Optimizing Convolutional Neural Network-Long Short-Term Memory Architecture with Additive Attention Mechanism for Stock Price Prediction

Binghui Wu*, Yaxin Qi and Guanhao Fu

International Business School, Shaanxi Normal University, Xi'an, 710119, China

INFORMATION

Keywords:
Stock price forecast
CNN-LSTM
additive attention mechanism
comparative analysis

DOI: 10.23967/j.rimni.2024.10.56790

SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

RIMNI Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería

Optimizing Convolutional Neural Network-Long Short-Term Memory Architecture with Additive Attention Mechanism for Stock Price Prediction

Binghui Wu*, Yaxin Qi and Guanhao Fu

International Business School, Shaanxi Normal University, Xi'an, 710119, China

ABSTRACT

Stock price fluctuations reflect market expectations for the economic situation and company profits. Accurately predicting stock prices has become a hot topic in academia. With the rapid development of artificial intelligence, many researchers are starting to use machine learning algorithms to predict stock prices. In this paper, a new time series prediction model, the combination of the convolutional neural network and long short-term memory neural network with additive attention mechanism (CNN-LSTM-AAM), is proposed for stock price prediction. It can combine the advantages of the convolutional neural network (CNN), the long short-term memory (LSTM) neural network, and the additive attention mechanism (AAM), and better capture nonlinear features of time series data. In the simulation analysis, we select sample data of three stocks (Vanke A, Shanghai International Port Group, and China Merchants Bank) and three stock price indexes (China Securities 500 Index, Shanghai Stock Exchange 50 Index, and Growth Enterprise Index) in the Chinese stock market for comparative analysis, and use mean square error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) as the evaluation indexes. The CNN-LSTM-AAM model has the best prediction ability relative to the CNN and CNN-LSTM models. In addition, we also find that the prediction ability of the CNN-LSTM-AAM model for different stock data sets is different under the existing parameter conditions. For a specific dataset, the parameter conditions of the CNN-LSTM-AAM model need to be further adjusted to achieve the best prediction effect. Based on the above findings, the CNN-LSTM-AAM model has better performance and higher accuracy, and can provide credible decision-making basis and research methods for investors, financial institutions, and regulators.

OPEN ACCESS

Received: 31/07/2024

Accepted: 29/10/2024

Published: 07/04/2025

DOI
10.23967/j.rimni.2024.10.56790

Keywords:
Stock price forecast
CNN-LSTM
additive attention mechanism
comparative analysis

1 Introduction

For a long time, the stock market has been an important place for investment and financing, and the volatility of stock prices affects the income of investors and financial stability. In the existing research on stock price prediction methods, the common analysis methods are mainly econometric

*Correspondence: Binghui Wu (binghui.wu@snnu.edu.cn).

analysis and machine learning. Although econometric analysis is objective and supported by economic theories and statistical methods, traditional econometric models based on many strict assumptions, only extract linear features from data, and are often limited to the prediction of a single variable [1]. Since stock price samples generally have large amounts of data and nonlinear characteristics, traditional econometric methods cannot achieve good forecasting results [2]. Recently, machine learning has been widely used to process complex datasets and solve non-linear problems. Relative to econometric analysis, machine learning doesn't require many assumptions and shows tremendous advantages in extracting data features, so it is suitable for dealing with nonlinear and non-stationary data [3]. In addition, the non-parametric and nonlinear properties of deep learning can provide great help in fitting the complex, dynamic, and uncertain stock market [4].

Recurrent Neural Network (RNN) has remarkable advantages in mining data information, and is extensively used in computer vision, computational biology, and stock price prediction [5]. Due to the existence of long-term dependence problems, the phenomena of vanishing and exploding gradients probably occur in the ordinary RNN [6,7]. To get rid of this dilemma, a special RNN called LSTM is designed [8]. Based on recurrent networks and gradient-based learning algorithms, LSTM generates paths that enable gradients to flow by introducing self-circulation [9]. After predicting the returns of the S&P 500 index, Brazil-Bovespa50 index, and OMX Stockholm 30 index, the LSTM model performs best relative to ARMA and GARCH models [10]. When the sample data is the Dow Jones index, the LSTM model still has the best performance, the ARIMA model performs poorly in the long term, and the GARCH model performs poorly in both the long and short term [11]. Besides, CNN is a representative algorithm, which is widely used in image recognition, voice recognition, and natural language processing [12–15]. As CNN and LSTM have obvious advantages in extracting data characteristics and dealing with long-term dependencies of data, respectively, some scholars combine CNN with LSTM for prediction analysis [16,17]. However, CNN-LSTM still has some defects in time series analysis. First, CNN-LSTM cannot effectively process noise and outliers in time series data, leading to instability of prediction results. In this case, anomaly detection algorithms or data cleaning techniques can be used to improve the robustness of the model against noise and outliers. Second, CNN-LSTM cannot show the correlation between data at different time points, resulting in low accuracy of prediction results. With the application of in-depth learning algorithms in financial markets, long-distance information is weakened. In this case, the attention mechanism receives more concern. Recent studies on the integration of attention mechanism and neural networks are as follows: landslide displacement prediction [18], soft sensor application [19], and stock price prediction [20].

To combine the advantages of CNN, LSTM, and the attention mechanism, we construct a time series prediction model based on CNN-LSTM with AAM, namely the CNN-LSTM-AAM model. The methodological contributions of this model show that it integrates the advantages of CNN and LSTM, introduces an additive attention mechanism, and improves prediction performance and accuracy. By combining the feature extraction capability of CNN with the series modeling capability of LSTM, our model can more effectively capture local features and long-term dependencies of time series data. At the same time, we innovatively introduce an additive attention mechanism after the LSTM layer, which allows the model to adaptively compute the weights at different time steps, thus focusing more on the important information in the prediction task. Through the innovation of research methodology, our model has better performance and prediction accuracy for time series data in theory.

In addition, the recent literature shows that fusion models of neural networks usually have better performance, related to transformer, self-attention mechanism, and graph neural network (GNN). The transformer model and its variants have achieved great success in natural language processing and have recently been applied to stock price prediction [21]. By comparing the prediction performance

of LSTM, CNN-LSTM, and CNN-LSTM-Transformer models, it is found that the CNN-LSTM-Transformer model shows superior accuracy in ozone concentration prediction [22], vegetable sales forecasting [23], solar power forecasting [24], and stock price prediction [25]. The self-attention mechanism can directly calculate the strength of the association between any two locations in a sequence, thus better capturing long-range dependencies. Some scholars have found that the CNN-LSTM model with self-attention mechanism can extract spatial and temporal features from time series more comprehensively and has better performance compared to LSTM, LSTM with self-attention mechanism, and CNN-LSTM models [26–28]. When complex relationships or dependencies exist between data, CNN and LSTM may not be able to get satisfactory prediction results. However, GNN can better capture the complex relationships between data points, which is crucial for improving prediction performance [29]. For example, the CNN-GNN model can be used for image segmentation [30] and scene classification [31], and the LSTM-GNN model is suitable for time series prediction [32]. After combining the advantages of CNN, LSTM, and GNN, the CNN-LSTM-GNN model can be used for stock price prediction [33] and scene classification [34].

