WILEY | Hindawi

Research Article

A Stock Closing Price Prediction Model Based on CNN-BiSLSTM

Haiyao Wang,[1] **Jianxuan Wang,**[2] **Lihui Cao,**[3] **Yifan Li,**[2] **Qiuhong Sun,**[2]
and Jingyang Wang [iD][2]

[1]*School of Ocean Mechatronics, Xiamen Ocean Vocational College, Xiamen 361100, China*
[2]*School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China*
[3]*The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China*

Correspondence should be addressed to Jingyang Wang; jingyangw@hebust.edu.cn

As the stock market is an important part of the national economy, more and more investors have begun to pay attention to the methods to improve the return on investment and effectively avoid certain risks. Many factors affect the trend of the stock market, and the relevant information has the nature of time series. This paper proposes a composite model CNN-BiSLSTM to predict the closing price of the stock. Bidirectional special long short-term memory (BiSLSTM) improved on bidirectional long short-term memory (BiLSTM) adds $1 - \tanh(x)$ function in the output gate which makes the model better predict the stock price. The model extracts advanced features that influence stock price through convolutional neural network (CNN), and predicts the stock closing price through BiSLSTM after the data processed by CNN. To verify the effectiveness of the model, the historical data of the Shenzhen Component Index from July 1, 1991, to October 30, 2020, are used to train and test the CNN-BiSLSTM. CNN-BiSLSTM is compared with multilayer perceptron (MLP), recurrent neural network (RNN), long short-term memory (LSTM), BiLSTM, CNN-LSTM, and CNN-BiLSTM. The experimental results show that the mean absolute error (MAE), root-mean-squared error (RMSE), and R-square (R^2) evaluation indicators of the CNN-BiSLSTM are all optimal. Therefore, CNN-BiSLSTM can accurately predict the closing price of the Shenzhen Component Index of the next trading day, which can be used as a reference for the majority of investors to effectively avoid certain risks.

1. Introduction

Stock predicting research is an applied research direction of financial big data. With the rapid growth of China's economy and the continuous expansion of the financial market, more and more investors have begun to pay attention to the methods to improve return on investment and effectively avoid certain risks. Among these methods, the stock price prediction is of great significance in the commercial and financial fields [1, 2]. In the face of the rise and fall of stock price, investors will get unpredictable profits and even losses, so it has become an issue of concern for investors to predict stock price and select stock worthy of investment. In view of the complexity and instability of the stock market [3], a large number of variables and information sources need to be considered in the process of stock price prediction, which is a very difficult task, and are still the focus and discussion in

the financial sector [4]. The traditional analysis method is to use the existing stock data and relevant technical charts, combined with the investor's own experience to predict the stock price. But this method is not applicable in today's increasingly large and complex stock market. In addition to low efficiency and excessive reliance on manual experience, there are also a series of problems such as poor integrity of stock content information and feature data redundancy. The utilization rate of stock data is low, and the effect is not good, so it is difficult to meet the needs of market development.

Many factors affect the changing trend of the stock market, and the trend of stock price fluctuation which is showing a nonlinear change law is very complex, so it is often very difficult to predict the stock market [5]. With the increasing availability of high-frequency trading data and the increasing popularity of artificial intelligence, deep learning is favoured as an "upgraded version" of existing

models and methods without relying on econometric assumptions and expert experience [6, 7]. Deep learning neural network has a good fitting ability for nonlinear function relations [8, 9]. Building a deep neural network to predict the trend and price of stock has been widely concerned by people, and some scholars have also carried out in-depth research on this aspect [10–12]. In 2010, Nair et al. built a denoising hybrid stock price prediction model based on decision tree [13]. Firstly, the model was used to extract the relevant features of stock data, and then the decision tree algorithm was used to select the extracted features. Then the principal component analysis (PCA) algorithm was used to reduce the dimension. The reduced dimension data were input into a fuzzy model for stock price prediction. In 2016, Wang et al. used the support vector machine (SVM) to build a model to predict the trend of the CSI 300 index and verified the validity of the support vector machine in stock price index prediction. [14]. In 2019, Hoseinzade and Haratizadeh proposed a framework based on CNN and predicted the trend of the S&P 500 index, Nasdaq index, Dow Jones index, New York Stock Exchange index, and Russell index on the next day. The results show that the prediction performance is higher than the baseline algorithm. [15].

To predict stock closing price more accurately, this paper proposes a stock prediction model based on CNN-BiSLSTM, which uses stock data of the last five trading days to predict the closing price of the next trading day. BiSLSTM improved the output gate based on the BiLSTM model. The model consists of CNN and BiSLSTM. CNN is used to extract the characteristics of stock data, and BiSLSTM is used to predict the stock closing price. Compared with BiLSTM, BiSLSTM can make the output value of the output gate more accurate. CNN-BiSLSTM can more accurately predict the stock closing price of the next trading day, which can be used as a reference for the majority of investors to effectively avoid certain risks. The main contributions are as follows:

(1) CNN is proposed to extract the feature that affects the stock price. One-dimensional CNN can be well applied to time series analysis. The one-dimensional convolutional layer extracts advanced data features from sample data, makes full use of the feature information of the input data, adopts local links, weight sharing, and space or time-related downsampling method to gain better features, makes the extracted features more distinguishable, and improves the accuracy of model prediction results when the closing price of stock is being predicted.

(2) BiSLSTM is proposed to predict the closing price of stocks. BiSLSTM which is improved on BiLSTM adds $1 - \tanh(x)$ function to its output gate, so that the value range of the output gate is finally $(0.24, 1)$. Therefore, BiSLSTM not only has the strong learning ability of BiLSTM, but also has a better fitting effect than BiLSTM in model training process. As a result, BiSLSTM is suitable for analyzing the relationship between time series data.

