Hindawi
Scientific Programming
Volume 2021, Article ID 5113151, 9 pages
https://doi.org/10.1155/2021/5113151

Research Article

Forecasting Variation Trends of Stocks via Multiscale Feature Fusion and Long Short-Term Memory Learning

Yezhen Liu,[1] **Xilong Yu**,[2] **Yanhua Wu**,[2] **and Shuhong Song**[1]

[1]*College of Economics and Management, Nanjing Forestry University, Nanjing 210037, China*
[2]*China Ship Scientific Research Center, China Shipbuilding Industry Corporation, Wuxi 214000, China*

Correspondence should be addressed to Shuhong Song; songshuhong@njfu.edu.cn

Received 20 August 2021; Revised 8 September 2021; Accepted 11 September 2021; Published 22 September 2021

Academic Editor: Chenxi Huang

Forecasting stock price trends accurately appears a huge challenge because the environment of stock markets is extremely stochastic and complicated. This challenge persistently motivates us to seek reliable pathways to guide stock trading. While the Long Short-Term Memory (LSTM) network has the dedicated gate structure quite suitable for the prediction based on contextual features, we propose a novel LSTM-based model. Also, we devise a multiscale convolutional feature fusion mechanism for the model to extensively exploit the contextual relationships hidden in consecutive time steps. The significance of our designed scheme is twofold. (1) Benefiting from the gate structure designed for both long- and short-term memories, our model can use the given stock history data more adaptively than traditional models, which greatly guarantees the prediction performance in financial time series (FTS) scenarios and thus profits the prediction of stock trends. (2) The multiscale convolutional feature fusion mechanism can diversify the feature representation and more extensively capture the FTS feature essence than traditional models, which fairly facilitates the generalizability. Empirical studies conducted on three classic stock history data sets, i.e., S&P 500, DJIA, and VIX, demonstrated the effectiveness and stability superiority of the suggested method against a few state-of-the-art models using multiple validity indices. For example, our method achieved the highest average directional accuracy (around 0.71) on the three employed stock data sets.

1. Introduction

Forecasting the variation trend of stocks is always one of the hot topics in the academic and practical studies of stock markets. The innately dynamic, chaotic, and nonstationary properties of stock markets make it extremely challenging to predict the tendency of financial time series (FTS) precisely. Given that the fluctuation of the stock price is affected by multiple aspects of social economic life, it has great economic and social values to forecast the developing trend of the stock price effectively. Both investors and for-profit institutions require scientific and intelligent methods to analyze and evaluate the price history so as to facilitate establishing the appropriate trade strategies.

The ultimate goal, for a specific stock, is to sell out shares at the highest price and purchase shares during the lowest period, which means minimizing risks as well as maximizing profits. Admittedly, financial data often exhibit hybrid, nonlinear, and seemingly unrelated characteristics, which makes market hypothesis difficult to apply to predict the potential [1]. Also, some unexpected factors commonly make the stock market change dramatically, e.g., worldwide economic condition, national policies, public voices, investors' expectations, and the like.

The prevailing theory is that the stock market is largely random, especially in the case of the Iranian stock market, which is determined by certain criteria of closing price. In the past, the most traditional methods associated with time series were based on stationary trends, leading to an inherent difficulty of anticipation [2]. Therefore, plenty of researchers were devoted to conducting abundant experiments and tried to establish reliable stock price models. However, there are still massive difficulties and unsolved problems, such as those countless variables and factors. In addition, due to the

difference between short and long terms, one fine-turning model probably works well in the short-term prediction, whereas could be poor in a longer time series.

Numerous studies have been persistently seeking suitable pathways to address such challenges, and modern artificial intelligence technologies, e.g., machine learning (ML) algorithms, have particularly facilitated this category of studies. It is a consensus that ML is qualified to extract potential characteristics and discover relative patterns from price history data. With the high-speed development of machine learning, many approaches have obtained convincing and outstanding performance on some price history data, for instance, the S&P 500 Index (S&P 500), Hang Seng Index (HSI), Jones Industrial Average (DJIA), and Nikkei 225 (N 255). Even if none of them were invariably successful in practice, their working mechanisms are worth learning and modifying, such as logistic regression, Support Vector Machine (SVM), decision tree, Recurrent Neural Network (RNN), LSTM, and Temporal Convolutional Attention-Based Network (TCAN).

In the beginning, LSTM was designed to resolve the issue of error backflow, namely, error signals explode or vanish as they flow backwards on a certain time scale. Facing noise and incompressible input sequences, LSTM can learn intervals spanning more than 1,000 iterations. A gradient-based approach is used to ensure continuous error flows in special units, which warrants that the gradient computation would be truncated at certain architecture-specific points without affecting the error flow computing on long-term data [3]. LSTM was further improved by adding the gradient propagation path and forgetting gate structure. Compared with the traditional RNN, this improvement can solve the problem of gradient vanishment. It is not the total gradient vanishment, but the one dominated by the short term that makes it difficult for models to capture long-term features. In time series prediction, such deficiency of gradient vanishment will lead to continuous loss of remote information during consecutive learning and to the dependence decrease of model parameters to remote features. Whereas stock data are exactly long-term and noncyclical and long-term dependence is critical to forecast results, the gate structure of LSTM is very suitable to complete the task of stock forecasting.