The contributions of this paper mainly lie in constructing the CNN-LSTM-AAM model, verifying its applicability in the stock market, and comparing the predictive performance of different models. First, the CNN-LSTM-AAM model can automatically learn the features in time series data and strengthen the correlation between data in different time points by the AAM. Although existing studies have used CNN-LSTM for stock price prediction, fewer studies combine CNN-LSTM and AAM. Besides, some scholars have conducted prediction studies on time series data based on CNN-LSTM and attention mechanism, but it is not clear which type of attention mechanism is used in their research. Our model not only clarifies the type of attention mechanism, but also discusses the characteristics and advantages of AAM. Second, we integrate trading indexes and technical indexes into the research framework of stock price prediction, which better reflects the applicability of the model in the stock market. Existing studies mostly use trading indexes for prediction analysis, and few studies take both types of indexes into account. Third, we compare the prediction abilities of CNN, CNN-LSTM, and CNN-LSTM-AAM models using sample data from listed companies in different industries and from different types of stock price indexes. The above sample data not only focuses on the stock price volatility of different companies at the micro level, but also pays attention to the overall trend of changes in the stock market at the macro level.

Our research method is different from existing stock prediction models and has obvious innovation. Specifically, the research process involves data cleaning, principal component analysis (PCA), CNN feature extraction, LSTM sequence modeling, and AAM information capture. Our research method can more accurately predict stock prices, and improve the accuracy and stability of the prediction model. The research method presented in this paper enriches the theoretical research of stock price prediction models. The rest of the paper is organized as follows. The next section designs the CNN-LSTM-AAM model, after introducing PCA, CNN, LSTM, and AAM. Section 3 is the experimental design, including two parts: data and experimental setting. In the next section, the results of simulation experiments are shown. The conclusions are presented in the last section.

2 Models

2.1 PCA

PCA transforms a group of potentially relevant variables into a group of linearly independent variables, and the transformed variables are called principal components (PC) [35]. In addition, PCA

can compress and de-noise data by reducing data dimensions [36]. The data processing process of PCA consists of the following four steps:

X is the m-dimensional raw data including n sample points, denoted as Eq. (1).

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \tag{1}$$

• Step 1: Calculating the covariance of the matrix Z.

The mean (μ_i) and variance (σ_i^2) of each column in Eq. (1) can be expressed as Eqs. (2) and (3).

$$\mu_i = \frac{1}{n}\sum_{j=1}^{n} x_{ji} \tag{2}$$

$$\sigma_i^2 = \frac{1}{n-1}\sum_{j=1}^{n} \left(x_{ji} - \mu_i\right)^2 \tag{3}$$

The matrix Z is obtained after the normalization of the raw data by using Eq. (3), as shown in Eq. (4).

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nm} \end{bmatrix} \tag{4}$$

• Step 2: Calculating the correlation coefficient matrix R.

After calculating the correlation coefficient of the raw data, the matrix (R) is obtained and written as Eq. (5).

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mm} \end{bmatrix} \tag{5}$$

r_{ij} is the correlation coefficient between the sample sequence in column (i) and the sample sequence in column (j) of the matrix X in the range $[-1, 1]$. The R should be a symmetric matrix, $r_{ij} = r_{ji}$.

• Step 3: Calculating eigenvalues and eigenvectors of the matrix R.

The correlation coefficient matrix R is a real symmetric matrix with nonnegative eigenvalues, which are assumed to satisfy the following conditions: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_p \geq 0$. The corresponding orthogonalized unit eigenvector $(u_1, u_2, \cdots, u_p)$, is written as Eq. (6).

$$u_1 = \begin{bmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{n1} \end{bmatrix}; u_2 = \begin{bmatrix} u_{12} \\ u_{22} \\ \vdots \\ u_{n2} \end{bmatrix}; \cdots; u_m = \begin{bmatrix} u_{1m} \\ u_{2m} \\ \vdots \\ u_{nm} \end{bmatrix} \tag{6}$$

The column vectors of X are represented successively as $X_1, X_2, \ldots, X_m$, then the i-th principal component of X (F_i) is shown as Eq. (7).

$$F_i = u_{1i}X_1 + u_{2i}X_2 + \cdots + u_{mi}X_m \tag{7}$$

• Step 4: Calculating the contribution rate (CR) and cumulative contribution rate (CCR) of PC.

CCR determines the number of PC. When the CCR reaches above 85%, new variables contain most of the information about the initial variables. In Eqs. (8) and (9), b_j and α_p represent CR and CCR, respectively.

$$b_j = \frac{\lambda_j}{\sum_{k=1}^{m}\lambda_k} (j = 1, 2, \cdots, m) \tag{8}$$

$$\alpha_p = \frac{\sum_{k=1}^{j}\lambda_k}{\sum_{j=1}^{m}\lambda_k} (j = 1, 2, \cdots, m) \tag{9}$$

2.2 CNN

CNN is a deep learning algorithm in essence, having depth structure and convolutional computation [37]. In this network, the process of data processing is as follows:

In the convolution layer, the output after convolution calculation can be expressed as Y_t, which is a function of the input vector (x_t), weight (ω_t), and bias (b_t) of the convolution kernel, seen from Eq. (10).

$$Y_t = f(x_t\omega_t + b_t) \tag{10}$$

After convolution calculation is performed, the extracted features still have high dimensions. The pooling layer can reduce the dimension of extracted features and network training costs. The output after pooling is represented as C_t in Eq. (11). Besides, Y_t is the input vector and *pool* represents the pooling rule.

$$C_t = pool(Y_t) \tag{11}$$

The activation function is important for introducing nonlinear factors. Generally, ReLU is used as the activation function of CNN. After several rounds of convolution layer and pool layer processing, the final classification results are usually given by one or two dense layers.

2.3 LSTM

As an improved recurrent neural network, LSTM solves the long-term dependence problem. Specifically, LSTM can memorize long and short-term information, save important information, and regulate the flow of features [38]. The network structure of LSTM is represented in Fig. 1.

First, the forget gate decides what information should be discarded. It reads the previous information in a hidden state (h_{t-1}) and the current input information (x_t), and produces a value greater than 0 and less than 1. When the value is 1, it means fully preserved. Conversely, 0 means completely abandoned. The output of the forget gate (f_t) is shown in Eq. (12), where $f_t \in [0, 1]$. The weight and bias matrixes are expressed as W_f and b_f, respectively. The sigmoid function is usually chosen as the active function of the forget gate, written as σ.

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

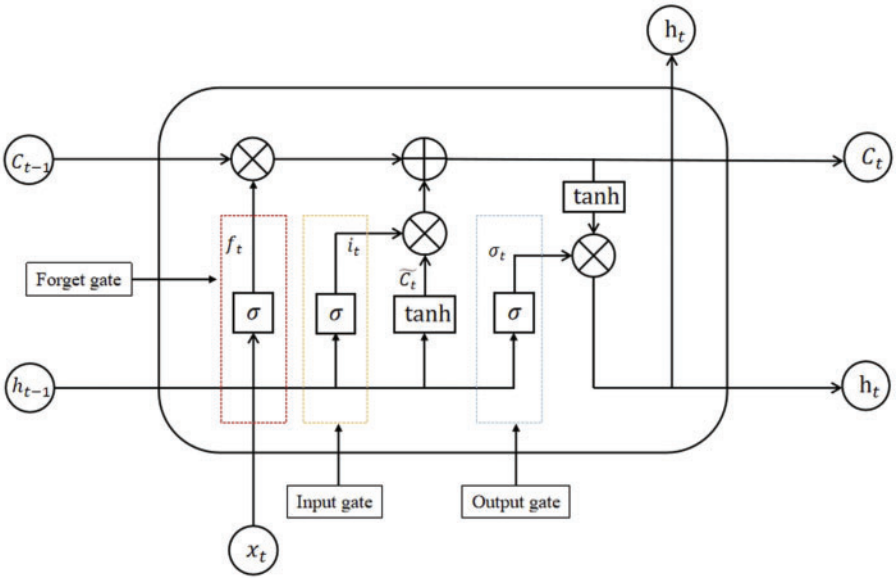


Figure 1: The network structure of LSTM

Second, the input gate determines the updating process of the cell state. Specifically, h_{t-1} and x_t are put into the active function. The function values determine which information needs to be updated. Next, h_{t-1} and x_t are entered into the tanh function, creating a candidate vector ($\tilde{C}_t$). In expressions (13) and (14), i_t and $\tilde{C}_t$ separately indicate the control variable and the updated value of the cell state. Besides, the weight and bias matrixes are expressed as W_i and b_i, respectively. In candidate input gates, W_C and b_C represent the weight and bias matrixes in turn.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{13}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{14}$$

