



# Fusion of Multiscale Convolution and LSTM for Stock Price Prediction

Hui Sheng<sup>1</sup>, Jiyong Hu<sup>1</sup>, Menglin Wang<sup>2</sup>, Min Liu<sup>1</sup>, Haoming Zhang<sup>1</sup>,  
and Longjun Huang<sup>1</sup>(✉)

<sup>1</sup> School of Software, Jiangxi Normal University, Nanchang, China  
{202140100843, liumin1, 202140100838}@jxnu.edu.cn, 63752738@qq.com

<sup>2</sup> Jiangxi Technology Business Polytechnic, Nanchang, China

**Abstract.** The stock market has long been a topic of great interest in the financial sector. The job of predicting stock prices has also proven challenging. It is challenging to capture such intricate correlations in time-series data because traditional approaches usually assume constant data, which does not account for the time-varying, dynamic, and extremely noisy nature of time-series data. Data obtained from the stock market is fundamentally a multi-scale, non-smooth, non-linear time series. To address the aforementioned issues, a stock prediction model (MCA\_LSTMNet) that combines multi-scale convolutional attention (MCA) and a Long Short-Term Memory network (LSTM) is proposed, which is able to better capture the long-term dependence and complicated feature connections of stock data. The daily closing price is used as the target sequence, and the remaining characteristics are used as the exogenous sequence in the model's first division of the many features of the stock time series into two sequences. The encoder then uses time convolution at various scales to extract the feature from the exogenous sequence's many time scales. The hidden layer state and multi-scale features of the decoder are then weighted and fused using the attention mechanism (AM) to produce the context vector at the corresponding moment, combine it with the value of the target sequence at each moment as the decoder's input, and finally acquire the prediction outcome. The MCA\_LSTMNet model is tested on three datasets in this study: SSE50, SZSE Component Index, and CSI300. The experimental findings demonstrate that the MCA\_LSTMNet model performs better in terms of prediction and generalization than other benchmark approaches.

**Keywords:** Multi-scale convolution · Attention mechanism · LSTM · Stock price forecasting

## 1 Introduction

China's financial market system is continuing to develop and improve as a result of economic globalization. The stock market, which is a fundamental element of financial growth, on the one hand offers businesses long-term, dependable, and affordable

H. Sheng and J. Hu—are Equal contribution.

funding, enabling them to grow their size, enhance their market share, and encourage industrial upgrading. The market's impression of a company's operational environment and potential development tendencies, on the other hand, might be reflected in stock prices and trading information on the stock exchange. Stocks enable investors to make better informed investment decisions since they allow them to participate in the gains brought about by the expansion of businesses [1].

Due to the exponential rise of the Internet, the stock market has become more transparent and liquid, which may also help to strengthen corporate governance by standardizing and institutionalizing company management [2]. Consequently, it is now practically necessary to accurately anticipate future stock movements, which has led to substantial research by academics both domestically and internationally. Different stock market forecasting techniques have been developed over time using historical stock time series data as samples. These techniques include the autoregressive moving average model (ARIMA), autoregressive conditional heteroskedasticity model, stochastic volatility model, and improved stock market forecasting models based on widely used mathematical models. These models, to some extent, reveal the characteristics of the stock market, its direction, and its opportunistic behavior [3–5]. Nevertheless, the aforementioned models explain and evaluate the correlation between variables by supposing that they have a linear connection with one another and applying statistical inference techniques. They disregard the numerous non-linear components and their intricate interrelationships as a result [6]. However, since stock market-derived data is essentially a nonlinear, nonsmooth, multiscale interval time series, conventional approaches based on constant data assumptions find it challenging to capture such complex relationships in time series data and also ignore the highly noisy, time-varying, and dynamic nature of stock time series data [7].

Deep learning, a very complicated nonlinear system, has seen previously unheard-of performance gains in feature representation learning in the recent past because of the dramatically increased quantity of data and advancements in computer technology [8]. As a consequence, research researchers have successfully applied a variety of neural network models to forecast the rising and downward trends of stock prices [9–11]. For instance, Sarode et al. [12] proposal to employ machine learning (ML) techniques for stock price prediction assessed the efficacy of different ML algorithms, such as Support Vector Machines (SVM), Random Forest, and Neural Networks. Jin et al. [13] improved the accuracy of stock price prediction by incorporating investors' emotional tendencies into stock prediction while using Empirical Mode Decomposition (EMD) for stepwise decomposition of the sequence and further correcting the LSTM by AM. In order to forecast stock price, ingeniously merging technical and financial market aspects, Qi et al.'s [14] Temporal Convolutional Attention-based Network (TCAN) forecasts the Chinese stock market.

