JUSTC

Article

http://justc.ustc.edu.cn

Received: November 16, 2022; Accepted: May 29, 2023

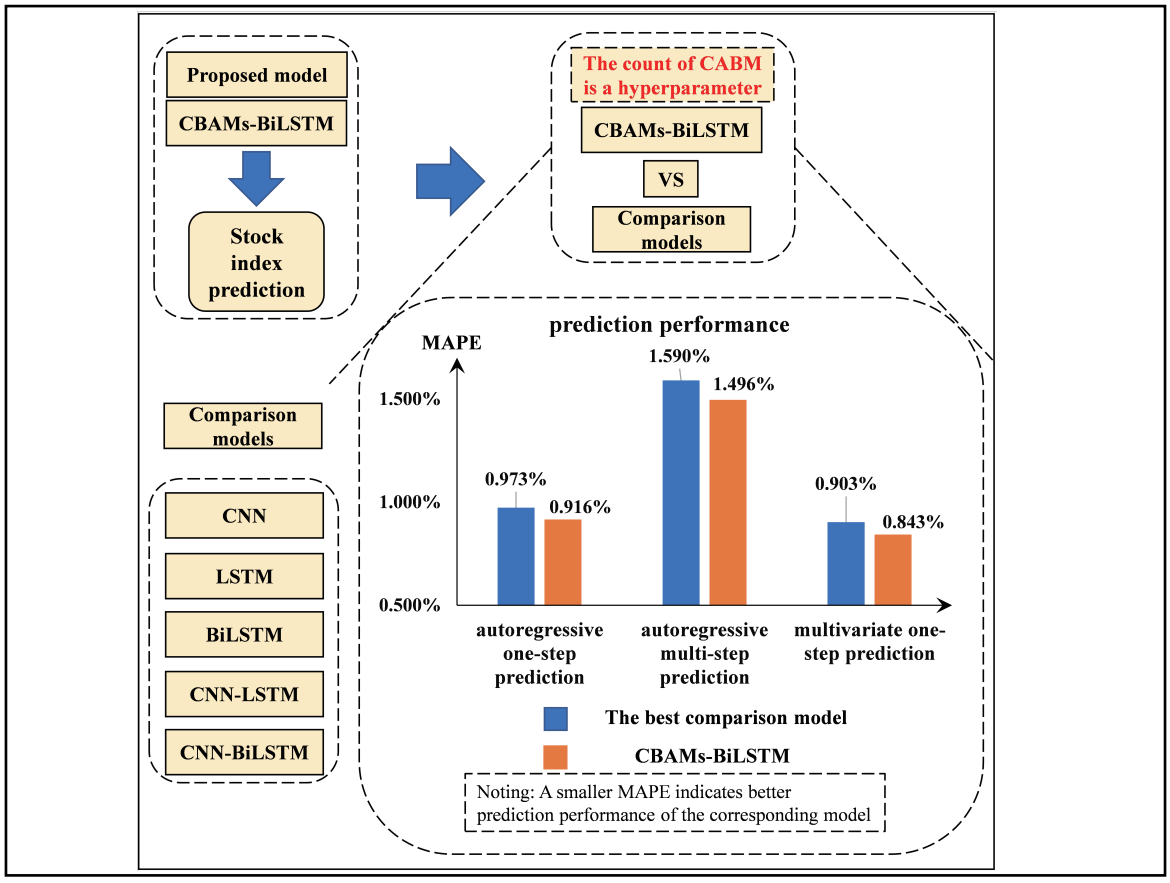
A novel CBAMs-BiLSTM model for Chinese stock market forecasting

Chenhao Cui, and Yong Li ✉

School of Management, University of Science and Technology of China, Hefei 230026, China

✉Correspondence: Yong Li, E-mail: yonglee@ustc.edu.cn

Graphical abstract



The proposed CBAMs-BiLSTM model has robustness and superiority in prediction performance.

Public summary

- The impact of the position and quantity of CBAM on the prediction performance of the stock index is analyzed.
- A novel stock index forecasting model, CBAMs-BiLSTM, with superiority and robustness in prediction performance and simulated returns, is proposed.
- The MCS test is introduced to measure the prediction performance of different models.

Citation: Cui C H, Li Y. A novel CBAMs-BiLSTM model for Chinese stock market forecasting. *JUSTC*, **2024**, 54(2): 0204. DOI: 10.52396/JUSTC-2023-0066

JUSTC

Article

http://justc.ustc.edu.cn

Received: November 16, 2022; Accepted: May 29, 2023

A novel CBAMs-BiLSTM model for Chinese stock market forecasting

Chenhao Cui, and Yong Li ✉

School of Management, University of Science and Technology of China, Hefei 230026, China

✉Correspondence: Yong Li, E-mail: yonglee@ustc.edu.cn

Cite This: *JUSTC*, **2024**, 54(2): 0204 (14pp)

Read Online

Abstract: The convolutional block attention module (CBAM) has demonstrated its superiority in various prediction problems, as it effectively enhances the prediction accuracy of deep learning models. However, there has been limited research testing the effectiveness of CBAM in predicting stock indexes. To fill this gap and improve the prediction accuracy of stock indexes, we propose a novel model called CBAMs-BiLSTM, which combines multiple CBAM modules with a bidirectional long short-term memory network (BiLSTM). In this study, we employ the standard metric evaluation method (SME) and the model confidence set test (MCS) to comprehensively evaluate the superiority and robustness of our model. We utilize two representative Chinese stock index data sets, namely, the SSE Composite Index and the SZSE Composite Index, as our experimental data. The numerical results demonstrate that CBAMs-BiLSTM outperforms BiLSTM alone, achieving average reductions of 13.06%, 13.39%, and 12.48% in MAE, RMSE, and MAPE, respectively. These findings confirm that CBAM can effectively enhance the prediction accuracy of BiLSTM. Furthermore, we compare our proposed model with other popular models and examine the impact of changing data sets, prediction methods, and the size of the training set. The results consistently demonstrate the superiority and robustness of our proposed model in terms of prediction accuracy and investment returns.

Keywords: stock index prediction; BiLSTM; CBAM; MCS; SME

CLC number: F830.91 **Document code:** A

1 Introduction

Stocks play a pivotal role in financial markets, making stock indexes of great interest to regulators and investors alike. At the macro level, stock indexes are influential factors in the stability of the financial environment and economic development, as well as serving as early warning indicators for the economic climate[1]. Thus, stock indexes hold significant importance for regulators. On a micro level, the fluctuations of stock indexes directly impact investment risks and returns. Accurate prediction of stock indexes not only aids regulators in overseeing stock markets but also assists investors in making informed investment decisions. However, the prediction of stock indexes is a challenging task due to the complex factors influencing them, such as price levels, monetary policies, and market interest rates[2, 3].

Traditionally, researchers in stock index prediction have favored statistical methods, such as regression analysis[4], generalized autoregressive conditional heteroscedasticity (GARCH)[5], autoregressive integrated moving average (ARIMA)[6, 7], and smooth transition autoregressive model (STAR)[8]. However, these methods rely on assumptions of time series stationarity and linearity among normally distributed variables, which are not satisfied in real stock markets[9]. Consequently, these models exhibit poor prediction accuracy when dealing with nonlinear and nonstationary stock data[10].

Machine learning models, particularly neural networks, have shown better performance in extracting nonlinearity and nonstationarity from financial time series compared to classical statistical models[11]. Neural networks leverage nonlinear activation functions to capture complex information in the data[12, 15]. For instance, Yu et al.[13] utilized a local linear embedding dimensionality reduction algorithm (LLE) to reduce the dimensionality of factors influencing stock indexes. They then employed a back-propagation (BP) neural network to optimize stock index prediction. Recurrent neural networks (RNNs) in deep learning can effectively extract autocorrelation information due to their recurrent structure[16]. Long short-term memory (LSTM), a type of RNN, not only extracts autocorrelation information but also addresses the vanishing or exploding gradient problem through gating functions[17]. Bidirectional LSTM (BiLSTM) differs from LSTM by considering both historical and future information, enhancing sequence analysis[18]. Several studies have confirmed the predictive superiority of BiLSTM over LSTM in stock data[19–21].

The attention mechanism (AM) is a network module that dynamically learns the weights of each feature, while the convolutional block attention module (CBAM) represents an enhanced version of the attention mechanism. CBAM introduces an attention mechanism for spaces and channels, enabling models to focus on essential features and disregard irrelevant ones, thereby improving the prediction accuracy of

DOI: 10.52396/JUSTC-2023-0066

JUSTC, **2024**, 54(2): 0204

network models[14]. Additionally, CBAM effectively reduces the interference caused by redundant features[22]. Cheng et al.[23] integrated CBAM into a temporal convolutional network (TCN) to create the hybrid model TCN-CBAM for predicting chaotic time series. The experimental results demonstrate that incorporating CBAM significantly enhances the prediction accuracy of the TCN. Li et al.[24] proposed a fault diagnosis model for rolling bearings that combines a dual-stage attention-based recurrent neural network (DA-RNN), CBAM, and convolutional neural network (CNN). By utilizing two vibration data sets from rolling bearings, they confirmed that the proposed DARNN-CBAM-CNN method improves the fault diagnosis accuracy by 1.90% compared to a DARNN-CNN method without CBAM. In the domain of gold price prediction, Liang et al.[14] highlighted that CBAM, unlike the attention mechanism, allocated weights across the two independent dimensions of channel and space, leading to better prediction accuracy in theory. Moreover, CBAM has proven effective in improving prediction accuracy in other areas, such as global horizontal irradiance (GHI) prediction[25] and PM2.5 concentration prediction[26]. However, despite the extensive research on CBAM's effectiveness in other fields, it has been relatively underutilized in stock index prediction. Furthermore, existing studies lack a detailed analysis of whether the position and quantity of CBAM in models affect prediction accuracy.