Third, the new cell state (C_t) saves information on the current state and transfers it to the LSTM at the next moment. If the product ($f_t \odot C_{t-1}$) is close to 0, this information will not appear in the new cell state. After $f_t \odot C_{t-1}$ and $i_t \odot \tilde{C}_t$ are added, the cell state gets the new information. Thus, C_t is shown as Eq. (15), where $\odot$ represents the Hadamard product.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{15}$$

Fourth, the output gate determines the value of the next hidden state. In Eq. (16), o_t represents the output of the output gate, where $o_t \in [0, 1]$. In Eq.(17), h_t shows the hidden state. Besides, W_o and b_o represent the weight and bias matrixes, respectively.

$$o_t = \sigma\left(W_o [h_{t-1}, x_t] + b_o\right) \tag{16}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{17}$$

2.4 AAM

Current information has a more significant impact on current prices. Conversely, earlier information has less impact on current prices. AAM can be used to improve the long-distance memory

capability of the LSTM and enable it to better remember previous information [39]. Specifically, the introduction of AAM in LSTM has the following advantages. First, considering the traditional LSTM can only give the same weight to all time steps, AAM assigns different weights to different time steps and helps LSTM to be more flexible in choosing what information to focus on. Second, AAM improves the long-distance memory ability of LSTM by calculating the contribution of different inputs, so it helps LSTM to remember the previous information better. Third, AAM helps LSTM discard unnecessary information and save computing resources, so it can improve the computational efficiency of LSTM. Specifically, we apply AAM to the output of the LSTM to better capture important context information in processing sequence data [40].

The output of LSTM is assumed to be $h_1, h_2, \ldots, h_n$, and the query vector is represented as q. The output of the AAM is obtained by the following three steps:

First, the attention score (e_i) is calculated through Eq. (18). q^T, W_1, and W_2 are learnable weight matrixes, and b is a learnable bias vector. i represents the i-th time step of LSTM, $i \in [1, n]$.

$$e_i = q^T \tanh\left(W_1 h_i + W_2 q + b\right) \tag{18}$$

Second, the attention weight (α_i) is calculated through Eq. (19).

$$\alpha_i = \frac{\exp\left(e_i\right)}{\sum_{j=1}^{n} \exp\left(e_j\right)} \tag{19}$$

Third, the output of the AAM (c) is shown in Eq. (20).

$$c = \sum_{i=1}^{n} \alpha_i h_i \tag{20}$$

2.5 The Construction of the CNN-LSTM-AAM Model

Based on AAM and CNN-LSTM, we construct the CNN-LSTM-AAM model. Fig. 2 shows the network structure of this model. First, the preprocess and feature engineering are carried out, including data cleaning, PCA, and data normalization. Second, the CNN layer extracts data features. Third, the LSTM layer learns the long-term dependence in the time series to reduce the learning difficulty of cyclic networks. Fourth, the AAM layer weights the output of the previous layer to increase the focus on important features and further enhance the prediction accuracy. Fifth, the output of the AAM layer is input to the dense layer. Finally, the predicted values of the time series are exported through the output layer.

In CNN-LSTM architecture, the CNN layer is used to extract local features in time series data, while the LSTM layer is used to capture long-term dependencies in the sequence. This combination enables the model to simultaneously deal with the spatial and temporal dimensions of time series. In the meantime, the CNN-LSTM architecture is flexible enough. We can optimize the model performance by adjusting hyperparameters such as convolutional kernel size, number of LSTM layers, and number of neurons. In addition, AAM can capture the long-term dependence and key features in time series data, while being adaptable enough to be used in conjunction with other types of models (such as CNN and LSTM). In short, the CNN-LSTM-AAM model has significant theoretical advantages for stock price prediction.

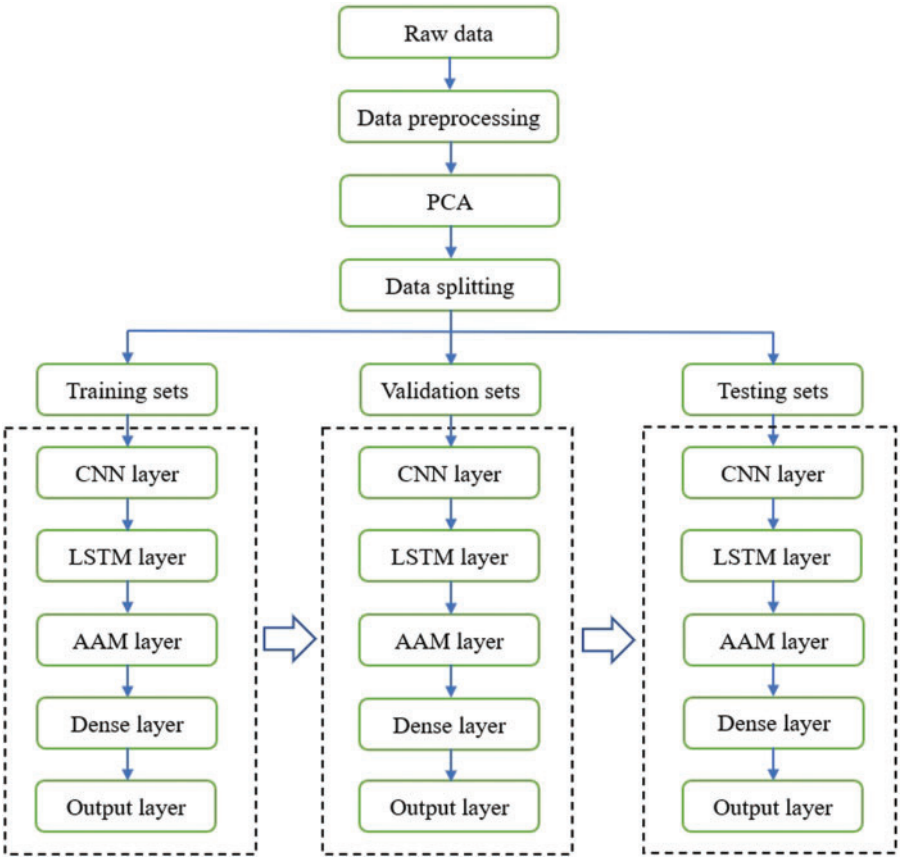


Figure 2: The network structure of the CNN-LSTM-AAM model

3 Experimental Design

3.1 Data Sources

To ensure the accuracy of data sources, stock price data is downloaded from the free BaoStock financial interface package in Python (www.baostock.com) (accessed on 28 October 2024). This paper selects six sample data from the Chinese stock market, including three stocks and three stock price indexes. They are from 01 January 2001, to 31 December 2021. First, three stocks include Vanke A (VA), Shanghai International Port Group (SIPG), and China Merchants Bank (CMB) in the Shanghai Stock Exchange (SSE). Second, three stock price indexes include the China Securities 500 Index (CSI 500), the Shanghai Stock Exchange 50 Index (SSE 50), and the Growth Enterprise Index (GEI). The CSI 500 shows the stock price performance of small-cap companies in the Shanghai and Shenzhen stock markets (SSE and SZSE). The SSE 50 reflects the overall performance of 50 stocks with larger market capitalization and better liquidity in the SSE. The GEI reports the general trend of 100 stocks with large market value and good liquidity in SZSE.

