and MAE of 103.2152 and 81.9858.

5.3 | The performance of the autoregression model

Concerning the previous models, we also evaluated the performance of the autoregression model by training for several years and leveraged the advantages of the learned coefficients to forecast the next stock prices. The performance of the autoregression model is reported detailedly in Table 7. Observation from Table 7, the autoregression model obtained the optimal RMSE and MAE values of 121.3567 and 93.9823 on 11 training years. Compared with the two previous models, the autoregression model is still close but not higher than the RF or SVR model. Figure 9 depicts the difference of the real values of stock prices and the measured values computed by the autoregression model on the considered dataset.

TABLE 4 The performance of the random forest model with different parameters

Configuration	Val RMSE	Val MAE	Configuration	Val RMSE	Val MAE	Configuration	Val RMSE	Val MAE
K:10 - N:10	201.1687	170.4733	K:100 - N:10	170.8141	141.9117	K:200 - N:10	167.7423	138.9014
K:10 - N:30	92.5001	69.4667	K:100 - N:30	90.1495	67.7124	K:200 - N:30	90.0889	67.4032
K:10 - N:50	87.8727	65.8746	K:100 - N:50	86.7735	64.5923	K:200 - N:50	86.6305	64.4783
K:10 - N:70	86.6601	64.7225	K:100 - N:70	**86.0326**	**63.7075**	K:200 - N:70	86.2499	63.9968
K:10 - N:90	86.8479	65.0156	K:100 - N:90	86.5581	64.1477	K:200 - N:90	86.2946	64.0759
K:300 - N:10	172.9872	144.6602	K:500 - N:10	170.0273	141.6941	–	–	
K:300 - N:30	90.1459	67.5361	K:500 - N:30	89.6365	67.0878	–	–	
K:300 - N:50	86.7013	64.4721	K:500 - N:50	86.7915	64.6238	–	–	
K:300 - N:70	86.1592	63.8306	K:500 - N:70	85.9055	63.6416	–	–	
K:300 - N:90	86.3491	64.0126	K:500 - N:90	86.4068	64.0463	–	–	

Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

TABLE 5 The performance of the support vector regression model over several training years

No. of training years	Training RMSE	Val RMSE	Training MAE	Val MAE
9 years	216.2698	151.5243	155.3026	112.9230
10 years	216.1302	128.9971	157.1193	95.7017
11 years	213.5332	108.5686	155.7509	81.7351

Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

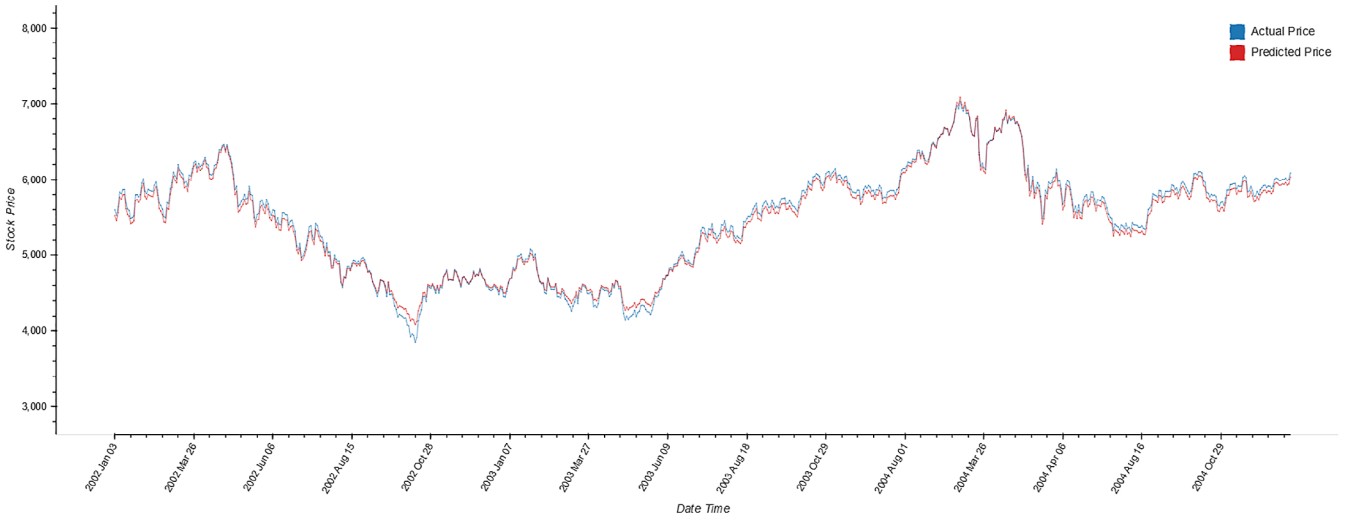

FIGURE 8 The visualization of the actual and the predicted stock prices performed by the support vector regression model with the number of training years is 11

TABLE 6 The performance of the support vector regression model with RBF, linear, and polynomial kernel function

Kernel function	Val RMSE	Val MAE
RBF	103.2152	81.9585
Linear	424.1574	364.0909
Poly	515.1757	465.0076

Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

TABLE 7 The performance of the autoregression model over several training years

No. of training years	Training RMSE	Val RMSE	Training MAE	Val MAE
9 years	181.5242	154.5174	118.7935	116.4623
10 years	186.2001	125.9007	124.6389	97.7775
11 years	183.1184	121.3567	123.4806	93.9823

Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

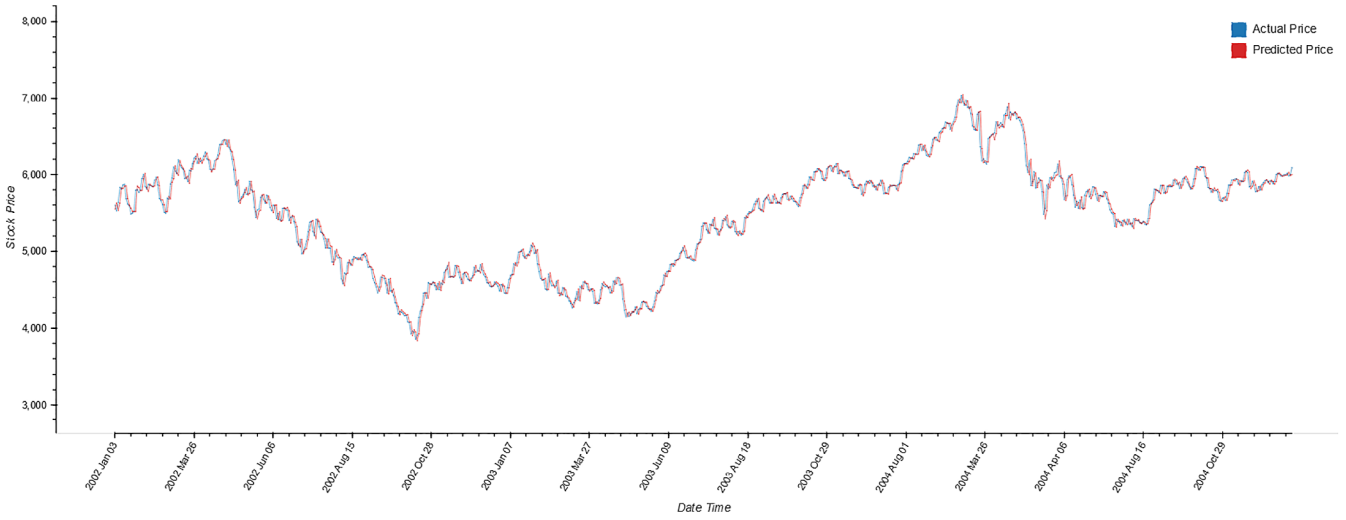

FIGURE 9 The visualization of the actual and the predicted stock prices performed by the autoregression model with the number of training years is 11

5.4 | The performance of the LSTM model

As we mentioned in Section 3.4, we utilized the time-step technique for the LSTM model in this study. We also evaluated the performance by calculating the RMSE and MAE on 9, 10, and 11 training years. The results of the LSTM with one time-step are presented in Table 8. Compared with the RF, SVR, and autoregression model, the short-term long memory exhibits the best performance over 9, 10, and 11 training years. More specifically, the LSTM obtained an RMSE and MAE of 84.4049 and 62.8812 on 11 training years, whereas the RF achieved 107.3416, 82.7181, the SVR acquired 108.5686, 81.7351, and the autoregression model gained 121.3567 and 93.98.23, respectively. The predicted price differentials of the LSTM model are depicted in Figure 10. It is difficult to recognize the disagreement between the LSTM model results and the ground-true values on the dataset.

