Research Article

Stock Price Forecast Based on CNN-BiLSTM-ECA Model

Yu Chen [ID],[1] **Ruixin Fang** [ID],[1] **Ting Liang** [ID],[1] **Zongyu Sha** [ID],[1] **Shicheng Li** [ID],[1] **Yugen Yi** [ID],[1] **Wei Zhou** [ID],[2] **and Huilin Song** [ID][3]

[1]*School of Software, Jiangxi Normal University, Nanchang 330022, China*
[2]*School of Computer, Shenyang Aerospace University, Shenyang 110136, China*
[3]*School of International Economics and Trade, Jiangxi University of Finance and Economics, Nanchang 330022, China*

Correspondence should be addressed to Yugen Yi; yiyg510@jxnu.edu.cn and Wei Zhou; zhouweineu@outlook.com

Financial data as a kind of multimedia data contains rich information, which has been widely used for data analysis task. However, how to predict the stock price is still a hot research problem for investors and researchers in financial field. Forecasting stock prices becomes an extremely challenging task due to high noise, nonlinearity, and volatility of the stock price time series data. In order to provide better prediction results of stock price, a new stock price prediction model named as CNN-BiLSTM-ECA is proposed, which combines Convolutional Neural Network (CNN), Bidirectional Long Short-term Memory (BiLSTM) network, and Attention Mechanism (AM). More specifically, CNN is utilized to extract the deep features of stock data for reducing the influence of high noise and nonlinearity. Then, BiLSTM network is employed to predict the stock price based on the extracted deep features. Meanwhile, a novel Efficient Channel Attention (ECA) module is introduced into the network model to further improve the sensitivity of the network to the important features and key information. Finally, extensive experiments are conducted on the three stock datasets such as Shanghai Composite Index, China Unicom, and CSI 300. Compared with the existing methods, the experimental results verify the effectiveness and feasibility of the proposed CNN-BILSTM-ECA network model, which can provide an important reference for investors to make decisions.

1. Introduction

With the unprecedented development of the network, multimedia data such as text, image, video, financial data from mobile phones, social networking sites, news, and financial websites are growing at a rapid pace and also affecting our real daily life. In the era of big data, how to make full use of these data providing the relevant and valuable information for us becomes a great significance task [1, 2]. For example, investors can employ the financial data to predict the future price trend of financial assets to reduce the decision-making risk [3, 4]. However, investors can hardly acquire the useful information of the budget allocation timely. In order to make right investment decisions for investors, some technical or quantitative methods are

necessary and important to use to predict the fluctuation of asset prices [5].

There are many methods that have been proposed to predict the financial stock data, which achieve excellent performance [5, 6]. For instance, traditional methods based on econometric statistical model aim to find the best estimation for time series prediction, such as Auto Regressive (AR) model, Moving Average (MA) model, Auto Regressive and Moving Average (ARMA) model, and Autoregressive Integrated Moving Average (ARIMA) model [7]. Although the abovementioned approaches can describe and evaluate the relationship between variables through statistical inference, there are still some limitations. On one hand, since these methods are based on the assumption of linear relationship of model structure, they can hardly capture the

nonlinear variation of the stock price [8, 9]. On the other hand, these approaches assume that the data have constant variance, while the financial time series have high-noisy, time-varying, dynamic properties, and so on [10].

In order to solve the aforementioned shortcomings, many machine learning technologies have been applied to simulate the nonlinear relationship in financial time series. Among them, Artificial Neural Network (ANN) owing to its excellent ability in nonlinear mapping and generalization has been widely used for dealing with the financial time series [11, 12]. Different from the econometric statistical models, the strict model structure and additional series of assumptions are not required for ANN model. For example, Hajizadeh et al. [13] put forward a hybrid model, which combined ANN with Exponential Generalized Autoregressive Conditional Heteroscedasticity (EGACH) model to predict the volatility of S&P500. The experimental results show that the test error of the hybrid model is lower than that of any single econometric model. Rather et al. [14] realized stock return prediction via taking ARMA and Exponential Smoothing (SE) linear models with Recurrent Neural Network (RNN) [15] into consideration. The experimental results indicate that RNN can further improve the prediction performance.

In recent years, Long Short-Term Memory (LSTM) network has been proposed which overcomes the vanishing gradient problem in RNN and uses storage cells and gates to learn long-term correlation in time series data [16]. LSTM has been widely used for forecasting time series due to its advantages [17–19]. Nelson et al. [20] proposed a stock price prediction method based on LSTM. From the experimental results [20], we can find that the LSTM model is more accurate than other machine learning models, such as Support Vector Machine (SVM) [21], Genetic Algorithm (GA) [22], and BP Neural Network [23, 24]. Graves and Schmidhuber [25] proposed a Bidirectional LSTM (BiLSTM) network. BiLSTM contains a forward LSTM and a reverse LSTM, in which the forward LSTM takes advantage of past information and the reverse LSTM takes advantage of future information. Then, many researchers applied BiLSTM to solve the problem of time series prediction [26–28]. Because BiLSTM network can use the information of both the past and the future, the final prediction is more accurate than unidirectional LSTM. In addition, multitask RNNs methods also have been proposed for forecasting time series such as EEG-based motion intention recognition and dynamic illness severity prediction [29, 30].

Attention Mechanism (AM) is derived from human vision [31]. By scanning the target area quickly, the human eyes can pay more attention on the key areas to obtain the useful information and suppress other useless information. Thus, we can use the limited resources to quickly screen out the valuable information based on the target from the massive information [31]. At present, attention mechanism has been well applied in computer vision question answering [32, 33], natural language processing [34, 35], and speech recognition [36, 37]. In addition, some researchers have also successfully applied the attention mechanism to the related research of time series [38–41].

Inspired by the successful applications of deep learning and attention mechanism on stock data analysis [42–44], this paper proposes a time series prediction model named as CNN-BiLSTM-ECA, which integers Convolutional Neural Networks (CNN) [10, 12] and BiLSTM to predict the closing price of stock data. First, our model adopts CNN to extract the deep features. Second, the feature vector is constructed in time series regarded as the input of the BiLSTM network. Third, in order to further improve the prediction performance of BiLSTM network, a novel attention mechanism termed as Effective Channel Attention (ECA) [45] module is introduced. Compared with other attention mechanisms, ECA is lighter and less complex, which can greatly improve network performance. Finally, we compare the proposed CNN-BiLSTM-ECA model with the LSTM, BiLSTM, and CNN-BiLSTM models on three stocks' data including Shanghai Composite Index, China Unicom, and CSI 300 to verify its effectiveness.

The outline of this paper is as follows: Section 2 reviews the related works and backgrounds are introduced in Section 3. Section 4 gives the proposed network model structure in detail. Section 5 shows extensive experiments to prove the effectiveness of the proposed approach. Section 6 presents some conclusions.

2. Related Works

Recently, a large number of stock prediction methods have been proposed. This paper mainly reviews two kinds of methods based on machine learning and deep learning.

2.1. Machine Learning Methods. As machine learning techniques are becoming more and more popular, Machine Learning (ML) methods based on financial time series forecasting have been studied extensively. Specifically, Nayak et al. [46] proposed a hybrid model based on SVM and K-Nearest Neighbor (KNN) to predict the Indian Stock Market Index. Combining weight SVM and KNN, Chen and Hao [47] proposed a new model for predicting the trend of the Chinese stock market. In order to find the optimal solution of neural network, Chiang et al. [48] developed a novel model to predict stock trend by introducing the Particle Swarm Optimization (PSO) into the neural network. Furthermore, Zhang et al. [49] proposed a novel ensemble method by taking AdaBoost, generic algorithm, and probabilistic SVM into consideration to acquire better prediction performance. Moreover, several hybrid based approaches have been proposed for stock trend prediction, which have achieved excellent performance. For example, Marković et al. [50] proposed a new hybrid method that integrates the analytic hierarchy process and weighted kernel least squares SVM. Lei [51] developed a hybrid method by combining the rough set and wavelet neural network. Many researchers have found that fusion model based on different techniques plays a vital role in prediction. Based on technical analysis and sentiment embedding, Picasso et al. [52] proposed a fusion model, which integrates Random Forest (RF), ANN, and SVM. Similar to [52], Parray et al. [53] applied several

methods based on machine learning including Logistic Regression (LR), SVM, and ANN to predict the trend of the stocks for the next day. Xu et al. [54] proposed a new fusion method by combining the k-mean clustering and ensemble method (i.e., SVM and RF). In order to reduce the influence of parameters, Dash et al. [55] proposed a new stock price prediction method named fine-tuned SVR, which combines the grid search technique and SVR. The grid search technique is used to select the best kernel function and tune the optimized parameters through training and validation datasets.

Apart from the aforementioned approaches based on machine learning, a variety of Deep Learning (DL) techniques have emerged in recent years for financial time series forecasting researches. In the next section, we will introduce some stock price prediction methods based on DL techniques.

2.2. Deep Learning Methods. DL technique based on ANNs is a branch of ML, which can extract high-level abstraction feature for data representation. DL methods can achieve excellent performance compared to conventional ML methods; thus, they have been widely applied in many application fields such as image processing and compute vision. Recently, DL methods have been proposed for analyzing the financial time series data. Specifically, combining the stock technical indicators and the historical price of stock data, Nelson et al. [20] applied LSTM model to predict the movement of the stock market's price. In order to forecast the short-term stock market trend, Liang et al. [56] constructed a new prediction model by combing Restricted Boltzmann Machine (RBM) and several classifiers. Kim and Won [57] combined LSTM with Generalized Autoregressive Conditional Heteroscedasticity (GACH) model to identify the stock price volatility. Moews et al. [58] designed a forecasting method based on Deep Feed-forward Neural Network (DFNN) and exponential smoothing. In order to reduce the training complexity and improve the prediction accuracy, Li et al. [59] constructed a forecasting model by integrating Feature Selection (FS) and LSTM method. For selecting and focusing on key information of stock data, Zhao et al. [60] introduced AM into RNN and proposed three predication frameworks named AT-RRR, AT-LSTM, and AT-GRU, respectively.

CNN utilizes the local perception and weighted sharing to greatly reduce the number of parameters. Therefore, many methods based on CNN are recently proposed to predict the stock trend and achieve good performance. For example, Sezer and Ozbayoglu [61] first transformed stock technical indicators into 2D images and then designed a novel method based on CNN for the stock price prediction task. Wen et al. [62] first exploited sequences reconstruction method to reduce noise of the financial time series data. And then, they employed CNN model to extract spatial structure from the time series data for stock prediction. Different from [62], Barra et al. [63] first utilized Gramian angular field technique to obtain 2D images from the time series data. Then, an ensemble learning framework of CNNs was exploited to forecast the trend of US market. Long et al. [64] defined three matrices to describe the trading behavior pattern named as the transaction number matrix, buying volume matrix, and selling volume matrix. Next, they exploited CNN to extract deep features. To capture the characteristics of time series, Hao and Gao [65] proposed a novel method by extracting multiscale CNN features of price series. Different from the existing methods, Lu et al. [66] proposed a new network model to predict stock price by combining CNN and LSTM.

3. Backgrounds

In this section, some basic knowledge of the proposed method will be reviewed briefly.

