WEP Warwick Evans Publishing

International Journal of Global Economics and Management

ISSN: 3005-9690 (Print), ISSN: 3005-8090 (Online) | Volume 4, Number 1, Year 2024
DOI: https://doi.org/10.62051/ijgem.v4n1.17
Journal homepage: https://wepub.org/index.php/IJGEM/index

A Study on Stock Price Prediction Based on Attention GCN-BiLSTM Modeling

Xiaofang Zhao *

School of Statistics and Applied Mathematics, Anhui University of Finance and Economics, Bengbu, China

*Corresponding Author: Xiaofang Zhao

ABSTRACT

It is of great significance to accurately predict stock price movements. By combining the R-Vine-Copula structure with the relationship between the conceptual sectors to which the stocks belong, the price correlation between stocks is measured and the adjacency matrix is constructed, and the model AGLSTM, which is constructed by combining the temporal attention mechanism, Graph Convolutional Neural Network and Bi-directional Long and Short-Term Memory Neural Network, is proposed to predict the price of 46 stocks from the constituent stocks of the SSE 50 by modeling them. The experimental results show that, compared with the baseline, the AGLSTM can predict the stock price. The experimental results show that compared with the baseline model, the MAE and RMSE results of the AGLSTM model predicted one step forward outperform all the comparison models, and the MAPE results are close to the best baseline model results. The results of MAPE are close to those of the best baseline model. Moreover, good results are also achieved in the experiment of multi-step forward prediction, which demonstrates the ability of AGLSTM model in long-term prediction and can provide some reference for investors to make investment decisions.

KEYWORDS

GCN; Bi-LSTM; Vine-Copula; Stock price prediction

1. INTRODUCTION

As one of the most important investment methods in the securities market, stocks have always been widely recognized. Stock trading is often accompanied by high profits and high risks. With the continuous growth of the financial market and the gradual increase in people's awareness of investment and finance, how to accurately predict the future price of stocks, so as to timely buy stocks that will rise in price and sell stocks that will fall in order to obtain higher investment profits has become the focus of people's attention. However, the stock market is characterized by high volatility, time-varying and complex dependence, which makes the accurate prediction of stock prices an extremely challenging task. Stock price series is a typical financial time series, characterized by nonlinearity, non-stationarity and high noise. Traditional stock price forecasting methods are mainly carried out with the help of stock trading data and technical indicators. Since stock prices are affected by various factors such as macroeconomics, political winds, news events, etc., scholars have studied and achieved good results by introducing external information such as text data to predict stock movements. However, the stock market is a highly interdependent and dynamic system, and these studies have neglected the correlation between the stock prices of different companies. Roll proposed stock price synchronization [1], where the price change of one stock may show up simultaneously or subsequently to other stocks connected to it in the network. With the boom of Graph Neural Networks

(GNN), scholars began to introduce graph structures into the stock market to represent the correlations between stocks, contributing to the study of stock price prediction. The crucial step in stock price prediction using Graph Neural Networks lies in the construction of the adjacency matrix, and choosing an appropriate adjacency matrix may be more important than model tuning. Only a relational network that conforms to realistic mechanisms can truly model the conduction path of stock price fluctuations and make correct predictions of stock prices [2], thus helping investors to obtain higher investment returns.

In order to get better stock price prediction results, the author proposes a model AGLSTM (temporal Attention Graph convolution bidirectional Long Short-Term Memory network) for stock price prediction. We introduce tail correlation coefficients and conceptual relationships for the construction of inter-stock adjacency matrices, extract features of stock prices from spatial and temporal dimensions, and assist in predicting stock prices on the next trading day and multiple trading days in the future by using the attention mechanism. By conducting experiments on 46 stocks from SSE 50 constituents and comparing the baseline model, it is verified that the proposed AGLSTM model can help investors make better investment decisions by predicting the future prices of specific stocks. The main contributions of this paper are as follows: First, combining the tail correlation coefficient to measure the correlation between stocks, it explores a new direction for the construction of adjacency matrix. Second, compared with the baseline model, the proposed model performs well in long-term forecasting and can provide investors with a longer-term investment reference. Third, the proposed model uses only stock closing price data to model the relationship with conceptual sectors, which reduces the workload when forecasting large-scale stock closing prices.

2. LITERATURE REVIEW

2.1. Time Series Based Stock Price Forecasting

Forecasting models based on time series are mainly divided into traditional time series models based on statistical analysis and machine learning models based on the laws of the data itself.