We also estimated the LSTM model with number of time-step from 2 to 20 and visualized the validation RMSE and MAE in Figure 11 and Tables 9 and 10. Observing from the results, the performance of LSTM over 20 time-step from 1 to 20 are regularly similar. The minimum validation RMSE achieved 83.9451 with 11 training years, and the number of time-step is 1, whereas the minimum value of MAE reached 61.5934 with the number of time-step is 1 and 11 training years. The LSTM obtained an average overall RMSE and MAE of 101.5443 and 76.7559, respectively. The training and validation loss during the training section is visualized in Figure 12. The obtained loss after 50 epochs demonstrates that the LSTM model seems to be converged. The training section finished with both inadequate training and validation loss. It almost equals 0.

5.5 | The performance comparison of learning models

We recognize that the error rates are achieved with the number of training years is of 11. Table 11 exhibits the validation RMSE and MAE of the RF, SVR, autoregression, and LSTM model on 11 training years comparing to the work in Reference 1. In summary, the LSTM obtained optimal error rates in comparison with the others. As we mentioned above, the visualizations of actual and predicted stock prices performed by the LSTM (Figure 10) are relatively similar, just containing a few different points. In comparison with the work in Reference 1, an investigation on the use of time-step can improve the performance reducing the error rate in MAE from 67.7686 to 61.5934 and from 88.7651 to 88.9451.

We also applied the proposed methods to the different Taiwan stock datasets. The dataset is public on the Internet and available at https://finmind.github.io/quickstart/.

TABLE 8 The performance of long short-term memory model over several training years

No. of training years	Training RMSE	Val RMSE	Training MAE	Val MAE
9 years	115.9212	106.144	79.9474	77.0891
10 years	121.1366	87.5507	85.7271	66.3436
11 years	119.6921	83.9451	84.4711	61.5934

Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

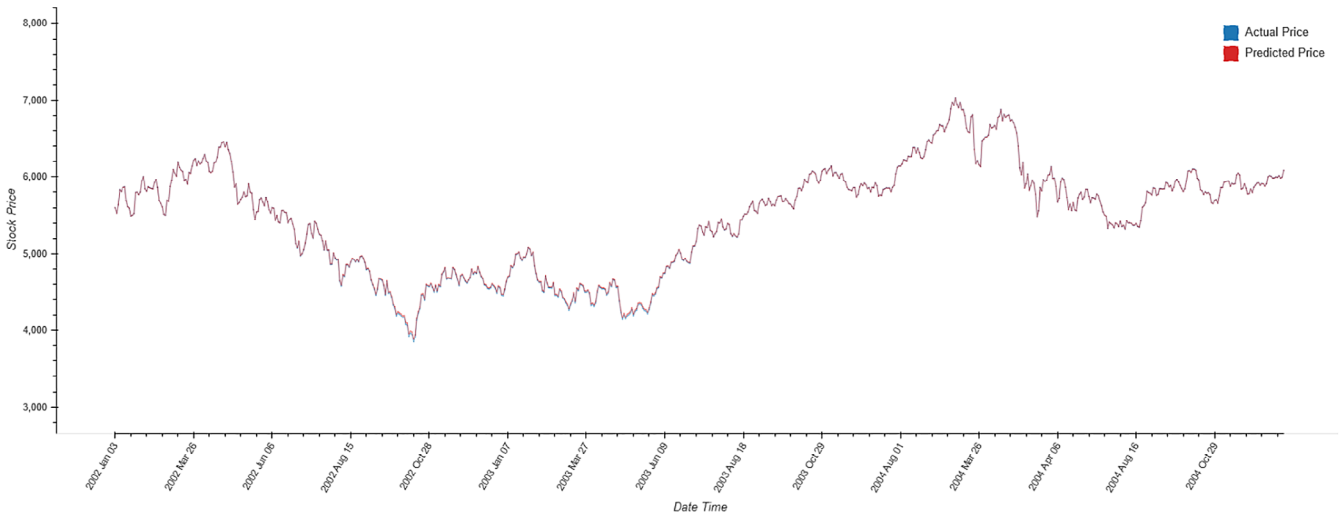

FIGURE 10 The visualization of the actual and the predicted stock prices performed by the long short-term memory model with the number of training years is 11 and the time-step is of 1

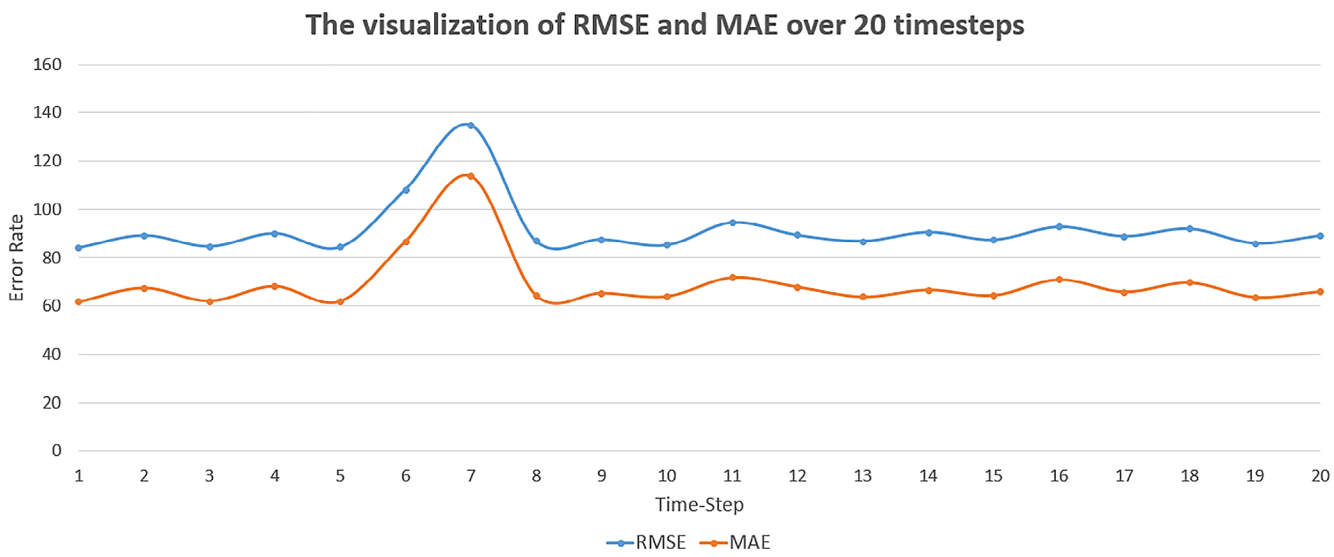

FIGURE 11 The visualization of validation RMSE and MAE over 20 time-step. The connected blue dot denotes for RMSE, the orange stands for MAE. MAE, mean absolute error; RMSE, root mean square error

TABLE 9 The validation performance in RMSE over 20 time-steps

	Training years			
Time-step	9	10	11	Average
1	106.1447	87.5507	***83.9451***	92.5469
2	108.4254	96.631	89.1412	98.0659
3	111.2583	92.8823	84.4905	96.2104
4	111.7044	93.3708	89.9908	98.3553
5	138.7352	87.7572	84.2242	103.5722
6	112.5225	96.8802	108.0795	105.8274
7	109.2656	90.2728	134.7467	111.4284
8	118.8437	98.7512	86.7149	101.4366
9	109.1723	97.6578	87.4620	98.0974
10	110.6104	92.8669	85.0704	96.1826
11	114.2438	90.8909	94.4527	99.8625
12	112.7287	94.0648	89.2258	98.6731
13	120.1542	92.03	86.6911	99.6251
14	111.2596	94.6085	90.4519	98.7733
15	110.3995	93.1069	87.2789	96.9284
16	112.1014	111.8307	92.6727	105.5349
17	121.8041	92.0078	88.7714	100.8611
18	115.5356	100.5281	91.8454	102.6364
19	113.8811	114.0903	85.8303	104.6006
20	123.8614	152.0729	89.0667	121.667
Average	**114.6326**	**98.4926**	**91.5076**	**101.5443**

Note: The best result is formatted in bold and italic text.
Abbreviation: RMSE, root mean square error.