3.1. Long Short-Term Memory Network. RNN model is widely used to analyze and predict time series data [15]. However, RNN model often suffers from the vanishing gradient problem in the training processing. Therefore, it is difficult to remember the previous information, namely, long dependence problem [16, 67]. To deal with this issue, Greff et al. [67] have proposed a LSTM network model, which has the function of memory in a longer time span. The model utilizes gate control mechanism to adjust information flow and systematically determines the amount of incoming information to be retained in each time step. Figure 1 shows the structure of the basis LSTM unit composed of a storage unit and three control gates (named as input gate, output gate, and forgetting gate). x_t and h_t correspond to the input and hidden states of time t, respectively. f_t, i_t, and o_t are forgetting gate, input gate, and output gate, respectively. $\widetilde{C}_t$ is the candidate information for the input to be stored, and the amount of storage is then controlled by the input gate. The calculation processes of each gate, input candidate, cell state, and hidden state are shown in the following formulas:

$$
\begin{aligned}
f_t &= \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big), \\
i_t &= \sigma\big(W_i \cdot [h_{t-1}, x_t] + b_i\big), \\
o_t &= \sigma\big(W_o \cdot [h_{t-1}, x_t] + b_o\big), \\
\widetilde{C}_t &= \tanh\big(W_C \cdot [h_{t-1}, x_t] + b_C\big), \\
C_t &= f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t, \\
h_t &= o_t \cdot \tanh\big(C_t\big),
\end{aligned}
\tag{1}
$$

where W_f, W_i, W_o, and W_c represent the weight matrix of forgetting gate, input gate, output gate, and update state, respectively. b_f, b_i, b_o, and b_c represent the bias vectors of forgetting gate, input gate, output gate, and update state, respectively. x^t represents the time series data of current time interval t, and h_{t-1} is the output of memory unit in the previous time interval $t-1$.

3.2. Attention Mechanism. Squeeze-and-Excitation (SENet) is an efficient attention mechanism, which can improve the representation ability of the network by modeling the dependency of each channel and can adjust the features
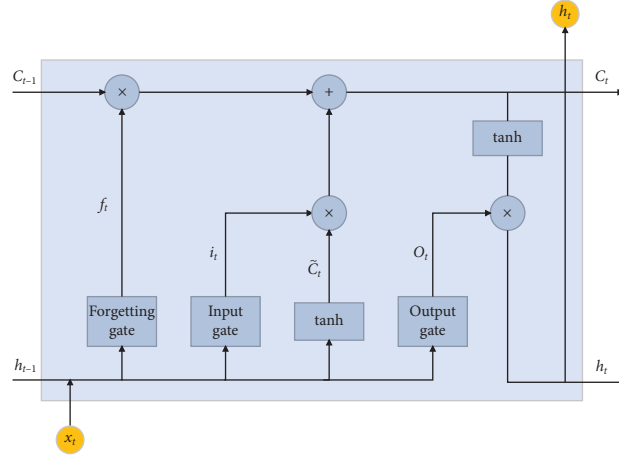
FIGURE 1: The structure diagram of LSTM unit.

channel by channel, so that the network can select more useful features [68]. The basic structure of SE block is shown in Figure 2. The first step is the squeeze operation, which takes the global spatial features of each channel as the representation of the channel to get the global feature description. The second step is the extraction operation, which learns the dependence degree of each channel and adjusts it, and the obtained feature graph is the output of SE block.

4. The Proposed Network Model Structure

In order to extract features efficiently and improve prediction accuracy, we combine CNN, BiLSTM network, and a lightweight ECA attention module into a unified framework and propose a new time series prediction network model named as CNN-BiLSTM-ECA. Our proposed model can automatically learn and extract local features and long memory features in the time series by making full use of data information to minimize the complexity of the model. The network model structure is shown in Figure 3.

In this model, firstly, the CNN model is utilized to extract deep feature vectors from the input origin time series data. Then, the BiLSTM model is employed to learn the temporal features from the new time series data constructed by the deep feature vectors. Moreover, the attention mechanism named as ECA is further introduced to extract more important features. Finally, the Dense model consisting of several fully connected layers is employed to perform the prediction task.

4.1. Convolutional Neural Network. In this paper, Convolution Neural Network (CNN) is utilized to extract data features efficiently. Compared with the traditional neural network structure, CNN is a local connection between neurons, which reduces the number of the parameters between the connection layers. In other words, it contains a part of the connections between $n-1$ layer and n layer in CNN. Figure 4 shows the difference between full connection and local connection, where Figure 4(a) is a schematic

diagram of a full connection and Figure 4(b) is a local connection diagram.

4.2. Bidirectional LSTM. In order to build a more accurate prediction model, the bidirectional LSTM (BiLSTM) network is employed, which acts as forward and backward LSTM networks for each training sequence. The two LSTM networks are connected to the same output layer to provide complete context information to each sequence point. Figure 5 shows the structure of BiLSTM.

4.3. Attention Module ECA. Channel Attention (CA) mechanism has great potential in improving the performance of Deep Convolutional Neural Networks (DCNNs). However, most of the existing methods are committed to designing more complex attention modules to achieve better performance, which inevitably increases the complexity and computational burden of the model. In order to avoid overfitting of the model and reduce the computation, a lightweight and low complexity module called as Effective Channel Attention (ECA) [32] is introduced. ECA can not only generate the weights for each channel, but also learn the correlation among the different channels. For the time series data, the larger weights will be assigned for the key features and smaller weights for the irrelevant features. Therefore, ECA focuses on the useful information, which improves the sensitivity of the network to the main features. Figure 6 shows the structure of ECA.

As shown in Figure 6, ECA is first to carry out channel Global Average Pooling (GAP). Then, ECA employs each channel and its k adjacent channels to capture the local cross channel interactions. ECA generates channel weights by performing fast 1D convolution as follows:

$$\omega = \sigma\left(C1D_k(y)\right), \tag{2}$$

where $C1D$ is denoted as 1D convolution and k is the kernel size of 1D convolution.

In order to avoid manually adjusting k, ECA uses channel dimension adaptively mapping way to
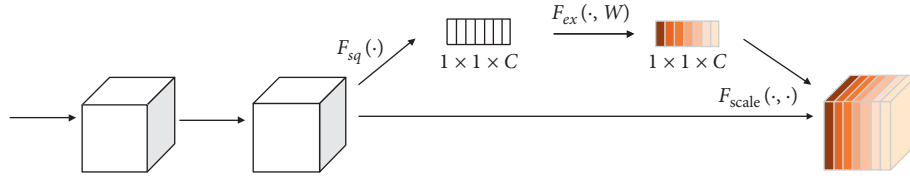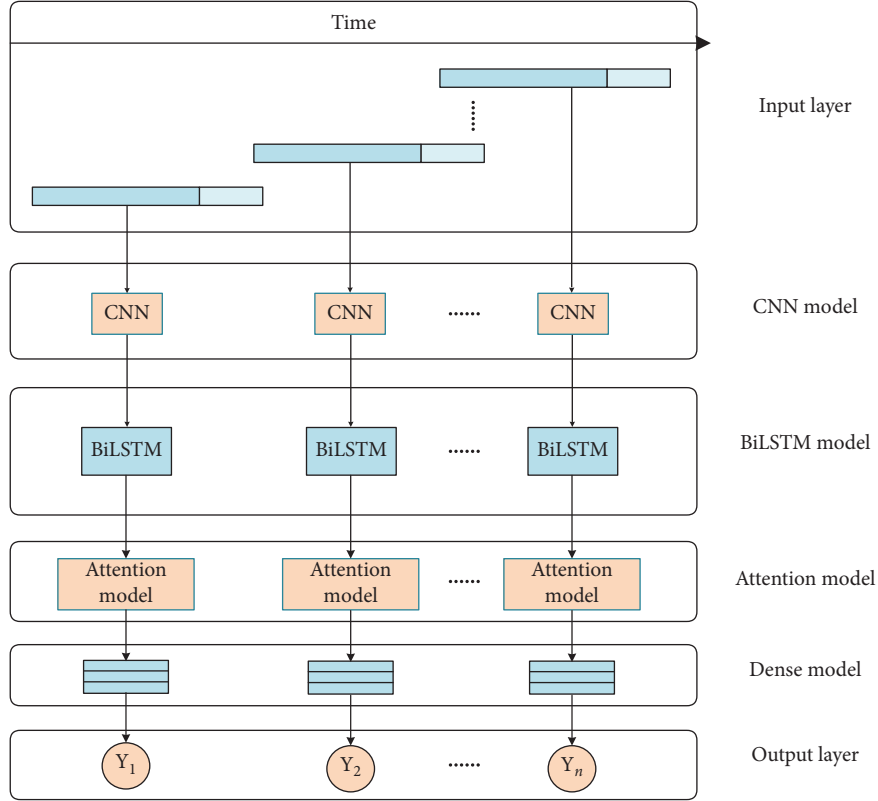
FIGURE 2: SE block structure diagram.

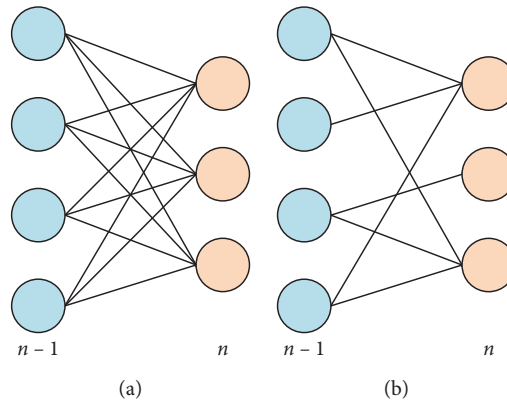FIGURE 3: Network model structure diagram.

(a)   (b)

FIGURE 4: The diagram of full connection and local connection.

determine the value of k. Since the kernel size k of 1D convolution is directly proportional to the channel dimension C, the corresponding relationship is defined as follows:

$$C = \phi(k) = 2^{(\gamma * k - b)}. \tag{3}$$

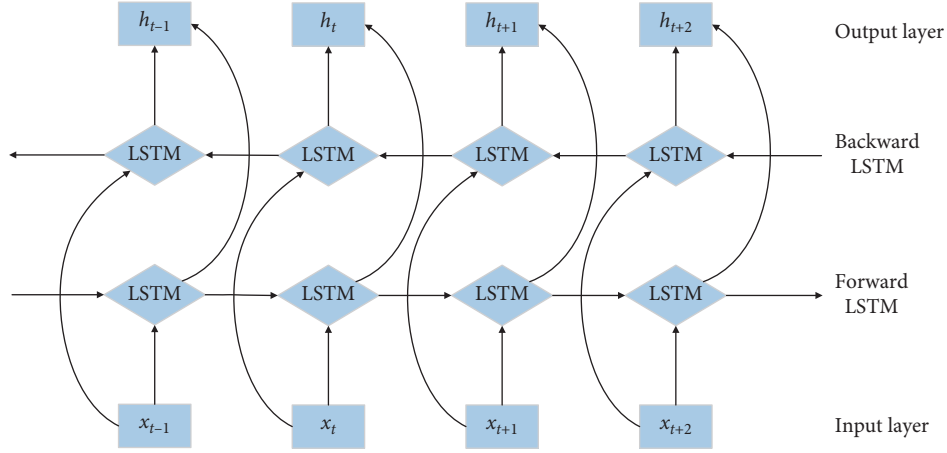Therefore, given the channel dimension C, the kernel size k can be adaptively determined by
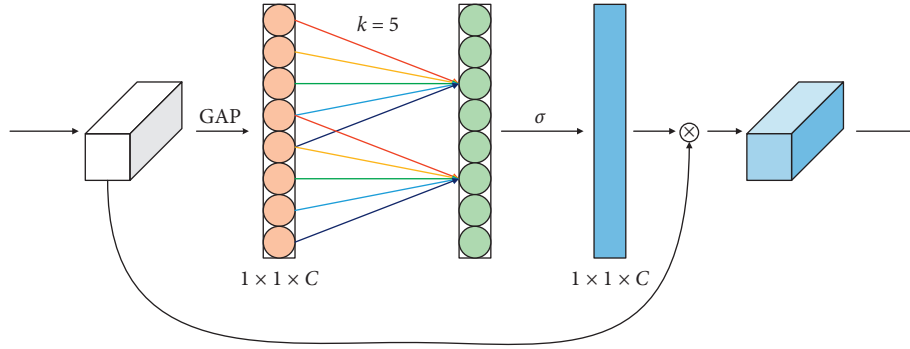
Figure 5: BiLSTM structure diagram.