Among the traditional models, the Autoregressive Model (AR), Autoregressive Moving Average Model (ARMA) and their variants are often used for forecasting stock prices in financial markets. These models fit univariate time series in historical stock data by assuming that the future values of the series variables are linear functions of the past observations of the variables and random errors. However, financial markets are characterized by a high degree of complexity and nonlinear dependence, and assumptions based on linearity can lead to large biases in forecasts. Although the Autoregressive Conditional Heteroskedasticity Model (ARCH), Generalized Autoregressive Conditional Heteroskedasticity Model (GARCH) and its variants make nonlinear assumptions, they often assume that the model parameters or the random errors of the time series satisfy a specific distribution, and in reality, it is difficult to accurately judge the actual distribution.

Machine learning-based stock price movement prediction mainly searches for patterns from time-varying multivariate datasets and trains classifiers to discriminate the price movement of target stocks. In recent years, machine learning methods such as Random Forest (RF) [3, 4], XGBoost [5] and Support Vector Machine (SVM) [6] have been applied to stock market prediction. These cases have achieved good results in predicting stock price movement and greatly enriched the research in this field. With the development of deep learning, Recurrent Neural Networks (RNN) came into the public eye for its ability to capture nonlinear dependencies. As a variant of RNN, Long Short-Term Memory Neural Network (LSTM) has powerful feature extraction capability and is good at dealing with temporal data and latency problems, improving the gradient vanishing problem that exists in RNN. At the same time, the stacked network structure of LSTM can extract deeper market features and improve the prediction performance, which is widely used in financial markets. Fischer and Krauss first introduced LSTM models into stock forecasting in the financial field [9] and explained the applicability of LSTM models in stock forecasting based on the long-period sequence memory and recognition properties of LSTM models themselves. In order to improve the forecasting performance, some hybrid models combined with other models have been proposed in academia. Wang et al. proposed an Attention Variational Sequence to Sequence model based on Convolutional LSTM [10] to dynamically extract potential representations of the financial market trends from the trading data. Wu et al. combined Attention-RNNs with a complex network approach [11] to convert market price data into time series price graphs and extract structural information from them for stock forecasting. Other scholars have combined LSTM with SAC model to construct adaptive stock trading strategies, and the results show that they have high annualized returns.

However, financial markets have high-dimensional interdependence, and changes in the price of one stock can have an impact on other stocks associated with it, and the above methods only consider the characteristics of the stock time series itself, ignoring the correlation between the stock prices of different companies.

2.2. Stock Price Prediction Based on Graph Convolutional Neural Networks

With the development of Graph Neural Networks (GNN), scholars have improved the stock price prediction methods by using graph structure to represent the correlations among stocks, and taking the networked transmission paths of stock price fluctuations into consideration. In the following literature, LSTM models and GCN models are combined to forecast the stock market, and the differences lie in the methods of portraying the correlations among stocks, and the selection of metrics for node characteristics and perspectives. Among the stock relationship metrics, knowledge graph is the most widely used method. Wang et al. calculated the correlation values between directly related stocks through industry relationship, conceptual stock relationship, and shareholder relationship to construct the adjacency matrix, and used the stock price features as the node features [13]. Shi et al. constructed the adjacency matrix by considering the large number of stock trades and measuring stock similarity from four aspects: industry, region, concept, and volatility [14]. Ma et al. considered news information and constructed neighbor matrix by calculating cosine similarity through supply chain relationship, equity relationship and industry competition relationship, and used GCN to fuse the news of related companies, and then input it into LSTM model to predict the stock price movement with the news and technical indicators of the target company [15]. Among other methods of stock correlation construction, Xu et al. synthesized different levels of market states and constructed an HGNN model to classify the types of stock stops using industry relationships as the adjacency matrix, which achieved good results [16]. Jafari and Haratizadeh used 23 technical indicators to discriminate classes by QDA, calculated the price discrimination direction accuracy to calculate the mutual influence scores among stocks, and constructed the adjacency matrix by using the influence scores greater than 0 as the entitled edges, and the node features selected a large number of basic indicators and momentum factors [17]. The stock relationship graphs constructed in the above literature are static, Song et al. used the DTW algorithm to construct price relationship graphs among stocks based on opening price, closing price, high price, low price and trading volume, used GCN to train price relationship, industry relationship and Wiki relationship respectively, learned the weights through the adaptive attention mechanism to get dynamic stock relationship embedding, and combined it with LSTM to make stock ranking prediction [18]. The analysis using traditional linear correlation coefficients is no longer suitable for the study of increasingly complex dependencies in the financial market, in order to measure the nonlinear relationship between stocks, Feng et al. used the Detrended Cross-Correlation Analysis (DCCA), which analyzes the power law interrelationships between non-smooth time series, to calculate the correlation coefficients between stock returns as an adjacency matrix and combined with a momentum factor for stock return prediction and recommendation [19].