2. Related Work

Artificial neural network (ANN) has been proved to be able to deal with complex nonlinear problems well, but the testing and training speed of neural networks are slow [16]. In addition, overfitting and falling into the local minimum are the disadvantages of neural networks. Huang et al. took LSTM as the main model of stock prediction and adopted the Bayesian optimization method to dynamically select parameters to determine the optimal number of units, and the prediction accuracy was improved by 25% compared with traditional LSTM [17]. Gunduz et al. sent relevant technical indicators of each sample into CNN to improve the accuracy of prediction [18]. Generalized autoregressive conditional heteroskedasticity (GARCH) is a classic model widely used in time series prediction, and GARCH assumes that values for time series are a linear generation process. However, market features are nonlinear, so making GARCH assumption is unsuitable for many financial time series applications [19]. Wen et al. proposed a new method to simplify noisy-filled financial temporal series via sequence reconstruction by leveraging motifs (frequent patterns) and then utilized a CNN to capture the spatial structure of time series [20]. But most conventional time series analysis studies rely on the linear relationship between stock prices, which is more suitable for sequences with stable trends and regular, so this relationship makes them insufficient to deal with more complex nonlinear relationships. However, stock price shows the feature of uncertainty and nonlinearity, and the influencing factors of stock price volatility are very complex. All of these are ignored in simple time series analysis. As a result, the prediction effect is poor [21].

After RNN was proposed, most scholars found that the RNN would forget the previous state information over time, and then the LSTM was proposed. In deep learning, the LSTM network structure is suitable for learning data of time type and is widely used in various tasks of time series analysis [22, 23]. LSTM is better than the traditional recurrent neural network [24, 25]. It overcomes the problem of gradient disappearance or gradient explosion [26]. Many financial time series studies use LSTM modelling [27]. Zhang et al. used the generative adversarial network (GAN) to predict the stock market [28]. MLP was used as the discriminator and LSTM network as the generator to predict the closing price. This is a breakthrough of a new method, which is worth further deepening and improving. The advantage of this method is that it can capture the time series feature of stock data. Akita et al. used the text data of Nikkei News as the input of LSTM, combined with market time series numerical data to predict the opening prices of 10 companies [29]. Under the simulated trading strategy, a model trained with numerical data and text data was used, which could obtain a higher profit rate than a model trained with only numerical data. Hyun et al. proposed a stock price prediction model based on CNN. Nine technical indicators were selected as predictors of the prediction model, and the technical indicators were converted into images of time series graph to verify the applicability of the new learning method in the stock market [30]. Yang et al. proposed a hybrid prediction method based on LSTM and ensemble empirical mode decomposition

(EMD). Firstly, the comprehensive EMD method was used to decompose the complex original stock price time series into several subsequences, and then the LSTM method was used to train and predict each subsequence. Finally, we obtained the prediction values of the original stock price time series by fusing the prediction values of several subsequences [31]. Lu et al. proposed a CNN-LSTM-based model to predict stock prices. CNN was used to efficiently extract features from the data, and LSTM was used to predict the stock price with the extracted feature data. This forecasting method not only provided a new research idea for stock price forecasting but also provided practical experience for scholars to study financial time series data [32].

3. CNN-BiSLSTM

3.1. Convolutional Neural Network. CNN was proposed by Lecun et al. in 1998 [33]. CNN is a multilayer neural network structure with a deep supervised learning structure, which is able to process time series data and image data. Since CNN has been successfully applied to the preprocessing of two-dimensional images, the same idea can also be used to process one-dimensional data [34]. CNN uses a small number of parameters to capture the features of input data and combine them to form advanced data features. Finally, these advanced data features are put into the full connection layer for further regression or classification prediction. The typical CNN structure consists of the input layer, convolutional layer, pooling layer, fully connected layer, and output layer. Among them, the convolution layer mainly performs convolution operations on the samples through the convolution kernel to obtain the input of the next layer. The pooling layer is an important part of CNN, which can effectively reduce the number of model parameters and reduce the complexity of operations while the useful information of the feature map is retained. CNN can extract data features through layer-by-layer convolution and pooling operations. The filter can set appropriate window size and window sliding step size according to the size of the input data and the need to extract features.

In the one-dimensional convolutional neural network, a one-dimensional array is used as the convolution kernel. In the traditional two-dimensional backpropagation algorithm, the dimensions need to be adjusted to match the convolution kernel. In the process of forward propagation, the output of the current convolutional layer l can be expressed as follows:

$$x_j^l = f\left(\sum_{i=1}^{I} x_i^{l-1} \otimes k_{ij}^l + b_j^l\right),\tag{1}$$

where x_j^l is the output feature map of the j-th neuron of the current layer (layer l); l is the number of input features of the l-th convolutional layer; x_i^{l-1} is the output feature map of the previous layer (layer l-1), x_i^{l-1} is also the input of the current l layer; $\otimes$ represents the convolution operation; k_{ij}^l represents the convolution kernel of the i-th neuron of the l-1 layer to the j-th neuron of the l layer; b_j^l is the j-th neuron of the l layer standard deviation; and f is the activation function, which is obtained by using the following formula:

$$f(x) = \frac{1}{1 + e^{-x}}.\tag{2}$$

As a subsampling layer, the pooling layer can ensure the invariance of the mapping, and max-pooling can be expressed as follows:

$$y_i^l = \max\text{ pooling}\left(x_i^{l-1}, s_{\text{scale}}, s_{\text{stride}}\right),\tag{3}$$

where y_i^l is the output of the i-th neuron of the current layer l; max-pooling () is the down-sampling function, taking the maximum value within a certain range; s_{scale} is the scale of pooling; and s_{stride} is the step length of pooling.

3.2. Long Short-Term Memory. LSTM was first proposed by Hochreater and Schmidhuber in 1997 [35]. In 2000, Gers et al. improved the LSTM network and proposed the forget gate method, which was suitable for continuous prediction [36]. Later in 2012, Grave improved and promoted LSTM [37]. On many issues, LSTM has achieved considerable success and has been widely used.