Figure 6: The structure diagram of ECA.

$$k = \psi(C) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{\text{odd}}, \qquad (4)$$

where $|\ \ |_{\text{odd}}$ is the nearest odd number. The parameters of γ and b are set to 2 and 1, respectively, in this paper. Obviously, the high-dimensional channel has a longer range of interaction through nonlinear mapping, while the low dimensional channel has a shorter range of interaction.

4.4. Dense Model. The purpose of Dense model is first to exploit full connection layer to extract the correlation among these features changed by nonlinear mapping and then maps them to the output space. In the proposed network structure model, three layers of full connection layer are added to solve the nonlinear problem well, which can achieve accurate prediction. Figure 7 shows the structure of Dense model.

5. Experiments

5.1. Experimental Process. The experiment consists of six parts: data colleting, data preprocessing, model training, model saving, model testing, and prediction results. The schematic diagram of the experiment process is shown in Figure 8.

5.2. Data Description and Preprocessing. The experimental data are collected from NetEase Finance (data collected from the website http://quotes.money.163.com/0000001.html#1b01), which includes Shanghai Composite Index (SSE Index for short, stock symbol: 000001), China Unicom (stock symbol: 600050), and CSI 300 (stock symbol: 399300). SSE stock data was collected from December 20, 1990, to November 23, 2020 (7304 groups); China Unicom stock data was collected from October 9, 2002, to March 17, 2021 (4340 groups); CSI 300 stock data was collected from January 7, 2002, to March 17, 2021 (4567 groups). Each stock data includes the closing price, the highest price, the lowest price, the opening price, previous day's closing price, change, ups and downs, and other time series data. Table 1 is an overview of the three stock's data, and Tables 2–4, respectively, show partial data of the three stocks.

From these tables, the original dataset obtained from the network cannot be directly used for model training and testing, so the data preprocessing is required. In view of the missing or disordered attribute values in the original dataset, we first use interpolation and sorting by date and other operations to complete time series data in our experiment. Then, data normalization is employed to deal with the
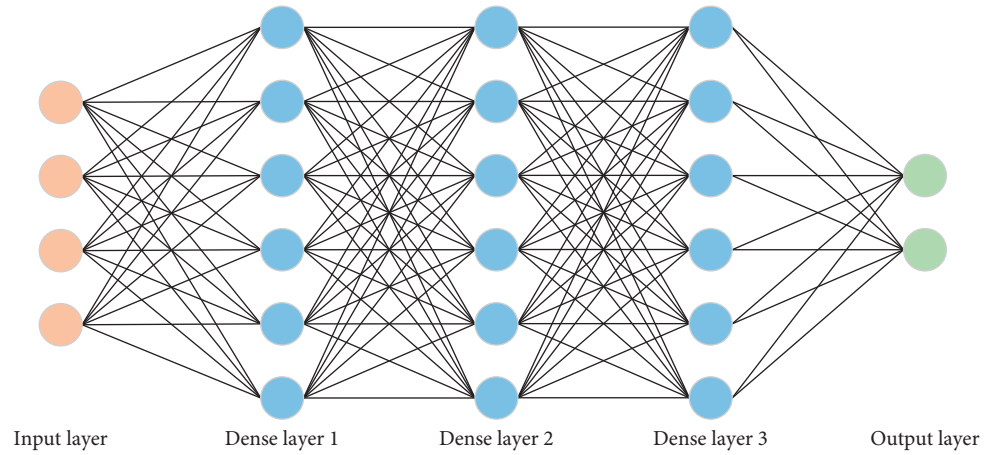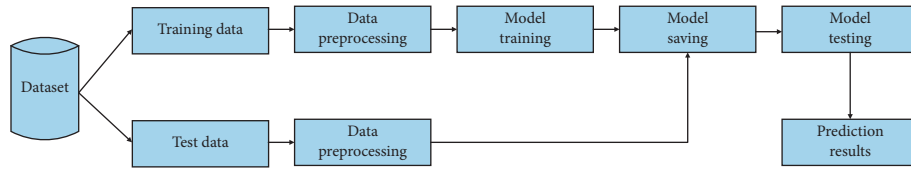
FIGURE 7: The structure diagram of Dense model.

Input layer   Dense layer 1   Dense layer 2   Dense layer 3   Output layer

FIGURE 8: The flow chart of the experiment.

TABLE 1: The detailed information of three stocks' data.

Stock code	Stock name	Start and end time	Total amount of data/group
000001	SSE	1990.12.20–2020.11.23	7304
600050	China Unicom	2002.10.09–2021.03.17	4340
399300	CSI 300	2002.01.07–2021.03.17	4657

TABLE 2: Some data come from SSE stock.

Data	Closing price	Highest price	Lowest price	Opening price	Previous day's closing price	Change	Ups and downs
1990/12/20	104.39	104.39	99.98	104.3	99.98	4.41	4.4109
1990/12/21	109.13	109.13	103.73	109.07	104.39	4.74	4.5407
1990/12/24	114.55	114.55	109.13	113.57	109.13	5.42	4.9666
1990/12/25	120.25	120.25	114.55	120.09	114.55	5.7	4.976
1990/12/26	125.27	125.27	120.25	125.27	120.25	5.02	4.1746
1990/12/27	125.28	125.28	125.27	125.27	125.27	0.01	0.008
1990/12/28	126.45	126.45	125.28	126.39	125.28	1.17	0.9339
1990/12/31	127.61	127.61	126.48	126.56	126.45	1.16	0.9174

TABLE 3: Some data come from China Unicom stock.

Data	Closing price	Highest price	Lowest price	Opening price	Previous day's closing price	Change	Ups and downs
2002/10/09	2.87	3.15	2.86	3.05	2.3	0.57	24.7826
2002/10/10	2.83	2.89	2.79	2.83	2.87	−0.04	−1.3937
2002/10/11	2.94	2.96	2.83	2.85	2.83	0.11	3.8869
2002/10/14	3.01	3.02	2.94	2.96	2.94	0.07	2.381
2002/10/15	2.98	3.01	2.96	3.01	3.01	−0.03	−0.9967
2002/10/16	2.99	3.05	2.98	2.99	2.98	0.01	0.3356
2002/10/17	2.98	3.01	2.97	2.99	2.99	−0.01	−0.3344
2002/10/18	3.01	3.03	2.97	2.98	2.98	0.03	1.0067

TABLE 4: Some data come from CSI 300 stock.

Data	Closing price	Highest price	Lowest price	Opening price	Previous day's closing price	Change	Ups and downs
2002/1/7	1302.08	1302.08	1302.08	1302.08	1316.46	−14.38	−1.0923
2002/1/8	1292.71	1292.71	1292.71	1292.71	1302.08	−9.37	−0.7196
2002/1/9	1272.65	1272.65	1272.65	1272.65	1292.71	−20.06	−1.5518
2002/1/10	1281.26	1281.26	1281.26	1281.26	1272.65	8.61	0.6765
2002/1/11	1249.81	1249.81	1249.81	1249.81	1281.26	−31.45	−2.4546
2002/1/14	1205.15	1205.15	1205.15	1205.15	1249.81	−44.66	−3.5733
2002/1/15	1186.43	1186.43	1186.43	1186.43	1205.15	−18.72	−1.5533
2002/1/16	1201.88	1201.88	1201.88	1201.88	1186.43	15.45	1.3022

problem of inconsistency in the magnitude of the data. Finally, an effective dataset can be constructed for the experiment. Data normalization is to scale the data according to a certain scale and transform it into a specific interval. In this experiment, the value of data is converted to $[0, 1]$, which improves the convergence speed and accuracy of the model. The transformation function is defined as follows:

$$x^* = \frac{x - \min}{\max - \min}, \tag{5}$$

where max and min are denoted as the maximum value and minimum value of the sample data, respectively.

5.3. Experiment Setting and Implementation.

Many hyperparameters are required to be predefined, e.g., the neurons numbers of different layers, the learning rate, and the number of iterations. In our experiment, the neurons numbers of three fully connected layers in Dense model are set as 128, 32, and 1, respectively. The hidden neuron number of each layer in BiLSTM is all set to 64. The learning rate is 0.001, and the number of iterations is 200. Relu function is used as the activation function. Description of the parameter setting of the proposed method is shown in Table 5.

In this experiment, Keras is used as the framework of neural network and Python programming language is employed to implement network structure. According to the parameter setting of the proposed network, the specific CNN-BiLSTM-ECA model structure is shown in Figure 9.

5.4. Evaluation Criterion.

In order to analyze the performance of the proposed prediction model intuitively and quantitatively, three evaluation criteria including Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) are utilized to evaluate the prediction results comprehensively.

The Mean Square Error (MSE) represents the mean of the sum squares of the errors between the predicted data and the original data, which is defined as follows:

TABLE 5: Parameters setting of the proposed method.

	Description of the parameters	Value
CNN	The filter number of convolution layer in CNN	1
	The kernel_size of the convolution layer in CNN	1
	The activation function in CNN	Relu
	The way of convolution layer padding in CNN	Same
BiLSTM	The number of hidden units of BiLSTM	64
	The activation function of BiLSTM	Relu
Dense	The layers number of the Dense model	3
	The hidden neurons number of the first Dense layer	128
	The hidden neurons number of the second Dense layer	32
	The hidden neurons number of the third Dense layer	1
	The activation function of Dense model	Relu
Other	Time step	10
	Batch size	256
	Epochs	200
	Optimizer	Adam
	Loss function	MSE
	Learning rate	0.001

$$\text{MSE} = \frac{1}{n} \sum_{t=1}^{n} \left(X_t - X_t' \right)^2, \tag{6}$$

where X_t and X_t' represent the predicted value and the true value, respectively.