Copula methods have contributed greatly to the study of complex dependence structures in financial markets.Aas et al. gave the first statistical inference method for Vine Copula [20], which is capable of estimating joint distributions of multidimensional variables, capturing nonlinear dependence and tail dependence among variables, and has found great application in modeling the dependence of high-dimensional variables in the field of finance. However, in studies using Graph Convolutional Neural Networks to predict stock prices, few scholars have combined the R-Vine Copula structure with the conceptual sector to which the stock belongs in order to construct the stock adjacency matrix.

3. METHOD

This section details the construction process of the AGLSTM model proposed in this paper, providing the necessary mathematical definitions and notation. The construction of the model is divided into four parts. First, a Vine-Copula model is used to measure the correlation between the closing prices of stocks, which is combined with the information of the conceptual sector to which the stock belongs to construct the entitled adjacency matrix between stock nodes. Then, the features of each stock are input into the model to extract the stock's own features through the temporal attention mechanism. Next, the price sequences and adjacency matrices of the stocks after being processed by the temporal attention aggregator are input into the GCN model to aggregate the relevant stock closing price information of the target stock. Finally, the obtained sequence is input into the Bi-LSTM model to predict the stock closing price based on the stock closing price history information.

3.1. Vine-Copula Combined with Conceptual Boards to Construct the Adjacency Matrix

3.1.1. Vine-Copula dependence metrics

Vine-Copula can portray the nonlinear, asymmetric dependence structure between variables in the following steps. Firstly, a more accurate marginal distribution is established for the random variables, on the basis of which the dependence between variables is estimated and the optimal binary Copula function is chosen for modeling. Then the dependence structure between the variables is determined by uniting them through the Vine structure.

The first marginal distribution construction: remember $x_{n,t}$ denotes the closing price of stock $n\ (n = 1,2,\cdots,N)$ at the moment of $t\ (t = 1,2,\cdots,T)$, then the logarithmic return of the nth stock at the moment of the stock's closing price at t is:

$$r_{n,t} = \ln\left(\frac{x_{n,t}}{x_{n,t-1}}\right) \tag{1}$$

The logarithmic return series of stock n is $r_n = \{r_{n,1}, r_{n,2}, \dots, r_{n,T}\} \in \mathbb{R}^T$, and the logarithmic return series of N stocks is $r = \{r_1, r_2, \dots, r_N\} \in \mathbb{R}^{T\times N}$. The $ARMA(p,q) - GARCH(1,1) - Skew - t$ distribution model is constructed separately for each stock's log return series:

$$\begin{cases} x_t = \phi_0 + \sum_{i=1}^{p} \phi_i x_{t-i} + \varepsilon_t - \sum_{j=1}^{q} \varphi_j x_{t-j} \\ \varepsilon_t = \sigma_t e_t, e_t \sim SkT(skew_i, shape_i) \\ \sigma_t = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2 \end{cases} \tag{2}$$

The residual sequence after smoothing is obtained, and the residuals are normalized and subjected to a probability integral transform to obtain a sequence that meets the needs of Copula modeling.

Bedford and Cooke proposed the decomposition of multidimensional Copula structures using R-Vine structures to characterize the dependence of multivariate time series [21]. A Vine structure consists of many layers of trees. Each tree contains many nodes and edges, where nodes represent variables or conditional variables and edges represent binary Copula functions describing the dependencies between variables. Generally, the Maximum Spanning Tree algorithm is used to construct an R-Vine, and the core idea of this method is to apply the principle of the strongest dependency to construct the first tree with the strongest dependency. If an R-Vine has n nodes, the degree of dependence between two nodes can be measured by the absolute value of the empirical Kendall rank correlation coefficient (τ) between them. The $n-1$ edges with the largest absolute value of τ are selected and each node is guaranteed to have at least one of these edges connected to it to construct the R-Vine.