The predecessor of LSTM is RNN. RNN is a neural network that learns sequence patterns through internal loops. In the RNN backpropagation process, the value is propagated back to the activation function, so the slope will become extremely small or extremely large, and the problem of gradient disappearance or gradient explosion occurs. In 2013, Hochreiter et al. proposed memory cells and gates, and these gate structures could solve the gradient problem of RNN and add or delete cell information [38]. Such gate structures could store information for a long time, and unnecessary information was forgotten [39, 40]. The LSTM uses memory units instead of neurons. The structure of LSTM memory cell is shown in Figure 1. The LSTM cell consists of a memory cell (C_t) and three gate structures. The three gate structures include input gate (i_t), forget gate (f_t), and output gate (o_t). The input gate is used to calculate the input information at that moment and control the input of new information into the internal memory unit. The forget gate is used to control the internal memory unit, which needs to save the information of the previous time. The output gate is used to control the amount of information output by the internal memory unit.

In Figure 1, x is the input; h is the hidden state that gives the network memory ability; and the subscripts $t-1$ and t represent different time steps. The connections between its nodes form a directed graph along the sequence, and h_t is calculated based on the output of the hidden state of the previous layer and the input of the current moment. The calculation principle of LSTM is as follows.

Firstly, the value of the input gate i_t is calculated by using formula (4), and the candidate state value $\widetilde{C}_t$ of the input cell at time t is calculated using formula (5):

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right),\tag{4}$$

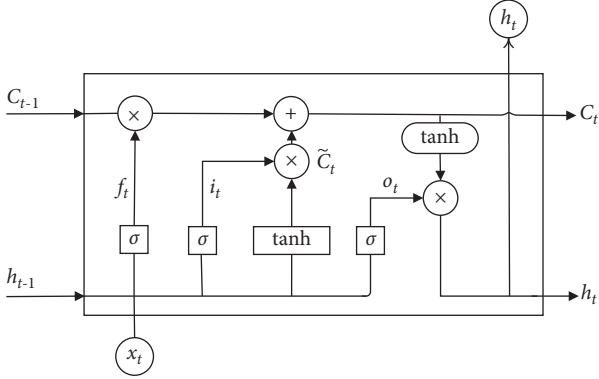$$\widetilde{C}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right).\tag{5}$$

FIGURE 1: Schematic diagram of the LSTM structure.

Secondly, the following formula is used to calculate the activation value of forget gate f_t at time t:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right). \tag{6}$$

Thirdly, the original information and the newly increased information are, respectively, controlled by the forget gate and the input gate. The $i_t, \widetilde{C}_t$, and f_t, calculated in the first two steps, are used to calculate the updated value C_t of the cell state at time t using the following formula:.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t. \tag{7}$$

After the new cell state is obtained, formula (8) is used to calculate the output gate value, and the updated memory cell uses formula (9) to calculate the current hidden state h_t:

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_0\right), \tag{8}$$

$$h_t = o_t * \tanh\left(C_t\right). \tag{9}$$

In formulas (4)–(9), W_i, W_c, W_f, and W_o represent four different matrix weights, b_i, b_c, b_f, and b_0 represent the offset, σ is the sigmoid function, and the symbol $*$ represents the vector outer product.

Finally, backpropagation is performed to obtain the LSTM, which composed of these storage blocks. Through the above calculation, the LSTM can effectively use the input time series data to make it have the function of long-term memory.

3.3. Bidirectional Long Short-Term Memory. Although LSTM can obtain the feature information of long distance, the obtained information is the information before the output time, and it does not use the reverse information. In time series prediction, the forward and backward information law of time series data should be fully considered, which can effectively improve the prediction accuracy. BiLSTM consists of two LSTM, forward and reverse. Compared with the one-way-state transmission in the standard LSTM, BiLSTM considers the changing laws of the data before and after data transmission and can make more complete and detailed decisions using the past and future information. It has shown superior performance. BiLSTM consists of forward

calculation and backward calculation, from the BiLSTM structure diagram in Figure 2. In Figure 2, the horizontal direction arrow indicates the two-way flow of time series information in the model, while the data information flows in one direction vertically from the input layer to the hidden layer to the output layer.

3.4. CNN-BiSLSTM. CNN-BiSLSTM is a hybrid of CNN and BiSLSTM. BiSLSTM is improved on BiLSTM, and $1 - \tanh(x)$ function is added to the output gate, so that the value range of the output gate is about (0.24, 1). Therefore, BiSLSTM not only has the strong learning ability of BiLSTM, but also has a better fitting effect than BiLSTM in the model training process. As a result, BiSLSTM is suitable for analyzing the relationship between time series data. SLSTM unit structure diagram is shown in Figure 3. CNN-BiSLSTM network structure is shown in Figure 4. The stock historical trading information is time series and belongs to time series data. In the CNN-BiSLSTM, CNN is used to extract the local features of the data layer by layer. Advanced features with strong expression ability can be extracted from the data, effectively avoiding subjectivity and limitations of manual feature extraction. The BiSLSTM has the feature of retaining contextual historical information for a long time, which can realize feature extraction of time dimension and long-distance dependent data. In addition, BiSLSTM can mine the long-term time series relationship between the influencing factors of stock and the closing price. Therefore, the data from the CNN output place are put into the BiSLSTM to model the bidirectional time structure through the calculation of formulas (10)–(15) where f_t is used as the forgetting gate, sigmoid function σ is used to judge whether the past memory needs to be retained for the current memory state through formula (12); i_t is used as the input gate to calculate whether the current input data are worth retaining through formula (10); $\widetilde{C}_t$ is used to calculate the data that need to be updated by formula (11), and i_t is used to control whether it needs to be updated or not; and C_t calculates whether the state at the current moment needs to be updated by formula (13). After the new state is obtained, formula (14) is used to calculate the output gate value O_t; compared with BiLSTM, BiSLSTM adds $1 - \tanh(x)$ function here. The updated memory cell can calculate the current hidden state h_t through the following formula:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right), \tag{10}$$

$$\widetilde{C}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right), \tag{11}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \tag{12}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t, \tag{13}$$

$$o_t = 1 - \tanh\left(\sigma\left(W_o \cdot [h_{t-1}, x_t] + b_0\right)\right), \tag{14}$$

$$h_t = o_t * \tanh\left(C_t\right). \tag{15}$$

Since BiSLSTM is composed of two SLSTM, one is forward and the other is backward, and the above calculation needs to be calculated in reverse. Finally, through the full connection layer, we calculate the closing price of the stock and make a more accurate forecast.