Due to the limitation of the size of the memory cells and the influence of the change of memory information in the short term of updating, however, it is difficult for the existing temporal data prediction models to effectively preserve the long-term dependence between time steps. According to the multivariate prediction of univariate stock price prediction model, all feature variables pass through the nodes of each hidden layer with equal weights within each time cell. As a result, the model is unable to determine

how the information about the target sequence to be predicted is distributed among the nodes of various hidden layers. To tackle this problem, this paper proposes a stock prediction model MCA\_LSTMNet that fuses MCA and LSTM, which employs an encoder and decoder structure. Firstly, the stock time series with multivariate characteristics are split, and the daily closing price is utilized as the target series for forecasting, while other variable characteristics are categorized as exogenous series, such as opening price, minimum price, maximum price, and change amount. The contextual feature vector for that instant is then created by fusing it with the hidden state of the closing price sequence from the previous moment and combining it with it. In order to complete the prediction result, it is combined with the target sequence as input to the decoder and output. This approach takes into account the influence of exogenous series on the change of the target series at each instant in addition to capturing the dependencies of the target series over a lengthy period of time. When compared to the benchmark approaches currently in use, the method suggested in this research offers enhanced prediction accuracy for stock price prediction.

Our main contributions in this article are as follows:

1. We propose a new framework MCA\_LSTMNet to better capture data features and extract information for stock-specific attributes.
2. We split the stock data of multi-variable features and develop a new attention mechanism to adaptively extract the relevant driving series at each time step by referring to the previous encoder hidden state.
3. The effectiveness of the proposed method is verified on three real stock data sets.

## 2 Related Works

### 2.1 Stock Forecasting Based on Classical Methods

The most popular model among the classic time-series forecasting techniques, and one that is frequently used in forecasting in the economics sector, is ARIMA. For instance, Khanderwal et al. [15] used ARIMA model to make short-term forecasts for three industries selected from NSE to guide stock market investors to make profitable investment choices. The wavelet transform was utilized by Conejo et al. [16] to break down the often historically unstable price series into a collection of more stable component series. Then, a well fitted ARIMA model is used to forecast the future values of these constituent series. By recreating the future behavior of the price series using the inverse wavelet transform, ARIMA forecasting may then anticipate prices. The efficacy of standard time series analysis-based approaches for nonlinear, nonsmooth time series is constrained since existing methods focus on anticipating smooth stock time series data.

### 2.2 Deep Learning Based Stock Prediction

The expressive capability of the convolutional kernel is used by convolutional neural networks (CNN) to mine the nonlinear aspects of historical temporal data, which helps to boost prediction accuracy. Cao et al. [17] combined CNN and SVM to build stock prediction model. They not only analyzed in how the model parameters affect the prediction results, but also analyzed the market rules to predict the trend of future financial

activities. Wu et al. [18] proposed a convolutional neural network based on an array of stock sequences (SSACNN) in order to do feature extraction of financial sequences and prediction of price trends, and the experiments proved that the method can increase the accuracy of predictions.

Choi et al. [19] passed the residual values of stock time series data with linear trends removed using ARIMA model to LSTM model, and their experimental results proved that the ARIMA-LSTM hybrid model can considerably increase the model's predictive power in comparison to more conventional techniques. In addition, researchers and scholars have combined CNN with LSTM models to obtain information about different dimensional features and market reviews by overlaying multiple filters [20–22]. Liu et al. [23] coupled LSTM and Convolutional Autoencoder (CAE) to extract characteristics from stock price data. Compared to a single model that only accepted acceptable inputs from technical indicator data, this model's predictive power is enhanced. Liwei et al. [24] first pre-trained the data using XGBoost model and optimized the parameters using Bayesian algorithm, and then analyzed and predicted the results using LSTM model. Kanwal et al. [25] proposed the BiCuDNNLSTM model which forecasts more accurate stock price by combining the bi-directional CudaLSTM and 1D convolution.