In summary, this paper aims to leverage the proven superiority of CBAM in other prediction problems and the established effectiveness of BiLSTM in stock data. To achieve this, the paper proposes a novel model called CBAM-BiLSTM, which combines CBAM with BiLSTM to further enhance the prediction accuracy of stock indexes. The experimental data consist of two representative Chinese stock index data sets, namely, the SSE Composite Index and the SZSE Composite Index. The prediction accuracy of the models is assessed using standard metric evaluation methods (SME) and the model confidence set test (MCS). For comparison, classical models in time series prediction problems, such as BiLSTM, CNN, LSTM, CNN-LSTM, and CNN-BiLSTM, are chosen as benchmark models.

The initial experiments focus on conducting a detailed analysis of how the position and quantity of CBAM affect the prediction accuracy of BiLSTM. The numerical results demonstrate that the proposed model exhibits significant improvements compared to BiLSTM alone, with an average reduction of 13.06%, 13.39%, and 12.48% in MAE, RMSE, and MAPE, respectively, and an average improvement of 1.98% in R^2. These findings confirm that the combination of CBAM and BiLSTM can further enhance the prediction accuracy of BiLSTM.

Furthermore, the paper validates the superiority and robustness of the proposed CBAM-BiLSTM model by comparing it with other popular models and evaluating its performance under different data sets, prediction methods, and training set sizes. This analysis encompasses both prediction accuracy and investment returns.

The innovations and contributions of this paper can be summarized as follows. First, the paper introduces a rational strategy that combines CBAM and BiLSTM to propose the advanced CBAM-BiLSTM model, thereby further improving the accuracy of stock index prediction. Second, the paper conducts a detailed analysis to investigate the impact of the position and quantity of CBAM on the prediction accuracy of BiLSTM.

The rest of the paper is organized as follows. Section 2 presents the methodology, which provides a detailed explanation of the structure and principles of the proposed CBAM-BiLSTM model. Section 3 comprises an analysis of the experiments, including information about the experimental data, experimental design, and result analysis. Finally, Section 4 concludes the article by summarizing the key findings, discussing some shortcomings, and outlining future research plans.

2 Methodology

2.1 Structure and principle of CBAM-BiLSTM

The structure of CBAM-BiLSTM is shown in Fig. 1. Multiple CBAMs and a BiLSTM are mixed using a linear stacking approach, with multiple CBAMs placed in front of BiLSTM used to achieve a sufficiently rational distribution of attention weights to input features. In CBAM-BiLSTM, the number of CBAMs is a hyperparameter that needs to be set artificially. Similar to other hyperparameters in a deep learning model, the number of CBAMs can be chosen as an appropriate value by comparing the prediction accuracy of models on the validation set. CBAMn-BiLSTM is CBAM-BiLSTM containing n CBAM. For example, CBAM3-BiLSTM means that the CBAM-BiLSTM model contains three CBAM

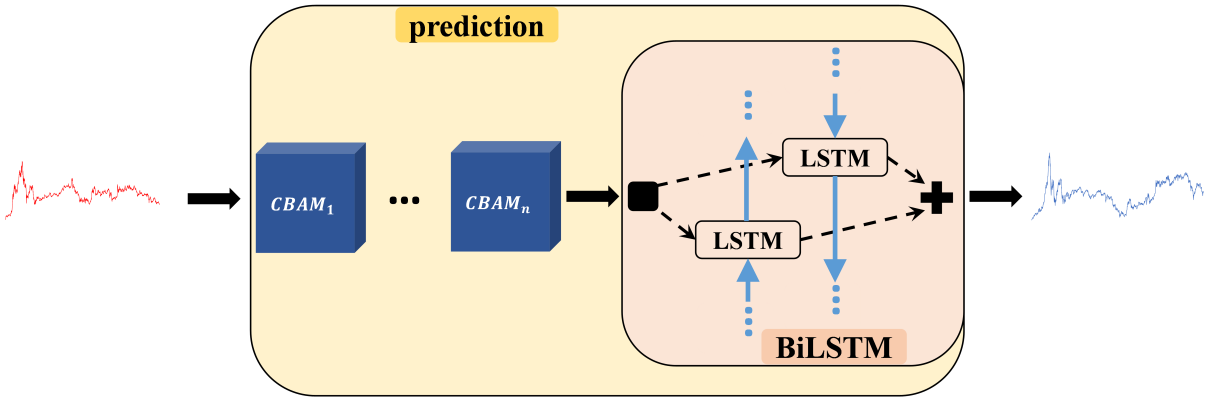


Fig. 1. Structure of CBAM-BiLSTM.

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

modules. The structures of CBAM and BiLSTM are described in detail below.

2.2 Structure and principle of BiLSTM

LSTM overcomes the problem of gradient disappearance or explosion that RNNs have by introducing long-term memory states and multiple gating functions. These gating functions selectively forget or remember new information in the long-term memory state, which in turn allows information useful for subsequent moments of computation to be passed and useless information to be discarded[27, 28]. BiLSTM consists of two independent LSTM layers that have the same input but transfer information in opposite directions. Therefore, compared to LSTM, BiLSTM can improve the prediction accuracy by fully considering both historical and future information. The cell structure of LSTM and the network structure of BiLSTM are shown in Fig. 2.

LSTM uses three gating functions to control the value of the long-term memory state c_t[29, 30]. The forgetting gating function f_t determines the amount of information retained from the previous time-unit state c_{t-1} to the current time-unit state c_t. The input gating function i_t controls the amount of information preserved in the current input x_t to the long-term memory state c_t. The output gating function o_t controls the amount of information input to the current output h_t from the long-term memory state c_t. LSTM has three inputs at moment t, including current time input value x_t, last time output value h_{t-1}, and last long-term memory state c_{t-1}. LSTM has two outputs at moment t: the current time output value h_t and the current long-term memory state c_t. Eqs. (1)–(6) represent the update process of LSTM at moment t.

$$o_t = \text{sigmoid}(w_o \cdot [h_{t-1}, x_t] + b_o), \tag{1}$$

$$f_t = \text{sigmoid}(w_f \cdot [h_{t-1}, x_t] + b_f), \tag{2}$$

$$i_t = \text{sigmoid}(w_i \cdot [h_{t-1}, x_t] + b_i), \tag{3}$$

$$\hat{c}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c), \tag{4}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \hat{c}_t, \tag{5}$$

$$h_t = \tanh(c_t) \circ o_t. \tag{6}$$

Here, w, b are the weight matrix and the deviation vector of the corresponding gating functions, respectively. $\hat{c}_t$ denotes the long-term memory state of the current input. "$\circ$" denotes the scalar product between vectors. "$\cdot$" denotes matrix multiplication.

In the network structure of BiLSTM, the same input data are fed to the forward LSTM layer and backward LSTM layer, and the hidden state h_t^f in the forward LSTM layer and the hidden state h_t^b in the backward LSTM layer are computed. In the forward LSTM layer, forward computation is performed from time 1 to time t. In the backward LSTM layer, backward computation is performed from time t to time 1. The outputs of the current prehidden state and posthidden state are obtained and saved at each time unit. Then, two hidden states are connected to calculate the output value of BiLSTM. Eqs. (7)–(9) represent the calculation process of BiLSTM.

$$h_t^f = \text{LSTM}(x_t, h_{t-1}^f), \tag{7}$$

$$h_t^b = \text{LSTM}(x_t, h_{t-1}^b), \tag{8}$$

$$o_t = w_f \cdot h_t^f + w_b \cdot h_t^b + b. \tag{9}$$

Here, LSTM$(\cdot)$ denotes the mapping of the already defined LSTM network layers. w_f and w_b denote the weight matrices of the forward LSTM layer and backward LSTM layer, respectively. b denotes the deviation vector of the output layer.

2.3 Structure and principle of CBAM

CBAM can implement an attention mechanism on both space and channel, which in turn can focus on key features and ignore useless features. After features are brought up by the convolutional neural network implementation, CBAM computes the weight mapping of feature mapping from both channel and spatial dimensions and then multiplies the weights with input features for adaptive learning. This lightweight

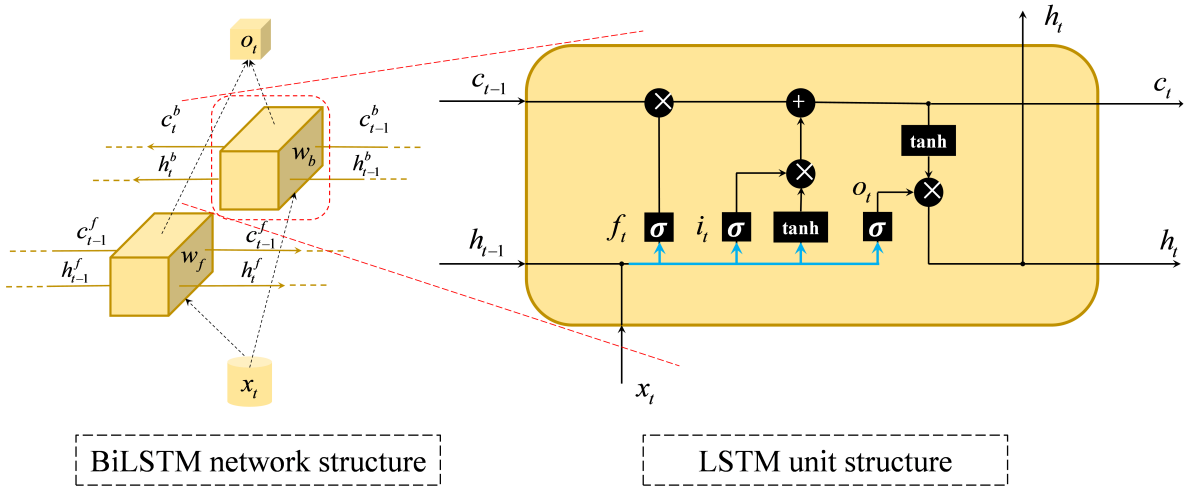


Fig. 2. Structure of LSTM and BiLSTM.