4. Experiments

4.1. Experimental Environment. To verify the effectiveness of the proposed model, Shenzhen Component Index is used as the experimental data in the experiment. All experiments are implemented on a computer equipped with Intel Core i5-6300HQ 2.30 GHz, 12.0 GB RAM, NVIDIA GeForce GTX 960m, and Windows 10 64-bit operating system. In this experiment, Python 3.7 is used as the programming language, PyCharm and Anaconda3 are used as the development tools, and Keras based on TensorFlow is used to construct the network model structure.

4.2. Experimental Data. Shenzhen Component Index is used as historical data for stock prediction in the experiment. Shenzhen Component Index is a constituent stock index compiled by Shenzhen Stock Exchange. It is a weighted stock index calculated by taking 40 representative listed companies from all listed stocks as the researching object and taking the outstanding shares as weight, which comprehensively reflects the stock price trend of A and B shares listed on Shenzhen Stock Exchange. The data used in the experiment come from the Wind-Economic database. The software ensures the accuracy of the data from the data source. The experimental data use the historical data of Shenzhen Component Index from July 1, 1991, to October 30, 2020. Some experimental data are shown in Table 1.

4.3. Experiment Process. The CNN-BiSLSTM is used to predict the stock closing price, and the experimental process is as follows:

(1) Perform preprocessing operations on experimental data, remove irrelevant items, serialize time data, standardize data, and divide training set and testing set.

(2) Input the preprocessed time series data into the CNN-BiSLSTM model for training. The training process is shown in Figure 5.

(3) Input the testing sample data into the trained model for prediction.

(4) Restore the predicted data through standardized formulas.

(5) Generate a comparison image between the true value and the predicted value of the stock closing price, and evaluate the prediction effect of the model through the true value and the predicted value.

4.4. Experimental Data Preprocessing. Firstly, the original data are checked, and the missing data are filled or eliminated to facilitate the training and testing of the model. For some special reasons, some intermittent data are vacant. Considering that the data are serial data. The data do not change much from one trading day to the next trading day. So, the average value of the data of the previous trading day and the next trading day will be used to make up. Secondly, the Chinese stock market stipulates that the market is closed all day on Saturdays, Sundays, and major holidays. Therefore, all data at these time nodes are removed, and only the trading day data are retained. Considering that some data in the data set have nothing to do with stock price prediction, they are excluded. The data of the index opening price, highest price, lowest price, closing price, volume, turnover, ups and downs, and change are selected as the influencing factors of stock closing price.

4.5. Experimental Model and Parameters. In this experiment, MLP, RNN, LSTM, BiLSTM, CNN-LSTM, and CNN-BiLSTM are used to compare with CNN-BiSLSTM. CNN-BiSLSTM model parameter settings are shown in Table 2. The comparison model parameters are the same as some of the CNN-BiSLSTM model parameters.

The model training parameters CNN-BiSLSTM used in this experiment are exactly the same as the comparison model. The sequence length is 5, and the delay is 1. The optimizer uses Adam, which not only calculates the adaptive parameter learning rate based on the mean of the first moment as the RMSProp algorithm, but also makes full use of the mean of the second moment of the gradient. The learning rate is 0.0001, and the loss function uses MAE. MAE is the sum of the absolute values of the difference between the true and predicted values. It only measures the mean modulus length of the predicted value error, without considering the direction, and has better robustness to outliers. Batch_size is 64, and epochs is 50.

4.6. Model Training and Prediction. The selected 6878 stock data are divided into training set and testing set, among which the training set is the first 6078 and the testing set is the last 800. Since the magnitude of data in different dimensions is not at the same level, the z-score standardization method is used to convert the data of different orders of magnitude in training set and testing set into the same level. The standardized operation is shown in the following formula:

$$y_i = \frac{x_i - \overline{x}}{s}, \tag{16}$$

where y_i is the standardized value, x_i is the input data, $\overline{x}$ is the average value of the data, and s is the standard deviation of the data.

After the parameters are set, CNN-BiSLSTM is initialized, and the training set data standardized by z-score are put into the model. The forward calculation of the neural network is performed. The model structure is shown in Figure 6. After the calculation is completed, MAE is used to calculate the error between the result of the forward calculation and the true value, and then the Adam algorithm is used for backpropagation to update the weight parameters. The CNN-BiSLSTM stock prediction model is obtained through repeated training of 6078 training samples.
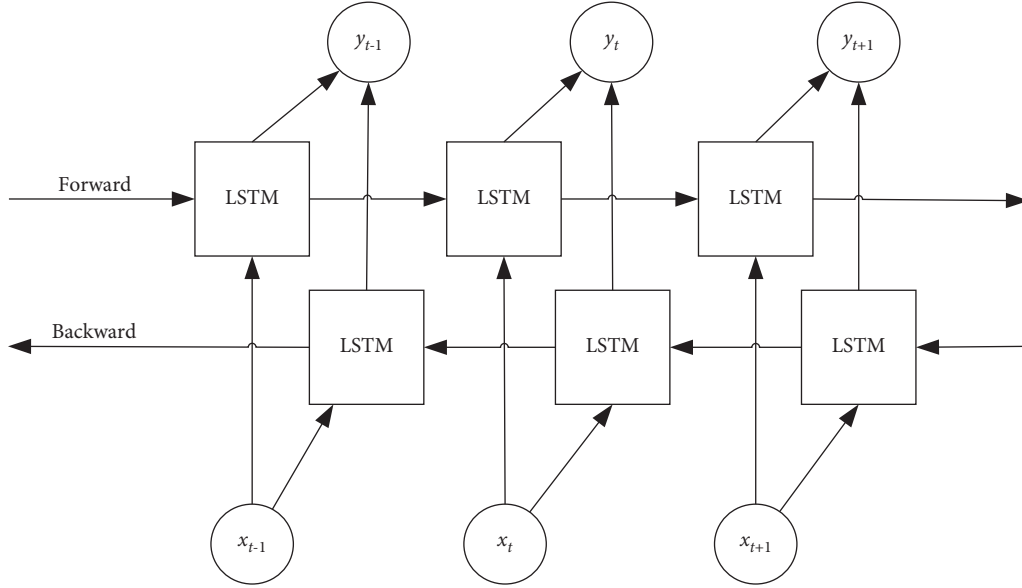
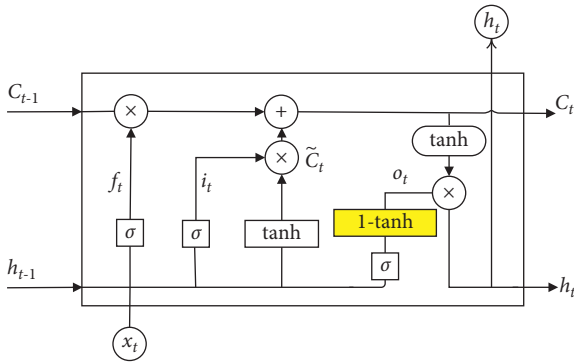Figure 2: Schematic diagram of BiLSTM structure.