3.2 Index Selection

To predict the stock price more accurately, the data set is based on daily trading information, involving two categories: trading indexes and technical indexes. Considering the availability and completeness of the data set, we have excluded the following indexes: P/E ratio, P/B ratio, market

capitalization, total equity, and limit-up price. Trading indexes are ultimately selected as opening and closing prices (Open-p and Close-p), highest and lowest prices (High-p and Low-p), volume (Vol-p), turnover rate (Turn-r), transaction amount (Trans-a), and price fluctuation rate (Fr-p). It should be noted that the closing price is the forecast target. In the selection of technical indexes, we fully consider the relevance, information capacity, and prediction ability of indexes. By analyzing the characteristics of the different indexes, we finally select 23 technical indexes: Bollinger Bands, Exponential Moving Average, True Range, Accumulation/Distribution, AROON, Money Flow Index, Price Ratio-of-Change, Relative Strength Index, Stochastic Fast, Williams Index, Accumulation/Distribution Oscillator, On Balance Volume, Hilbert Transform—Dominant Cycle Period, Average Price, Typical Price, Weighted Close Price, Average Directional Movement Index, Average Directional Movement Index, Absolute Price Oscillator, Moving Average Convergence/Divergence, Momentum, Volume Rate-of-Change, Stochastic Oscillator, and Ultimate Oscillator. For the above technical indexes, we conduct data cleaning and PCA below.

3.3 Data Cleaning

Due to the complexity and uncertainty of the stock market, sample data may have duplicate values, missing values, and outliers. Therefore, data cleaning is necessary, and includes the following content:

(1) Duplicate data removal. In the stock market, due to the high frequency of data updates, there may indeed be duplicate data. These data should be removed to ensure the uniqueness of the data.

(2) Missing data processing. Missing values indicate that an indicator does not have a value at a certain time. Considering the characteristics of stock price data, including its continuity and time dependence, we choose exponential smoothing interpolation to fill in missing values in time series data. This method takes full account of the temporal relationship between data points and gives higher weight to the most recent data to better capture the dynamics of stock prices.

(3) Outlier detection. Outliers refer to some data in the sample sequence that deviates from the normal range. We adopt the 3-sigma principle to identify and mark possible outliers.

(4) Outlier processing. For data points that are marked as outliers, we perform further analysis to confirm whether they are indeed outliers. After confirmation, we remove them from the dataset.

(5) Re-interpolation. After removing outliers, we check the dataset again for new missing values and use exponential smoothing interpolation to fill them in.

Through the above data processing, we can ensure the integrity and accuracy of the dataset to the maximum extent possible, while reducing the bias introduced due to missing data and improper handling of outliers.

3.4 Principal Component Selection

Data with a high correlation usually harms the learning ability of neural networks. For example, the neural network falls into the local minimum solution and the training burden increases. In the stock market, a variety of indicators are usually used to analyze the trend of stock prices. However, a high correlation is likely between these indicators, which may lead to redundant information and misleading conclusions. PCA transforms highly correlated indicators into mutually independent PCs to reduce dimension and redundant information. Besides, PCA can reduce the calculation cost and analysis

difficulty by screening out major indicators with greater contributions. By selecting the first few PCs with large eigenvalues, most of the variance information is retained to achieve better dimensionality reduction. Therefore, we select the PC according to whether the eigenvalue is greater than 1.

Tables 1–3 show that the first seven PCs are selected for CMB, CSI 500, and GEI, the first eight PCs are selected for VA and SSE 50, and the first nine PCs are selected for SIPG. For CMB, CSI 500, and GEI, the eigenvalue of the seventh PC equals 1.1746, 1.1241, and 1.1231, respectively. Next, the eigenvalue of the eighth PC equals 1.1112 and 1.0305 for VA and SSE 50 in turn. Finally, the eigenvalue of the ninth PC is 1.0024 for SIPG. The CCR of each dataset is greater than 90%, indicating PCA extracts more data features and can effectively preserve the information of the original data.

Table 1: Eigenvalue and CCR of VA and SIPG

Components	VA		SIPG	
	Eigenvalue	CCR	Eigenvalue	CCR
1	14.7932	0.4049	15.324	0.4188
2	9.5466	0.6663	8.7968	0.6522
3	2.8930	0.7455	3.1341	0.7448
4	2.5449	0.8152	2.2869	0.8073
5	1.4022	0.8536	1.8052	0.8566
6	1.3085	0.8894	1.3440	0.8934
7	1.1731	0.9215	1.1782	0.9256
8	1.1112	0.9520	1.0437	0.9541
9	0.9568	0.9781	1.0024	0.9815
10	0.7996	1.0000	0.6778	1.0000

Table 2: Eigenvalue and CCR of CMB and CSI 500

Components	CMB		CSI 500	
	Eigenvalue	CCR	Eigenvalue	CCR
1	15.3095	0.4213	16.4491	0.4390
2	9.3478	0.6785	10.5429	0.7204
3	3.0422	0.7622	2.6808	0.7920
4	2.1814	0.8222	1.7543	0.8388
5	1.3716	0.8600	1.3988	0.8761
6	1.2987	0.8957	1.2853	0.9104
7	1.1746	0.9280	1.1241	0.9404
8	0.9271	0.9536	0.8916	0.9642
9	0.8603	0.9772	0.7936	0.9854
10	0.8274	1.0000	0.5465	1.0000

Table 3: Eigenvalue and CCR of SSE 50 and GEI

Components	SSE 50		GEI	
	Eigenvalue	CCR	Eigenvalue	CCR
1	15.4980	0.4202	16.9911	0.4569
2	9.9952	0.6912	9.7501	0.7191
3	2.7620	0.7661	2.6416	0.7902
4	2.1317	0.8239	1.5436	0.8317
5	1.4867	0.8642	1.4669	0.8711
6	1.2289	0.8975	1.2725	0.9053
7	1.1292	0.9281	1.1231	0.9355
8	1.0305	0.9561	0.9415	0.9609
9	0.8451	0.9790	0.8384	0.9834
10	0.7746	1.0000	0.6170	1.0000

3.5 Data Normalization

Before using machine learning models, the data should be normalized to avoid the deviation of results caused by different dimensions. All sample data (x) are converted to normalized values (x^*) between 0 and 1 by Eq. (21). The largest and smallest x are represented as x_{max} and x_{min}, respectively.

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{21}$$

3.6 Experimental Setting

Before training, verifying, and testing machine learning models, sample data are split into three parts in chronological order: training sets, validation sets, and testing sets. Specifically, the first dataset contains the first 80% data, the second dataset includes the subsequent 10% data, and the third dataset consists of the last 10% data. The data splitting method is a one-time split rather than cross-validation in this paper. This split method is chosen to keep our experiments simple and reproducible. By dividing the data into training, validation, and test sets, we can train, tune, and evaluate the performance of the model separately, which helps us to get a more accurate picture of how the model performs on unseen data.