In this study, we aim to propose a novel, multiscale, convolutional feature, fusion-based LSTM model for the FTS forecasting issue. Our efforts lie in the following two aspects:

(1) Owing to the dedicated gate structure designed for short-term and long-term memory, our model can make full use of the given stock history data to adaptively train the forecasting model. Also, the LSTM network structure effectively avoids the gradient explosion and gradient vanishment problems of deep learning. Therefore, our scheme greatly warrants the prediction performance in time series scenarios and thereby benefits the prediction of stock trends.

(2) The multiscale, convolutional feature, fusion mechanism embedded in the proposed LSTM-based model can diversify the feature representation and thereby can more extensively capture the feature essence of time series than traditional models. These two improvements facilitate the generalizability of our novel LSTM model to a certain extent.

The rest of the article is organized as follows. Section 2 briefly introduces the works related to stock price prediction. Section 3 illustrates our suggested methodology in details. In Section 4, experimental results and discussions are presented. Finally, we conclude the article in Section 5.

2. Related Works

Numerous studies anticipating stock price variation trends have been performed based on the FTS analysis. The widely used techniques can be roughly divided into three categories: statistical econometric models or tools, regression algorithms, and deep learning methods. This study focuses on machine learning-based techniques, so we primarily review the latter two in the following.

Timbó et al. proposed a multiple linear regression algorithm with a data processing methodology, named Knowledge Discovery in Databases (KDD) [4]. KDD is a multistep process to capture useful, vital information within massive price data sets, including selection, preprocessing, transformation, data mining, and interpretation, and in which linear regression can benefit from the precise and low-noise data. Lin et al. proposed a stock forecasting method using SVM, including two functions: feature selection and trend forecasting [5]. In terms of the technique of support vector regression (SVR), this method can forecast the tendency of stock prices well. The authors also proved the superior generalizability of this method versus others.

Compared with other machine learning methods, artificial neural networks (ANNs), particularly deep neural networks, have showed their validity in practice, such as in stock pricing prediction [6]. In the study by Wanjawa and Muchemi [7], ANN was utilized to forecast stock pricing by a feed-forward multilayer perceptron with inverse loss propagation and thus obtained good performance. Nonetheless, researchers noticed that ANN scarcely establishes a correlative connection between current and previous data, leading to poor robustness and low universality. It is a consensus that the predicted results are correlative not only to current data but also to previous data. To tackle this problem, RNN was devised. In contrast to traditional ANN, RNN proved more convincing performance in the financial field. During its iterations, earlier time series data are beneficial to the model's precise learning via the feed-forward and back-forward looping.

The earlier stock prediction used traditional RNNs. These methods were frequently combined with other technologies for denoising data, such as Discrete Wavelet Transform (DWT). Contrary to the limitations of Fourier Transform, DWT originally uses wavelet basis to describe the signal [8]. The wavelet basis is a very small scale of a signal, so the wavelet transform has the ability to describe time series. One of the keynotes of wavelet transform is to use different resources to

describe different frequency ranges. For different frequency scales, like trees, the richness of the description is very different. The higher the frequency of sampling, the finer the description. Compared with DWT, b-Spline Wavelets of High-Order-d (BSd) can achieve better results on certain data sets. BSd-RNN was proposed to forecast the high-frequency time series in the study by Hajiabotorabi et al. [9]. With the combination of Bsd and RNN, the time series was decomposed into numerous smooth data sets using a multiresolution technique, which made it possible to generate distinctly detailed data sets with modest wave amplitudes. Due to the local properties, the suggested BSd-RNN model was capable of accurately approximating more smooth patterns than other common models.

Among all RNN-based models, LSTM could be the most effective model for time series prediction. LSTM used a set of memory cells with the gate structure to replace hidden neurons of RNN. As such, through the gate structure feature, the information was retained and persistently updated in the following training iterations. For instance, Zhao et al. used LSTM to achieve extraordinary performance in stock trend prediction [10]. Seng et al. used the ordinary three-layer LSTM structure, instead of utilizing too complex network structures, to forecast LQ45 financial sectors indices and obtain nice results [11].

Also, many researchers sought the manners to further improve LSTM's performance by assembling other learning models. Autoregressive Fractional Integrated Moving Average (ARFIMA) was first used to predict the weather's seasonal change. Afterwards, some researchers attempted to forecast stock prices using ARFIMA because it is fairly suitable for predicting the results of time series data. ARIMA-LSTM model was proposed by Bukhari et al. [12]. In the field of deep learning, Convolution Neural Network (CNN) is another outstanding network qualified for forecasting tasks using varied convolution blocks. Nonetheless, CNN can hardly tackle time series data separately. Thus, Qiu et al. combined RNN with CNN to put forward a novel network called Deep Wide Area Neutral Network (DWNN) [13]. Experiments showed that this model can reduce the mean square error of prediction by 30% compared with the conventional RNN structures. In order to capture the time-dependent characteristics, Zhang et al. proposed another alternative fusion of CNN and RNN strategy in [14]. During the hidden state transfer, CNN's convolution layers were introduced to extract the correlation features and RNN, meanwhile, proceeded in time steps. As such, this design had not only the depth of RNN in the temporal dimension but also the width of temporal data.