Figure 3: SLSTM unit structure diagram.

The data of the testing samples are put into the CNN-BiSLSTM after the training for prediction. Since the data in the testing set are standardized data, formula (17) is required to restore the data. MAE, RMSE, and R^2 are used to evaluate the predicted value and the true value after restoration:

$$x_i = y_i * s + \overline{x}. \tag{17}$$

4.7. Analysis of Results. The preprocessed stock data are put into the CNN-BiSLSTM, MLP, RNN, LSTM, BiLSTM, CNN-LSTM, and CNN-BiLSTM models for training. After the training is completed, the divided testing set is used for prediction. The comparison result of the predicted value and the true value in the last 200 days is shown in Figures 7–13. Models' evaluation index contrast is shown in Table 3.

From Figures 7–13, among MLP, RNN, LSTM, BiLSTM, CNN-LSTM, CNN-BiLSTM, and CNN-BiSLSTM, we can conclude that the errors between the predicted and true value of CNN-BiSLSTM have the best degree of small fitting, and the MLP model has the largest error and the worst fitting degree.
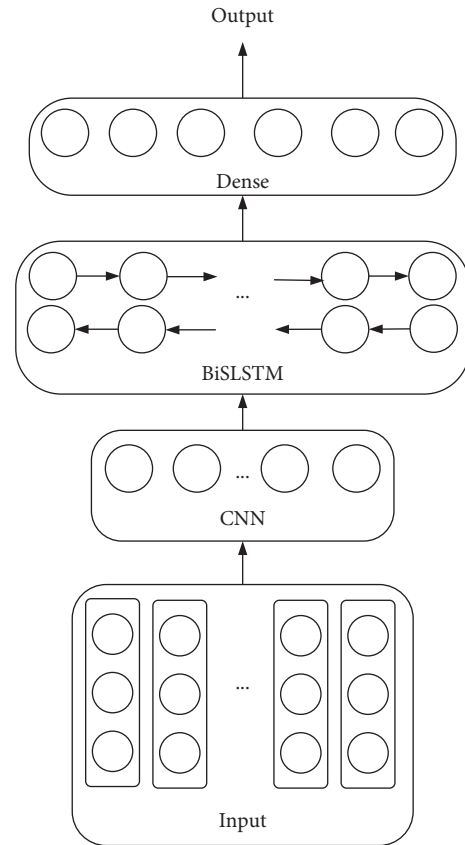

Figure 4: Schematic diagram of CNN-BiSLSTM structure.

The basic evaluation indicators of the regression model are used in the experiment. The basic evaluation indicators of the regression model include MAE, RMSE, and R^2. These three indicators are used to measure the error between the predicted value and the true value.

TABLE 1: Part of the experimental data.

Date	Opening price	Highest price	Lowest price	Closing price	Volume	Turnover	Ups and downs	Change
1991/7/1	685.65	687.48	680.83	684.01	979500	5471735	−7.4	−1.0703
1991/7/2	680.4	683.34	665.39	670.15	566700	3061411	−13.86	−2.0263
1991/7/3	666.81	671.95	660.93	669.06	849400	4369820	−1.09	−0.1627
1991/7/4	665.85	667.54	660.42	661.61	525200	2938094	−7.45	−1.1135
1991/7/5	659.38	660.66	648.22	652.38	635500	3521849	−9.23	−1.3951
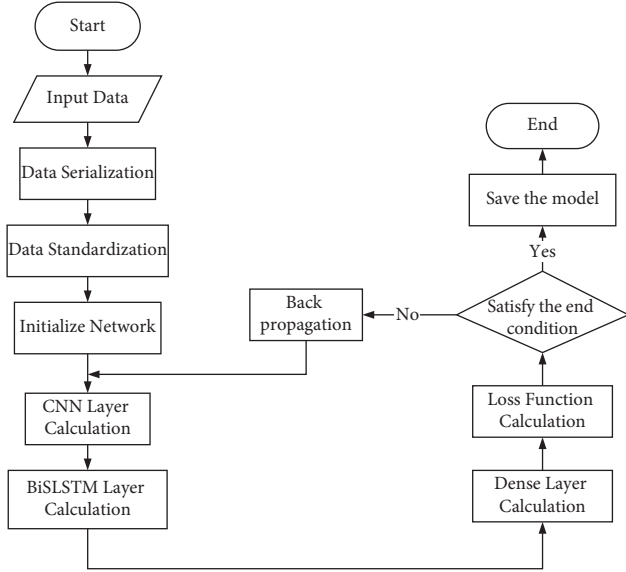
FIGURE 5: Training process of CNN-BiSLSTM.
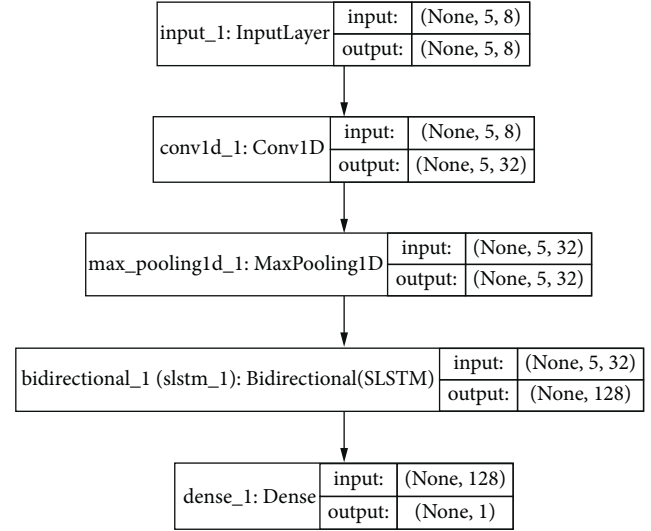
FIGURE 6: CNN-BiSLSTM model structure.