The model parameters and parameter values are summarized in Table 4. In the CNN layer, the five parameters have different setting values. The parameters of the LSTM layer are one more than those of the Dense layer and AAM layer, respectively. Finally, training parameters mainly include Time step, Batch size, Epoch, optimizer, Loss function, Dropout, and Learning rate. It should be noted that we have performed sensitivity analyses on the parameters in Table 4.

Table 4: The parameter settings

Category	Parameter	Setting value
CNN layer	Filter	64
	Kernel size	1

(Continued)

Table 4 (continued)

Category	Parameter	Setting value
	Activation function	ReLU
	Padding	Same
	Pooling size	2
LSTM layer	Layer	3
	Unit	64
	Activation function	ReLU
AAM layer	Layer	1
	Unit	64
Dense layer	Layer	1
	Unit	1
Training parameter	Time step	20
	Batch size	64
	Epoch	100
	Optimizer	Adam
	Loss function	MSE
	Dropout	0.2
	Learning rate	0.001

As the CNN-LSTM-AAM model has many parameters, a direct comprehensive search and tuning may encounter computational resources and time constraints. Therefore, we adopt a two-step approach to determine the parameter values: experience setting and random search. For some key parameters, such as convolutional kernel size and learning rate, we set initial values based on existing research and experience. These parameters have a significant impact on the performance of the model and usually have recommended value ranges in the related literature. Specifically, the convolutional kernel size is usually set to a small value (such as 1) to capture local features, while the learning rate is set to a common range (such as 0.001, 0.01, etc.) based on the choice of optimization algorithm (such as Adam, SGD, etc.). For other parameters such as Batch size, Epoch, and Time step, we use random search to determine their optimal values. The random search can explore the parameter space in a short time to find the parameter combinations with better performance. The batch size usually affects the training speed and stability of the model. For small datasets, the choice of batch size is relatively flexible, and smaller batches can be selected, such as 16, 32, or 64, etc. On the contrary, larger batches should be selected to improve training efficiency, such as 1024 or more. Considering the stability and speed of the training process, we first choose a smaller batch and then gradually increase it to finally determine the optimal value. In addition, Epoch determines the learning level of the model on the training data. We determine the optimal Epoch based on the performance of the validation set. Since the time step can affect the capacity of the model to capture time series data, we choose an appropriate value based on the temporal resolution and features of the data.

Thus, the process of determining parameters includes initial parameter settings and random search for tuning. Based on experience and literature, initial values of key parameters (such as the kernel size and learning rate) are set. Next, several experiments are conducted using a random search method over a range of candidate values for batch size, number of iterations, and time step. After each experiment,

we need to record the performance metrics of the model on the validation set (such as MSE, MAE, and MAPE). Finally, based on the experimental results, the parameter combination with the best performance is selected. Through the above two-step method of determining the parameter values, we can find a more optimal combination of parameters within the limited computational resources and time. This method combines the advantages of experience and random search, which ensures the rationality of parameter setting and improves the tuning efficiency.

MSE, MAE, and MAPE are often used to evaluate the prediction ability of the model [41]. A smaller evaluation index indicates a higher prediction accuracy of the model. Three evaluating indexes are shown in expressions (22)–(24), where y_i and $\hat{y}_i$ represent actual and predictive values, respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \tag{22}$$

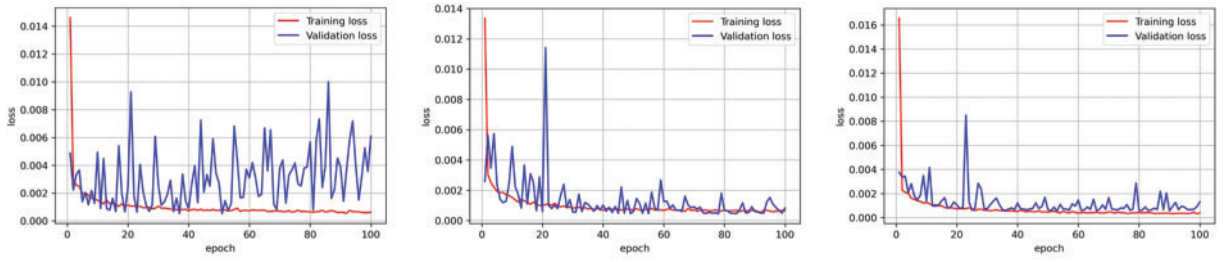
$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|\hat{y}_i - y_i\right| \tag{23}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{24}$$

4 Results

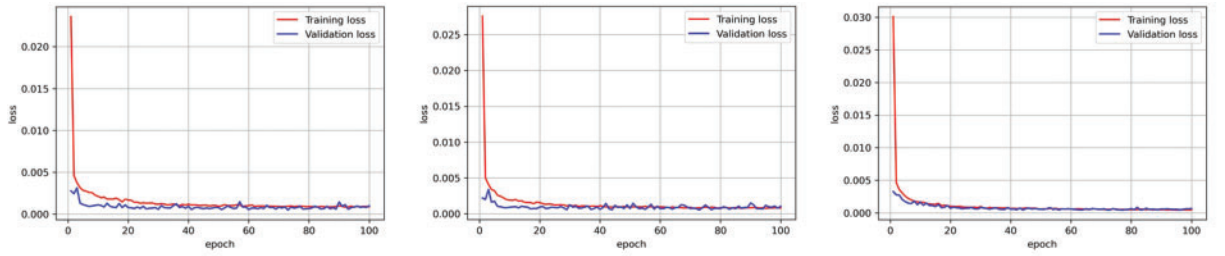
In the simulation experiment, we compare the prediction ability of the CNN-LSTM-AAM model (Model-3rd) with that of the LSTM model (Model-1st) and the CNN-LSTM model (Model-2nd). The analysis process includes two steps. We first analyze the prediction ability with three stock price data sets, and then verify it again with three stock price index data sets.

Figs. 3–5 show the loss curves of three models in the data sets of VA, SIPG, and CMB. It can be found that the Model-3rd has the most stable performance during the training process and the smallest final loss value compared to the other two models in the three stock data sets. Thus, the Model-3rd has stronger ability to reflect the characteristics of stock price volatility and can predict the trend of stock prices more accurately. For the SIPG dataset, the loss function curves of the three models perform similarly in Fig. 4. However, the Model-3rd performs better compared to the other two models for VA and CMB data sets in Figs. 3 and 5. The above differences probably come from individual differences in the stock data, but the performance of Model-3rd is still validated.