In order to anticipate stock opening prices, Qiu et al. [26] analyzed stock data using the wavelet transform and an attention-based LSTM neural network. Lin et al. [27] combined the AM and the LSTM model to propose a new AT-LSTM model with four stock market indices with six features as inputs, and the results show that the prediction results of this method outperforms other methods in terms of stability. Lu et al. [28] proposed a CNN-BiLSTM-AM method, which uses CNN to perform feature extraction on the input data, and BiLSTM will employ the taken characteristics to forecast the stock's closing price in the future moments, and uses the AM to capture the effect of feature states on different moments to improve the prediction accuracy.

### 3 Methodology

#### 3.1 Problem Definition

Given an exogenous sequence  $X$ , where  $T$  is the time step and  $n$  is the number of variables,  $X = (x^1, x^2, x^3, \dots, x^n)^T = (x_1, x_2, x_3, \dots, x_T) \in R^{n \times T}$ . The model aims to pass the past values of the target series  $(y_1, y_2, \dots, y_T)$ , and the current and past values of the extraneous series  $(x_1, x_2, \dots, x_T)$ , a nonlinear mapping to a target sequence of current values is learned, as shown in Eq. 1.

$$\hat{y}_{T+1} = F(y_1, y_2, \dots, y_T; x_1, x_2, \dots, x_T) \quad (1)$$

where  $F(\cdot)$  is the target mapping function for learning.

When decoding the target sequence, the residual term of the target value at moment  $T + 1$  is preset to be the mean value of the previous  $T$  moments, and the input at each moment is to be subtracted from the mean value of the target sequence. The last output of the decoder plus the predetermined residual term constitutes the ultimate prediction outcome.

### 3.2 Model Architecture Overall Framework

This study suggests a novel stock price prediction model, MCA\_LSTMNet, to incorporate the impacts of exogenous sequences on the target sequence at various points and, at the same time, prevent the degradation problem resulting from the model’s depth. The multiscale convolution module is used in the encoder stage to acquire characteristics from the exogenous input at several temporal scales in order to provide a higher quality feature representation. Context vectors are the features produced by executing a weighted fusion operation of the output of the encoder with the hidden state of the decoder at the previous instant through the attention layer. In the decoder step, the context vector is mixed with the target sequence’s historical values as the input, and a set of LSTM units may better learn the present semantic information to better capture the input sequence’s long-term dependencies. Figure 1 depicts the specific model structure.

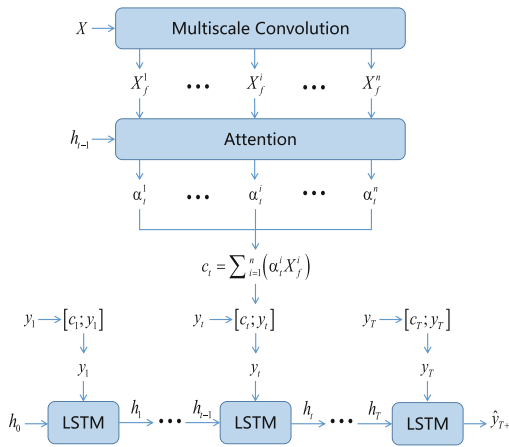


Fig. 1. MCA\_LSTMNet model structure diagram

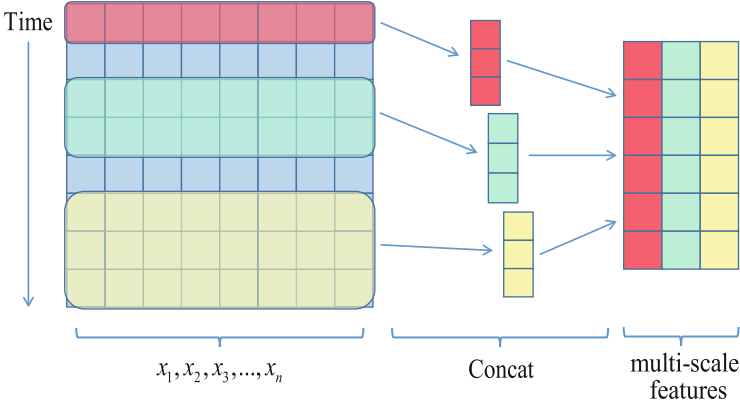
#### Encoder

Multiple time frames of change are represented in stock data, including both short-term and long-term trends. Therefore, feature extraction by multi-scale convolution may gather data on several time scales at once in order to more comprehensively reflect the dynamic changes in stock prices. In Fig. 2, the structure is displayed. The variables of each exogenous sequence affect the variation of the target sequence and vary with the value of the target variable at different times, even though the exogenous sequences are not the predictive task of the model and the interactions between the exogenous features are not required to be taken into account. As a result, by using several 1D convolutions with varying sized convolution kernels, this work extracts increasingly complex time-dimensional characteristics from exogenous sequences and merges features of various scales as the encoder’s output. As demonstrated in Eqs. 2 and 3:

$$f_{i,t}^d = Conv \left( \sum_{k=1}^d (W_{i,k-1}, x_{t+k-1}) + b_i^d \right) \quad (2)$$

$$X_f = \text{Concat}(f^{d1}, f^{d2}, f^{d3}, \dots)^T \tag{3}$$

where  $f_{i,t}^l$  indicates the eigenvalue of the  $i$ th convolution kernel with step  $d$  of the exogenous sequence, and  $X_f$  enotes the feature after fusing the multi-scale information.



**Fig. 2.** Multi-scale convolutional structure

**LSTM**

A recurrent neural network (RNN) model that is frequently applied to analyze sequence data is called LSTM [35]. LSTM models are able to better capture and simulate long-term relationships because they offer a gating mechanism that addresses the issue of gradient vanishing during training of lengthy sequences, as opposed to typical RNN models. The core component of LSTM is the memory unit, which comprises in a forgetting gate  $f_t$ , an input gate  $i_t$ , and an output gate  $o_t$ . The memory unit’s retention and forgetting of information are governed by the forget gate, its rate of adding new information is governed by the input gate, and its selective output from the hidden state is governed by the output gate. Figure 3 depicts the LSTM unit’s structural diagram, while formulae 4 to 9 outline its operation:

$$f_t = \sigma(W_f h_{t-1} + W_f x_t + b_f) \tag{4}$$

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

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

$$w_t = \tanh(W_w h_{t-1} + W_w x_t + b_w) \tag{7}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes w_t \tag{8}$$

$$h_t = o_t \otimes \tanh(C_t) \tag{9}$$

where  $x_t$  denotes the input at the present time,  $h_{t-1}$  denotes the concealed condition at the preceding time,  $W_f, W_i, W_o, W_w$  and  $b_f, b_i, b_o, b_w$  are the parameters to be learned.  $\sigma$  and  $\otimes$  represent the sigmoid function and element-wise multiplication, respectively.

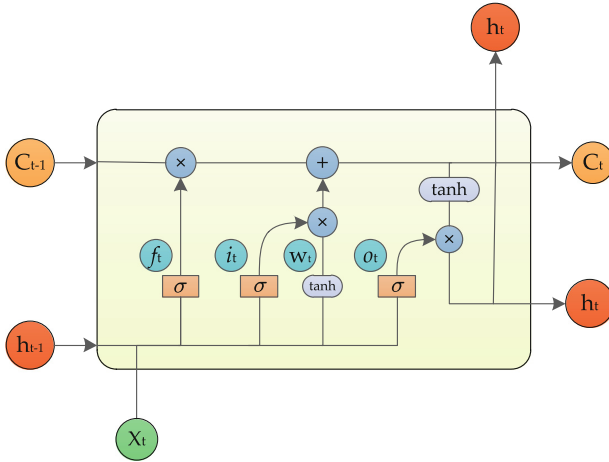


Fig. 3. LSTM structure

**Attention Mechanism**

AM is often used in deep learning for a range of tasks, including computer vision’s image classification, target detection, and picture synthesis as well as natural language processing’s machine translation and text summarization [29]. By automatically learning significant characteristics or correlations in the data, it helps the model to concentrate more on task-relevant components and less on extraneous information. Considering that each external characteristic has distinct semantic data and affects the target sequence in various ways at various times. By using the AM to adaptively alter the attention to information on multiple time scales and learn the most pertinent features to the expected objective, it is possible to weight and integrate these external impacts of various dimensions. It is vital to rely on the hidden state knowledge of the decoder at the preceding instant for computing the weights. The fused features are the context vectors, denoted by and computed as shown in Eqs. 10, 11, 12:

$$e_t^i = v_e^T \tanh\left(W_e[h_{t-1}; s_{t-1}] + U_e X_f^i\right) \tag{10}$$

$$\alpha_t^i = \frac{\exp(e_t^i)}{\sum_{k=1}^n \exp(e_t^k)} \tag{11}$$

$$c_t = \sum_{i=1}^n (\alpha_t^i X_f^i) \tag{12}$$

where  $e_t^i$  denotes the score of the  $i$ th exogenous feature at time  $t$ ,  $\alpha_t^i$  is the weight coefficient of the  $i$ th exogenous feature at time  $t$ , and  $c_t$  is the context vector computed from the sequence of exogenous features at time  $t$ .  $v_e, W_e, U_e$  are parameters to be learned