TABLE. 2: CNN-BiSLSTM model parameter settings.

Model parameters	Value
Conv1D filters	32
Conv1D kernel_size	1
Conv1D activation function	Sigmoid
Conv1D padding	Same
MaxPooling1D pool_size	1
MaxPooling1D padding	Same
BiLSTM units	64
BiSLSTM activation function	Tanh
Dense units	1

5. Discussion

According to the data in Table 3 and Figures 7–13, the errors between the predicted value and the true value of the CNN-BiSLSTM are listed, and the comparison models are arranged from high to low as MLP, RNN, LSTM, BiLSTM, CNN-LSTM, CNN-BiLSTM, and CNN-BiSLSTM. From the errors between the predicted value and the true value, we can conclude that the CNN-BiSLSTM has the best fitting degree, and MLP is the worst. The MLP is not suitable for processing time series data. The MAE, RMSE, and R^2 performance of the MLP are all worse than other models. Compared with the RNN, the LSTM can predict the closing price of stock more accurately through the precise gate structure. The prediction effect has been significantly improved by MAE, RMSE, and R^2.

The structure of BiLSTM is more complex than LSTM, and the change rule of historical and future data can be considered. Compared with LSTM, MAE of BiLSTM is reduced by about 5.89 and RMSE by about 5.24. In R^2, BiLSTM is closer to 1, so the prediction effect of BiLSTM is better. The prediction data of LSTM and BiLSTM are added with CNN for feature extraction, and advanced features with stronger expressive ability are learned to form CNN-LSTM and CNN-BiLSTM. Compared with LSTM and BiLSTM, the prediction results have once again been significantly improved. Compared with CNN-LSTM, MAE and RMSE of CNN-BiLSTM decrease by about 0.774 and 0.793, respectively, and R^2 is about 0.985, which is closer to 1. CNN-BiSLSTM adds $1 − \tanh(x)$ function to the output gate of CNN-BiSLSTM. Compared with CNN-BiLSTM, MAE of CNN-BiSLSTM decreases by about 6.46, RMSE decreases by about 4.23, and R^2 is about 0.986, which is closer to 1. The prediction effect of more complex CNN-BiSLSTM is better than that of CNN-BiLSTM, and it is more suitable for stock price prediction.
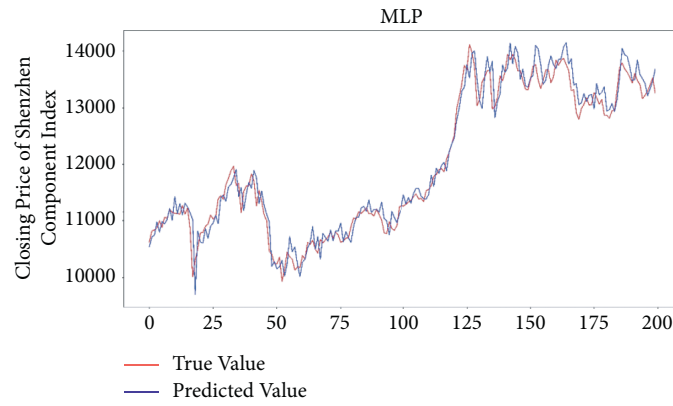
FIGURE 7: Comparison of predicted value and true value in the last 200 days of MLP.
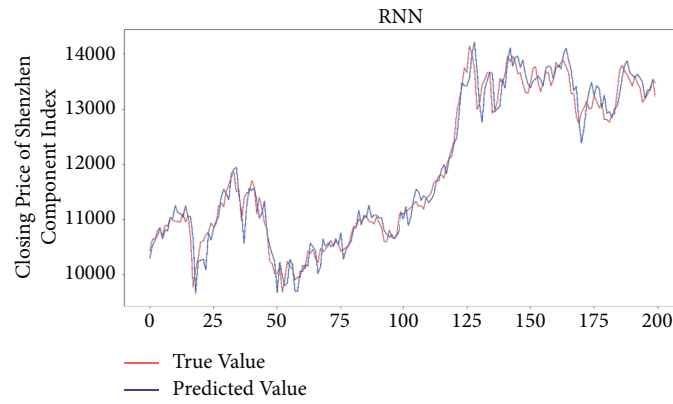

FIGURE 8: Comparison of predicted value and true value in the last 200 days of RNN.
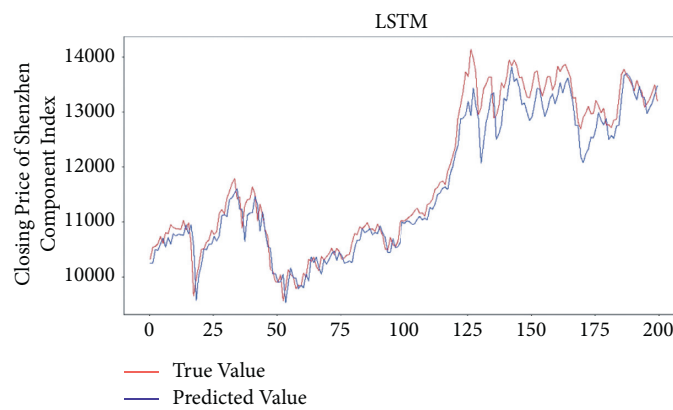