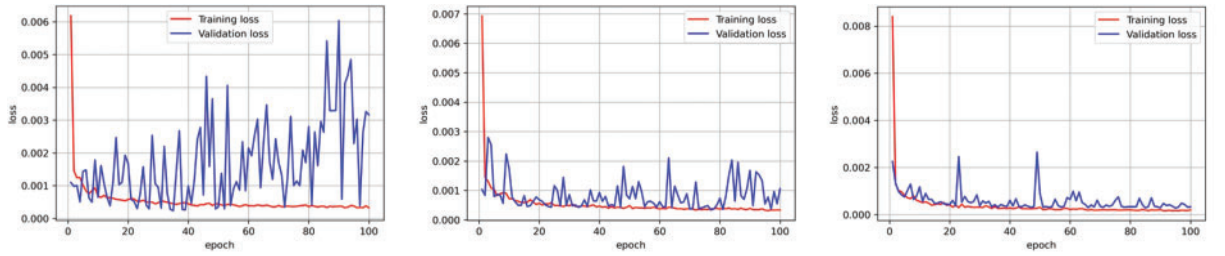
(a) The loss curves of the LSTM model (b) The loss curves of the CNN-LSTM model (c) The loss curves of the CNN-LSTM-AAM model

Figure 3: The loss curves for VA



(a) The loss curves of the LSTM model (b) The loss curves of the CNN-LSTM model (c) The loss curves of the CNN-LSTM-AAM model

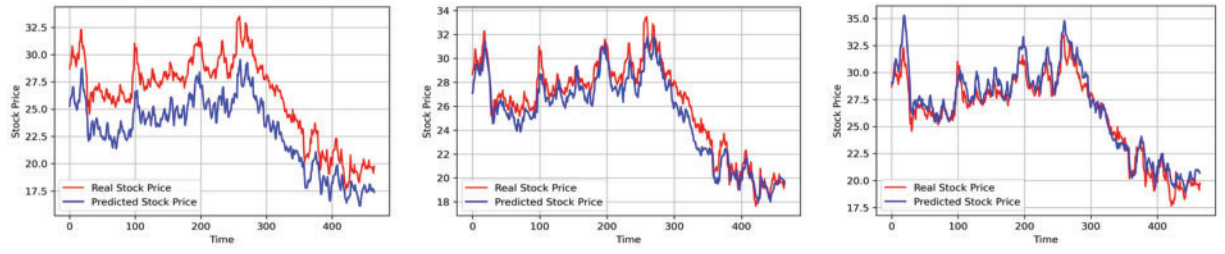
Figure 4: The loss curves for SIPG



(a) The loss curves of the LSTM model (b) The loss curves of the CNN-LSTM model (c) The loss curves of the CNN-LSTM-AAM model

Figure 5: The loss curves for CMB

Figs. 6–8 show the prediction results of different models in the data sets of VA, SIPG, and CMB, respectively. As can be seen from these three figures, the predicted stock prices are closer to the actual stock prices and have less volatility relative to the actual values in the Model-3rd. Specifically, the prediction results of the Model-3rd are closer to the actual results with less fluctuation for VA and CMB data sets in Figs. 6 and 8, demonstrating better stability and accuracy. For the SIPG dataset, the difference in prediction results is not too great in Fig. 7, but the Model-3rd model is still superior to the other two models. In addition, we find that each model has different prediction effects for the data sets of VA, SIPG, and CMB. Thus, the parameter environment of the Model-3rd needs to be adjusted for different stock data sets to achieve the best prediction effects.

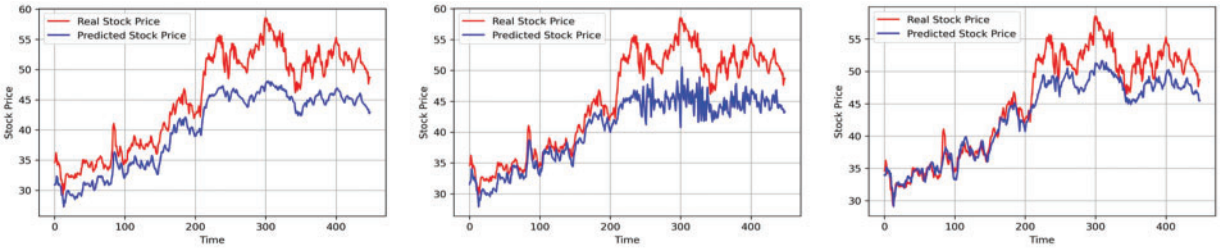


(a) The predicition results of the LSTM model (b) The predicition results of the CNN-LSTM model (c) The predicition results of the CNN-LSTM-AAM model

Figure 6: The prediction results for VA

(a) The predicition results of the LSTM model (b) The predicition results of the CNN-LSTM model (c) The predicition results of the CNN-LSTM-AAM model

Figure 7: The prediction results for SIPG



(a) The predicition results of the LSTM model (b) The predicition results of the CNN-LSTM model (c) The predicition results of the CNN-LSTM-AAM model

Figure 8: The prediction results for CMB

Table 5 displays the evaluation results in data sets of VA, SIPG, and CMB, including MSE, MAE, and MAPE. The introduction of CNN structure is beneficial to extract data features and enhance the prediction abilities of models. For VA, MSE decreases from 0.0066 to 0.0011, MAE decreases from 0.0782 to 0.0277, and MAPE decreases from 12.80% to 4.50%. For SIPG and CMB, three evaluation indexes show an obvious decline. Because the prediction results of the Model-2nd are closer to the actual results, this model has better accuracy in the three data sets, compared to the Model-1st. Besides, Table 5 shows that Model-3rd has the smallest MSE, MAE, and MAPE in the three data sets, relative to the other two models. For example, the MSE, MAE, and MAPE are 0.0008%, 0.0229%, and 3.86% in turn for VA. They are reduced by 87.88%, 70.72%, and 69.84% relative to Model-1st, and by 27.27%, 17.33%, and 14.22% relative to Model-2nd. For the SIPG data set, the MSE, MAE, and MAPE are respectively 0.0002%, 0.0113%, and 2.91%, which are reduced by 71.43%, 54.44%, and 56.63% in Model-1st, and 50.00%, 28.93% and 31.85% in Model-2nd. For the CMB data set, the same conclusion shows that the MSE, MAE, and MAPE have downward trends. In addition, the prediction abilities of the three models are different for different stock data sets. Among the three models, the data set with the smallest MSE, MAE, and MAPE is the SIPG data set. Specifically, the SIPG data set has the smallest MSE, which equals 0.0007 in Model-1st, 0.0004 in Model-2nd, and 0.0002 in Model-3rd. Likewise, it has the smallest MAE and MAPE in these models. Therefore, the prediction of the SIPG data set is relatively more accurate in the existing parameter conditions.

Table 5: The evaluation results for three stocks

Data sets	Models	MSE	MAE	MAPE (%)
VA	Model-1st	0.0066	0.0782	12.80
	Model-2nd	0.0011	0.0277	4.50

(Continued)

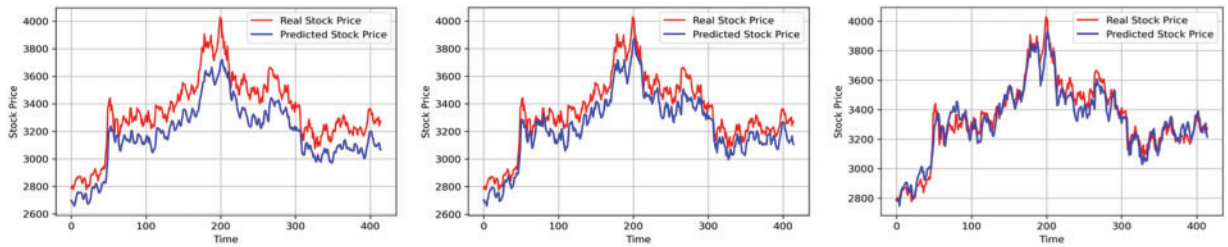
Table 5 (continued)

Data sets	Models	MSE	MAE	MAPE (%)
	Model-3rd	0.0008	0.0229	3.86
SIPG	Model-1st	0.0007	0.0248	6.71
	Model-2nd	0.0004	0.0159	4.27
	Model-3rd	0.0002	0.0113	2.91
CMB	Model-1st	0.0108	0.0962	12.45
	Model-2nd	0.0115	0.0877	10.69
	Model-3rd	0.0039	0.0486	5.87

To verify the prediction ability of the Model-3rd again, three stock price indexes are selected for further analysis, including CSI 500, SSE 50, and GEI. Figs. 9–11 show the prediction results of different models in the data sets of CSI 500, SSE 50, and GEI, respectively. The prediction ability of the Model-2nd is significantly better than that of Model 1 for CSI 500, SSE 50, and GEI. Specifically, the predicted curve is closer to the real curve in the Model-2nd. In Figs. 10 and 11, the prediction ability of the Model-3rd is superior to that of the Model-2nd, but this advantage is not particularly evident in Fig. 9. Overall, the prediction ability of the models is ranked from high to low, followed by Model-3rd, Model-2nd, and Model-1st. For a specific dataset, the parameter environment of Model-3rd needs to be continuously optimized to achieve better prediction ability.