If $f(x_1, x_2, \dots, x_n)$ is the joint density function of n variables, then the joint probability density expression of R-Vine-Copula is:

$$f(x_1, x_2, \dots, x_n) = \prod_{k=1}^{n} f(x_k) \times \left[\prod_{i}^{n-1} \prod_{e \in E_i} c_{j(e),k(e)|D(e)} \left(F\left(x_{j(e)} \middle| x_{D(e)}\right) F\left(x_{j(e)} \middle| x_{D(e)}\right)\right)\right] \quad (3)$$

Where e is each edge in E_i connecting the random variables R-Vine-Copula, $e = j(e), k(e)|D(e)$, the nodes at the ends of the edge e are $j(e)$ and $k(e)$, $c_{j(e),k(e)|D(e)}$ is the binary Copula density function, and $x_{D(e)}$ denotes the subvector of the labeling contained in $x = (x_1, x_2, \dots, x_n)$ in the subvector labeled as contained in $D(e)$.

Considering only the estimation results of the first layer of the tree here, the Kendall's tau correlation coefficient, the upper tail dependence coefficient, and the lower tail dependence coefficient are weighted and averaged with a weight of 4:3:3, and the weights of the connecting edges between the stocks are calculated to construct the adjacency matrix A_1.

3.1.2. Conceptual plate dependency measure

If stock i and stock j belong to the same conceptual sector, a connecting edge is added to the two stocks, and the weight of the connecting edge, $e_{i,j}$, is the inverse of the number of stocks belonging to that conceptual sector in the study stock. If stock i and stock j do not belong to any of the same conceptual sectors, there is no connecting edge for the two stocks, i.e., the weight is zero.

$$e_{i,j} = \begin{cases} \frac{1}{num_k}, & i,j \in Concept_k \\ 0, & else \end{cases} \quad (4)$$

Where $Concept_k$ denotes the kth conceptual segment, and num_k denotes the number of stocks belonging to the conceptual segment in the study stocks. If two stocks belong to more than one conceptual segment, the weights of the connected edges are simply summed up to get the adjacency matrix A_2 constructed according to the conceptual segments.

Finally, the adjacency matrices obtained from both methods are weighted and averaged to obtain the final adjacency matrix A:

$$A = 0.3A_1 + 0.7A_2 \quad (5)$$

3.2. Temporal Attention Mechanism

Denote $X_t = \{x_t^1, x_t^2, \dots, x_t^N\} \in \mathbb{R}^N$ as the closing price of all stocks at moment t, and $X = \{X_1, X_2, \dots, X_T\} \in \mathbb{R}^{N \times T}$ as the closing price characteristics of all stocks at all moments. Denote by $\mathcal{X}_h = \{X_{t_0-h+1}, X_{t_0-h+2}, \dots, X_{t_0}\} \in \mathbb{R}^{N \times h}$ the sequence of closing prices of all stocks at the past h moments.

Stock prices vary across time slices, but there is autocorrelation in stock prices, and a time-attention aggregator is used to aggregate information in the time dimension of the stock closing price series [22].

$$E = V_e \cdot \sigma\big(((\mathcal{X}_h)^T U_1)U_2(U_3\mathcal{X}_h) + b_e\big) \tag{6}$$

$$E'_{i,j} = \frac{\exp(E_{i,j})}{\sum_{j=1}^{h} \exp(E_{i,j})} \tag{7}$$

Where $U_1 \in \mathbb{R}^N$, $U_2 \in \mathbb{R}^{N\times h}$ and $U_3 \in \mathbb{R}^h$ are learnable parameters and $V_e, b_e \in \mathbb{R}^{h\times h}$. E is the temporal attention matrix, and the elements $E_{i,j}$ denote the degree of dependence of the closing price of the stock at moment i and moment j. The matrix E' is the time-attention matrix. Softmax normalization of the attention matrix yields the matrix E'. Multiply the input sequence $\mathcal{X}_h$ with it to obtain the stock closing price sequence with temporal attention input $\widehat{\mathcal{X}}_h$, where $\widehat{\mathcal{X}}_h \in \mathbb{R}^{N\times h}$:

$$\widehat{\mathcal{X}}_h = \mathcal{X}_h E' \tag{8}$$

3.3. GCN Convolutional Neighboring Stock Price Information

For the graph $G = (V, E)$, the Laplacian matrix $L = D - A$, the symmetrically normalized Laplacian matrix L and its spectral decomposition are $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = U\Lambda U^{-1} = U\Lambda U^T$. where $I_N \in \mathbb{R}^{N\times N}$ is the unit array, $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix, $D \in \mathbb{R}^{N\times N}$ is the degree matrix, Λ is the diagonal matrix consisting of eigenvalues, and U is the eigenvector.