One of the major drawbacks of aforementioned methods is their inability to forecast highly dynamic and transforming patterns of stock price variation, whereas TCAN solved this issue to a certain extent. Hao et al. proposed the TCAN algorithm that integrated an attention mechanism into the time series neural network [15]. By jointly introducing Temporal Attention (TA) and Enhanced Residual (ER), TCAN was enabled to extract both the shallow layers' pivotal features and correlative characteristics hidden in the time series.

3. Methodology

In this section, we detail our novel LSTM-based structure and scheme for stock price prediction as follows.

As shown in Figure 1, our method is mainly composed of two parts: data preprocessing and model construction. First, the raw data set is preprocessed with wavelet denoising, normalized time step data, and data set division. For the latter, the prediction model employs the three-branch structure to constitute the multiscale feature fusion-based convolutional LSTMs, followed by a dense layer for eventually denormalizing the output.

3.1. Preprocessing. The Yahoo Finance data (https://finance.yahoo.com) was downloaded to act as our experimental data sets, which includes the data of open, high, low, and closing prices of stocks, trading volume, and adjusted prices.

To capture the essential characteristics of stock price time series data, besides, the six originally contained variables (i.e., *open*, *closing*, *high*, *low*, *trading volume*, and *adjusted price*), the *moving average* (MA), and *exponent moving average* (EMA) are also calculated in our study. These two can reflect the trend of stock price variation exponentially or at a constant level and are proved effective to guide stock investment. In this way, the stock price history data are represented as the form of eight-dimensional time series and further used as the input for our LSTM-based model.

The complexity and volatility of the stock market and the dynamic trading criterion usually cause the stock price data obtained to be noisy [16] and nonstationary because enormous factors, either explicitly or implicitly, influence the variation tendency of stock prices. Classic denoising algorithms, such as Fourier analysis, are prone to being ineffective in the case of massive information fusion. Therefore, we employ the wavelet transform to denoise the financial time series data. The wavelet transform has the capacity to conduct time series analysis in both the time and frequency domains. Specifically, the *db4* wavelet function, having four decomposition layers, is used to remove the noise hidden in high frequencies.

It is explicit that single stock price data cannot reflect the tendency of stock price variation and are not qualified to forecast the future pricing. Therefore, the data utilized in our model are extracted from the whole data set at all time steps. The time interval is set to 20 days in our current study. After the whole preprocessing, the sequential data would be represented as a $b \times t \times d$ matrix, in which b, t, and d represent the batch size, time step number, and feature dimensionality of the input data, respectively.

Among all the eight adopted FTS variables, the closing price could be the most straightforward one because it impacts investment strategies to a great extent, and thus, the closing price is regarded as one primary prediction target in our study.

3.2. The Proposed LSTM-Based Model Structure. The chaotic, nonstationary, and nonlinear characteristics of stock pricing limit the feasibility of conventional neural networks. RNN

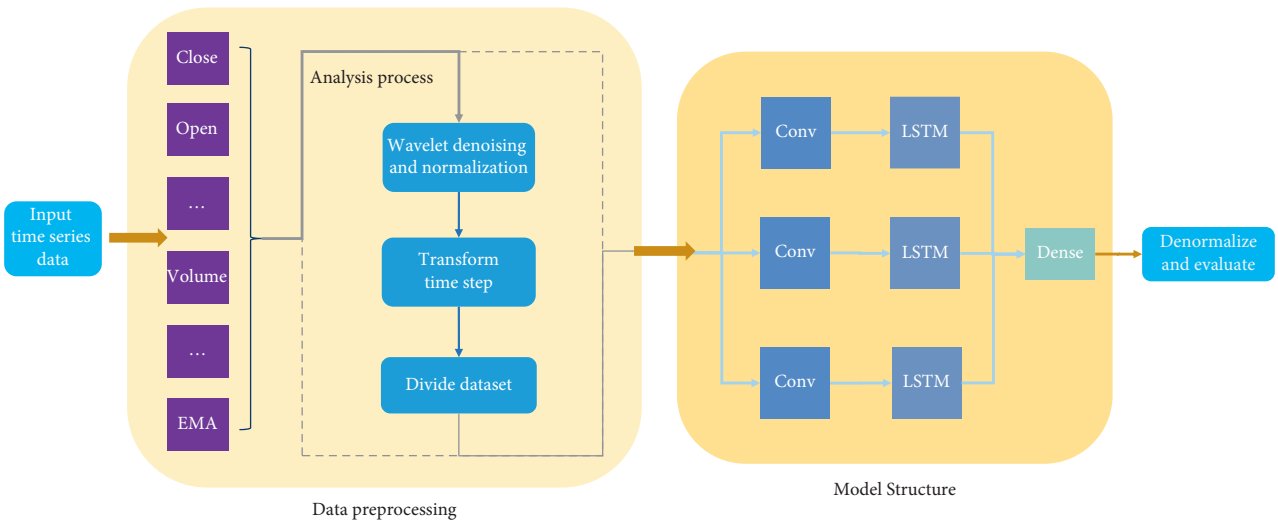

Figure 1: Scheme of our FTS prediction model.

was designed for avoiding long-term dependence problems and using the correlated information between various time intervals when tackling the time series prediction problem. However, original RNN can hardly conquer the challenge of gradient vanishment. Different from classic RNN, the LSTM neural network consisting of special memory cells was proposed by Abedinia et al. [17]. In LSTM, the memory cells are used to replace the hidden layer neurons in RNN, and the states of memory cells play the pivotal roles in the forecasting task. Moreover, the gate structures are utilized to convey the feature information and to update the state of memory cells. As shown in Figure 2, each LSTM cell is composed of three dynamic gates: the input, forget, and output gates. The basic structure of any memory cell includes one add layer, two tanh layers, three sigmoid layers, and three concatenation layers.