The  $\tilde{y}_t$  obtained by combining the context vector  $c_t$  and the target sequence  $y_t$  can be utilized to modify the decoder’s hidden state at time  $t$ , as shown in Eq. 13.

$$\tilde{y}_t = \tilde{W}^T [y_t; c_t] + \tilde{b} \tag{13}$$

**Decoder**

This part is mainly composed of LSTM units that take as input the  $\tilde{y}_t$  obtained by combining the context vector  $c_t$  with the target sequence  $y_t$ . Therefore, the target variable  $\hat{y}_{T+1}$  to be predicted is obtained by computing the formula shown in 14:

$$\begin{aligned} \hat{y}_{T+1} &= F(y_1, y_2, \dots, y_T; x_1, x_2, \dots, x_T) \\ &= v_y^T (W_y [h_T; \tilde{y}_T] + b_y) + b_v \end{aligned} \tag{14}$$

where  $[h_T; \tilde{y}_T]$  indicates the decoder’s hidden state and the cascade of decoder inputs,  $W_y$  and  $b_y$  map the cascade to the size of the hidden states in the decoder.  $v_y$  represents the weights and  $b_v$  represents the deviations.

**4 Experiments**

**4.1 Description of the Dataset**

The three stocks used in this paper’s experiment are SSE 50 (stock code: 000016), SZSE Component Index (stock code: 399001), and CSI 300 (stock code: 399300), and their data were obtained from NetEase Finance. Each stock contains seven variables, of which the high price, low price, opening price, previous closing price, change amount, and fluctuation rafngear exogenous series variables, the closing price is the target series variable, and the data are in days as time interval. Table 1 gives the starting and ending times and the total amount of data for the three stocks.

**Table 1.** Data start and end times and total data for the three stocks

Stock Name	Beginning and Ending Time	Total volume of data
SSE 50	2004.01.02–2021.06.23	4245
SZSE Component Index	1991.04.04–2021.06.23	7349
CSI 300	2002.01.07–2021.03.07	4657

Initially, the stock data is min-max normalized (MinMaxScaler), which maps the different variables to the interval [0, 1], as shown in Eq. 15:

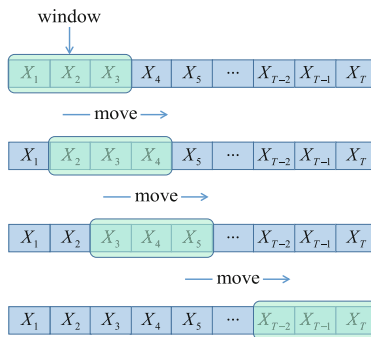
$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{15}$$

In addition to lowering feature scale variation that affects model stability, normalization also accelerates gradient descent during backpropagation, which speeds up the trained network's convergence.

The preprocessed data are then divided into three subsets: the train set, validation set, and test set, which are separated into 60%, 20%, and 20% portions, respectively. As shown in Fig. 4, sliding window processing is applied to each subset's data. The window size of the samples is set to  $t$ , and the closing price on day  $t + 1$  is used as the prediction target. Table 2 lists the number of samples for various stocks, with  $t$  set to 10.