TABLE 10 The validation performance in MAE over 20 time-steps

Time-step	Training years			Average
	9	10	11	
1	77.089	66.3436	***61.5934***	68.342
2	79.3048	75.5475	67.3621	74.0715
3	82.6232	70.4876	61.9717	71.6942
4	81.9322	72.4427	68.0781	74.151
5	109.3925	65.9424	61.797	79.044
6	83.2372	74.5731	86.5627	81.4576
7	79.8634	68.7973	113.6494	87.4367
8	90.0363	76.2617	64.268	76.8553
9	79.8025	75.7315	65.2263	73.5867
10	80.4354	70.9388	63.8342	71.7361
11	84.2804	68.6782	71.6951	74.8846
12	83.0872	72.0609	67.7612	74.3031
13	90.3397	70.587	63.6555	74.8607
14	80.709	72.4062	66.5699	73.2283
15	80.5949	71.3473	64.26	72.0674
16	81.9883	88.678	70.8006	80.489
17	88.8493	69.4435	65.7169	74.6699
18	85.1394	77.8762	69.5418	77.5191
19	83.1029	91.7674	63.5188	79.4631
20	92.8399	127.0257	65.9058	95.2572
Average	**84.7324**	**76.3468**	**69.1884**	**76.7559**

Note: The best result is formatted in bold and italic text.
Abbreviation: MAE, mean absolute error.

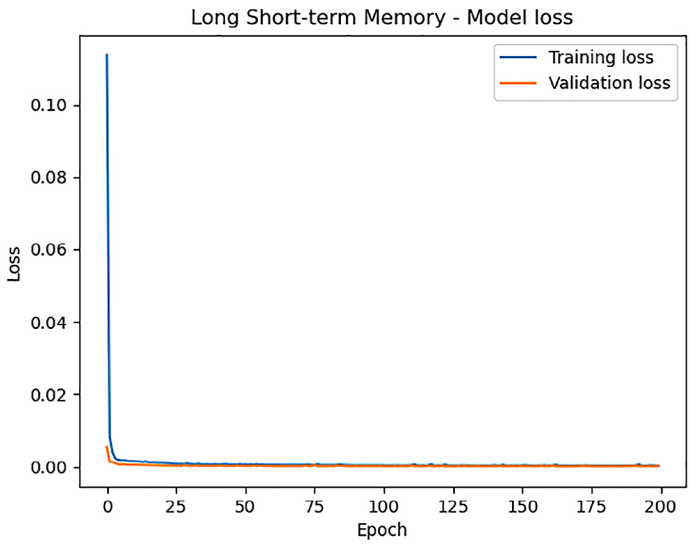

FIGURE 12 The visualization of training and validation loss of long short-term memory model with 11 training years and 1 time-step The blue line represents for the training loss whereas the orange stands for the validation loss

The data is split into training and testing with ratios of 0.7 and 0.3, respectively. Table 12 reports the comparison performance of the proposed method on the Taiwan stock dataset. The performance on the external dataset reveals that the LSTM, RF, and autoregression model can be applied for the stock prices prediction problems. Furthermore, the LSTM outperformed the others on the proposed dataset and the external dataset.

TABLE 11 The performance of the considered algorithms compared with the-state-of-the-art

Learning model	Validation RMSE	Validation MAE
Long short-term memory	83.9451	61.5934
Random forest	86.1209	63.7802
Support vector regression	108.5686	81.7351
Autoregression	121.3567	93.9823
Long short-term memory[1]	88.7651	67.7686
Random forest[1]	91.1585	69.8861
Support vector regression[1]	392.1609	358.0659

Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

TABLE 12 The performance of the proposed method on the Taiwan stock dataset

Learning model	Validation RMSE	Validation MAE
Long short-term memory	833.6448	110.7638
Random forest	892.4436	120.6644
Support vector regression	1799.2473	1662.5326
Autoregression	847.7123	134.7622

Abbreviations: MAE, mean absolute error; RMSE, root mean square error.