The Fourier transform of the signal x on the graph is $\hat{x} = U^T x$. Since the inverse of an orthogonal matrix is equal to the transpose of an orthogonal matrix, the Fourier inverse transform is $x = U\hat{x}$. The Fourier transform of a functional convolution is the product of the Fourier transforms of the functions, thus: $g *_G x = U(\widehat{g *_G x}) = U((U^T g)\odot(U^T x))$, where $\odot$ is the product of Hadamard products, i.e., element-by-element products. Noting that the convolution kernel after the Fourier transform is $g_\theta = U^T g$, and writing g_θ in diagonal form is $g_\theta(\Lambda)$, the final convolution formula is shown in Eq. 9, where $g_\theta(\cdot)$ can be any function.

$$g *_G x = Ug_\theta(\Lambda)U^T x = g_\theta(U\Lambda U^T)x = g_\theta(L)x \tag{9}$$

Defferrard et al. proposed a method to reduce the computational complexity by defining the Chebyshev polynomials of the diagonal matrices of the eigenvectors as filters [23]:

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k\big(\tilde{\Lambda}\big) \tag{10}$$

where θ'_k are the coefficients of the Chebyshev polynomial, and the transformation $\tilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I_N$ is done on Λ such that the size of the element is between -1 and 1. $T_k(\tilde{\Lambda})$ is the Chebyshev polynomial that takes $\tilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I_N$. where $T_0(x) = 1$, $T_1(x) = x$, and $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$.

Thus the graph convolution formula can be written as Eq. 11:

$$g *_G x \approx \sum_{k=0}^{K} \theta'_k T_k\big(\tilde{L}\big)\, x \tag{11}$$

Where: $(U\Lambda U^T)^k = U\Lambda^k U^T$, $\tilde{L} = \frac{2}{\lambda_{max}}L - I_N = U\tilde{\Lambda}U^T$.

The stock closing price sequence $\widehat{\mathcal{X}}_h$ with time attention is input to the GCN model, and the information of the surrounding K-order neighbors centered on each stock in the graph is extracted by the convolution kernel, and the output can be obtained after the Relu activation function:

$$\widehat{\mathcal{X}}_{g*h} = Relu\left(g *_G \widehat{\mathcal{X}}_h\right) \tag{12}$$

3.4. Bi-LSTM Capturing Stock Time Trends

Bi-LSTM captures the features of the sequence in two directions, forward and reverse, through two independent LSTM layers, one processing the inputs in chronological order and the other in reverse chronological order. The hidden state h_t and the cell state c_t for each time step of the forward LSTM are computed from Eq. 13 to 18. The hidden state h_t' and cell state c_t' for each time step of the reverse LSTM are obtained by the same equations.

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

$$f_t = \sigma\left(W_f[x_t; h_{t-1}] + b_f\right) \tag{14}$$

$$o_t = \sigma(W_o[x_t; h_{t-1}] + b_o) \tag{15}$$

$$\tilde{c}_t = tanh(W_c[x_t; h_{t-1}] + b_c) \tag{16}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{17}$$

$$h_t = o_t \tanh \odot (c_t) \tag{18}$$

Where: i_t, f_t, o_t are the input gate, forgetting gate and output gate respectively, x_t is the sequence state of the input at moment t, h_t is the hidden state of the sequence at moment t, and c_t is the staging state of the sequence at moment t.

The Bi-LSTM model inputs x_t and h_{t-1} to get the vector $y_t = [h_t; h_t']$ as the output after splicing the hidden states of forward LSTM and reverse LSTM.