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

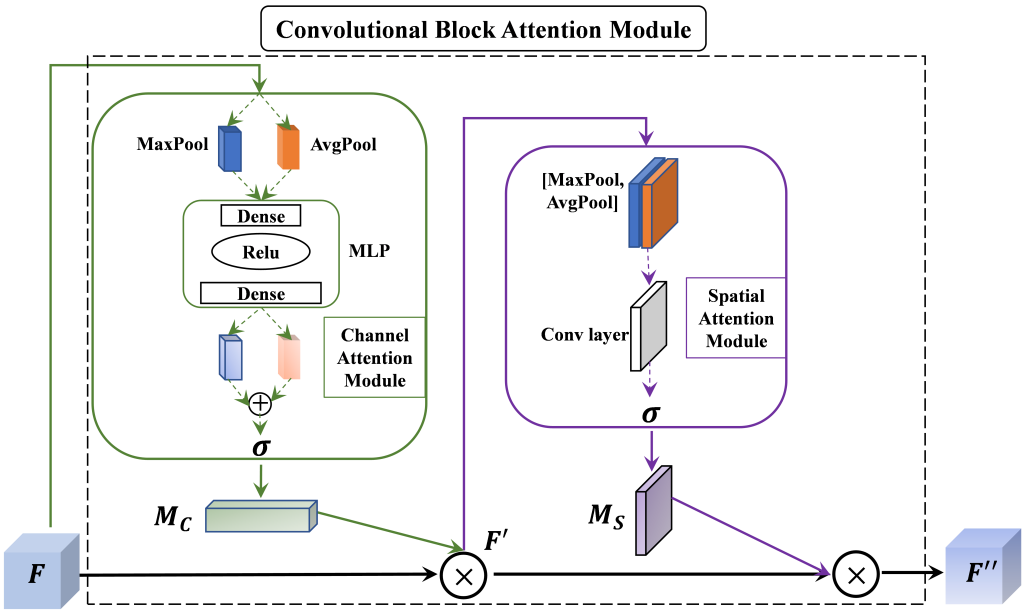


Fig. 3. Structure of CBAM.

general-purpose module can be integrated into a variety of convolutional neural networks for end-to-end training[31]. Fig. 3 illustrates the network structure of CBAM.

From Fig. 3, the channel attention module (CAM) outputs a one-dimensional channel attention vector M_C, which is used to assign weights to each channel, indicating the importance of each channel. The spatial attention module outputs a three-dimensional spatial attention tensor M_S, which indicates which features at which locations in the three-dimensional space are key features and which are secondary features. Eqs. (10) and (11) represent the whole calculation process.

$$F' = M_C(F) \otimes F, \tag{10}$$

$$F'' = M_S(F') \otimes F'. \tag{11}$$

Here, $M_C(F)$ represents the output of the channel attention module when the input is F. $M_S(F')$ represents the output of the spatial attention module when the input is F'. $\otimes$ represents element multiplication. Pooling operations in CBAM include two types: "MaxPool" and "AvgPool". Pooling can extract high-level features, and different pooling methods mean that the extracted high-level features are richer. From Fig. 3, we can see that Eq. (12) represents the computation process of the channel attention module, and that Eq. (13) represents the computation process of the spatial attention module.

$$M_C(F) = \text{sigmoid}(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))), \tag{12}$$

$$M_S(F) = \text{sigmoid}(\text{Conv}([\text{AvgPool}(F), \text{MaxPool}(F)])). \tag{13}$$

Here, AvgPool(·) is the average pooling of input features. MaxPool(·) maximizes the pooling of input features. MLP(·) is the output of a multilayer perceptron. Conv(·) is the output of a convolutional layer.

2.4 Evaluation of model performance

The standard metric evaluation method (SME) and model confidence set test (MCS) are used to comprehensively evaluate the performance of the models.

2.4.1 Standard metric evaluation method

Loss error is the difference between the observed and predicted values and is used to evaluate the prediction accuracy of models. The mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), and coefficient of fit (R^2) are chosen to comprehensively evaluate the prediction accuracy of the models. Smaller MAE, RMSE, and MAPE values indicate a higher prediction accuracy of the models; larger R^2 values indicate a higher prediction accuracy of the models. The true value is $y = (y_1, y_2, \cdots, y_n)$, and the predicted value is $\hat{y} = (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n)$. The equations below are expressions of metrics.

$$\text{MAE} = \frac{1}{n}\sum_i |y_i - \hat{y}_i|, \tag{14}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_i (y_i - \hat{y}_i)^2}, \tag{15}$$

$$\text{MAPE} = \frac{1}{n}\sum_i \frac{|y_i - \hat{y}_i|}{|y_i| + 10^{-7}}, \tag{16}$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}. \tag{17}$$

Here, $\bar{y} = \frac{1}{n}\sum_i y_i$. RMSE is the arithmetic square root of the mean squared error (MSE), which is more intuitive in order of magnitude than MSE; hence, RMSE was chosen for the

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

experiments. RMSE and MAE reflect absolute errors, so MAPE and R^2 are introduced to measure the relative errors of models. A small perturbation term, 10^{-7}, is added to MAPE to ensure that the denominator is not 0. This has no impact on the evaluation of models by this metric. For R^2, the closer the value is to 1, the better the variance of the target feature is explained by the models, which further indicates that the models are more accurate in predicting the target feature.

The Sharpe ratio is an indicator that combines the returns and risk of an investment. Experiments use the Sharpe ratio to evaluate the superiority of models in terms of investment returns. The expression of the Sharpe ratio is given as

$$\text{SharpeRatio} = \frac{E(R_p) - R_f}{\sigma_p}. \tag{18}$$

Here, R_p is the return sequence. $E(R_p)$ is the mean of R_p. R_f is the risk-free rate. σ_p is the standard deviation of R_p. To facilitate calculation, let $R_f = 0$.

2.4.2 Model confidence set test

The model confidence set (MCS) test proposed by Hansen et al.[33] is used to test whether there is a significant difference between the prediction accuracy of different models. More conveniently, we can calculate MCS p values of models to quantify the model's prediction accuracy and to visually compare the strengths and weaknesses of different models' prediction accuracy. This method is widely used to test differences in prediction performance between different predictive models[34–36].

We assume that there are m predictive models and s samples to be predicted. $M = \{m_1, m_2, \cdots, m_m\}$ is a set of all predictive models. For each predictive model, there are s predictions $\hat{y}$. In MCS, loss functions need to be selected for calculating the loss generated by each sample. $\text{Loss} = \text{Loss}(y_i, \hat{y}_i)$ is a loss function. The loss functions chosen for the experiments are as follows. $L_{i,j}$ is the loss value calculated for the ith model with the jth sample according to Loss. $d_{(u,v),j} = L_{u,j} - L_{v,j}$ is the loss difference between the uth model and vth model computed on the jth sample according to Loss.

$$\text{Loss1} : \text{MSE} = \frac{1}{n}\sum_i (y_i - \hat{y}_i)^2,$$

$$\text{Loss2} : \text{MAE} = \frac{1}{n}\sum_i |y_i - \hat{y}_i|,$$

$$\text{Loss3} : \text{HMSE} = \frac{1}{n}\sum_i \left(1 - \frac{\hat{y}_i}{y_i}\right)^2,$$

$$\text{Loss4} : \text{HMAE} = \frac{1}{n}\sum_i \left|1 - \frac{\hat{y}_i}{y_i}\right|.$$

The MCS test is designed to test the significance of differences in the prediction accuracy of models in a set and to eliminate the model with poor prediction accuracy. Therefore, in each test, the null hypothesis is that all models have the same prediction accuracy. That is.

$$H_{0,N} : E(d_{(u,v),j}) = 0, \quad u, v \in N \subset M.$$

Here, $E(d_{(u,v),j})$ is the mean of $d_{(u,v),j}$. When the null hypothesis is rejected, the model with poor prediction accuracy in N is removed. The MCS steps are as follows.

Step 1: Let $N = M$.

Step 2: At significance level α, the null hypothesis is tested to see if it holds.

Step 3: If the null hypothesis is accepted, then let $N^*_{1-\alpha} = M$; otherwise, remove the model with poor prediction accuracy from N until the null hypothesis is not rejected.