The forget gate determines how much cell state information would be discarded from the previous cell. As shown in Figure 2, one memory cell accepts the output of previous information h_{t-1} and external input of x_t in a concatenated vector $[h_{t-1}, x_t]$ via σ transformation, as listed in Equation (1) in which W_f and b_f separately represent the weight matrix and bias of the forget gate. After Equation (1) calculation, f_t ranging from 0 to 1 determines the reserved percentage of the previous cell state C_{t-1}, where 0 indicates the entire abandonment and 1 indicates the entire acceptance.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right). \tag{1}$$

In LSTM's cell model, the input gate can determine the proportion of the new input x_t reserved to generate the eventual cell state C_t. This gate extracts pivotal information from current input as well as prevents unconsidered content from entering current cell. The calculation of the input gate is detailed in Equation (2), where W_i and b_i are separately the weight matrix and bias.

The updated information of cell state $\widetilde{C}_t$ is generated through the tanh layer and using Equation (3) in which W_c and b_c denote the weight matrix and bias, respectively. The current cell state C_t can be obtained using Equation (4).

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right), \tag{2}$$

$$\widetilde{C}_t = \tan h\left(W_c \cdot [h_{t-1}, x_t] + b_c\right), \tag{3}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t. \tag{4}$$

The output gate determines how much cell state C_t can be transformed into the output h_t using Equation (6).

$$O_t = \sigma\left(W_\sigma \cdot [h_{t-1}, x_t] + b_\sigma\right), \tag{5}$$

$$h_t = O_t \times \tan h\left(C_t\right). \tag{6}$$

In our empirical studies, the final output of our designed model is the predicted value of the $(t+1)$-th day's closing stock price using the previous t days' feature data. Mean square error (MSE) measurement is used to constitute the ultimate loss function of our LSTM-based model.

Convolutional LSTM (ConvLSTM) [18] has proven the excellent performance while participating in time series prediction problems because it is good at capturing the spatio-temporal relations well. Moreover, the multiscale feature fusion strategy overall outperforms other basic structures on deep feature extraction. Therefore, to further improve the time series feature extraction effectiveness, in our LSTM-based model (see Figure 1), we devise a multiscale convolutional feature fusion mechanism to extensively extract the features of stock pricing history, i.e., the three-branch structure on the right in Figure 1. However, due to the characteristics of stock pricing data, one-dimensional convolutions are used in our model. It is worth mentioning that the three convolutional layers use the uniform number of filters (e.g., 100), whereas the kernel sizes and strides are set differently, e.g., 6 (kernel size) and 3 (stride) for the first convolutional layer, 12 (kernel size) and 3 (stride) for the second, and 6 (kernel size) and 2 (stride) for the third. As such,

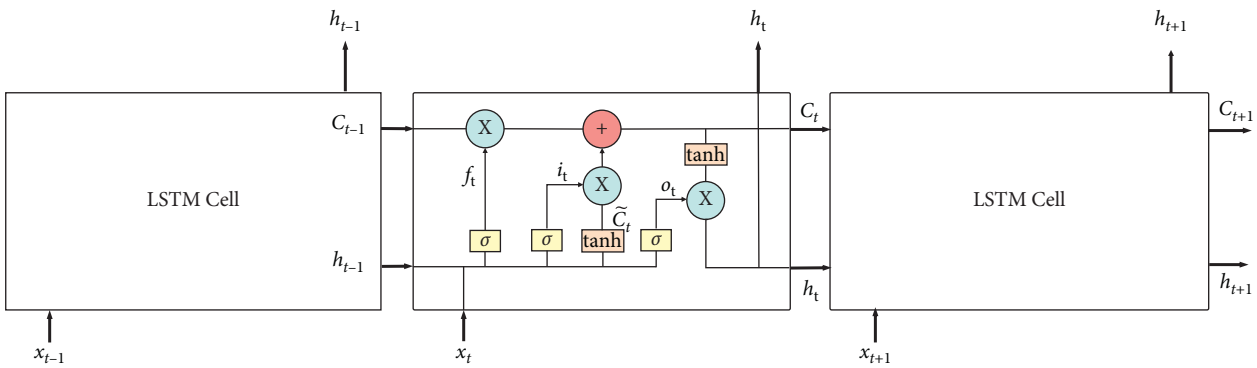

Figure 2: Gate structure of LSTM.

the three convolutional layers input different scales of features into the following LSTMs, and we achieve the goal of FTS feature presentation diversity. Finally, the outputs of the three branches are concatenated as the input of the Dense layer.