**Table 2.** Number of samples for different experimental data

	SSE 50	SZSE Component Index	CSI 300
train	2537	4399	2783
validation	839	1460	922
test	839	1460	922



**Fig. 4.** Data sliding window process

## 4.2 Experimental Setup and Evaluation Indicators

In this study, PyTorch is utilized to create the network model, while Python 3.7 serves as the experimental environment. The random seed is specified as a constant in the experiment to lessen the effect of random initialization on the deep learning model. The Batch size and iterations are set to 128 and 100, respectively, to balance the model's training speed, model performance, and generalization energy. Moreover, as demonstrated in Eq. 16, the learning rate employs a dynamic adjustment method.

$$lr = lr_{start} \times factor^{epoch-1} \quad (16)$$

where,  $factor$  is the change factor of learning rate, which takes the value range of (0,1)

In consideration of the effectiveness of various prediction methods, four different evaluation metrics are employed, including mean square error (*MSE*), root mean square error (*RMSE*), mean absolute error (*MAE*), mean absolute percentage error (*MAPE*). The formulas are shown in 17 to 20:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y - \hat{y})^2 \quad (17)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y - \hat{y})^2} \quad (18)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y - \hat{y}| \quad (19)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y - \hat{y}|}{y} \quad (20)$$

where variables  $\hat{y}$  and  $y$  denote the value predicted by the model and the true stock value, respectively. As the figures for each of the aforementioned four indicators decrease, the forecasting model's accuracy rises.

### 4.3 Experimental Results Analysis

This work chooses a number of time series prediction models to compare with the MCA\_LSTMNet model in order to assess its efficacy, including RNN [30], LSTM [31], TCN [32], CNN-LSTM-ECA [33], and AE-LSTM-ECA [34]. To assure the fairness of the trials, the same number of hidden layer neurons was set for all comparison techniques. The various prediction error metrics for various models on the three datasets are listed in Tables 3 through 5.

**Table 3.** Predictors of SSE 50 on different models

Models	MSE	RMSE	MAE	MAPE
RNN	1756.749	41.914	30.262	1.023
LSTM	1647.593	40.591	29.231	0.993
TCN	1570.607	39.631	28.547	0.969
CNN-LSTM-ECA	1603.747	40.047	29.006	0.984
AE-LSTM-ECA	1559.434	39.490	28.508	0.968
<b>MCA_LSTMNet</b>	<b>1536.708</b>	<b>39.201</b>	<b>28.507</b>	<b>0.967</b>

**Table 4.** Predictors of SZSE Component Index on different models

Models	MSE	RMSE	MAE	MAPE
RNN	44369.441	210.641	139.835	1.284
LSTM	42359.344	205.814	137.105	1.262
TCN	37694.051	194.150	130.374	1.196
CNN-LSTM-ECA	39848.137	199.620	132.680	1.218
AE-LSTM-ECA	37301.305	193.135	128.512	1.180
<b>MCA_LSTMNet</b>	<b>36977.637</b>	<b>192.295</b>	<b>128.296</b>	<b>1.178</b>

**Table 5.** Predictors of CSI 300 on different models

Models	MSE	RMSE	MAE	MAPE
RNN	3154.284	56.163	39.819	0.989
LSTM	2982.426	54.612	38.646	0.966
TCN	2798.167	52.898	37.001	0.927
CNN-LSTM-ECA	2790.289	52.823	37.784	0.947
AE-LSTM-ECA	2719.459	52.148	36.995	0.926
<b>MCA_LSTMNet</b>	<b>2706.281</b>	<b>52.022</b>	<b>36.223</b>	<b>0.923</b>

Tables 3, 4, 5 show that the MCA\_LSTMNet model's indicators are all lower than those of the other comparison models, demonstrating the model's superiority in stock price prediction.

In this study, the prediction results of all models on the test set are recorded and mapped to the original price interval after inverse normalization, in which 200 days of data are randomly picked for curve charting. Figure 5 illustrate the result plots of this methodology, where the predicted values of the closing price are represented in orange and the real values are displayed in blue. The closing price in RMB is the vertical coordinate.

The prediction results of the model proposed in this study in different datasets roughly match their true values, as illustrated in Fig. 5, and the degree of fitting is essentially consistent with the trend of the error indicator, proving that the model has a strong capacity for prediction.