After implementing the steps, $N^*_{1-\alpha}$ will contain the models that are optimal at the confidence level of $1-\alpha$. To test the null hypothesis, the following two statistics are experimentally selected.

$$T_{\text{R}} = \max_{u,v \in N} \frac{|\bar{d}_{(u,v)}|}{\sqrt{\widehat{\text{var}}(\bar{d}_{(u,v)})}}, \tag{19}$$

$$T_{\text{MAX}} = \max_{u \in N} \frac{\bar{d}_{(u,\cdot)}}{\sqrt{\widehat{\text{var}}(\bar{d}_{(u,\cdot)})}}. \tag{20}$$

Here, $|N|$ is the count of elements in N. $\bar{d}_{(u,v)} = \frac{1}{s}\sum_j d_{(u,v),j}$. $\bar{d}_{(u,\cdot)} = \frac{1}{|N|}\sum_{v \in N} \bar{d}_{(u,v)}$. The true distributions of statistics T_{R} (R-statistic) and T_{MAX} (MAX-statistic) are very complex, so the distributions of the R-statistic and MAX-statistic are simulated in the MCS test using the bootstrap method. See Ref. [33] for details of the bootstrap steps.

A major advantage of MCS is the ability to quantify the prediction accuracy of the model by calculating the MCS p values of models. The MCS p values of the models are not the same as the p values of the hypothesis testing. The MCS p values of the models are not the result of a probability calculation and do not have probabilistic significance. The process of calculating the MCS p values of models is as follows[33]. Denote $m_{(k)}$ for the model with prediction accuracy ranked k among all models. The ranking order of the prediction accuracy of all models is $m_{(1)} < m_{(2)} < \cdots < m_{(m)}$. Let $Q_k = \{m_{(k)}, m_{(k+1)}, \cdots, m_{(m)}\}$. Thus, $Q_1 = M$ and $Q_m = \{m_{(m)}\}$. Apparently, if H_{0,Q_k} is rejected, $m_{(k)}$ will be removed from Q_k. Let p_k be the p value of H_{0,Q_k} based on the above statistics. Therefore, define the MCS p values of $m_{(g)}$ to be $\max_{k \leqslant g} p_k$. Because Q_m only contains the model with the highest prediction accuracy, we cannot test H_{0,Q_m}. Thus, we define the MCS p values of $m_{(m)}$ to be 1.00.

From the above analysis, it can be seen that the MCS p value of the model with the highest prediction accuracy is always 1.00 and that the larger MCS p values of the model indicate the higher accuracy of the model.

3 Analysis of experiments

3.1 Data introduction

Two representative Chinese stock index data sets, the SSE Composite Index (index code: 000001; the abbreviation for it is SHCI in this article) and the SZSE Composite Index (index code: 399106; the abbreviation for it is SZCI in this article), have been carefully selected as the experimental data

DOI: 10.52396/JUSTC-2023-0066
JUSTC, 2024, 54(2): 0204

for this study. The SHCI data set consists of all stocks listed on the Shanghai Stock Exchange, including A shares and B shares. It effectively captures the price movements of stocks listed on the Shanghai Stock Exchange. The SZCI data set represents a weighted composite stock index compiled by the Shenzhen Stock Exchange. It is calculated based on all stocks listed on the Shenzhen Stock Exchange, with each stock's issue weight taken into account. The time period for the two data sets spans from 2012-06-14 to 2022-08-31. Daily data for the study are obtained from the Wind database.

In this study, the closing price of the stock indexes is chosen as the experimental data. The data are divided into three parts: training data, validation data, and test data, as illustrated in Fig. 4. The training data are utilized to calculate the weights and biases of the models. The validation data are employed to determine the optimal number of CBAMs if necessary. Finally, the trained models are evaluated on the test set. Notably, all models have been trained and converged, ensuring that there is no overfitting issue.

Table 1 summarizes the statistics of the two data sets. Based on the results of the Jarque–Bera (JB) test, the null hypothesis of a normal distribution is rejected at the 5% significance level for both the SHCI and SZCI. Additionally, the results of the Ljung–Box test suggest that the null hypothesis of no autocorrelation up to the 20th order is rejected at the 5% significance level for both data sets, indicating the presence of long-term serial autocorrelation in SHCI and SZCI.

3.2 Preprocessing of data

Max-min normalization can speed up the training process of models and facilitate the convergence of models[14]. Therefore, the data are first normalized. The normalization formula is as follows.

$$y_i \leftarrow \frac{y_i - \min\{y_i\}}{\max\{y_i\} - \min\{y_i\}} \in (0,1). \tag{21}$$

Here, $\leftarrow$ denotes assignment. y_i is the closing price at moment i. The prediction method in the experiments is to use 60 days as a time step to predict the next day and then keep sliding forward. The form of the data is shown in Fig. 5. The expression is shown in Eq. (22).

$$y_{t_{61}} = f(y_{t_1}, y_{t_2}, \cdots, y_{t_{60}}). \tag{22}$$

3.3 Related hyperparameter settings

The experiments were conducted using *Python* 3.8.1 as the programming language and *PyCharm* 2020.1.2 (Community Edition) as the compiler. The *Python* libraries used include numpy 1.23.3, pandas 1.4.4, matplotlib 3.6.1, TensorFlow 2.10.0, keras 2.10.0, sklearn 1.1.3, and arch 5.3.1. To ensure reproducibility of the results, random seeds were set to 12, 1234, and 2345. To focus on the performance of the models rather than the influence of hyperparameters on the prediction results, consistent hyperparameter values were used for different models, as indicated in Table 2. The default values were retained for the remaining hyperparameters.

3.4 The impact of CBAM on BiLSTM

Despite the demonstrated superiority of CBAM in various domains, there is a lack of detailed analysis in existing studies regarding the influence of CBAM's position and quantity on prediction accuracy. To fill this gap, the experiments are divided into two parts. The first part examines the impact of CBAM position on the prediction accuracy of BiLSTM, while the second part investigates the effect of CBAM quantity on the prediction accuracy of BiLSTM. This approach allows for a comprehensive understanding of how the position and amount of CBAM in models can affect prediction accuracy.

3.4.1 The impact of the position of CBAM on BiLSTM

First, the impact of CBAM position on prediction accuracy is

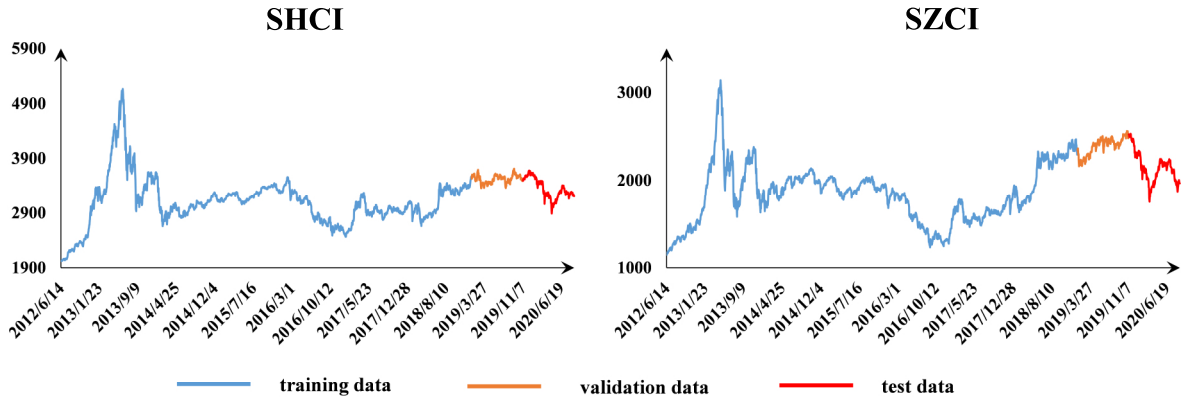


Fig. 4. Two stock indexes.

Table 1. Summary statistics of data sets.

Data set	Count	Mean	Min	Max	Std	JB_pvalues	LB(20)_pvalues
SHCI	2000	3165.98	2023.74	5166.35	425.45	0.0000	0.0000
SZCI	2000	1925.27	1148.29	3140.66	347.39	0.0273	0.0000

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

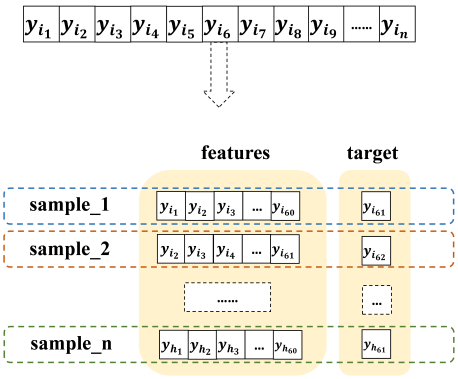


Fig. 5. Data formats.

analyzed. Two modeling schemes are considered: CBAM in front of BiLSTM (CBAM_f_BiLSTM) and CBAM behind BiLSTM (CBAM_b_BiLSTM). Fig. 6 illustrates these two modeling schemes, and Table 3 presents the results on the test sets for both schemes. Furthermore, the robustness analysis of the two modeling schemes on the test sets is shown in Table 4.

The values in Table 4 are explained as follows. The "model 1 / model 2" column represents the percentage optimization of model 1 compared to model 2. The "Mean" column displays the mean optimization percentage for the two modeling schemes across different data sets, while the "Std" column represents the standard deviation of the optimization percentage for the two schemes across different data sets.