3.3. Data Set. As mentioned in Section 3.1, we conducted experiments on three influential stock index data sets, i.e., S&P 500, DJIA and CBOE Volatility Index (VIX). To obtain relatively low-noisy data, the experimental data sets were constituted by discrete time series data coming from the three stock history data sets. As previously introduced, eight FTS variables are used for model training and forecasting, i.e., open, closing, high, low, trading volume, adjusted price, MA, and EMA. Among them, the former six are from the original data sets, and the last two are regenerated using the following equations:

Let

$$\text{Avg}\, M_i = \frac{1}{N}\sum_{i-N}^{i}(\text{close}_i), \tag{7}$$

where $AvgM_i$and $close_i$ represent i-th day's moving average and closing price, respectively, and N denote the time step length. Then,

$$MA = \ln\frac{\text{close}}{\text{Avg}\, M}. \tag{8}$$

Let

$$E_i = \frac{2(\text{close}_i - E_{i-1})}{N+1} + E_{i-1}, \tag{9}$$

where E_i represents i-th day's exponent moving average; then,

$$EMA = \ln\frac{\text{close}}{E}. \tag{10}$$

4. Experimental Studies

4.1. Setup. To evaluate the realistic performance of our devisal, three well-established machine learning algorithms were adopted to make comparisons with our proposed LSTM-based model, including ANN, SVR, and linear regression. Besides, five validity indices were used for performance measurement: mean square error (MSE), mean absolute percentage error (MAPE), mean absolute error (MAE), coefficient of determination (R^2), and directional accuracy (DA). Their detailed definitions are as follows:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i - y_i)^2, \tag{11}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\widehat{y}_i - y_i\right|, \tag{12}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\left|\widehat{y}_i - y_i\right|}{y_i}\right|, \tag{13}$$

$$R^2 = \frac{\sum_{i=1}^{N}(\widehat{y}_i - \overline{y}_i)^2}{\sum_{i=1}^{N}(\widehat{y}_i - \overline{y}_i)^2}, \tag{14}$$

$$DA = \frac{1}{n}\sum_{i=1}^{n} a_i, \tag{15}$$

in which N represents the sample size, $y_i$represents the truth value, $\overline{y}_i$ represents the mean value of the truth value, and $\widehat{y}_i$ represents the predicted value. a_i in Eq. (15) signifies whether the rising and falling forecasts are correct, and 0 for false and 1 for true.

As mentioned in Section 3, for the three input convolutional layers in Figure 1, we uniformly set the filter number to 100, employing different kernel sizes and strides. Specifically, the upper convolutional layer has the kernel size 6 and stride 3, the medium layer has the kernel size 12 and stride 3, and the bottom layer has the kernel size 6 and stride 2. The Leaky ReLU was used as the activation function in these convolutional layers.

Our model was trained by the Adam optimizer with an initial learning rate of 2e-6, which drops every five iterations at 0.95. Meanwhile, three adjacent LSTM layers were equipped with 128 units and 0.2 dropout rate.

In addition, ANN was also trained using the Adam optimizer, consisting of five dense layers with units 500, 500, 250, 250, and 1, respectively. SVR employed the radial basis function (RBF) as the kernel function.

Table 1: Experiment of different time step lengths of LSTM on S&P 500.

Time step	MSE	MAE	MAPE	R^2
5	$4.9663e-3$	$4.6879e-3$	$3.7178e-3$	0.9747
10	$6.3305e-5$	$5.7630e-3$	$4.4984e-3$	0.9804
20	$3.7410e-5$	$4.1263e-3$	$3.2351e-3$	0.9645
30	$2.2613e-5$	$3.6213e-3$	$2.6458e-3$	1.0343
40	$1.4465e-4$	$9.2449e-3$	$7.2079e-3$	0.9133

Table 2: Performance comparisons among four methods on S&P 500.

Method	MSE	MAE	MAPE	R^2
Proposed LSTM-based model	$2.2613e-5$	$3.6213e-3$	$2.6458e-3$	1.0343
ANN	$2.9482e-5$	$4.7260e-3$	$1.6978e-3$	1.0299
SVR	$4.4264e-3$	$6.5222e-2$	$4.7541e-3$	0.7755
Linear regression	$1.0189e-4$	$9.9822e-3$	$7.2890e-3$	1.0087

Table 3: Performance comparisons among four methods on DJIA.

Method	MSE	MAE	MAPE	R^2
Proposed LSTM-based model	$1.2958e-3$	$8.8381e-3$	$5.9901e-3$	0.9721
ANN	$5.1614e-4$	$9.9758e-3$	$6.7933e-3$	0.9605
SVR	$3.7029e-3$	$5.6042e-2$	$3.7629e-2$	0.7696
Linear regression	$1.0295e-4$	$9.3711e-3$	$6.3274e-3$	0.9899

Table 4: Performance comparisons among four methods on VIX.