The Root Mean Square Error (RMSE) defines the expected value of the square of the error caused by the predicted value and the true value, and its range is $[0, +\infty)$. When the predicted value is completely consistent with the true value, the value of RMSE is equal to 0, which is a perfect model. On the contrary, when the prediction error is larger, the value of the RMSE will be larger. The calculation formula is as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} \left(X_t - X_t' \right)^2}. \tag{7}$$
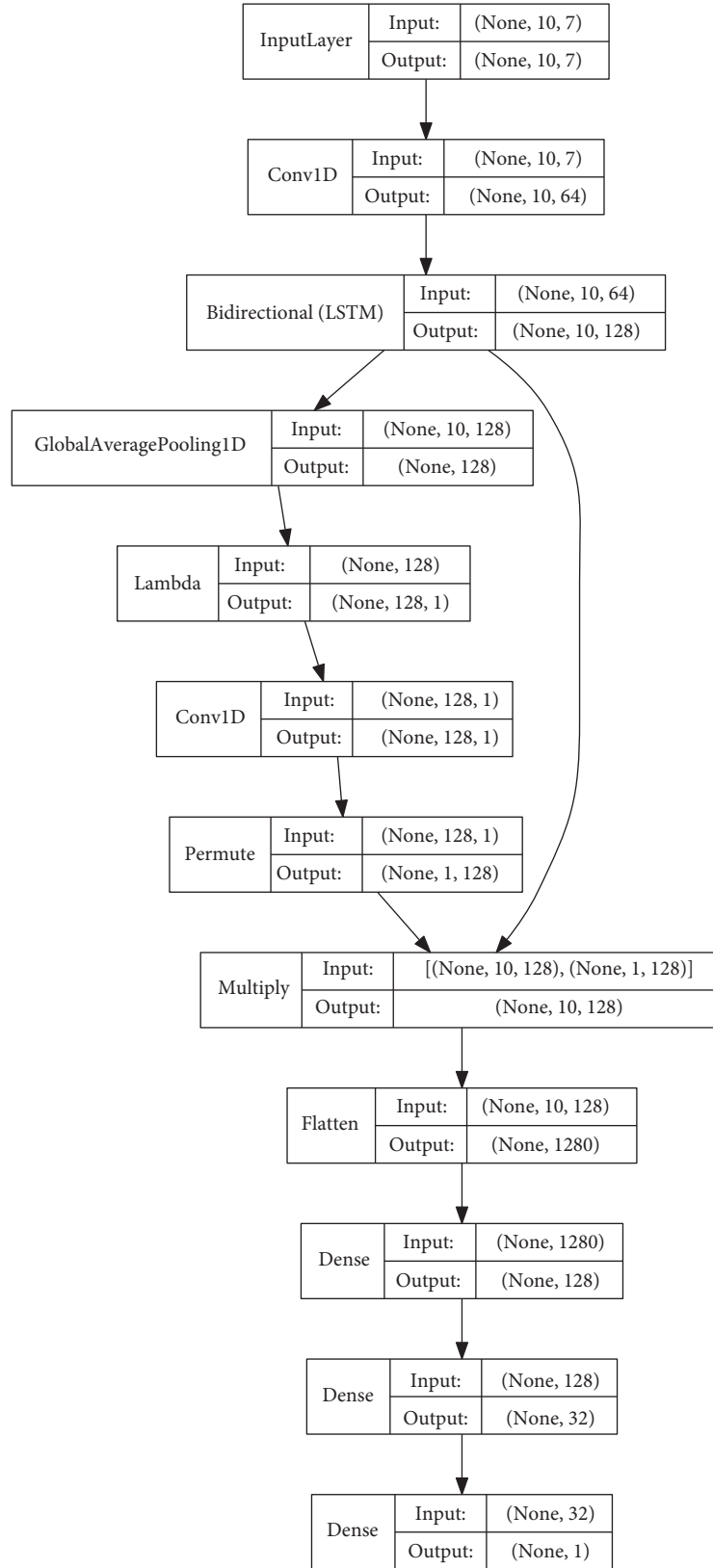
FIGURE 9: The model structure of the proposed CNN-BiLSTM-ECA using Keras to implement.

The mean absolute error (MAE) represents the average of the absolute value of the deviation of all single observations, which can avoid the problem of mutual cancellation of errors and accurately reflect the size of the actual forecast error as below:

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^{n} |X_t - X_t'|. \tag{8}$$

5.5. Experiment Results and Analysis

5.5.1. Parameter Analysis. Time step is a key parameter of the time series neural network, which affects the prediction performance of the network. Therefore, we analyze the time step parameters on the three datasets including the Shanghai Composite Index, China Mobile, and CSI 300. Firstly, the top 85% of the dataset is used for the training set, and the remaining 15% is used for the test set. Secondly, we conduct comparative experiments under the conditions that when the time steps are set as 5, 10, 15, and 20. The experimental results under different parameters are shown in Figures 10–12. The prediction error values of each time step are shown in Table 6.

From these figures, when the time step is set to 5, the prediction results have a large deviation. It means that the small-time step setting will ignore the influence of global factors. When the time step is set to 20, the time setting span is too large. Therefore, the data have certain fluctuations resulting in inaccurate prediction results. When the time step is set to 10, the error of the model is the smallest and the accuracy is the highest. Therefore, the time step is set to 10; in other words, the stock data of the previous 10 days is used as the input layer of the neural unit, and the closing price of the 11th day is used as the label to train the model.

Table 6 visually analyzes and describes the forecast error value of each time step. It is found that the error trend between each time step is consistent with the error trend between the predicted price curve and the actual price curve in Figures 10–12. Therefore, according to these results, when the time step is set to 10, the prediction model can achieve the best performance.

5.5.2. Experimental Results and Analysis. In order to verify the effectiveness of the proposed model, different models including CNN [10], LSTM [15], BiLSTM [25, 26], CNN-LSTM [20], BiLSTM-ECA, CNN-LSTM-ECA, and CNN-BiLSTM [27] are compared on the three stock datasets collected from the Shanghai Composite Index, China Unicom, and CSI 300. The prediction results are shown in Figures 13–15. As shown in these figures, the blue curve is the predicted value of the closing price of the stock, and the red curve is the true value of the closing price of the stock. Meanwhile, the x-axis is the time, and the y-axis is the normalization value of the stock price.

As shown in Figures 13–15, firstly, we can find that the prediction results of BiLSTM and LSTM methods are better than CNN model. It indicates that LSTM and BiLSTM take the trend of the stock price as time series in consideration, which is useful to improve the accuracy of the forecast. Moreover, the performance of BiLSTM is better than LSTM due to the fact that BiLSTM model can exploit the subsequent information of price stock time series. Secondly, introducing the CNN model to reduce the noise and capture nonlinear structure of the stock data well, CNN-LSTM and CNN-BiLSTM methods can outperform CNN, LSTM, and BiLSTM. Finally, by integrating the ECA model into LSTM and BiLSTM to select the important features and key information, BiLSTM-ECA, CNN-LSTM-ECA, and CNN-BiLSTM-ECA methods can achieve better performance than other methods. To sum up, the proposed CNN-BiLSTM-ECA model obtains the best prediction results.

The prediction errors of different methods on the three stock datasets are listed in Table 7. As shown in this table, firstly, the CNN model has the low prediction performance among all the compared approaches. Since BiLSTM uses bidirectional information, BiLSTM model can further improve more prediction ability than LSTM. Then, owing to the deep feature extraction of CNN, CNN-LSTM and CNN-BiLSTM model have higher predictive power. At last, among all the compared methods, the CNN-BiLSTM-ECA model can achieve the best performance, which verifies the effectiveness of the CNN-BiLSTM-ECA network model.

In order to show the model performance more clearly, some prediction results of the LSTM model and the CNN-BiLSTM-ECA model on the part days of the three datasets are selected, as shown in Figures 16–18. From these figures, we can learn that the prediction results of the CNN-BiLSTM-ECA model are closer to the true value of the stock price, and the error is smaller. In summary, the experiment results verify the feasibility and effectiveness of the proposed network model.
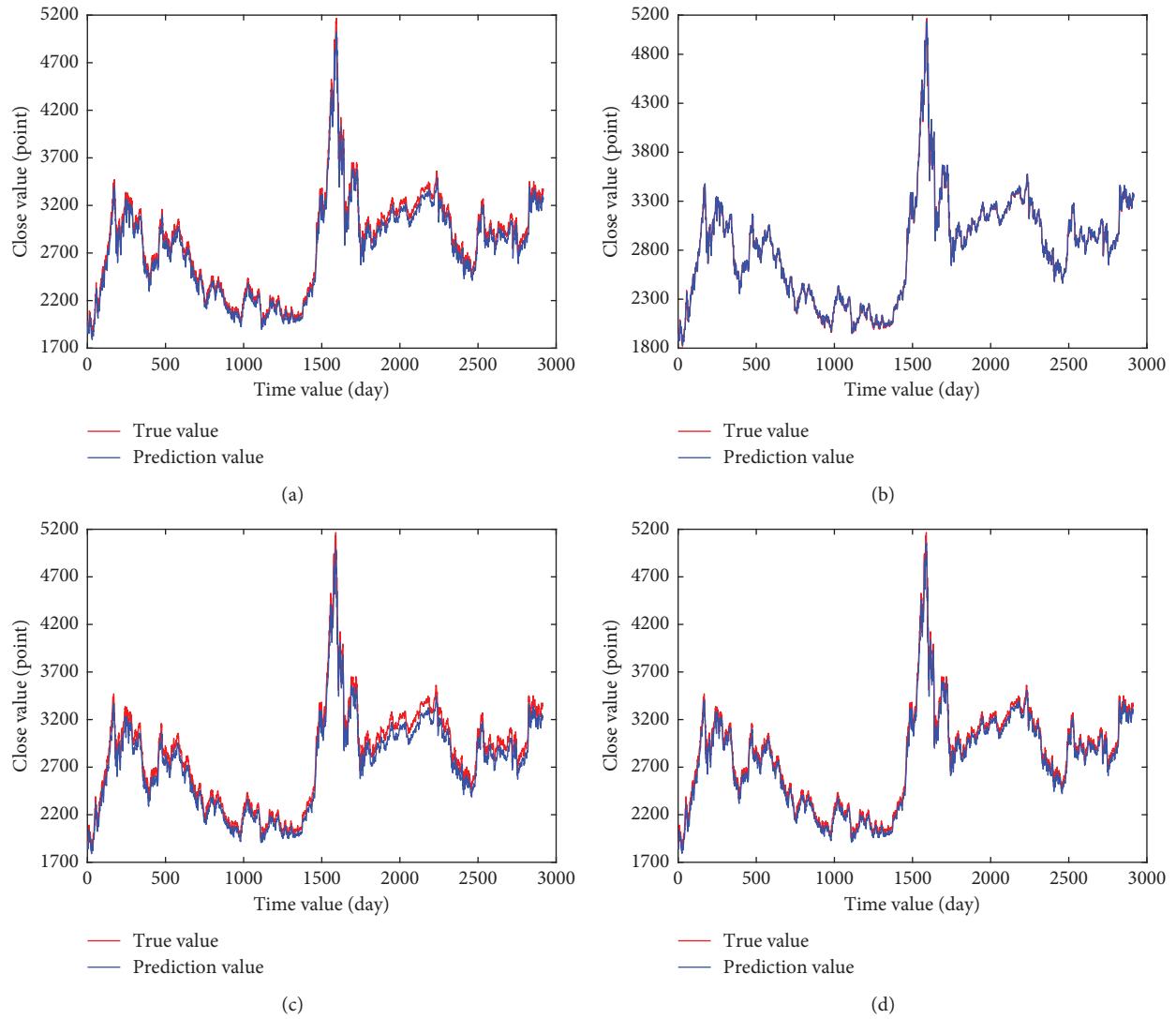
(a)

(b)

(c)

(d)