To illustrate the calculation process, let us consider the example of the MAE optimization percentage in the mean column (−1.65%). For the SHCI data set, CBAM_f_BiLSTM/BiLSTM = (34.0270 − 35.3833)/34.0270 × 100% = −3.99%, where 34.0270 and 35.3833 are values from Table 3. Similarly, for the SZCI data set, CBAM_f_BiLSTM/BiLSTM = (30.8811 − 30.6652)/30.8811× 100% = 0.70%, where 30.8811 and 30.6652 are values from Table 3. Thus, for MAE, the mean of CBAM_f_BiLSTM/BiLSTM is calculated as (−3.99% + 0.70%)/2 = −1.65%. The standard deviation (Std) of CBAM_f_BiLSTM/BiLSTM is computed as follows:

$$\begin{aligned}\text{Std} = \text{Sqrt}\{&\text{Square}[-3.99\% - (-1.65\%)] + \\ &\text{Square}[0.70\% - (-1.65\%)]\} = 2.3450.\end{aligned}$$

Here, Sqrt{ } represents the arithmetic square root function, and Square[ ] represents the square function.

The optimization percentages for RMSE and MAPE are calculated in the same manner as MAE. However, the optimization percentage for R^2 is calculated in the opposite way to MAE. Let us consider the example of the R^2 optimization percentage in the mean column (0.11%). For the SHCI data set, CBAM_f_BiLSTM/BiLSTM = (0.9404 − 0.9448)/0.9448 × 100% = −0.47%, where 0.9404 and 0.9448 are values from Table 3. Similarly, for the SZCI data set, CBAM_f_BiLSTM/BiLSTM = (0.9466−0.9400)/0.9400×100% = 0.70%, where 0.9466 and 0.9400 are values from Table 3. Therefore, for R^2, the mean of CBAM_f_BiLSTM/BiLSTM is calculated as (−0.47%+0.70%)/2 = 0.11%. The standard deviation is computed as Std = Sqrt{Square[−0.47% − 0.11%] + Square[0.70% − 0.11%]} = 0.5850.

Table 2. Hyperparameters of models.

Model	Filters	Kernel_size	Pool_size	Units	Loss	Optimizer	Batch_size	Epochs
CNN	64	3	2	–	MSE	Adam	128	200
LSTM	–	–	–	64	MSE	Adam	128	200
BiLSTM	–	–	–	64	MSE	Adam	128	200
CNN-LSTM	64	3	2	64	MSE	Adam	128	200
CNN-BiLSTM	64	3	2	64	MSE	Adam	128	200
CBAMs-BiLSTM	–	–	–	64	MSE	Adam	128	200

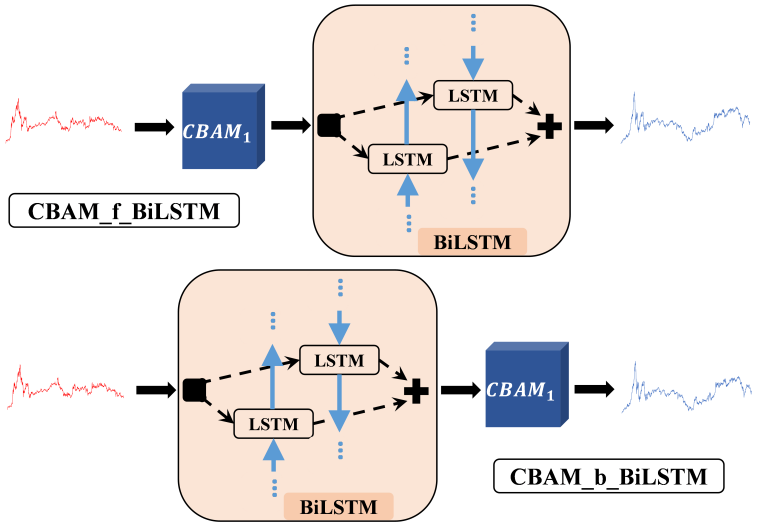


Fig. 6. Two modeling schemes.

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

Table 3. SME of two modeling schemes on test sets.

Model	SHCI				SZCI			
	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
BiLSTM	34.0270	43.0277	0.01038	0.9448	30.8811	38.9196	0.01449	0.9400
CBAM_f_BiLSTM	35.3833	44.0236	0.01077	0.9404	30.6652	37.4941	0.01448	0.9466
CBAM_b_BiLSTM	38.9827	48.7188	0.01190	0.9279	30.1073	37.2790	0.01430	0.9484

The above results are calculated based on the data without normalization. Five decimal places are retained because the MAPE values are too small.

Table 4. Robustness analysis of two modeling schemes on test sets.

Model	Mean				Std			
	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
CBAM_f_BiLSTM/BiLSTM	−1.65%	0.68%	−1.85%	0.11%	2.3450	2.9850	1.9150	0.5850
CBAM_b_BiLSTM/BiLSTM	−6.03%	−4.51%	−6.67%	−0.45%	8.5350	8.7250	7.9750	1.8457

The above results are calculated based on the data without normalization.

Table 4 clearly shows that CBAM_f_BiLSTM outperforms CBAM_b_BiLSTM in all metrics. Additionally, CBAM_f_BiLSTM exhibits smaller standard deviations for each metric. These findings indicate that CBAM_f_BiLSTM not only achieves better prediction accuracy but also demonstrates stronger robustness. Consequently, the modeling scheme with CBAM in front of BiLSTM is considered superior.

3.4.2 The impact of the amount of CBAM on BiLSTM

Although CBAM_f_BiLSTM demonstrates better prediction accuracy and robustness than CBAM_b_BiLSTM, it is slightly less effective than the standalone BiLSTM model. Therefore, the analysis now focuses on increasing the amount of CBAM to examine its impact on prediction accuracy.

Fig. 7 displays the results of models with varying amounts of CBAM on the test sets, ranging from 1 to 15. Table 5 presents the corresponding results, while Table 6 provides the robustness analysis of these models. The symbols and calculation procedure in Table 6 are consistent with those in Table 4.

From Fig. 7, it is evident that when the amount of CBAM is set to 6 or 15, the proposed model performs poorly on SZCI, despite its good performance on SHCI. Table 6 reveals that in such cases, the standard deviation for each metric is higher compared to the other results. Additionally, the mean for each metric is negative. These observations indicate that when the amount of CBAM is 6 or 15, the model not only predicts worse than BiLSTM but also exhibits poor robustness. Based on these experimental findings, it is apparent that the prediction accuracy of CBAM-BiLSTM significantly improves compared to BiLSTM when the amount of CBAM is

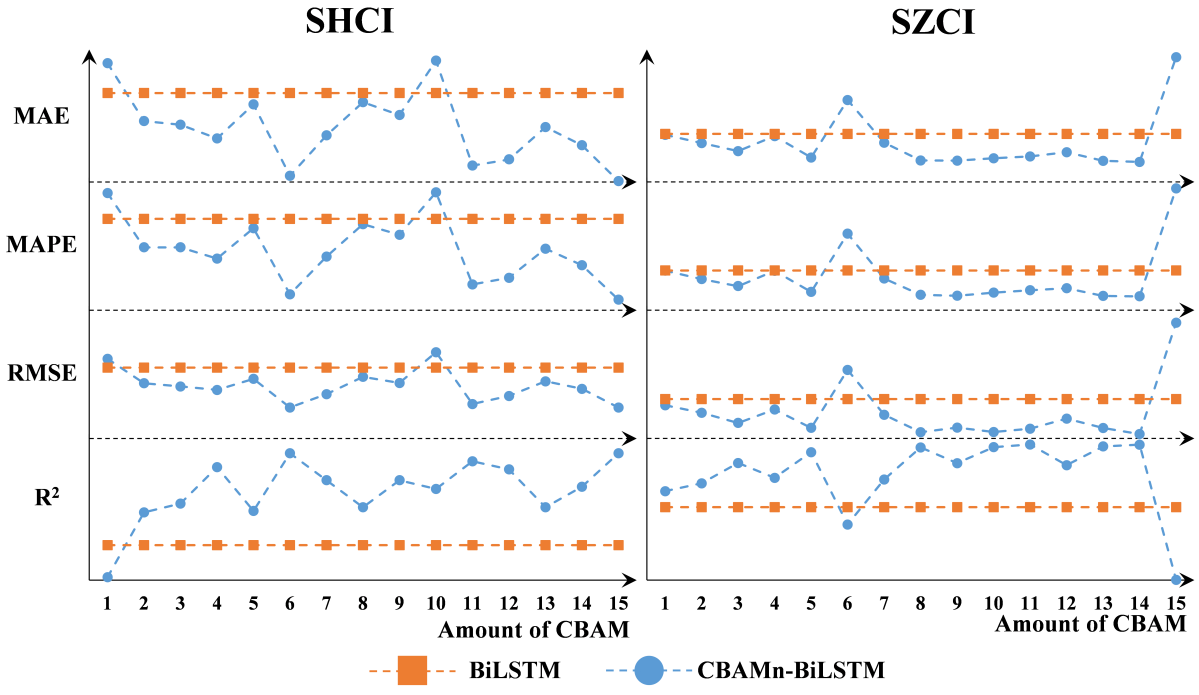


Fig. 7. Results for models with different amounts of CBAM on test sets, which are calculated based on the data without normalization.

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

Table 5. SME of models with different amounts of CBAM on test sets.