Method	MSE	MAE	MAPE	R^2
Proposed LSTM-based model	$3.0961e-5$	$3.8656e-3$	$9.9029e-3$	0.9958
ANN	$1.4427e-4$	$2.0201e-2$	$6.4719e-2$	1.0222
SVR	$2.3282e-3$	$4.3949e-2$	$1.2084e-1$	0.9218
Linear regression	$8.7053e-5$	$6.0373e-3$	$1.5141e-2$	0.9736

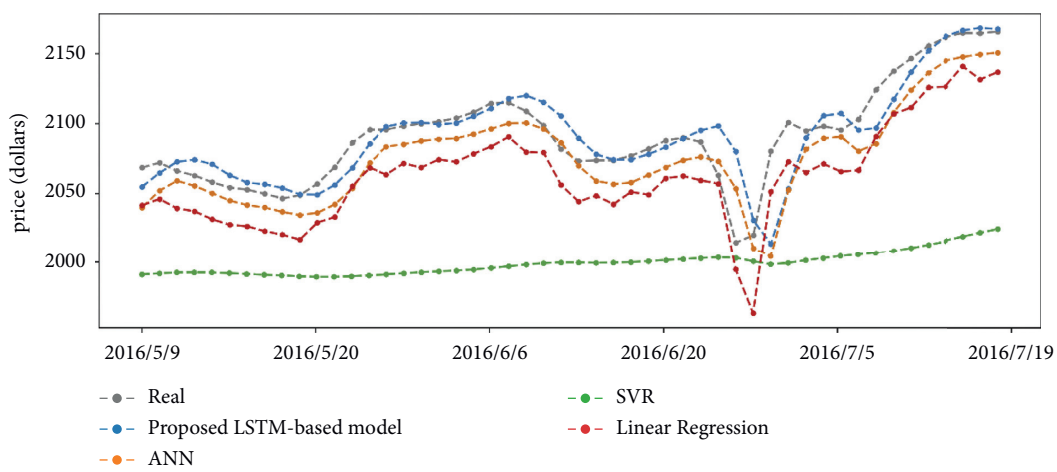

Figure 3: Comparison of predicted closing prices of four methods on the S&P 500 data set.

4.2. Experimental Results and Analyses. The time step length of time series is usually a core parameter determining FTS forecasting efficiency. Table 1 shows the relationships of various time step numbers with LSTM's realistic performance. Usually, overlong time step lengths are prone to the gradient vanishment, whereas too short ones easily lose the vital information embedded in time series. As revealed in Table 1, the step-length 5 obtained the worst score, which implies that it is too short for FTS forecasting tasks, whereas the step-length 40 seemed overfitting a bit. Generally, he

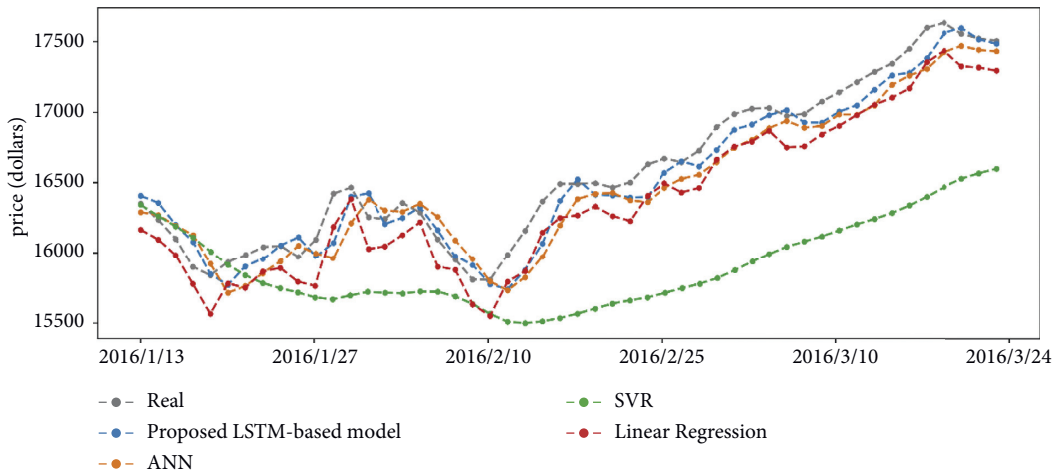

Figure 4: Comparison of predicted closing prices of four methods on the DJIA data set.

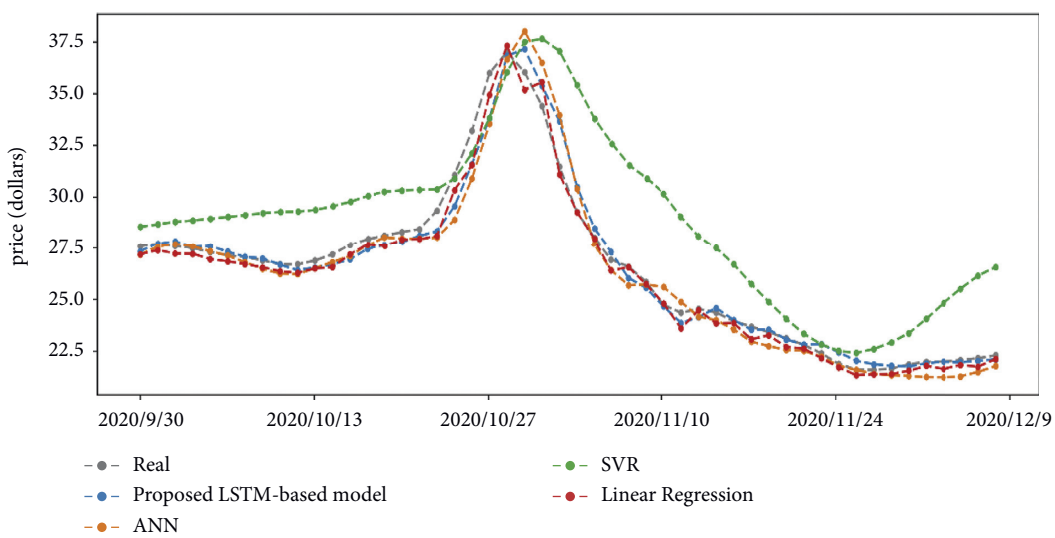

Figure 5: Comparison of predicted closing prices of four methods on the VIX data set.