Input $\widehat{\mathcal{X}}_{g*h}$ into the Bi-LSTM model and get the model output $\widehat{\mathcal{X}}_{b*g*h}$ after Relu activation function.

$$\widehat{\mathcal{X}}_{b*g*h} = Relu\left(BiLSTM\left(\widehat{\mathcal{X}}_{g*h}\right)\right) \tag{19}$$

3.5. Output Layer

Finally, the stock price sequence that aggregates the temporal and spatial information is passed through the Dropout layer and the fully connected layer to obtain the predicted output $\widehat{\mathcal{Y}}_h$ for the next time window of the stock's closing price.

$$\widehat{\mathcal{Y}}_h = FC\left(Dropout\left(\widehat{\mathcal{X}}_{b*g*h}\right)\right) \tag{20}$$

The structural framework diagram of the model is shown in Fig. 1 and the algorithmic flow of AGLSTM is shown in Algorithm 1:

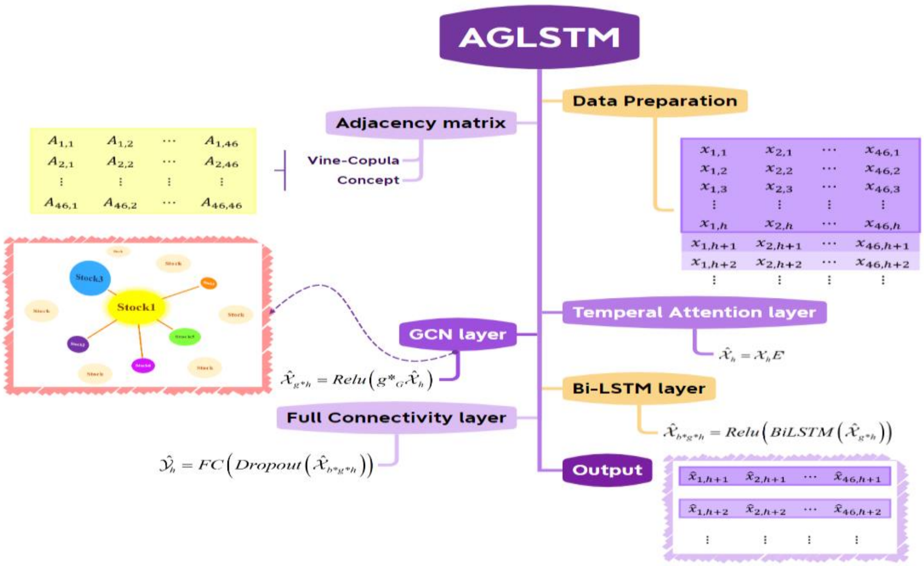

Figure 1. Structure of AGLSTM Model Framework

Table 1. Algorithm 1

Algorithm AGLSTM algorithm
Input: closing prices of 46 stocks for 1214 observation days $X = \{X_1, X_2, \dots, X_T\}$, adjacency matrix A
Output: predicted closing prices of stocks for the next m days $\hat{\mathcal{Y}}_m$
1: The original dataset is repartitioned to generate a new dataset based on the sliding time window length h 2: Initialize global parameters 3: for epoch to n_epochs do
4: for time_step to n_timesteps do
5: According to Eq. 8 the closing price sequence with attention is obtained $\hat{\mathcal{X}}_h = AT(\mathcal{X}_h)$.
6: According to Eq. 12, we get the closing price sequence that aggregates the neighboring stock price information.
7: According to Eq. 19 the closing price sequence containing historical information is obtained $\hat{\mathcal{X}}_{g*h} = GCN(\hat{\mathcal{X}}_h)$ 8: According to Eq. 20 the predicted output of the closing price of the stock is obtained $\hat{\mathcal{Y}}_m$
9: end for
10: Calculate loss, perform backpropagation to update model parameters, save optimal parameters
11: end for 12: Load the optimal parameters to test the test set and output the predictions 13: end

4. EXPERIMENTAL SETUP

In this section, we describe the dataset, the selection of the baseline model, the setting of the model parameters, and the evaluation metrics of the model.

4.1. Data Set

We collect the adjusted closing price data of the SSE 50 constituents of the Chinese stock market from September 3, 2018 to September 1, 2023, excluding four stocks, Trina Solar, Kingsoft Office, China Telecom, and Three Gorges Energy, which were not listed before September 3, 2018. Each of the remaining 46 stocks has 1,214 observations each. The data are obtained from the Wind Financial database. Our goal is to predict stock prices for the next m days based on historical trading data for the past h days. The dataset is divided into a training set (September 3, 2018 to September 1, 2022), a validation set (September 2, 2022 to March 8, 2023), and a test set (March 9, 2023 to September 1, 2023) in the ratio of 8:1:1. For the stock-relationship graph, we have 46 nodes and 140 connected edges are generated based on the information of the conceptual sector to which the stocks belong and the correlation coefficients derived from the Vine-Copula model.