(a) The predicition results of the LSTM model (b) The predicition results of the CNN-LSTM model (c) The predicition results of the CNN-LSTM-AAM model

Figure 9: The prediction results for CSI 500



(a) The predicition results of the LSTM model (b) The predicition results of the CNN-LSTM model (c) The predicition results of the CNN-LSTM-AAM model

Figure 10: The prediction results for SSE 50

(a) The predicition results of the LSTM model (b) The predicition results of the CNN-LSTM model (c) The predicition results of the CNN-LSTM-AAM model

Figure 11: The prediction results for GEI

Table 6 displays the evaluation results in data sets of CSI 500, SSE 50, and GEI, including MSE, MAE, and MAPE. For CSI 500, SSE 50, and GEI, three evaluation indexes have the smallest values in the Model-3rd. In particular, they equal 0.00099%, 0.0279%, and 5.02% for CSI 500; 0.0002%, 0.0115%, and 4.18% for SSE 50; 0.0027%, 0.0464%, and 5.95% for GEI. By comparing the evaluation results, the prediction abilities of the three models are ranked from high to low as Model-3rd, Model-2nd, and Model-1st. The conclusion again shows that the Model-3rd has the strongest prediction ability. Among the three models, the SSE 50 data set has the smallest MSE, MAE, and MAPE. It shows that Model-3rd is relatively more accurate in predicting the SSE 50 data set under the existing parameter conditions.

Table 6: The evaluation results for three stock price indexes

Data sets	Models	MSE	MAE	MAPE (%)
CSI 500	Model-1st	0.0026	0.0494	8.98
	Model-2nd	0.00099	0.0288	5.22
	Model-3rd	0.00099	0.0279	5.02
SSE 50	Model-1st	0.0010	0.0285	11.98
	Model-2nd	0.0007	0.0231	9.63
	Model-3rd	0.0002	0.0115	4.18
GEI	Model-1st	0.0176	0.1280	16.54
	Model-2nd	0.0057	0.0700	9.01
	Model-3rd	0.0027	0.0464	5.95

5 Conclusions

The CNN-LSTM-AAM model constructed in this paper has a strong ability to predict stock prices. We use the data sets of three stock prices (VA, SIPG, and CMB) to compare the prediction abilities of the three models. The results indicate that the CNN-LSTM-AAM model has the greatest advantages in prediction ability among the three models. In the three stock data sets, the CNN-LSTM-AAM model has the smallest MSE, MAE, and MAPE relative to the other two models. In addition, we use the data sets of three stock price indexes (CSI 500, SSE 50, and GEI) to verify the prediction abilities of the three models. The results again indicate that our model has the strongest prediction ability and accuracy. Besides, we also find that the prediction ability of the CNN-LSTM-AAM model for different stock data sets is different under the existing parameter conditions. In short,

the CNN-LSTM-AAM model has bright application prospects and presents a new method to predict stock prices. For a specific stock data set, we need to optimize the parameter values of the model to achieve the best prediction effect. Given the strong predictive ability of the CNN-LSTM-AAM model, it can be widely applied in many fields, such as text classification, water quality prediction, facial expression recognition, mechanical fault identification, earthquake prediction, human action recognition, passenger flow prediction, trajectory prediction, and so on. For the financial industry, the CNN-LSTM-AAM model can be used for risk management and investment decisions. Specifically, our model can forecast financial market trends and risk changes by learning from historical data, helping financial institutions evaluate financial risks promptly. Besides, our model can predict the price trend of financial assets and optimize investors' investment decisions by analyzing and processing data such as financial news and company financial statements.

The CNN-LSTM-AAM model proposed in this paper has better performance and accuracy in stock price prediction, but still has some limitations. First, this model only considers daily trading information, without incorporating other factors such as investor sentiment characteristics, national policies, economic cycles, and industry factors into the training features. Therefore, future research should consider increasing the training features of the model to improve its predictive performance. We plan to integrate more factors from the perspective of feature selection to capture characteristics of stock price volatility more comprehensively. Second, our model shows better stability and accuracy in stock price prediction, but still has some prediction errors. These prediction errors probably result from the complexity and uncertainty of the stock market. Thus, we intend to further optimize the structure of the model by incorporating more machine learning algorithms, ultimately improving the accuracy and robustness of the predictions. Third, our research only focuses on stock price prediction, but the potential applications of our model in other fields are also worth exploring. We should further expand the application scope of the CNN-LSTM-AAM model in biology, medicine, sociology, economics, and other disciplines.

Acknowledgement: The authors are grateful to all the editors and anonymous reviewers for their comments and suggestions.