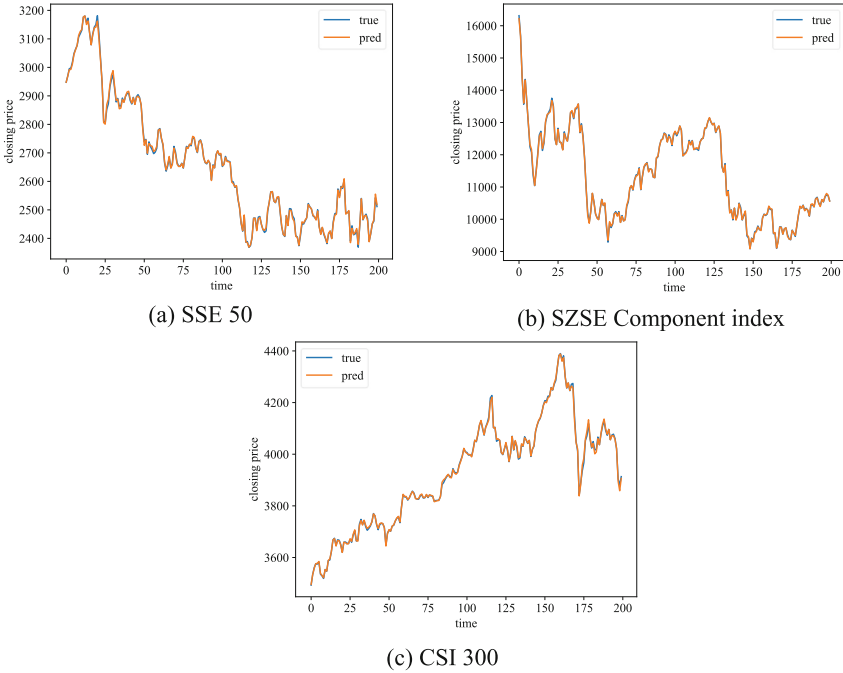


Fig. 5. Prediction results of MCA\_LSTMNet model in three datasets

## 5 Conclusion

Given the limitations of current multivariate prediction univariate methods, this study provides a better framework MCA\_LSTMNet. Firstly, feature information of the exogenous sequences is extracted using convolutional kernels of different sizes. Next, context vectors are obtained by AM-weighted fusion, which are combined with the target sequences as input to the decoder. Finally, the decoder produces the prediction results. Despite the fact that the study's approach makes stock price forecasting more accurate, the stock market's large and complex data involves many aspects, which are usually insufficiently quantitative and difficult to obtain. In the future, collecting more exogenous information to be added to the model will be considered to enhance forecasting precision.

**Acknowledgment.** This work is supported in part by grants from the National Natural Science Foundation of China (No. 62062040), the Outstanding Youth Project of Jiangxi Natural Science Foundation (No. 20212ACB212003), the Jiangxi Province Key Subject Academic and Technical Leader Funding Project (No. 20212BCJ23017), Jiangxi Provincial Department of Education Science and Technology Research Key Projects (No. GJJ2208802).

## References

1. Thakkar, A., Chaudhari, K.: Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions. *Inf. Fusion* **65**, 95–107 (2021)
2. Gandhmal, D.P., Kumar, K.: Systematic analysis and review of stock market prediction techniques. *Comput. Sci. Rev.* **34**, 100190 (2019)
3. Madhuri, C.R., Chinta, M., Kumar, V.P.: Stock market prediction for time-series forecasting using prophet upon arima. In: 2020 7th International Conference on Smart Structures and Systems (ICSSS), pp. 1–5. IEEE (2020)
4. Gao, J.: Research on stock price forecast based on ARIMA-GARCH Model//MSIEID 2022. In: Proceedings of the 4th Management Science Informatization and Economic Innovation Development Conference, MSIEID 2022, December 9–11, 2022, Chongqing, China. European Alliance for Innovation, p. 317 (2023)
5. Wang, C.T., Lin, Y.Y.: The prediction system for data analysis of stock market by using genetic algorithm. In: 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 1721–1725 (2015)
6. Wang, J.Z., Wang, J.J., Zhang, Z.G., et al.: Forecasting stock indices with back propagation neural network. *Expert Syst. Appl.* **38**(11), 14346–14355 (2011)
7. Kumar, G., Jain, S., Singh, U.P.: Stock market forecasting using computational intelligence: a survey. *Arch. Comput. Methods Eng.* **28**, 1069–1101 (2021)
8. Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. *Appl. Soft Comput.* **90**, 106181 (2020)
9. Ding, G., Qin, L.: Study on the prediction of stock price based on the associated network model of LSTM. *Int. J. Mach. Learn. Cybern.* **11**(6), 1307–1317 (2020)
10. Chong, E., Han, C., Park, F.C.: Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies. *Expert Syst. Appl.* **83**, 187–205 (2017)
11. Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **270**, 654–669 (2018)
12. Sarode, S., Tolani, H.G., Kak, P., et al.: Stock price prediction using machine learning techniques. In: 2019 International Conference on Intelligent Sustainable Systems (ICISS), pp. 177–181. IEEE (2019)
13. Jin, Z., Yang, Y., Liu, Y.: Stock closing price prediction based on sentiment analysis and LSTM. *Neural Comput. Appl.* **32**, 9713–9729 (2020)
14. Qi, R., Pan, W.: Stock prediction with adaptive-factors temporal convolutional attention-based network. In: Cao, W., Ozcan, A., Xie, H., Guan, B. (eds.) Computing and Data Science. CONF-CDS 2021. Communications in Computer and Information Science, vol. 1513, pp. 91–104 (2021). Springer, Singapore. [https://doi.org/10.1007/978-981-16-8885-0\\_9](https://doi.org/10.1007/978-981-16-8885-0_9)
15. Khanderwal, S., Mohanty, D.: Stock price prediction using ARIMA model. *Int. J. Mark. Hum. Resour. Res.* **2**(2), 98–107 (2021)
16. Conejo, A.J., Plazas, M.A., Espinola, R., et al.: Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. *IEEE Trans. Power Syst.* **20**(2), 1035–1042 (2005)
17. Cao, J., Wang, J.: Stock price forecasting model based on modified convolution neural network and financial time series analysis. *Int. J. Commun. Syst.* **32**(12), e3987 (2019)
18. Wu, J.M.T, Li, Z., Srivastava, G., et al.: A graph-based convolutional neural network stock price prediction with leading indicators. *Softw. Pract. Exper.* **51**(3), 628–644 (2021)
19. Choi, H.K.: Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model. arXiv preprint [arXiv:1808.01560](https://arxiv.org/abs/1808.01560) (2018)
20. Long, W., Lu, Z., Cui, L.: Deep learning-based feature engineering for stock price movement prediction. *Knowl.-Based Syst.* **164**, 163–173 (2019)