FIGURE 9: Comparison of predicted value and true value in the last 200 days of LSTM.
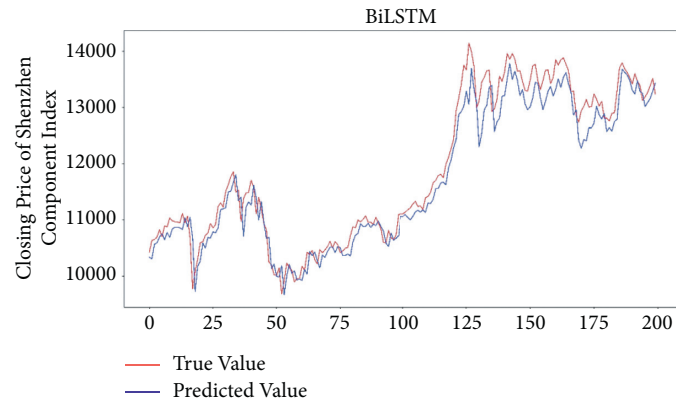
FIGURE 10: Comparison of predicted value and true value in the last 200 days of BiLSTM.
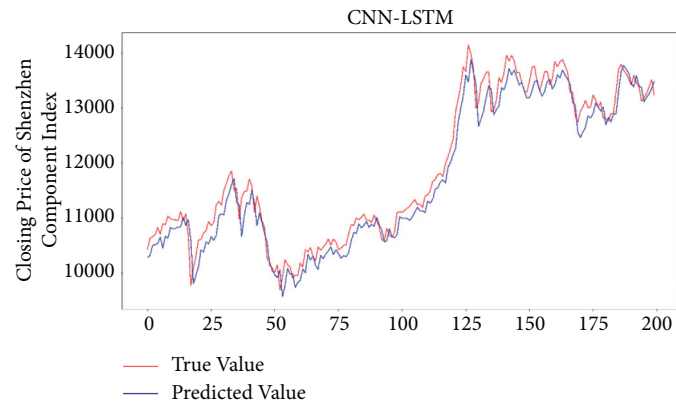

FIGURE 11: Comparison of predicted value and true value in the last 200 days of CNN-LSTM.
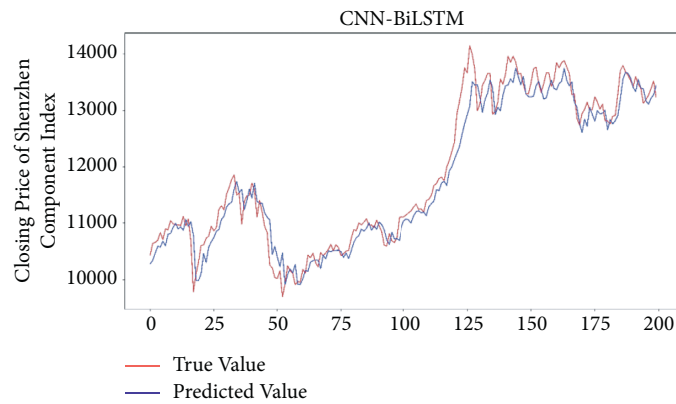

FIGURE 12: Comparison of predicted value and true value in the last 200 days of CNN-BiLSTM.
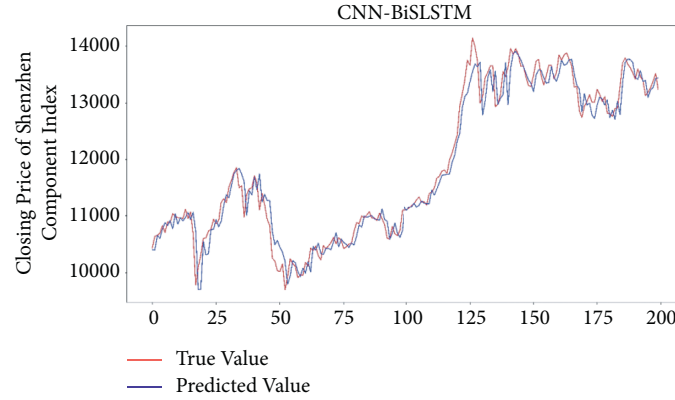
Figure 13: Comparison of predicted value and true value in the last 200 days of CNN-BiSLSTM.

Table 3: Model evaluation index contrast.

Model	MAE	RMSE	R^2
MLP	139.40961	194.99599	0.98094
RNN	132.54118	184.73302	0.98303
LSTM	127.75855	181.01318	0.98352
BiLSTM	121.87029	175.77207	0.98353
CNN-LSTM	120.70497	167.55882	0.98493
CNN-BiLSTM	119.93147	166.76596	0.98535
CNN-BiSLSTM	**113.47137**	**162.53164**	**0.98634**

The bold values can show CNN-BiSLSTM is the best.

6. Conclusions

A hybrid stock predicting model based on CNN-BiSLSTM is proposed. The model consists of two parts. First, CNN is used to capture the features of the input data and combine them to form high-level data features. BiSLSTM adds $1 - \tanh(x)$ function to the output gate calculation based on BiLSTM. Second, BiSLSTM is used to consider the change rule of historical data at the same time, and the stock data in the past are used to predict the closing price of the stock of the next trading day. CNN-BiSLSTM is compared with the reference models of MLP, RNN, LSTM, BiLSTM, CNN-LSTM, and CNN-BiLSTM. The experimental results show that the CNN-BiSLSTM stock prediction model has a better prediction effect than the reference models.

There are still some details to be improved in this paper, which need to be further studied. The future work can be divided into two parts:

(1) Investors are an indispensable part of the stock market. To some extent, investors are also understanding and controlling the stock market. Therefore, through the investors' evaluation and views on individual stock, we can analyze the opinions and emotions held by most investors and further infer the future trend of stock, which can provide guidance for investment strategies.

(2) The prediction of the closing price of stock in this paper has limitations. It only predicts the closing price of stock in the next trading day, which has limited reference value for investment. Regarding investors, they prefer to predict the price and trend of the stock in the next period of time, so they need to conduct more in-depth research on the stock changes.

Data Availability

The data presented in this study are available on request from the corresponding author due to restrictions privacy.

Conflicts of Interest

The authors declare that they have no known conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

References

[1] S. Zhang, Q. He, H. Zhang, and K. Ouyang, "Doppler correction using short-time MUSIC and angle interpolation resampling for wayside acoustic defective bearing diagnosis,"

IEEE Transactions on Instrumentation and Measurement, vol. 66, no. 4, pp. 671–680, 2017.