Model	SHCI				SZCI			
	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
BiLSTM	34.0270	43.0277	0.01038	0.9448	30.8811	38.9196	0.01449	0.9400
CBAM1-BiLSTM	35.3833	44.0236	0.01077	0.9404	30.6652	37.4941	0.01448	0.9466
CBAM2-BiLSTM	32.7574	41.2633	0.00995	0.9493	28.8200	35.8072	0.01352	0.9498
CBAM3-BiLSTM	32.5954	40.9043	0.00995	0.9505	26.9906	33.5097	0.01272	0.9582
CBAM4-BiLSTM	31.9710	40.5019	0.00978	0.9555	30.3758	36.5677	0.01446	0.9521
CBAM5-BiLSTM	33.5174	41.7608	0.01024	0.9495	25.5348	32.3920	0.01206	0.9626
CBAM6-BiLSTM	30.2781	38.5197	0.00924	0.9574	38.5297	45.5080	0.01867	0.9329
CBAM7-BiLSTM	32.1149	40.0215	0.00981	0.9537	28.8973	35.3460	0.01358	0.9514
CBAM8-BiLSTM	33.6171	41.9892	0.01030	0.9500	24.8818	31.4335	0.01173	0.9646
CBAM9-BiLSTM	33.0362	41.2902	0.01014	0.9537	24.8027	32.4378	0.01163	0.9581
CBAM10-BiLSTM	35.4938	44.7713	0.01078	0.9525	25.3482	31.4534	0.01198	0.9647
CBAM11-BiLSTM	30.7473	38.9105	0.00939	0.9563	25.7932	32.2218	0.01225	0.9658
CBAM12-BiLSTM	31.0215	39.8307	0.00949	0.9552	26.7466	34.5033	0.01249	0.9573
CBAM13-BiLSTM	32.4862	41.4804	0.00993	0.9500	24.7747	32.3736	0.01161	0.9650
CBAM14-BiLSTM	31.6652	40.6338	0.00968	0.9528	24.5299	30.9824	0.01156	0.9657
CBAM15-BiLSTM	30.0401	38.5079	0.00916	0.9574	48.2636	56.2220	0.02378	0.9101

The above results are calculated based on the data without normalization. Five decimal places are retained because the MAPE values are too small.

not equal to 1, 6, 10, or 15.

Furthermore, Table 6 indicates that when the amount of CBAM is 8, 9, 11, 12, 13, or 14, the standard deviation for each metric is smaller, while the mean for each metric is larger. This suggests that the model not only achieves higher prediction accuracy but also maintains good robustness.

To further validate the experimental findings presented in Table 6, the model confidence set (MCS) test is employed to analyze the prediction accuracy and robustness of the models with varying numbers of CBAM. Table 7 displays the MCS p values for the models obtained by summing errors from different test sets. The MCS test in these experiments is implemented using the *Python* library arch 5.3.1, with the random seed set to 12345. It is important to note that the MCS p values are not the result of a probability calculation and do not possess probabilistic significance. Instead, larger values indicate higher prediction accuracy for the corresponding model, with a maximum value of 1.

As observed in Table 7, when the amount of CBAM is set to 11, the MCS p values for the model reach 1.00 for both test statistics, indicating the model's superior prediction accuracy. In line with this, Table 6 demonstrates that compared to BiLSTM, CBAM11-BiLSTM exhibits an average reduction of 13.06%, 13.39%, and 12.48% in MAE, RMSE, and MAPE, respectively, while showcasing an average improvement of 1.98% in R^2.

3.5 Superiority and robustness of CBAM-BiLSTM

The aforementioned experiments provide a comprehensive analysis of the impact of the position and amount of CBAM on the prediction accuracy of BiLSTM. This section further explores the superiority and robustness of CBAM-BiLSTM in terms of prediction accuracy and investment returns.

First, the prediction accuracy of the models is examined using different prediction methods on the test sets. Subsequently, the influence of the training sample size on the prediction accuracy of the proposed model is analyzed. Finally, the experiments delve into the assessment of investment returns generated by the models on the test sets.

Tables 8 and 9 present the outcomes of the SME and MCS p values, respectively, on the test sets when models are based on different prediction methods. Eq. (22) illustrates the expression for autoregressive one-step prediction, while Eq. (23) showcases the expression for autoregressive multistep prediction. Additionally, Eq. (24) demonstrates the expression for multivariate one-step prediction.

$$(y_{t_{61}}, y_{t_{62}}, \cdots, y_{t_{68}}) = f(y_{t_1}, y_{t_2}, \cdots, y_{t_{60}}), \tag{23}$$

$$\begin{aligned} y_{t_{61}} = f\{&(y_{t_1}, \cdots, y_{t_{60}}), (x_{t_1}, \cdots, x_{t_{60}}), (h_{t_1}, \cdots, h_{t_{60}}), \\ &(l_{t_1}, \cdots, l_{t_{60}}), (v_{t_1}, \cdots, v_{t_{60}}), (t_{t_1}, \cdots, t_{t_{60}})\}. \end{aligned} \tag{24}$$

Here, y_i represents the closing price at moment i, x_i refers to the opening price at moment i, h_i denotes the highest price at moment i, l_i represents the lowest price at moment i, v_i signifies the volume at moment i, and t_i represents the turnover at moment i.

From Table 8, it is evident that the proposed model exhibits the minimum error in each prediction method. Moreover, in Table 9, the MCS p values for the proposed model are consistently 1 across all prediction methods. These findings reinforce that, compared to other popular models, the proposed model achieves the highest prediction accuracy across different data sets and prediction methods. Thus, the results in Table 8 and Table 9 validate the superiority and robustness of the proposed model in terms of prediction accuracy.

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

Table 6. Robustness analysis of models with different amounts of CBAM on test sets.

Model	Mean				Std			
	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
CBAM1-BiLSTM/BiLSTM	−1.64%	0.67%	−1.86%	0.12%	3.3130	4.2267	2.7203	0.8284
CBAM2-BiLSTM/BiLSTM	5.20%	6.05%	5.21%	0.76%	2.0813	2.7553	2.0567	0.3995
CBAM3-BiLSTM/BiLSTM	8.40%	9.42%	8.17%	1.27%	5.9335	6.3395	5.6919	0.9355
CBAM4-BiLSTM/BiLSTM	3.84%	5.96%	2.99%	1.21%	3.1154	0.1222	3.9342	0.1100
CBAM5-BiLSTM/BiLSTM	9.41%	9.86%	9.01%	1.46%	11.1829	9.7776	10.9240	1.3442
CBAM6-BiLSTM/BiLSTM	−6.88%	−3.23%	−8.96%	0.29%	25.3039	19.3782	28.1168	1.4780
CBAM7-BiLSTM/BiLSTM	6.02%	8.08%	5.88%	1.08%	0.5689	1.5523	0.6031	0.1906
CBAM8-BiLSTM/BiLSTM	10.32%	10.82%	9.88%	1.59%	12.8854	11.8944	12.9081	1.4632
CBAM9-BiLSTM/BiLSTM	11.30%	10.35%	11.00%	1.43%	11.8594	8.9209	12.3802	0.6940
CBAM10-BiLSTM/BiLSTM	6.80%	7.57%	6.75%	1.73%	15.7174	16.4304	14.9908	1.2796
CBAM11-BiLSTM/BiLSTM	13.06%	13.39%	12.48%	1.98%	4.8348	5.4028	4.1799	1.0833
CBAM12-BiLSTM/BiLSTM	11.11%	9.39%	11.18%	1.47%	3.2215	2.7700	3.7331	0.5238
CBAM13-BiLSTM/BiLSTM	12.15%	10.21%	12.10%	1.61%	10.7805	9.3502	10.9872	1.4890
CBAM14-BiLSTM/BiLSTM	13.75%	12.98%	13.43%	1.79%	9.6350	10.4866	9.5498	1.3349
CBAM15-BiLSTM/BiLSTM	−22.29%	−16.98%	−26.21%	−0.92%	48.0869	38.8633	53.6066	3.1929

The above results are calculated based on the data without normalization.

Table 7. MCS p values on test sets.

Model	R-statistic				MAX-statistic			
	MSE	MAE	HMSE	HMAE	MSE	MAE	HMSE	HMAE
BiLSTM	0.000	0.000	0.000	0.000	0.009	0.009	0.009	0.009
CBAM1-BiLSTM	0.000	0.000	0.000	0.000	0.009	0.009	0.009	0.009
CBAM2-BiLSTM	0.048	0.048	0.048	0.048	0.423	0.423	0.423	0.423
CBAM3-BiLSTM	0.186	0.186	0.186	0.186	0.831	0.831	0.831	0.831
CBAM4-BiLSTM	0.000	0.000	0.000	0.000	0.234	0.234	0.234	0.234
CBAM5-BiLSTM	0.045	0.045	0.045	0.045	0.807	0.807	0.807	0.807
CBAM6-BiLSTM	0.000	0.000	0.000	0.000	0.009	0.009	0.009	0.009
CBAM7-BiLSTM	0.155	0.155	0.155	0.155	0.807	0.807	0.807	0.807
CBAM8-BiLSTM	0.000	0.000	0.000	0.000	0.831	0.831	0.831	0.831
CBAM9-BiLSTM	0.186	0.186	0.186	0.186	0.831	0.831	0.831	0.831
CBAM10-BiLSTM	0.040	0.040	0.040	0.040	0.423	0.423	0.423	0.423
CBAM11-BiLSTM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
CBAM12-BiLSTM	0.186	0.186	0.186	0.186	0.831	0.831	0.831	0.831
CBAM13-BiLSTM	0.155	0.155	0.155	0.155	0.831	0.831	0.831	0.831
CBAM14-BiLSTM	0.385	0.385	0.385	0.385	0.831	0.831	0.831	0.831
CBAM15-BiLSTM	0.000	0.000	0.000	0.000	0.003	0.003	0.003	0.003

The MCS p values of the models are not the result of a probability calculation and do not have probabilistic significance. The above results are calculated based on the data without normalization.