Figure 10: Comparison of the experimental results of different time steps of the Shanghai Composite Index. (a) Time step = 5. (b) Time step = 10. (c) Time step = 15. (d) Time step = 20.
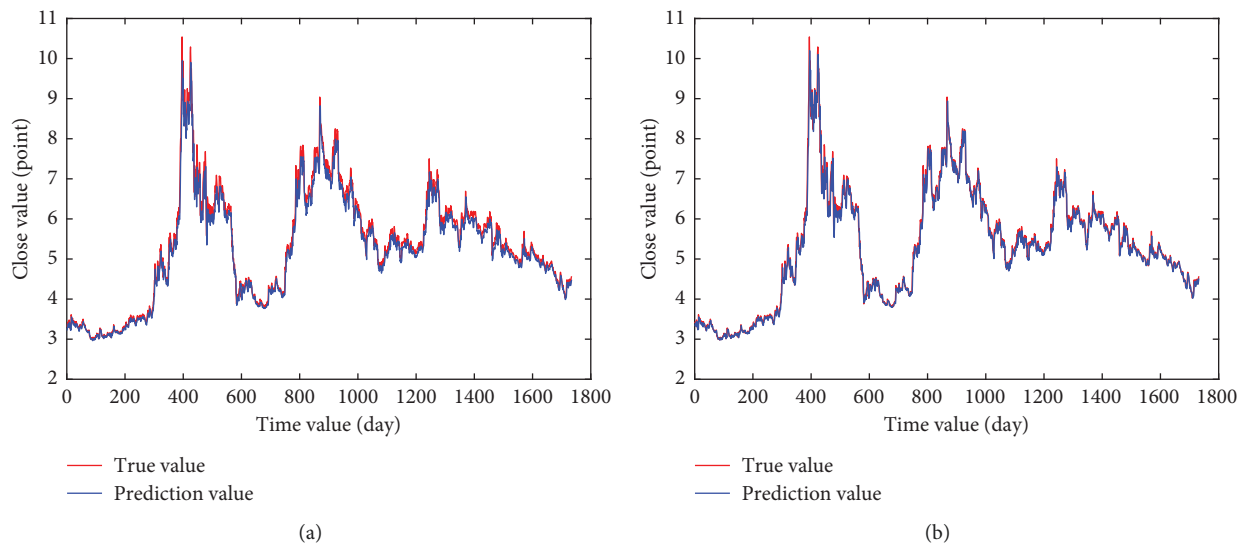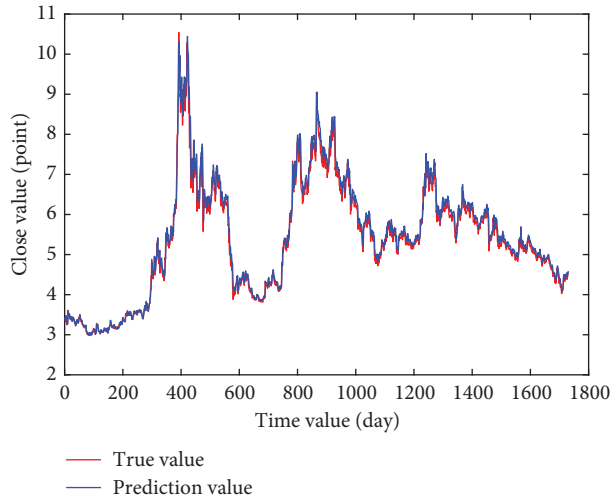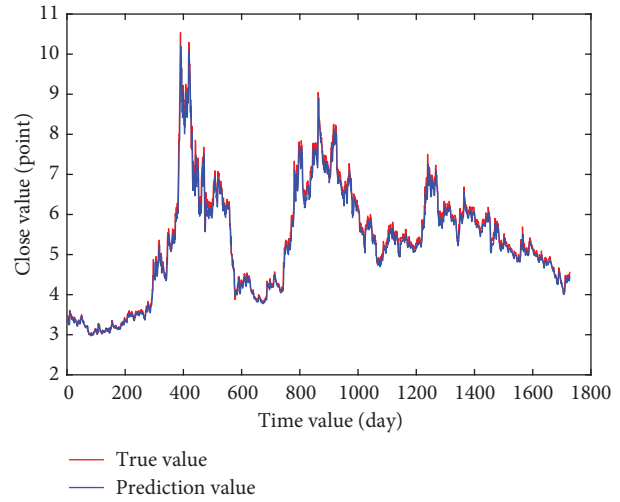

(a)

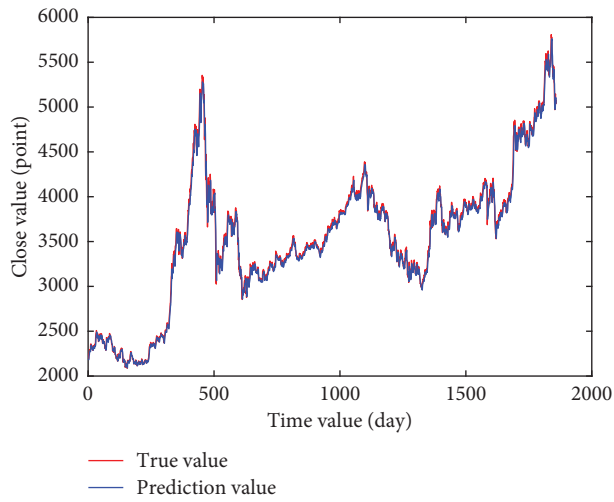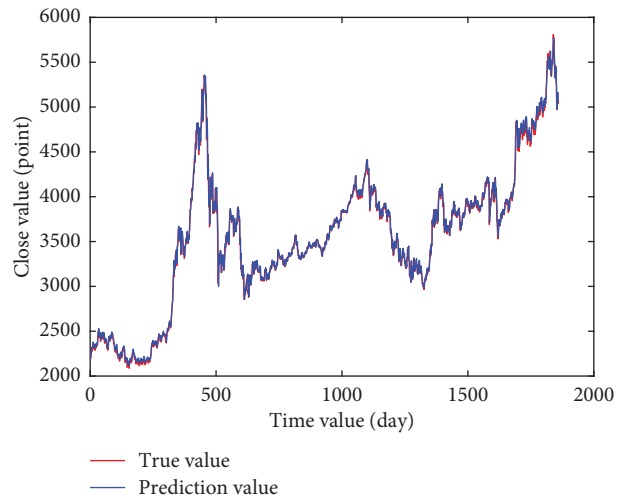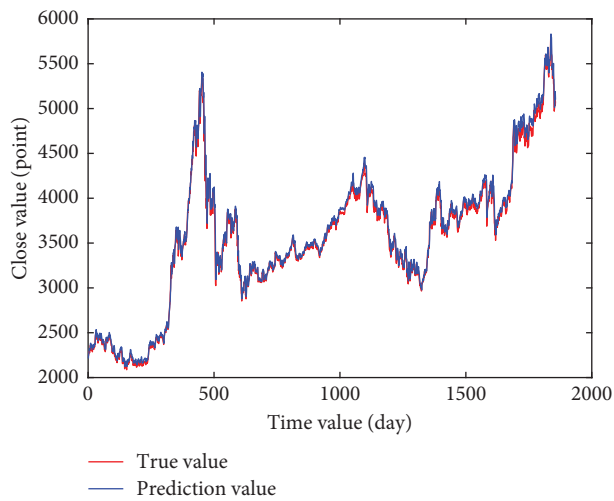(b)

Figure 11: Continued.

(c)

(d)

FIGURE 11: Comparison experiment results of different time steps of China Unicom. (a) Time step = 5. (b) Time step = 10. (c) Time step = 15. (d) Time step = 20.
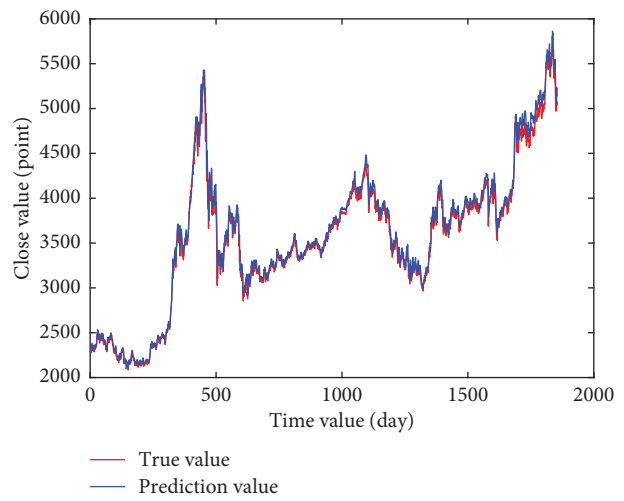

(a)

(b)

(c)

(d)

FIGURE 12: Comparison of experimental results of different time steps of CSI 300. (a) Time step = 5. (b) Time step = 10. (c) Time step = 15. (d) Time step = 20.

TABLE 6: Forecast errors at different time steps.

Time step	Shanghai Composite Index			China Unicom			CSI 300		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
5	6431.222	80.195	70.004	0.044	0.209	0.144	3949.252	62.843	43.987
10	**1956.036**	**44.227**	**28.349**	**0.028**	**0.167**	**0.103**	**3434.408**	**58.604**	**39.111**
15	10162.775	100.811	91.489	0.038	0.196	0.177	5013.827	70.808	51.035
20	4711.037	68.637	58.377	0.035	0.186	0.123	6409.845	80.062	57.951

(a)
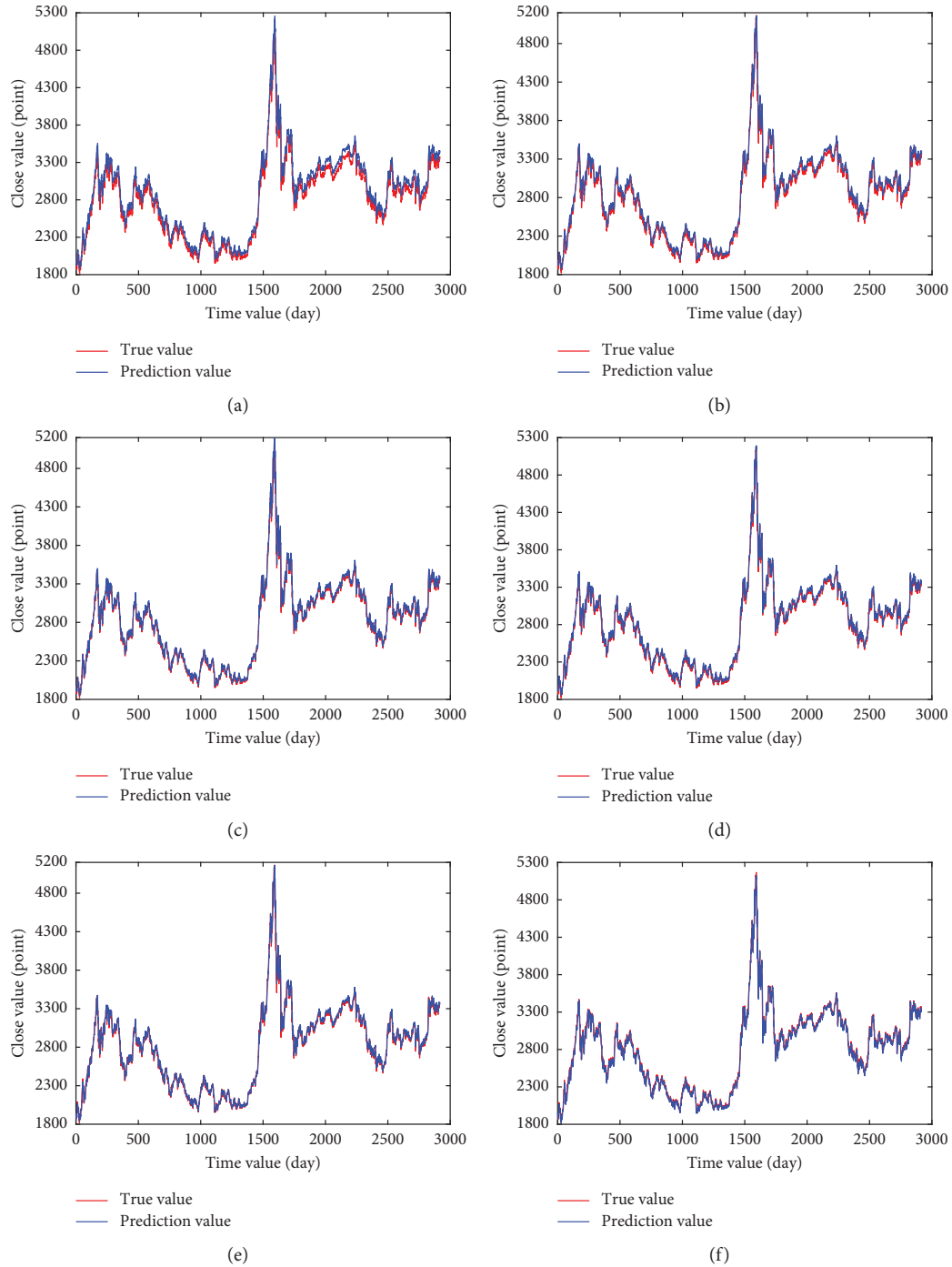

(b)

(c)

(d)

(e)

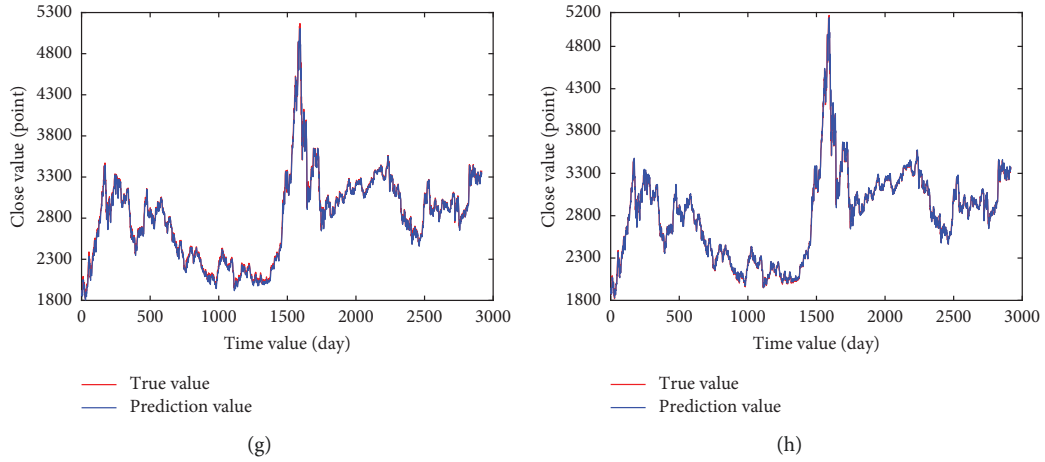

(f)

FIGURE 13: Continued.