6 | CONCLUSION

We have presented the experimental results of stock prices on the Taiwanese stock market with various methods. During a period from 1990 to 2004, we investigated the performance of forecast models on various ways for dividing training and test sets. As observed, LTSM outperforms other considered machine learning algorithms. Some investigations on the configuration of time-step in LSTM were introduced. We can see that there are some fluctuations in the performance when we change the number of time-step. As illustrated from the RF results, the parameter of several top leaf nodes can have a more significant effect on the performance compared with the number of trees. When we use more than 30 nodes, the performance tends to be saturated. In an attempt to reduce the regression errors, we compared the performance on 11 training years. The results demonstrate LSTM achieved the best performance with the number of time-step is of 1. More specifically, the LSTM obtained RMSE and MAE of 83.9451 and 61.5934 compared with the RF built by 100 trees, and 70 max-leaf nodes achieved the errors of the RMSE and MAE are 86.1209 and 63.7802. Furthermore, SVR and autoregression with 30 nodes exhibit regular performance, 108.5686 of RMSE, 81.7351 of MAE for SVR, and 121.3567 of RMSE, 93.9823 of MAE for autoregression, respectively.

Experimental results inline charts also show that the predicted values share the same pattern with the actual prices, and the proposed method is expected to provide future patterns for Taiwanese stock prices. Future work can investigate more profound architecture to enhance performance.

ACKNOWLEDGEMENTS

Tran B. Toan was funded by Vingroup Joint Stock Company and supported by the Domestic Master/PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Vingroup Big Data Institute (VINBIGDATA), code VINIF.2020.ThS63.

DATA AVAILABILITY STATEMENT

Data and experimental scripts of this study are published at the Github repository link Stock-Pred: https://github.com/thnguyencit/stock-pred. The data that support the findings of this study are openly available at Reference 1.

ORCID

Hai T. Nguyen https://orcid.org/0000-0002-1386-1390
Toan B. Tran https://orcid.org/0000-0002-0575-9783
Phuong H. D. Bui https://orcid.org/0000-0002-6179-3513

REFERENCES

1. Bui PHD, Tran TB, Nguyen HT. *Taiwanese Stock Market Forecasting with a Shallow Long Short-Term Memory Architecture*. Berlin, Germany: Springer International Publishing; 2021:192-202.