Fig. 8 illustrates the influence of the training set size on the prediction accuracy of CBAM-BiLSTM, with R^2 selected as the metric. It is observed that R^2 remains highly consistent as the size of the training set varies across each data set. This stability reinforces the notion that the proposed model exhibits strong robustness in relation to the size of the training set.

Ideally, a market prediction system can be integrated as a module within a trading system, where improved prediction accuracy is expected to yield higher profits. In this context, we present experiments that utilize the proposed model as the prediction subsystem of a simple trading system. It is important to note that the overall performance of the system depends on how the predictions are utilized for trading. The trading strategy employed in our experiments is as follows: If

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

Table 8. SME for different prediction methods on test sets.

Prediction method	Model	SHCI				SZCI			
		MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
Autoregressive one-step prediction	CNN	37.6814	47.6687	0.01146	0.9283	28.3603	36.9023	0.01345	0.9515
	LSTM	32.0276	42.6244	0.00973	0.9451	33.7055	40.9462	0.01597	0.9362
	BiLSTM	34.0270	42.0277	0.01038	0.9448	30.8811	38.9196	0.01449	0.9400
	CNN-LSTM	34.3729	43.0901	0.01046	0.9428	26.8299	34.0354	0.01266	0.9569
	CNN-BiLSTM	33.6111	42.6511	0.01024	0.9447	26.0393	33.6091	0.01229	0.9586
	CBAMs-BiLSTM	30.0401	38.5079	0.00916	0.9574	24.5299	30.9824	0.01156	0.9657
Autoregressive multistep prediction	CNN	62.6576	75.6104	0.01899	0.8106	57.1727	69.0520	0.02756	0.8369
	LSTM	55.8078	70.1163	0.01694	0.8515	58.3186	70.6678	0.02797	0.8244
	BiLSTM	52.5381	67.0115	0.01590	0.8621	58.2896	70.2769	0.02803	0.8158
	CNN-LSTM	60.7637	72.1551	0.01840	0.8263	55.8181	68.5404	0.02673	0.8487
	CNN-BiLSTM	54.2639	67.6053	0.01648	0.8622	54.5331	66.8516	0.02630	0.8503
	CBAMs-BiLSTM	49.7695	64.2500	0.01496	0.8817	50.6251	62.4469	0.02410	0.8789
Multivariate one-step prediction	CNN	40.6929	56.1362	0.01217	0.9024	36.8902	45.5928	0.01735	0.9256
	LSTM	30.0031	41.7864	0.00903	0.9466	35.3793	42.5291	0.01667	0.9279
	BiLSTM	29.7076	41.8334	0.00892	0.9418	30.2780	38.2994	0.01424	0.9433
	CNN-LSTM	34.0978	44.6628	0.01034	0.9455	34.3397	42.5123	0.01609	0.9256
	CNN-BiLSTM	31.3454	43.5541	0.00944	0.9369	29.7848	37.3321	0.01408	0.9476
	CBAMs-BiLSTM	27.9779	39.0993	0.00843	0.9500	27.6592	35.5245	0.01284	0.9482

The above results are calculated based on the data without normalization. Five decimal places are retained because the MAPE values are too small.

Table 9. MCS p values for different prediction methods on test sets.

Prediction method	Model	R-statistic				MAX-statistic			
		MSE	MAE	HMSE	HMAE	MSE	MAE	HMSE	HMAE
Autoregressive one-step prediction	CNN	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.001
	LSTM	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.001
	BiLSTM	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.001
	CNN-LSTM	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	CNN-BiLSTM	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	CBAMs-BiLSTM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Autoregressive multistep prediction	CNN	0.024	0.024	0.024	0.024	0.313	0.313	0.313	0.313
	LSTM	0.041	0.041	0.041	0.041	0.555	0.555	0.555	0.555
	BiLSTM	0.041	0.041	0.041	0.041	0.555	0.555	0.555	0.555
	CNN-LSTM	0.023	0.023	0.023	0.023	0.555	0.555	0.555	0.555
	CNN-BiLSTM	0.073	0.073	0.073	0.073	0.555	0.555	0.555	0.555
	CBAMs-BiLSTM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Multivariate one-step prediction	CNN	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.001
	LSTM	0.000	0.000	0.000	0.000	0.020	0.020	0.020	0.020
	BiLSTM	0.018	0.018	0.018	0.018	0.110	0.110	0.110	0.110
	CNN-LSTM	0.000	0.000	0.000	0.000	0.020	0.020	0.020	0.020
	CNN-BiLSTM	0.018	0.018	0.018	0.018	0.110	0.110	0.110	0.110
	CBAMs-BiLSTM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

The above results are calculated based on the data without normalization. The MCS p values of the models are not the result of a probability calculation and do not have probabilistic significance.

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

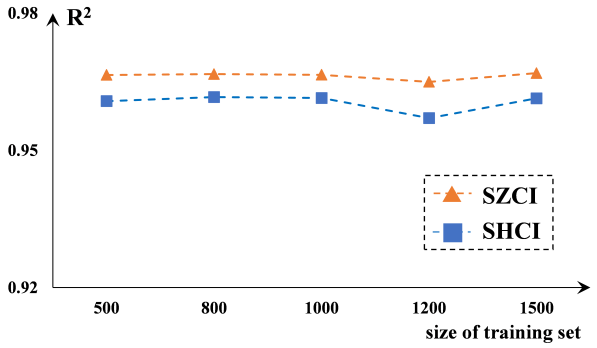


Fig. 8. Impact of the size of the training set on the prediction accuracy of CBAM-BiLSTM.

the predicted price for day $t+1$ is higher than the true price for day t, the predicted label for day t is considered "up"; otherwise, it is considered "down". When the predicted label for the next day is "up", the trading system fully invests in the corresponding index and holds the shares until a "down" label is encountered, at which point the system closes the position.

In this trading strategy, each individual prediction made by the models influences the trading performance and ultimately impacts the overall profit. To assess the performance of the trading system, we utilize the Sharpe ratio. Table 10 presents the Sharpe ratios of different predictive models on the test sets. In the table, R_p represents the return sequences of the models, $E(R_p)$ indicates the mean of the models' return sequences, σ_p represents the standard deviation of the models' return sequences, and "Sharpe" denotes the Sharpe ratio of the models. The numerical values in Table 10 demonstrate that employing the predictions of the proposed model as the foundation for the trading strategy leads to satisfactory results. This further confirms that, in terms of investment returns, the proposed model exhibits superiority and robustness when compared to other popular models.

3.6 Findings

In summary, the experiments conducted a detailed analysis on the impact of the position and number of CBAMs on BiLSTM. The results confirmed that CBAM has the ability to enhance the prediction accuracy of BiLSTM, and this improvement exhibits good robustness. Furthermore, the proposed model's superiority and robustness in terms of prediction accuracy were confirmed through comparisons with other popular models, as well as by varying the prediction method and data sets. Additionally, the experiments demonstrated the model's robustness in relation to the size of the training set. Finally, the experiments affirmed the model's superiority and robustness in terms of investment returns.

Overall, the experimental findings provide strong evidence supporting the effectiveness and reliability of the proposed model. The results indicate that integrating CBAM into the BiLSTM architecture enhances prediction accuracy and robustness across various scenarios and data sets. These findings contribute to advancing the understanding and applicability of the proposed model in real-world scenarios involving market prediction and trading systems.

4 Conclusions

4.1 Summary

To address the issue of low accuracy in stock index prediction, this paper introduces a novel model called CBAM-BiLSTM, which combines multiple CBAMs with a BiLSTM architecture. The experimental evaluation is conducted using the SSE Composite Index and the SZSE Composite Index as the data sets. The performance of various models is assessed using standard metric evaluation and model confidence set test methods. The final results demonstrate that CBAM-BiLSTM exhibits superior performance and robustness in terms of both prediction accuracy and investment returns. Moreover, the experiments include a comprehensive analysis of the impact of CBAM position and amount on the prediction accuracy of the BiLSTM model.

Overall, this research introduces a novel model that effectively addresses the challenge of accurate stock index prediction. Through rigorous evaluation and analysis, the proposed CBAM-BiLSTM model shows its superiority and robustness compared to other models. The findings provide valuable insights into improving prediction accuracy and investment returns in the field of stock market analysis.

4.2 Discussion and outlook

The proposed CBAM-BiLSTM model demonstrates its competence in predicting stock price indexes compared to other hybrid predictive models based on machine learning methods from the literature. In this study, our model achieves a minimum MAPE of 0.0084 (0.84%) and a maximum R^2 value of 0.9657. In previous studies, Md et al.[37] achieved an R^2 of

Table 10. Sharpe ratio of models on test sets.

Model	SHCI			SZCI		
	$E(R_p)$	σ_p	Sharpe	$E(R_p)$	σ_p	Sharpe
Buy and hold	−0.00046	0.01117	−0.04091	−0.00125	0.01478	−0.08427
CNN	−0.00025	0.00670	−0.03697	−0.00061	0.01140	−0.05366
LSTM	0.00002	0.00824	0.00293	−0.00001	0.00770	−0.00183
BiLSTM	0.00023	0.00656	0.03466	−0.00070	0.01059	−0.06633
CNN-LSTM	0.00025	0.00580	0.04339	0.00036	0.00947	0.03801
CNN-BiLSTM	0.00027	0.00599	0.04523	−0.00023	0.00971	−0.02401
CBAMs-BiLSTM	0.00032	0.00599	0.05354	0.00036	0.00426	0.08395

Five decimal places were retained due to small values.