(g)

(h)

FIGURE 13: The prediction results of different models on the Shanghai Composite Index. (a) CNN. (b) LSTM. (c) BiLSTM. (d) CNN-LSTM. (e) CNN-BiLSTM. (f) BiLSTM-ECA. (g) CNN-LSTM-ECA. (h) CNN-BiLSTM-ECA.
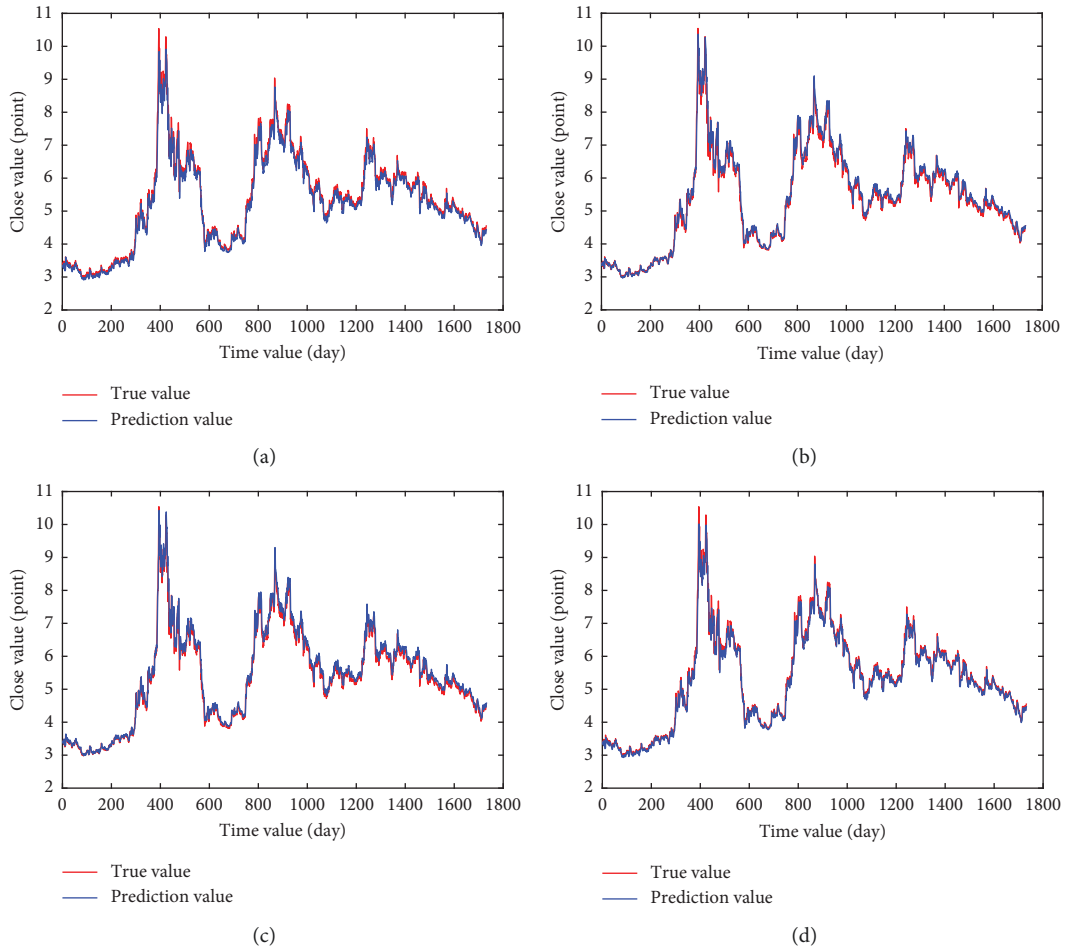

(a)

(b)

(c)

(d)

FIGURE 14: Continued.
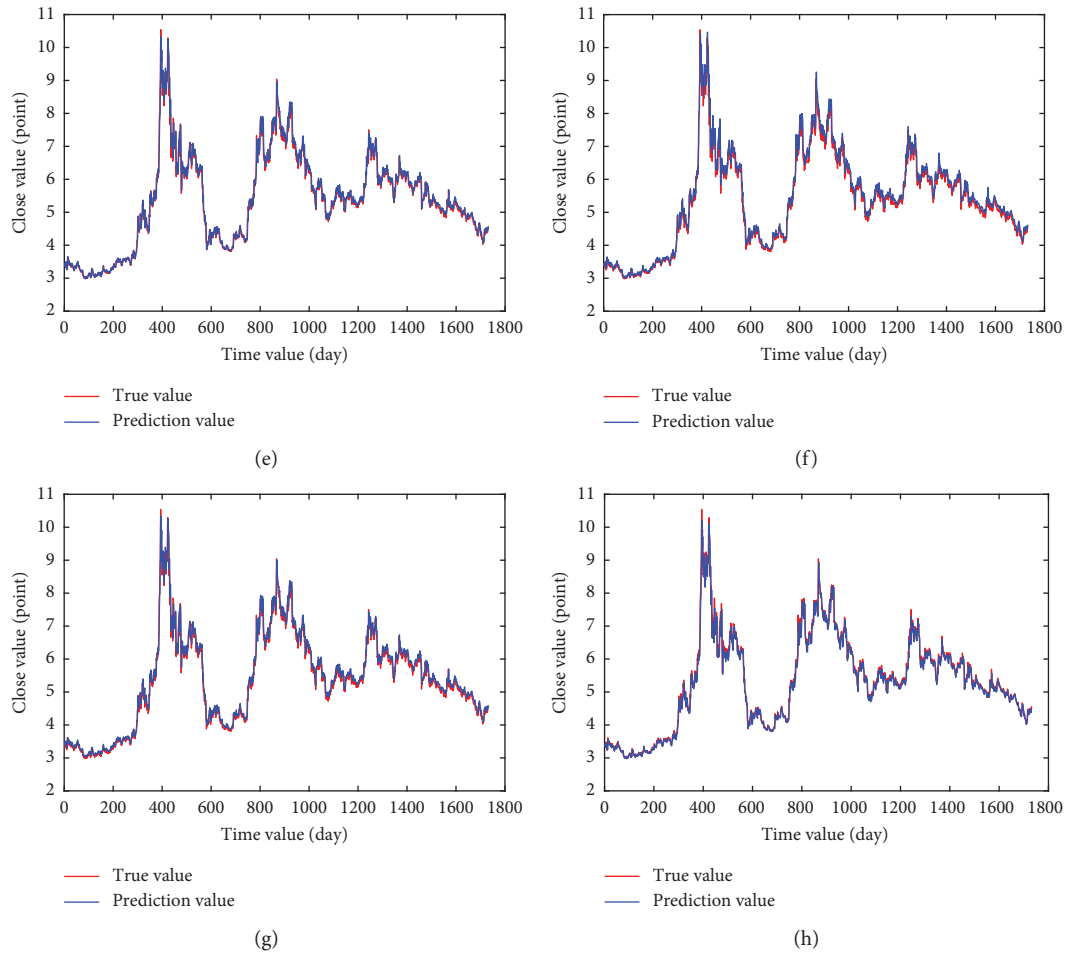
(e)

(f)

(g)

(h)

FIGURE 14: The prediction results of different models for China Unicom. (a) CNN. (b) LSTM. (c) BiLSTM. (d) CNN-LSTM. (e) CNN-BiLSTM. (f) BiLSTM-ECA. (g) CNN-LSTM-ECA. (h) CNN-BiLSTM-ECA.
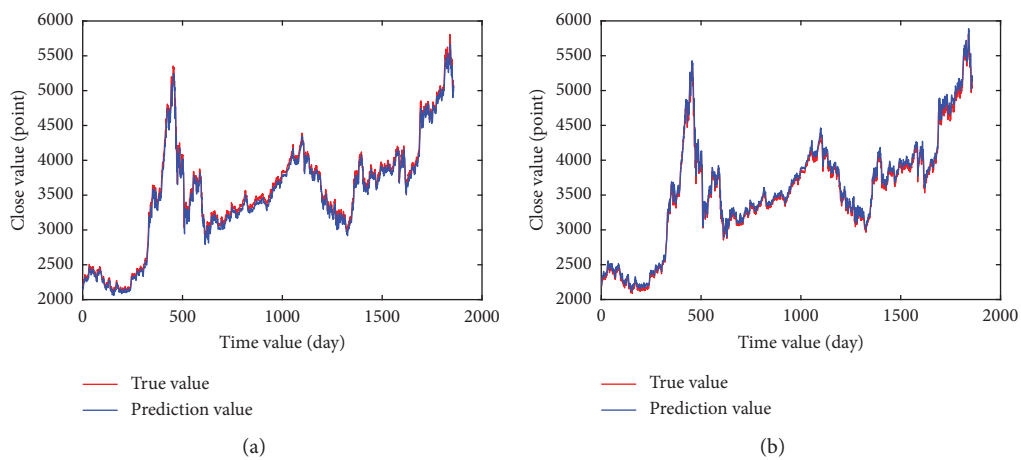

(a)

(b)