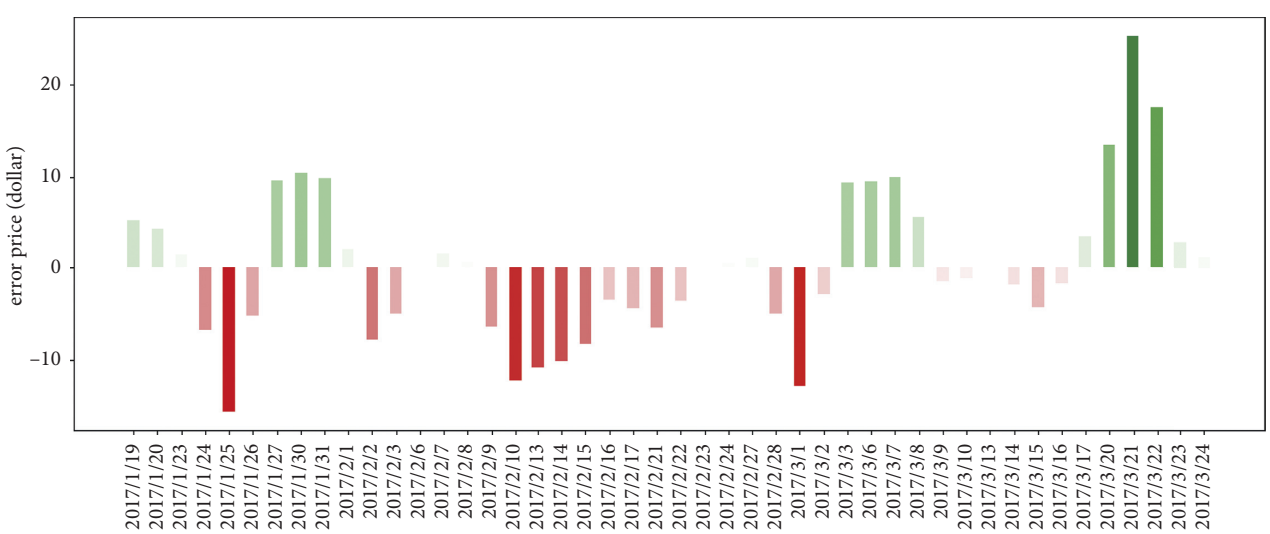

Figure 6: Difference between prediction and ground truth of our method on the S&P 500 data set.

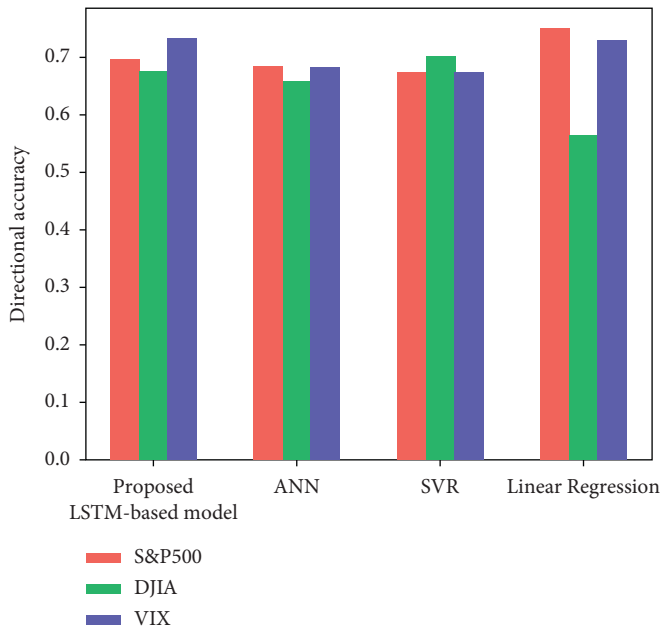

Figure 7: Comparisons of DA index of different methods.

step-length 30 obtained the best performance. Thus, we set the time step length to 30 in our empirical studies.

Tables 2–4 specifically display the obtained scores of the four methods on three stock data sets by means of used validity indices. Generally, our LSTM-based model scores best on all indices, particularly on S&P 500 and VIX data set. Despite that the results of our method on DJIA are not overwhelming, they still rank at top 2.

Figures 3–5 further intuitively show the predicted trends of closing prices on three stock data sets. As is revealed, the results of the proposed LSTM-based method are closer to the real market trends than those of the others, as overall the blue lines have the universality of the smallest offset/deviation from the ground truth. By the way, some blue lines even overlap with the grey lines at some points, which implies the preferable sensitivity to small variations of our designed, multiscale convolutional feature fusion-based LSTM model.