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

0.981 for Samsung stock. Maqbool et al.[38] obtained a MAPE of 1.55% for the HDFC bank stock price data set. Gülmez[39] achieved R^2 values ranging from 0.814 to 0.975 for various stock data sets. Cui et al.[40] obtained an MAPE of 0.62% for the SSE Composite Index.

It is acknowledged that the robustness of model forecasting can vary between tranquil and turbulent periods due to the idiosyncratic patterns of the data; however, this issue can be further addressed by considering hyperparameters. For instance, the number of BiLSTM modules and the number of neurons were not extensively explored in this study. Therefore, future research will aim to develop appropriate methods for selecting hyperparameters to further enhance the predictive performance of the proposed model.

Regarding applications, future work will involve incorporating more contributing feature variables to improve the predictive performance of the model. However, it is important to note that the variables used to predict composite stock indexes and individual stock prices differ significantly. Composite stock indexes may require macrolevel variables such as GDP growth rate, inflation rate, interest rate, government fiscal policy, and international trade situation. On the other hand, individual stock prices tend to focus on internal factors of the company and the market's supply and demand relationship. Therefore, future research plans involve utilizing natural language processing techniques to extract variables that impact stock indexes, thus boosting the prediction performance. Furthermore, since Ref. [32] suggests that solely constructing predictive models in terms of improving the precision may not yield good investment returns, future plans meanwhile cover investment returns as the primary aim to enhance the practical utility of the proposed model.

Acknowledgements

The authors thanks to Dr. Peiwan Wang for organizing the ideas in the discussion section of the article.

Conflict of interest

The authors declare that they have no conflict of interest.

Biographies

Chenhao Cui received his master's degree from the University of Science and Technology of China in 2023. His research mainly focuses on the application of deep learning in time series prediction.

Yong Li is an Associate Professor at the University of Science and Technology of China (USTC). He received his Ph.D. degree from USTC in 2012. His research mainly focuses on FinTech and data mining.

References

[1] Liu H, Long Z. An improved deep learning model for predicting stock market price time series. *Digital Signal Processing,* **2020**, *102*: 102741.

[2] Mokni K. A dynamic quantile regression model for the relationship between oil price and stock markets in oil-importing and oil-exporting countries. *Energy,* **2020**, *213*: 118639.

[3] Wang L, Ma F, Liu J, et al. Forecasting stock index volatility: New evidence from the GARCH-MIDAS model. *International Journal of Forecasting,* **2020**, *36* (2): 684–694.

[4] Olaniyi S A S, Adewole K S, Jimoh R G. Stock trend prediction using regression analysis: A data mining approach. *ARPN Journal of Systems and Software,* **2011**, *1* (4): 154–157.

[5] Franses P H, Ghijsels H. Additive outliers, GARCH and forecasting volatility. *International Journal of Forecasting,* **1999**, *15* (1): 1–9.

[6] Mondal P, Shift L, Goswami S. Study of effectiveness of time series modeling (ARIMA) in forecasting stock indexs. *International Journal of Computer Science, Engineering and Applications,* **2014**, *4* (2): 13–29.

[7] Challa M L, Malepati V, Kolusu S N R. S&P BSE Sensex and S&P BSE IT return forecasting using ARIMA. *Financial Innovation,* **2020**, *6*: 47.

[8] Sarantis N. Nonlinearities, cyclical behavior and predictability in stock markets: International evidence. *International Journal of Forecasting,* **2001**, *17* (3): 459–482.

[9] Long J, Chen Z, He W, et al. An integrated framework of deep learning and knowledge graph for prediction of stock index trend: An application in Chinese stock exchange market. *Applied Soft Computing,* **2020**, *91*: 106205.

[10] Chen Y, Wu J, Wu Z. China's commercial bank stock index prediction using a novel K-means-LSTM hybrid approach. *Expert Systems with Applications,* **2022**, *202*: 117370.

[11] Tay F E H, Cao L. Application of support vector machines in financial time series forecasting. *Omega,* **2001**, *29* (4): 309–317.

[12] Bishop C M. Neural networks and their applications. *Review of Scientific Instruments,* **1994**, *65* (6): 1803–1832.

[13] Yu Z, Qin L, Chen Y, et al. Stock index forecasting based on LLE-BP neural network model. *Physica A: Statistical Mechanics and Its Applications,* **2020**, *553*: 124197.

[14] Liang Y, Lin Y, Lu Q. Forecasting gold price using a novel hybrid model with ICEEMDAN and LSTM-CNN-CBAM. *Expert Systems with Applications,* **2022**, *206*: 117847.

[15] Cao J, Wang J. Stock index forecasting model based on modified convolution neural network and financial time series analysis. *International Journal of Communication Systems,* **2019**, *32* (12): e3987.

[16] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena,* **2020**, *404*: 132306.

[17] Yu Y, Si X, Hu C, et al. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation,* **2019**, *31* (7): 1235–1270.

[18] Xu G, Meng Y, Qiu X, et al. Sentiment analysis of comment texts based on BiLSTM. *IEEE Access,* **2019**, *7*: 51522–51532.

[19] Siami-Namini S, Tavakoli N, Namin A S. The performance of LSTM and BiLSTM in forecasting time series. In: 2019 IEEE International Conference on Big Data (Big Data). Los Angeles, USA: IEEE, **2019**: 3285–3292.

[20] Pirani M, Thakkar P, Jivrani P, et al. A comparative analysis of ARIMA, GRU, LSTM and BiLSTM on financial time series forecasting. In: 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). Ballari, India: IEEE, **2022**: 1–6.

[21] Lu W, Li J, Wang J, et al. A CNN-BiLSTM-AM method for stock index prediction. *Neural Computing and Applications,* **2021**, *33* (10): 4741–4753.

[22] Guo Y, Mao J, Zhao M. Rolling bearing fault diagnosis method based on attention CNN and BiLSTM network. *Neural Processing Letters,* **2022**, *55*: 3377–3410.

[23] Cheng W, Wang Y, Peng Z, et al. High-efficiency chaotic time series prediction based on time convolution neural network. *Chaos, Solitons & Fractals,* **2021**, *152*: 111304.

[24] Li J, Liu Y, Li Q. Intelligent fault diagnosis of rolling bearings under imbalanced data conditions using attention-based deep learning method. *Measurement,* **2022**, *189*: 110500.

[25] Song S, Yang Z, Goh H H, et al. A novel sky image-based solar irradiance nowcasting model with convolutional block attention mechanism. *Energy Reports,* **2022**, *8*: 125–132.

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204

[26] Li D, Liu J, Zhao Y. Prediction of multi-site $PM_{2.5}$ concentrations in Beijing using CNN-Bi LSTM with CBAM. *Atmosphere,* **2022**, *13* (10): 1719.
[27] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **2013**, *35* (8): 1798–1828.
[28] Ismail Fawaz H, Forestier G, Weber J, et al. Deep learning for time series classification: A review. *Data Mining and Knowledge Discovery,* **2019**, *33* (4): 917–963.
[29] Greff K, Srivastava R K, Koutník J, et al. LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems,* **2016**, *28* (10): 2222–2232.
[30] Huang C G, Huang H Z, Li Y F. A bidirectional LSTM prognostics method under multiple operational conditions. *IEEE Transactions on Industrial Electronics,* **2019**, *66* (11): 8792–8802.
[31] Woo S, Park J, Lee J Y, et al. CBAM: Convolutional block attention module. In: Computer Vision – ECCV 2018. Cham, Switzerland: Springer, **2018**: 3–19.
[32] Dessain J. Machine learning models predicting returns: Why most popular performance metrics are misleading and proposal for an efficient metric. *Expert Systems with Applications,* **2022**, *199*: 116970.
[33] Hansen P R, Lunde A, Nason J M. The model confidence set. *Econometrica,* **2011**, *79* (2): 453–497.
[34] Masini R P, Medeiros M C, Mendes E F. Machine learning advances for time series forecasting. *Journal of Economic Surveys,* **2023**, *37* (1): 76–111.
[35] Liang C, Umar M, Ma F, et al. Climate policy uncertainty and world renewable energy index volatility forecasting. *Technological Forecasting and Social Change,* **2022**, *182*: 121810.
[36] Vidal A, Kristjanpoller W. Gold volatility prediction using a CNN-LSTM approach. *Expert Systems with Applications,* **2020**, *157*: 113481.
[37] Md A Q, Kapoor S, AV C J, et al. Novel optimization approach for stock price forecasting using multilayered sequential LSTM. *Applied Soft Computing,* **2023**, *134*: 109830.
[38] Maqbool J, Aggarwal P, Kaur R, et al. Stock prediction by integrating sentiment scores of financial news and MLP-regressor: A machine learning approach. *Procedia Computer Science,* **2023**, *218*: 1067–1078.
[39] Gülmez B. Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Systems with Applications,* **2023**, *227* (C): 120346.
[40] Cui X, Shang W, Jiang F, et al. Stock index forecasting by hidden Markov models with trends recognition. In: 2019 IEEE International Conference on Big Data (Big Data). Los Angeles, USA: IEEE, **2019**: 5292–5297.

DOI: 10.52396/JUSTC-2023-0066
JUSTC, **2024**, 54(2): 0204