FIGURE 15: Continued.
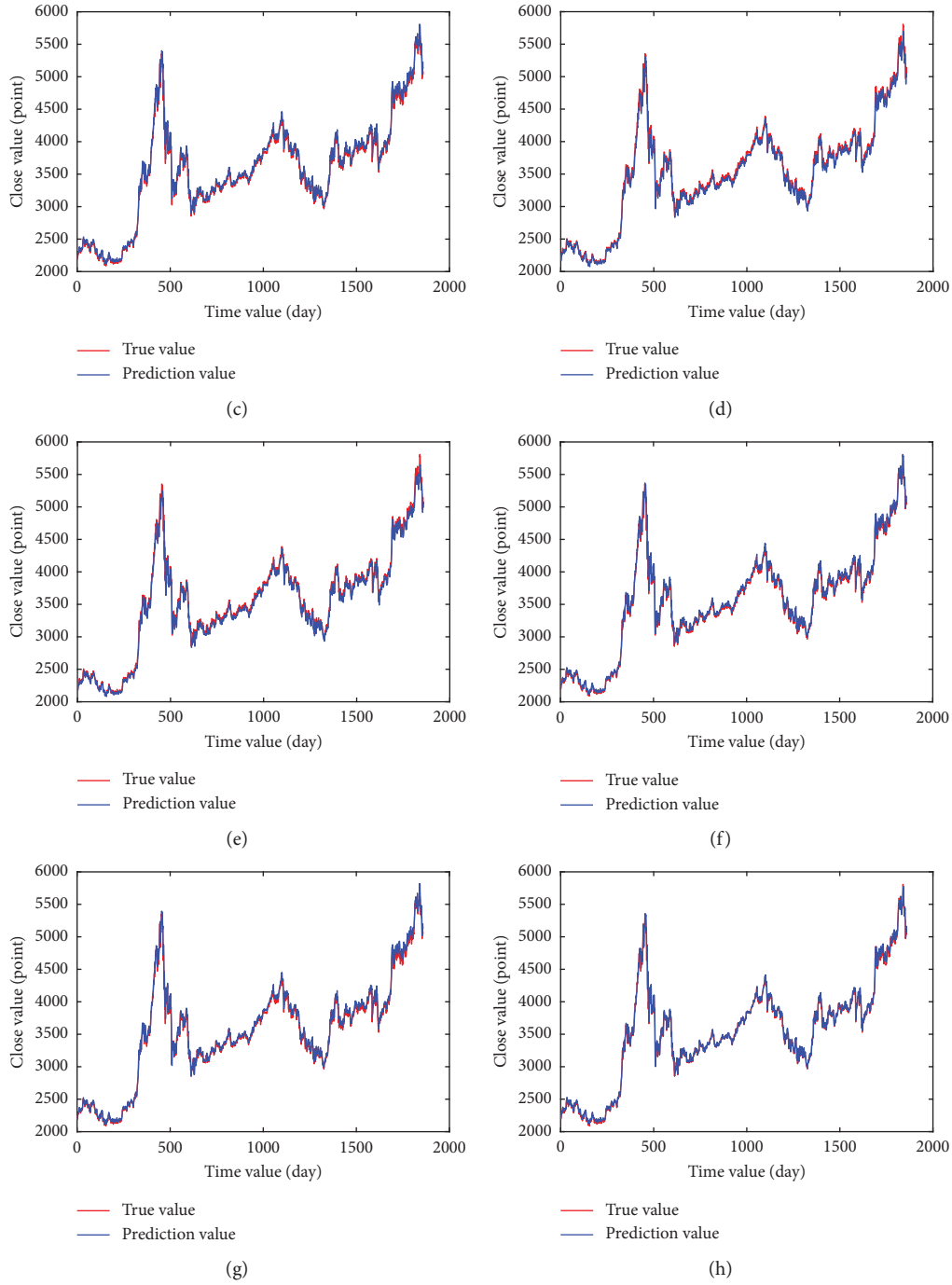
(c)

(d)

(e)

(f)

(g)

(h)

Figure 15: The prediction results of different models for CSI 300. (a) CNN. (b) LSTM. (c) BiLSTM. (d) CNN-LSTM. (e) CNN-BiLSTM. (f) BiLSTM-ECA. (g) CNN-LSTM-ECA. (h) CNN-BiLSTM-ECA.

Table 7: Forecast errors of different network models.

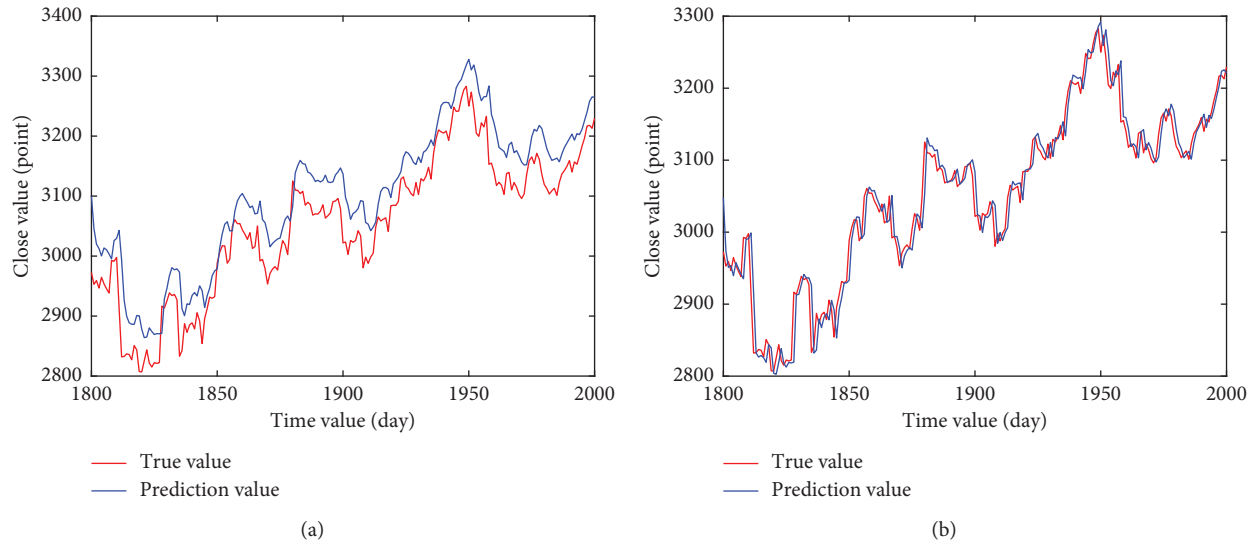| Model	Shanghai Composite Index			China Unicom			CSI 300		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
CNN	8447.149	91.908	79.914	0.037	0.193	0.134	6218.092	78.855	63.981
LSTM	4222.102	64.978	50.585	0.036	0.189	0.128	5809.153	76.218	58.679
BiLSTM	2603.726	51.027	35.419	0.035	0.187	0.132	5091.610	71.356	52.119
CNN-LSTM	2902.648	53.876	39.502	0.030	0.174	0.110	4905.472	70.039	52.457
CNN-BiLSTM	2321.235	48.179	32.289	0.029	0.170	0.110	4643.541	68.144	51.143
BiLSTM-ECA	2184.278	46.736	32.737	0.039	0.198	0.142	4161.203	64.507	46.453
CNN-LSTM-ECA	2200.705	46.911	32.353	0.032	0.180	0.127	4568.808	67.593	51.061
CNN-BiLSTM-ECA	**1956.036**	**44.227**	**28.349**	**0.028**	**0.167**	**0.103**	**3434.408**	**58.604**	**39.111**

FIGURE 16: Local prediction results of LSTM and CNN-BiLSTM-ECA models on the Shanghai composite index. (a) LSTM. (b) CNN-BiLSTM-ECA.
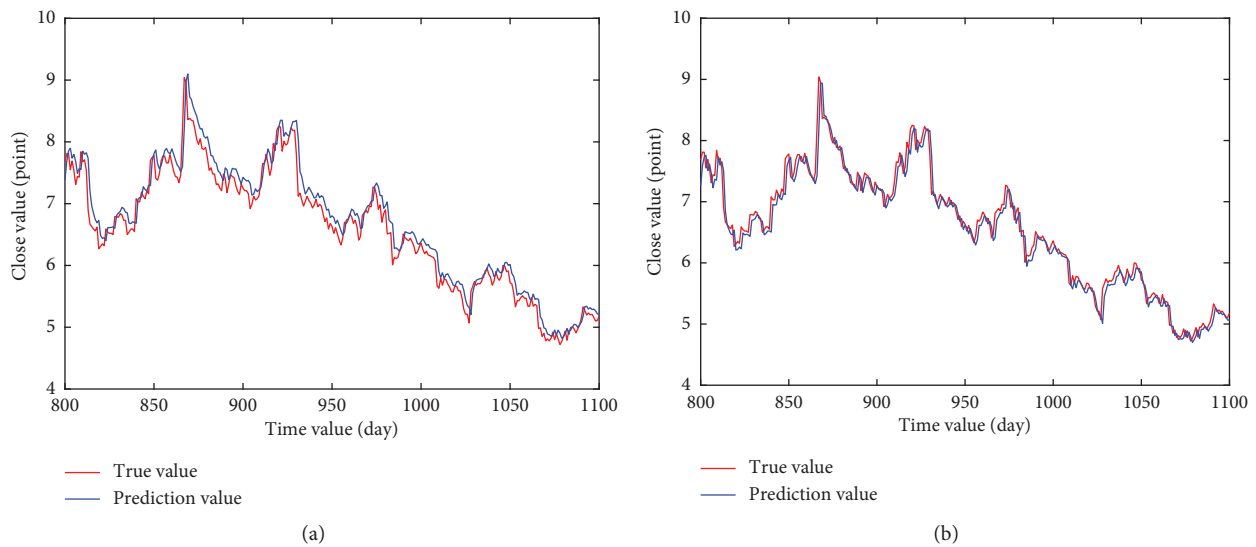

FIGURE 17: Local prediction results of LSTM and CNN-BiLSTM-ECA models on the China Unicom. (a) LSTM. (b) CNN-BiLSTM-ECA.
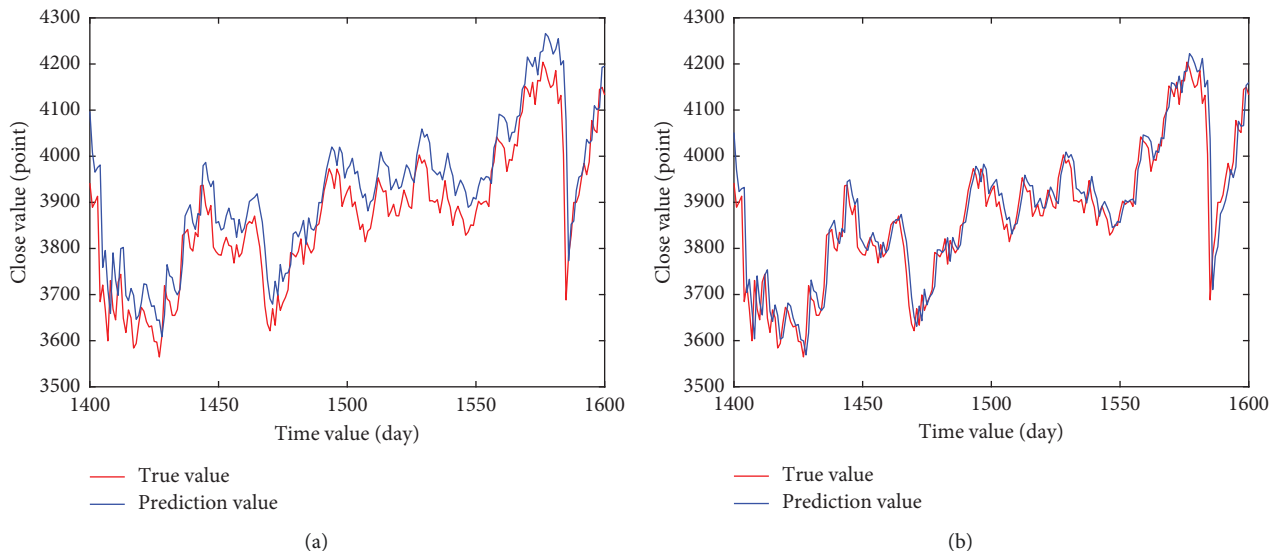

FIGURE 18: Local prediction results of LSTM and CNN-BiLSTM-ECA models on the CSI 300. (a) LSTM. (b) CNN-BiLSTM-ECA.

6. Conclusions

For the stock market, learning the future price is very important for making investment decisions. This paper first proposes a new stock price time series forecasting network model (CNN-BiLSTM-ECA), which takes the stock closing price, the highest price, the lowest price, the opening price, the previous day's closing price, the change, the rise and fall, and other time series data as the input to predict the next day's stock closing price. The proposed network model combines CNN and BiLSTM network models. Firstly, CNN is used to extract the deep features of the input data effectively. Secondly, feature vectors are constructed in time series as inputs to the BILSTM network for learning and prediction. At the same time, ECA attention model is introduced into the model to enhance the importance of learning features. Thus, the useless features are reduced and the accuracy of the model is further improved. The proposed model is compared with CNN, LSTM, BiLSTM, CNN-LSTM, CNN-BiLSTM, BiLSTM-ECA, and CNN-LSTM-ECA network models on three datasets. The experimental results show that the proposed model has the highest prediction accuracy and the best performance. The MSE, RMSE, and MAE of CNN-BiLSTM-ECA are the smallest among all methods, with the values of 1956.036, 44.227, and 28.349, respectively, for the Shanghai Composite Index. Compared with single LSTM model, the reduction was 53.67%, 31.94%, and 43.96%, respectively. It can be seen that it is difficult to achieve high prediction accuracy by using only a single network, but the complexity of the network can improve its prediction accuracy. The CNN-BiLSTM-ECA model proposed in this paper can effectively predict stock prices and provide relevant references for investors to maximize investment returns.