21. Wu, J.M.T., Li, Z., Herencsar, N., et al.: A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Syst.* 1–20 (2021)
22. Shah, J., Vaidya, D., Shah, M.: A comprehensive review on multiple hybrid deep learning approaches for stock prediction. *Intell. Syst. Appl.* 200111 (2022)
23. Liu, H., Qi, L., Sun, M.: Short-term stock price prediction based on CAE-LSTM method. *Wireless Commun. Mob. Comput.* 1–7 (2022)
24. Liwei, T., Li, F., Yu, S., et al.: Forecast of lstm-xgboost in stock price based on bayesian optimization. *Intell. Autom. Soft Comput* **29**(3), 855–868 (2021)
25. Kanwal, A., Lau, M.F., Ng, S.P.H., et al.: BiCuDNNLSTM-1dCNN—a hybrid deep learning-based predictive model for stock price prediction. *Expert Syst. Appl.* **202**, 117123 (2022)
26. Qiu, J., Wang, B., Zhou, C.: Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE* **15**(1), e0227222 (2020)
27. Lin, Y., Huang, Q., Zhong, Q., et al.: A new attention-based LSTM model for closing stock price prediction. *Int. J. Financ. Eng.* **9**(03), 2250014 (2022)
28. Lu, W., Li, J., Wang, J., et al.: A CNN-BiLSTM-AM method for stock price prediction. *Neural Comput. Appl.* **33**, 4741–4753 (2021)
29. Guo, M.H., Xu, T.X., Liu, J.J., et al.: Attention mechanisms in computer vision: a survey. *Computat. Vis. Media* **8**(3), 331–368 (2022)
30. Chen, W., Yeo, C.K., Lau, C.T., et al.: Leveraging social media news to predict stock index movement using RNN-boost. *Data Knowl. Eng.* **118**, 14–24 (2018)
31. Rokhsatyazdi, E., Rahnamayan, S., Amirinia, H., et al.: Optimizing LSTM based network for forecasting stock market. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–7. IEEE (2020)
32. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint [arXiv:1803.01271](https://arxiv.org/abs/1803.01271)*, (2018)
33. Chen, Y., Fang, R., Liang, T., et al.: Stock price forecast based on CNN-BiLSTM-ECA Model. *Sci. Program.* **2021**, 1–20 (2021)
34. Liu, M., Sheng, H., Zhang, N., et al.: A new deep network model for stock price prediction. In: *International Conference on Machine Learning for Cyber Security*, pp. 413–426 (2022)
35. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)