2. Cai Q, Zhang D, Zheng W, Leung SC. A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression. *Knowl-Based Syst*. 2015;74:61-68. https://doi.org/10.1016/j.knosys.2014.11.003.
3. Chen SM, Chang YC. Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques. *Inf Sci*. 2010;180(24):4772-4783. https://doi.org/10.1016/j.ins.2010.08.026.
4. Panigrahi S, Behera HS. *Fuzzy Time Series Forecasting: A Survey*. Singapore, Asia: Springer; 2019:641-651.
5. Chen SM, Chen CD. TAIEX forecasting based on fuzzy time series and fuzzy variation groups. *IEEE Trans Fuzzy Syst*. 2011;19(1):1-12. https://doi.org/10.1109/tfuzz.2010.2073712.
6. Chen SM, Jian WS. Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups, similarity measures and PSO techniques. *Inf Sci*. 2017;391-392:65-79. https://doi.org/10.1016/j.ins.2016.11.004.
7. Chen SM, Kao PY. Forecasting the TAIEX Based on Fuzzy Time Series, PSO Techniques and Support Vector Machines. Asian Conference on Intelligent Information and Database Systems; 2013:89-98; Springer, Berlin Heidelberg.
8. Chen SM, Manalu GMT, Pan JS, Liu HC. Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups and particle swarm optimization techniques. *IEEE Trans Cybern*. 2013;43(3):1102-1117. https://doi.org/10.1109/TSMCB.2012.2223815.
9. Chen SM, Phuong BDH. Fuzzy time series forecasting based on optimal partitions of intervals and optimal weighting vectors. *Knowl-Based Syst*. 2017;118:204-216. https://doi.org/10.1016/j.knosys.2016.11.019.
10. Huarng KH, Yu THK, Hsu YW. A multivariate heuristic model for fuzzy time-series forecasting. *IEEE Trans Syst Man Cybern B (Cybern)*. 2007;37(4):836-846. https://doi.org/10.1109/tsmcb.2006.890303.
11. Yu HK. Weighted fuzzy time series models for TAIEX forecasting. *Phys A Stat Mech Appl*. 2005;349(3-4):609-624. https://doi.org/10.1016/j.physa.2004.11.006.
12. Tanuwijaya B, Selvachandran G, Son LH, et al. A novel single valued neutrosophic hesitant fuzzy time series model: applications in Indonesian and Argentinian stock index forecasting. *IEEE Access*. 2020;8:60126-60141. https://doi.org/10.1109/access.2020.2982825.
13. Chen CC, Liu Y, Hsu TH. An analysis on investment performance of machine learning: an empirical examination on Taiwan stock market. *Int J Econom Financ Issues*. 2019;9(4):1-10. https://doi.org/10.32479/ijefi.8129.
14. Kocak C, Egrioglu E, Bas E. A new deep intuitionistic fuzzy time series forecasting method based on long short-term memory. *J Supercomput*. 2020. https://doi.org/10.1007/s11227-020-03503-8.
15. Su Z, Xie H, Han L. Multi-factor RFG-LSTM algorithm for stock sequence predicting. *Comput Econ*. 2020. https://doi.org/10.1007/s10614-020-10008-2.
16. Niu H, Xu K, Wang W. A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network. *Appl Intell*. 2020;50(12):4296-4309. https://doi.org/10.1007/s10489-020-01814-0.
17. Ghosh P, Neufeld A, Sahoo JK. Forecasting directional movements of stock prices for intraday trading using LSTM and random forests; 2020. https://arxiv.org/abs/2004.10178.
18. Bukhari AH, Raja MAZ, Sulaiman M, Islam S, Shoaib M, Kumam P. Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting. *IEEE Access*. 2020;8:71326-71338. https://doi.org/10.1109/ACCESS.2020.2985763.
19. Huang CS, Liu YS. Machine learning on stock price movement forecast: the sample of the Taiwan stock exchange. *Int J Econom Financ Issues*. 2019;9(2):189-201.
20. Li Q, Tan J, Wang J, Chen H. A multimodal event-driven LSTM model for stock prediction using online news. *IEEE Trans Knowl Data Eng*. 2020;1. https://doi.org/10.1109/TKDE.2020.2968894.
21. Moghar A, Hamiche M. Stock market prediction using LSTM recurrent neural network. *Proc Comput Sci*. 2020;170:1168-1173. https://doi.org/10.1016/j.procs.2020.03.049.
22. Chen S. A classification problem with python – homesite quote conversion; 2019. https://towardsdatascience.com/a-classification-problem-with-python-homesite-quote-conversion-15174bca09b8.
23. Singh J. Random forest: pros and cons; 2020. https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04.
24. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in python. *J Mach Learn Res*. 2011;12:2825-2830.
25. Kingma DP, Ba J. Adam: a method for stochastic optimization; 2017. https://arxiv.org/abs/1412.6980.
26. Fayyaz M, Saffar MH, Sabokrou M, Fathy M, Klette R. STFCN: spatio-temporal FCN for semantic video segmentation; 2016. ArXiv 2016; abs/1608.05971.
27. Abadi M, Agarwal A, Barham P, et al. TensorFlow: large-scale machine learning on heterogeneous systems; 2015. https://arxiv.org/abs/1603.04467. tensorflow.org.

How to cite this article: Nguyen HT, Tran TB, Bui PHD. An effective way for Taiwanese stock price prediction: Boosting the performance with machine learning techniques. *Concurrency Computat Pract Exper*. 2021;e6437. https://doi.org/10.1002/cpe.6437