Funding Statement: This research is supported by the Program of Social Science Foundation in Shaanxi Province of China (Grant No. 2020D048) and the Fundamental Research Funds for the Central Universities of China (Grant No. 2022ZYYB18).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Binghui Wu, Guanhao Fu; data collection: Yaxin Qi, Guanhao Fu; analysis and interpretation of results: Binghui Wu, Guanhao Fu; draft manuscript preparation: Binghui Wu, Yaxin Qi, Guanhao Fu; investigation: Yaxin Qi, Guanhao Fu; validation: Binghui Wu, Guanhao Fu; supervision: Binghui Wu; funding acquisition: Binghui Wu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Binghui Wu, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Hao J, Feng QQ, Li J, Sun X. A bi-level ensemble learning approach to complex time series forecasting: taking exchange rates as an example. J Forecast. Feb 2023;42(6):1385–406. doi:10.1002/for.2971.
2. Kumar G, Singh UP, Jain S. Hybrid evolutionary intelligent system and hybrid time series econometric model for stock price forecasting. Int J Intell Syst. 2021 May;36(9):4902–35. doi:10.1002/int.22495.
3. Liu Y, Xie T. Machine learning versus econometrics: prediction of box office. Appl Econ Lett. 2019 Jan;26(2):124–30. doi:10.1080/13504851.2018.1441499.
4. Jiang W. Applications of deep learning in stock market prediction: recent progress. Expert Syst Appl. 2021 Jul;184:115537. doi:10.1016/j.eswa.2021.115537.
5. Zhang G, Guo W, Xiong X, Guan Z. A hybrid approach combining data envelopment analysis and recurrent neural network for predicting the efficiency of research institutions. Expert Syst Appl. 2024 Oct;238(13):122150. doi:10.1016/j.eswa.2023.122150.
6. Roodschild M, Gotay Sardiñas J, Will A. A new approach for the vanishing gradient problem on sig-moid activation. Prog Artif Intell. 2020 Oct;9(4):351–60. doi:10.1007/s13748-020-00218-y.
7. Qin C, Chen L, Cai Z, Liu M, Jin L. Long short-term memory with activation on gradient. Neural Netw. 2023 Apr;164(2):135–45. doi:10.1016/j.neunet.2023.04.026.
8. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997 Nov;9(8):1735–80. doi:10.1162/neco.1997.9.8.1735.
9. Liu R, Vakharia V. Optimizing supply chain management through BO-CNN-LSTM for demand forecasting and inventory management. J Organ End User Comput. 2024;36(1):335591. doi:10.4018/JOEUC.
10. Hansson M. On stock return prediction with LSTM networks (M.S. Thesis). Lund University: Lund, Sweden; 2017.
11. Lu M, Xu X. TRNN: an efficient time-series recurrent neural network for stock price prediction. Inform Sci. 2024 Nov;657(3):119951. doi:10.1016/j.ins.2023.119951.
12. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. Pattern Recognit. 2018 Oct;77(11):354–77. doi:10.1016/j.patcog.2017.10.013.
13. Khurana D, Koli A, Khatter K, Singh S. Natural language processing: state of the art, current trends and challenges. Multimed Tools Appl. 2023 Jul;82(3):3713–44. doi:10.1007/s11042-022-13428-4.
14. Hourri S, Nikolov NS, Kharroubi J. Convolutional neural network vectors for speaker recognition. Int J Speech Technol. 2021 Jun;24(2):389–400. doi:10.1007/s10772-021-09795-2.
15. Karthikeyan V. Modified layer deep convolution neural network for text-independent speaker recogni-tion. J Exp Theor Artif Intell. 2024 Jul;36(2):273–85. doi:10.1080/0952813X.2022.2092560.
16. Kim T, Kim HY. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. PLoS One. 2019 Feb;14(2):e0212320. doi:10.1371/jour-nal.pone.0212320.
17. Baek H. A CNN-LSTM stock prediction model based on genetic algorithm optimization. Asia-Pac Financ Mark. 2023 Jun;31(2):205–20. doi:10.1007/s10690-023-09412-z.
18. Bai D, Lu G, Zhu Z, Tang J, Fang J, Wen A. Using time series analysis and dual-stage attention-based recurrent neural network to predict landslide displacement. Environ Earth Sci. 2022 Oct;81(21):509. doi:10.1007/s12665-022-10637-w.
19. Ma L, Zhao Y, Wang B, Shen F. A multi-step sequence-to-sequence model with attention LSTM neural networks for industrial soft sensor application. IEEE Sens J. 2023 Apr;23(10):10801–13. doi:10.1109/JSEN.2023.3266104.
20. Zhang Q, Zhang Y, Yao X, Li S, Zhang C, Liu P. A dynamic attributes-driven graph attention network modeling on behavioral finance for stock prediction. ACM Trans Knowl Discov Data. 2023 Sep;18(1):1–29. doi:10.1145/361131.
21. Wang C, Chen Y, Zhang S, Zhang Q. Stock market index prediction using deep Transformer model. Expert Syst Appl. 2022 Dec;208:118128. doi:10.1016/j.eswa.2022.118128.

22. Chen Y, Chen X, Xu A, Sun Q, Peng X. A hybrid CNN-Transformer model for ozone concentration prediction. Air Qual Atmos Health. 2022 Apr;15(9):1533–46. doi:10.1007/s11869-022-01197-w.
23. Tian A. Enhancing vegetable sales forecasting with a CNN-LSTM-Transformer hybrid model. In: 2nd International Conference Business Management Mark Foreign Trade (BMMFT 2023), 2023 Sep 2–3; Frankfurt, Germany. doi:10.54097/rr7t4d15.
24. Salman D, Direkoglu C, Kusaf M, Fahrioglu M. Hybrid deep learning models for time series forecasting of solar power. Neural Comput Appl. 2024 Feb;36(16):1–18. doi:10.1007/s00521-024-09558-5.
25. Asl GS, Thulasiram RK, Thavaneswaran A. Stock volatility forecasting with transformer network. In: Proceedings 2023 IEEE Symposium Series on Computer Intelligence (SSCI), 2023 Dec 5–8; Mexico City, Mexico. doi:10.1109/SSCI52147.2023.10371830
26. Sun L, Xu W, Liu J. Two-channel attention mechanism fusion model of stock price prediction based on CNN-LSTM. ACM Trans Asian Low-Resour Lang Inf Process. 2021 Jul;20(5):83. doi:10.1145/3453693.
27. Yi S, Liu H, Chen T, Zhang J, Fan Y. A deep LSTM-CNN based on self-attention mechanism with input data reduction for short-term load forecasting. IET Gener Transm Distrib. 2023 Apr;17(7):1538–52. doi:10.1049/gtd2.12763.
28. Deng S, Zhou J. Prediction of remaining useful life of aero-engines based on CNN-LSTM-Attention. Int J Comput Intell Syst. 2024 Sep;17(1):232. doi:10.1007/s44196-024-00639-w.
29. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst. 2020 Mar;32(1):4–24. doi:10.1109/TNNLS.2020.2978386.
30. Lu Y, Chen Y, Zhao D, Liu B, Lai Z, Chen J. CNN-G: convolutional neural network combined with graph for image segmentation with theoretical analysis. IEEE Trans Cogn Dev Syst. 2020 May;13(3):631–44. doi:10.1109/TCDS.2020.2998497.
31. Jiang J, He Z, Zhang S, Zhao X, Tan J. Learning to transfer focus of graph neural network for scene graph parsing. Pattern Recognit. 2021;112(2):107707. doi:10.1016/j.patcog.2020.107707.
32. Chen P, Fu X, Wang X. A graph convolutional stacked bidirectional unidirectional-LSTM neural network for metro ridership prediction. IEEE Trans Intell Transp Syst. 2021 Jul;23(7):6950–62. doi:10.1109/TITS.2021.3065404.
33. Wu JMT, Li Z, Herencsar N, Vo B, Lin JCW. A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. Multimedia Syst. 2023 Jun;29(3):1751–70. doi:10.1007/s00530-021-00758-w.
34. Pan Y, Zhang X, Jiang H, Li C. A network traffic classification method based on graph convolution and LSTM. IEEE Access. 2021 Nov;9:158261–72. doi:10.1109/ACCESS.2021.3128181.
35. Farahabadi FB, Vajargah KF, Farnoosh R. Dimension reduction big data using recognition of data features based on copula function and principal component analysis. Adv Math Phys. 2021 Jul;2021:9967368. doi:10.1155/2021/9967368.
36. Johnstone LM, Lu AY. On consistency and sparsity for principal components analysis in high dimensions. J Am Stat Assoc. 2009 Jan;104(486):682–93. doi:10.1198/jasa.2009.0121.
37. Rao X, Lin F, Chen Z, Zhao J. Distracted driving recognition method based on deep convolutional neural network. J Ambient Intell Humaniz Comput. 2021 Jan;12(1):193–200. doi:10.1007/s12652-019-01597-4.
38. Lindemann B, Müller T, Vietz H, Jazdi N, Weyrich M. A survey on long short-term memory networks for time series prediction. Procedia CIRP. 2021 May;99(2):650–5. doi:10.1016/j.procir.2021.03.088.
39. Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning. Neurocomputing. 2021 Sep;452:48–62. doi:10.1016/j.neucom.2021.03.091.
40. Xiang L, Wang P, Yang X, Hu A, Su H. Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism. Measurement. 2021 Apr;175(8):109094. doi:10.1016/j.measurement.2021.109094.
41. Nourbakhsh Z, Habibi N. Combining LSTM and CNN methods and fundamental analysis for stock price trend prediction. Multimed Tools Appl. 2023 May;82(12):17769–99. doi:10.1007/s11042-022-13963-0.