The existing problem is that the input characteristic parameters of the model selection are relatively single. Therefore, it can be considered to increase the training features of the model, such as the emotional or sentiment features of news events or social media [69], so as to improve the prediction performance of the model from the perspective of feature selection. And then, we further apply our model on more application fields, such as gold price prediction, oil price prediction, foreign exchange price prediction, novelty detection [70], and optic disc detection [71]. In addition, a graph-based embedding technology will be introduced to solve the problem of time series prediction [72].

Data Availability

The network code and data are available from the corresponding author upon request.

Conflicts of Interest

All the authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Y. Yi, J. Wang, W. Zhou, C. Zheng, J. Kong, and S. Qiao, "Non-negative matrix factorization with locality constrained adaptive graph," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 427–441, 2020.

[2] Y. Yi, J. Wang, W. Zhou, Y. Fang, J. Kong, and Y. Lu, "Joint graph optimization and projection learning for dimensionality reduction," *Pattern Recognition*, vol. 92, pp. 258–273, 2019.

[3] R. C. Cavalcante, R. C. Brasileiro, V. L. F. Souza, J. P. Nobrega, and A. L. I. Oliveira, "Computational intelligence and financial markets: a survey and future directions," *Expert Systems with Applications*, vol. 55, pp. 194–211, 2016.

[4] H. Song, D. Peng, and X. Huang, "Incorporating research reports and market sentiment for stock excess return prediction: a case of mainland china," *Scientific Programming*, vol. 2020, Article ID 8894757, 7 pages, 2020.

[5] A. Thakkar and K. Chaudhari, "Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions," *Information Fusion*, vol. 65, pp. 95–107, 2021.

[6] G. Kumar, S. Jain, and U. P. Singh, "Stock market forecasting using computational intelligence: a survey," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1069–1101, 2021.

[7] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A systematic review of fundamental and technical analysis of stock market predictions," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 3007–3057, 2020.

[8] L. De Haan, C. Mercadier, and C. Zhou, "Adapting extreme value statistics to financial time series: dealing with bias and serial dependence," *Finance and Stochastics*, vol. 20, no. 2, pp. 321–354, 2016.

[9] H. Tang, P. Dong, and Y. Shi, "A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points," *Applied Soft Computing*, vol. 78, pp. 685–696, 2019.

[10] W. Chen, M. Jiang, W.-G. Zhang, and Z. Chen, "A novel graph convolutional feature based convolutional neural network for stock trend prediction," *Information Sciences*, vol. 556, pp. 67–94, 2021.

[11] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: a systematic

literature review: 2005–2019," *Applied Soft Computing*, vol. 90, Article ID 106181, 2020.

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

[13] E. Hajizadeh, A. Seifi, M. H. Fazel Zarandi, and I. B. Turksen, "A hybrid modeling approach for forecasting the volatility of S&P 500 index return," *Expert Systems with Applications*, vol. 39, no. 1, pp. 431–436, 2012.

[14] A. M. Rather, A. Agarwal, and V. N. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3234–3241, 2015.

[15] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994.

[16] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 114–119, 2019.

[17] G. Ding and L. Qin, "Study on the prediction of stock price based on the associated network model of LSTM," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 6, pp. 1307–1317, 2020.

[18] X. Yan, W. Weihan, and M. Chang, "Research on financial assets transaction prediction model based on LSTM neural network," *Neural Computing and Applications*, vol. 33, no. 1, pp. 257–270, 2021.

[19] 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.

[20] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1419–1426, Anchorage, AK, USA, 2017.

[21] P.-Y. Hao, C.-F. Kung, C.-Y. Chang, and J.-B. Ou, "Predicting stock price trends based on financial news articles and using a novel twin support vector machine with fuzzy hyperplane," *Applied Soft Computing*, vol. 98, Article ID 106806, 2021.

[22] C.-H. Cheng, T.-L. Chen, and L.-Y. Wei, "A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting," *Information Sciences*, vol. 180, no. 9, pp. 1610–1629, 2010.

[23] Y. Zhang and L. Wu, "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network," *Expert Systems with Applications*, vol. 36, no. 5, pp. 8849–8854, 2009.

[24] D. Zhang and S. Lou, "The application research of neural network and BP algorithm in stock price pattern classification and prediction," *Future Generation Computer Systems*, vol. 115, pp. 872–879, 2021.

[25] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[26] Y. Xu, L. Chhim, B. Zheng, and Y. Nojima, "Stacked deep learning structure with bidirectional long-short term memory for stock market prediction," in *Proceedings of the 2020 International Conference on Neural Computing for Advanced Applications*, pp. 447–460, Shenzhen, China, 2020.

[27] Q. Chen, W. Zhang, and Y. Lou, "Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network," *IEEE Access*, vol. 8, pp. 117365–117376, 2020.

[28] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," *Neural Computing and Applications*, vol. 33, pp. 4741–4753, 2020.

[29] W. Chen, S. Wang, X. Zhang et al., "EEG-based motion intention recognition via multi-task RNNs," in *Proceedings of the 2018 SIAM International Conference on Data Mining*, San Diego, CA, USA, 2018.

[30] W. Chen, S. Wang, G. Long, L. Yao, Q. Z. Sheng, and X. Li, "Dynamic illness severity prediction via multi-task RNNS for intensive care unit," in *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*, IEEE, Singapore, 2018.

[31] S. Kastner and L. G. Ungerleider, "Mechanisms of visual attention in the human cortex," *Annual Review of Neuroscience*, vol. 23, no. 1, pp. 315–341, 2000.

[32] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016.

[33] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019.

[34] H. Choi, K. Cho, and Y. Bengio, "Fine-grained attention mechanism for neural machine translation," *Neurocomputing*, vol. 284, pp. 171–176, 2018.

[35] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–18, 2020.

[36] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep audio-visual speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–11, 2018.

[37] C. Li and Y. Qian, "Deep audio-visual speech separation with attention mechanism," in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7314–7318, Barcelona, Spain, 2020.

[38] D. T. Tran, A. Iosifidis, J. Kanniainen, and M. Gabbouj, "Temporal attention-augmented bilinear network for financial time-series data analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1407–1418, 2018.

[39] S. Y. Shih, F. K. Sun, and H. Lee, "Temporal pattern attention for multivariate time series forecasting," *Machine Learning*, vol. 108, no. 8, pp. 1421–1441, 2019.

[40] J. Hu and W. Zheng, "Multistage attention network for multivariate time series prediction," *Neurocomputing*, vol. 383, pp. 122–137, 2020.

[41] J. Bedi, "Attention based mechanism for load time series forecasting: AN-LSTM," in *Proceedings of the 2020 International Conference on Artificial Neural Networks*, pp. 838–849, Bratislava, Slovakia, 2020.

[42] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies," *Expert Systems with Applications*, vol. 83, pp. 187–205, 2017.

[43] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18569–18584, 2017.

[44] A. M. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, "Deep learning for financial applications: a survey," *Applied Soft Computing*, vol. 93, Article ID 106384, 2020.

[45] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-net: efficient channel attention for deep convolutional neural networks," in *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.

[46] R. K. Nayak, D. Mishra, and A. K. Rath, "A Naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices," *Applied Soft Computing*, vol. 35, pp. 670–680, 2015.

[47] Y. Chen and Y. Hao, "A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction," *Expert Systems with Applications*, vol. 80, pp. 340–355, 2017.

[48] W.-C. Chiang, D. Enke, T. Wu, and R. Wang, "An adaptive stock index trading decision support system," *Expert Systems with Applications*, vol. 59, pp. 195–207, 2016.

[49] X.-d. Zhang, A. Li, and R. Pan, "Stock trend prediction based on a new status box method and AdaBoost probabilistic support vector machine," *Applied Soft Computing*, vol. 49, pp. 385–398, 2016.

[50] I. Marković, M. Stojanović, J. Stanković et al., "Stock market trend prediction using AHP and weighted kernel LS-SVM," *Soft Computing*, vol. 21, no. 18, pp. 5387–5398, 2017.

[51] L. Lei, "Wavelet neural network prediction method of stock price trend based on rough set attribute reduction," *Applied Soft Computing*, vol. 62, pp. 923–932, 2018.

[52] A. Picasso, S. Merello, Y. Ma, L. Oneto, and E. Cambria, "Technical analysis and sentiment embeddings for market trend prediction," *Expert Systems with Applications*, vol. 135, pp. 60–70, 2019.

[53] I. R. Parray, S. S. Khurana, M. Kumar, and A. A. Altalbe, "Time series data analysis of stock price movement using machine learning techniques," *Soft Computing*, vol. 24, no. 21, pp. 16509–16517, 2020.

[54] Y. Xu, C. Yang, S. Peng, and Y. Nojima, "A hybrid two-stage financial stock forecasting algorithm based on clustering and ensemble learning," *Applied Intelligence*, vol. 50, no. 11, pp. 3852–3867, 2020.

[55] R. K. Dash, T. N. Nguyen, K. Cengiz, and A. Sharma, "Fine-tuned support vector regression model for stock predictions," *Neural Computing and Applications*, pp. 1–15, 2021.

[56] Q. Liang, W. Rong, J. Zhang, J. Liu, and Z. Xiong, "Restricted Boltzmann machine based stock market trend prediction," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1380–1387, Anchorage, AK, USA, 2017.

[57] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[58] B. Moews, J. M. Herrmann, and G. Ibikunle, "Lagged correlation-based deep learning for directional trend change prediction in financial time series," *Expert Systems with Applications*, vol. 120, pp. 197–206, 2019.

[59] H. Li, J. Hua, J. Li, and G. Li, "Stock forecasting model FS-LSTM based on the 5G internet of things," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 7681209, 7 pages, 2020.

[60] J. Zhao, D. Zeng, S. Liang, H. Kang, and Q. Liu, "Prediction model for stock price trend based on recurrent neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 745–753, 2021.

[61] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: time series to image conversion approach," *Applied Soft Computing*, vol. 70, pp. 525–538, 2018.

[62] 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.

[63] S. Barra, S. M. Carta, A. Corriga, A. S. Podda, and D. R. Recupero, "Deep learning and time series-to-image encoding for financial forecasting," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 683–692, 2020.

[64] J. Long, Z. Chen, W. He, T. Wu, and J. Ren, "An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market," *Applied Soft Computing*, vol. 91, Article ID 106205, 2020.

[65] Y. Hao and Q. Gao, "Predicting the trend of stock market index using the hybrid neural network based on multiple time scale feature learning," *Applied Sciences*, vol. 10, no. 11, p. 3961, 2020.

[66] 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.

[67] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[68] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.

[69] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media," *Knowledge and Information Systems*, vol. 60, no. 2, pp. 617–663, 2019.

[70] Y. Yi, Y. Shi, W. Wang, G. Lei, J. Dai, and H. Zheng, "Combining boundary detector and SND-SVM for fast learning," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 3, pp. 689–698, 2021.

[71] W. Zhou, S. Qiao, Y. Yi, N. Han, Y. Chen, and G. Lei, "Automatic optic disc detection using low-rank representation based semi-supervised extreme learning machine," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 1, pp. 55–69, 2020.

[72] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based poi embedding for location-based recommendation," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, IN, USA, 2016.