[2] S. Meng, H. Fang, and D. Yu, "Fractal characteristics, multiple bubbles, and jump anomalies in the Chinese stock market," *Complexity*, vol. 2020, Article ID 7176598, 12 pages, 2020.

[3] N. Naik and B. Mohan, "Intraday stock prediction based on deep neural network," *National Academy Science Letters*, vol. 43, no. 2, 2019.

[4] X. Cui, W. Shang, and F. Jiang, "Stock index forecasting by hidden Markov models with trends recognition," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019.

[5] K. Adam, A. Marcet, and J. Nicolini, "Stock market volatility and learning," *The Journal of Finance*, vol. 71, no. 1, pp. 419–438, 2016.

[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] M. Dixon, D. Klabjan, and J. Bang, *Classification-Based Financial Markets Prediction Using Deep Neural Networks*, Social Science Electronic Publishing, Rochester, NY, USA, 2017.

[8] D. Shah, W. Campbell, and F. Zulkernine, "A comparative study of LSTM and DNN for stock market forecasting," in *Proceedings of the IEEE International Conference on Big Data*, 2018.

[9] H. Rezaei, H. Faaljou, and G. Mansourfar, "Stock price prediction using deep learning and frequency decomposition," *Expert Systems with Applications*, vol. 169, Article ID 114332, 2020.

[10] R. Kusuma, T. Ho, W. Kao, Y.-Y. Ou, and K.-L. Hua, "Using deep learning neural networks and candlestick chart representation to predict stock market," 2019, https://arxiv.org/abs/1903.12258.

[11] X. Zhang, Y. Zhang, S. Wang, Y. Yao, B. Fang, and P. S. Yu, "Improving stock market prediction via heterogeneous information fusion," *Knowledge-Based Systems*, vol. 143, pp. 236–247, 2018.

[12] D. Nelson, A. Pereira, and R. Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, 2017.

[13] B. Nair, N. Dharini, and V. Mohandas, "A stock market trend prediction system using a hybrid decision tree-neuro-fuzzy system," in *Proceedings of the 2010 International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom)*, 2010.

[14] J. Wang, R. Hou, C. Wang, and L. Shen, "Improved v-support vector regression model based on variable selection and brain storm optimization for stock price forecasting," *Applied Soft Computing*, vol. 49, pp. 164–178, 2016.

[15] E. Hoseinzade and S. Haratizadeh, "CNNpred: CNN-based stock market prediction using a diverse set of variables," *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019.

[16] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, no. 1, pp. 11–24, 2014.

[17] B. Huang, Q. Ding, G. Sun, and H. Li, "Stock prediction based on Bayesian-LSTM," in *Proceedings of the International Conference on Machine Learning and Computing*, pp. 128–133, Zhuhai, China, 2018.

[18] H. Gunduz, Y. Yaslan, and Z. Cataltepe, "Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations," *Knowledge-Based Systems*, vol. 137, pp. 138–148, 2017.

[19] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: machine learning techniques applied to financial market prediction," *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.

[20] M. Wen, P. Li, L. Zhang, and Y. Chen, "Stock market trend prediction using high-order information of time series," *IEEE Access*, vol. 7, pp. 28299–28308, 2019.

[21] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Computing & Applications*, vol. 32, no. 13, pp. 9713–9729, 2020.

[22] S. Mehtab and J. Sen, "Stock price prediction using CNN and LSTM-based deep learning models," 2020, https://arxiv.org/abs/2010.13891.

[23] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, 2017.

[24] Y. Baek and H. Y. Kim, "ModAugNet: a new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Systems with Applications*, vol. 113, pp. 457–480, 2018.

[25] A. Moghar and M. Hamiche, "Stock market prediction using LSTM recurrent neural network," *Procedia Computer Science*, vol. 170, pp. 1168–1173, 2020.

[26] J. Wang, J. Li, X. Wang, and J. Wang, "Air quality prediction using CT-LSTM," *Neural Computing & Applications*, vol. 33, no. 3, pp. 1–14, 2020.

[27] I. Hasan, F. Setti, T. Tsesmelis, A. D. Bue, F. Galasso, and M. Cristani, "MXLSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6067–6076, Salt Lake City, UT, USA, 2018.

[28] K. Zhang, G. Zhong, J. Dong, S. Wang, and Y. Wang, "Stock market prediction based on generative adversarial network," *Procedia Computer Science*, vol. 147, pp. 400–406, 2019.

[29] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in *Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 945–950, IEEE, Okayama, Japan, 2016.

[30] S. S. Hyun, I. K. Hae, and J. A. Jae, "Is deep learning for image recognition applicable to stock market prediction," *Complexity*, vol. 2019, Article ID 4324878, 11 pages, 2019.

[31] Y. Yang, Y. Yang, and J. Xiao, "A hybrid prediction method for stock price using LSTM and ensemble EMD," *Complexity*, vol. 2020, Article ID 6431712, 16 pages, 2020.

[32] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," *Complexity*, vol. 2020, Article ID 6622927, 10 pages, 2020.

[33] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[34] D. Kuang and B. Xu, "Predicting kinetic triplets using a 1D convolutional neural network," *Thermochimica Acta*, vol. 669, pp. 8–15, 2018.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[37] A. Graves, "Supervised sequence labelling with recurrent neural networks," *Studies in Computational Intelligence*, vol. 385, 2012.

[38] N. Masseran, A. M. Razali, K. Ibrahim, and M. T. Latif, "Fitting a mixture of von Mises distributions in order to model data on wind direction in Peninsular Malaysia," *Energy Conversion and Management*, vol. 72, pp. 94–102, 2013.

[39] G. Wang, G. Yu, and X. Shen, "The effect of online investor sentiment on stock movements: an LSTM approach," *Complexity*, vol. 2020, Article ID 4754025, 11 pages, 2020.

[40] F. Jia and B. Yang, "Forecasting volatility of stock index: deep learning model with likelihood-based loss function," *Complexity*, vol. 2021, Article ID 5511802, 13 pages, 2021.