Because the stock price is almost noncyclical, it is reasonable that the predicted results of all employed methods have the characteristic of hysteresis. The adoption of wavelet transform can lighten such influence to a certain extent by removing noise as well as retaining inherent features; however, it is not enough. In our proposed method, with embedding the multiscale feature fusion mechanism, it is distinct that the hysteresis of our method is less than that of ANN. As the evidence, in Figure 3, the prediction curve of the ANN is approximately the back translation of our LSTM-based method. Figures 3–5 also illustrate that the overall accuracy and stability of our method are better than those of the other methods.

Figure 6 displays the bias between our LSTM-based model's predicted closing prices and the given closing prices on S&P 500 data set. In this figure, the green bar represents the case where the prediction is higher than the truth, whereas the red represents the inverse case. As is shown, the highest bias is less than 30 and generally varies between -10 and 10. Compared with the stock price high up to 2,500, these deviations actually reflect the forecasting stability and effectiveness of our method.

The DA index can reflect whether the forecasted trend conforms with the real movement tendency of closing prices of stocks. The rising or falling tendency is another remarkable indicator in stock trading. Hence, we utilized DA to prove the superiority of our efforts, as shown in Figure 7. We also achieve the similar conclusion that overall, our LSTM-based model has the higher prediction accuracy on stock tendency changes.

All the above results and analyses indicate that the dedicated gate structure of LSTM as well as the proposed multiscale feature fusion strategy greatly warrant the desirable preferable performance of our method in forecasting stock trends.

5. Conclusion

In this paper, we propose a multiscale convolutional feature fusion-based LSTM model to address the challenge of forecasting stock trends. With experiments on three classic stock data sets, it has been proved that the proposed method has superior effectiveness and stability than a few other state-of-the-art methods. For future study, we will contribute to further improving the prediction accuracy based on other deep learning techniques.

Data Availability

The labeled data sets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by Nanjing Forestry University.

References

[1] S. Akhter and M. A. Misir, "Capital markets efficiency: evidence from the emerging capital market with particular reference to Dhaka stock exchange," *South Asian Journal of Management*, vol. 12, pp. 35–51, 2005.

[2] M. Nabipour, P. Nayyeri, H. Jabani, and A. Mosavi, "Deep learning for stock market prediction," *Entropy*, vol. 22, 2020.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, 1997.

[4] N. S.Timbo, S. Labidi, T. P. d. Nascimento, M. L. Lima, G. Nunes Neto, and R. C.Matos, "Approach based on linear regression for stock exchange prediction–case study of Petr4 petrobrás, Brazil," *International Journal of Artificial Intelligence & Applications*, vol. 7, no. 1, pp. 21–31, 2016.

[5] Y. Lin, H. Guo, and J. Hu, "An svm-based approach for stock market trend prediction," in *Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN)*, Dallas, TX, USA, August 2013.

[6] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," *Neural Computing & Applications*, vol. 32, no. 6, pp. 1609–1628, 2020.

[7] B. W. Wanjawa and L. Muchemi, "Ann model to predict stock prices at stock exchange markets," 2014, https://arxiv.org/abs/1502.06434.

[8] L. Bai, S. Yan, X. Zheng, and B. M. Chen, "Market turning points forecasting using wavelet analysis," *Physica A Statistical Mechanics & Its Applications*, vol. 437, pp. 184–197, 2015.

[9] Z. Hajiabotorabi, A. Kazemi, F. F. Samavati, and F. M. M. Ghainia, "Improving DWT-RNN model via B-spline wavelet multiresolution to forecast a high-frequency time series," *Expert Systems with Applications*, vol. 138, Article ID 112842, 2019.

[10] Z. Zhao, R. Rao, S. Tu, and J. Shi, "Time-weighted LSTM model with redefined labeling for stock trend prediction," in *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, Boston, MA, USA, November 2017.

[11] H. Seng and J. C. Young, "Predicting LQ45 financial sector indices using RNN-LSTM," *Journal of Big Data*, vol. 8, 2021.

[12] A. H. Bukhari, M. Raja, M. Sulaiman, S. Islam, and P. Kumam, "Fractional neuro-sequential arfimalstm for financial market forecasting," *IEEE Access*, vol. 8, p. 1, 2020.

[13] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism," *PLoS One*, vol. 15, no. 1, Article ID e0227222, 2020.

[14] R. Zhang, Z. Yuan, and X. Shao, "A new combined CNN-RNN model for sector stock price analysis," in *Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Tokyo, Japan, July 2018.

[15] H. Hao, Y. Wang, Y. Xia, J. Zhao, and F. Shen, "Temporal convolutional attention-based network for sequence modeling," 2020, https://arxiv.org/abs/2002.12530.

[16] O. Dessaint, T. Foucault, L. Frésard, and A. Matray, *Noisy Stock Prices and Corporate Investment*, Social Science Electronic Publishing, Rochester, NY, USA, 2019.

[17] O. Abedinia, N. Amjady, and H. Zareipour, "A new feature selection technique for load and price forecast of electrical power systems," *IEEE Transactions on Power Systems*, vol. 32, p. 1, 2016.

[18] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Montreal, Canada, December 2015.