All stocks are first analyzed by descriptive statistics, and Table 2 shows in detail the five stocks with the smallest and five stocks with the largest mean stock prices:

Table 2. Results of Descriptive Statistics for Selected Stocks

Stock code	Mean	Minimum	1/4th Quartile	Median	3/4th Quartile	Maximum	Standard deviation
600010.SH	1.76	1.04	1.33	1.68	2.00	3.93	0.54
601288.SH	2.75	2.53	2.63	2.68	2.79	3.65	0.21
600028.SH	4.02	3.12	3.56	3.82	4.18	6.68	0.79
601398.SH	4.26	3.81	4.10	4.20	4.39	4.97	0.21
600050.SH	4.61	3.22	3.95	4.74	5.23	6.86	0.80
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
603501.SH	118.52	20.48	80.07	109.81	165.83	251.87	61.05
601888.SH	162.28	46.79	85.58	178.49	213.42	381.00	77.96
600809.SH	178.28	21.59	60.46	216.62	275.88	365.98	107.79
600436.SH	235.69	70.11	112.62	261.09	308.07	485.80	110.34
600519.SH	1459.25	486.48	1064.64	1647.98	1806.17	2479.11	458.16

As can be seen from Table 2, the numerical characteristics of the closing price series of different stocks vary greatly. The stock with the largest mean and standard deviation is Guizhou Moutai (600519), and the smallest stock is Baosteel (600010), with a difference of 1,457.49 yuan in the mean and 457.62 yuan in the standard deviation. In addition to Guizhou Moutai (600519), the average value of the daily closing price of other stocks did not reach 1,000 yuan.

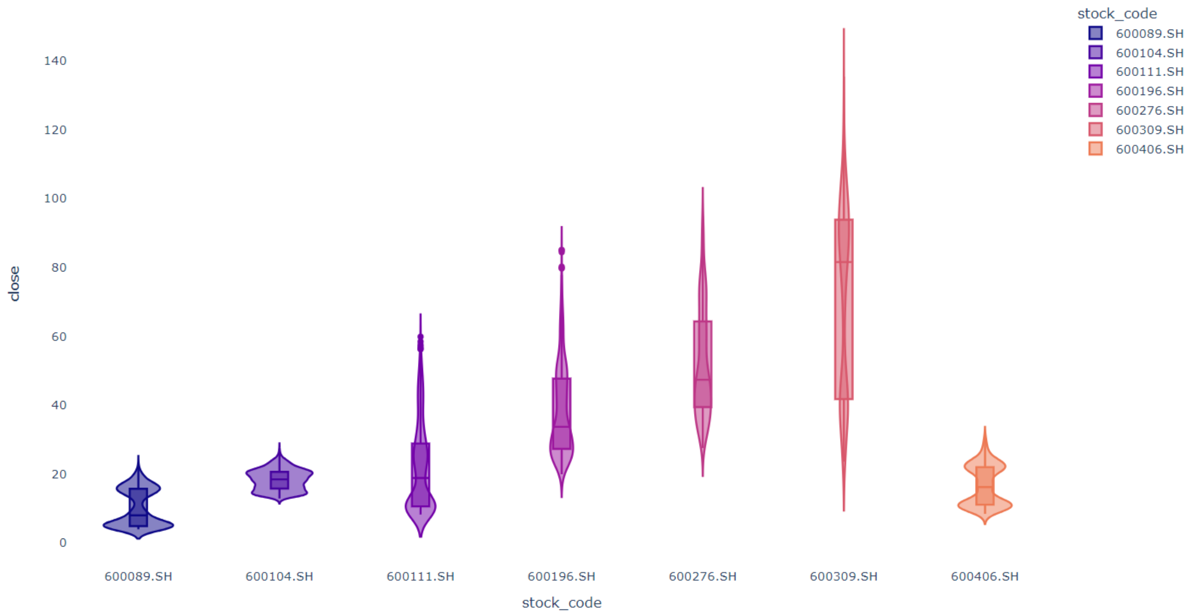

Figure 2. Violin Diagram

We have excerpted seven stocks to draw a violin chart as shown in Fig. 2, which shows that the closing prices of the stocks generally show a bimodal distribution. There are large differences in the closing prices of different stocks, but some stocks have more similar violin plots, such as Guodian Nanrui stock (600406) and TBEA stock (600089) shown in the figure.

The correlation relationship among stocks constructed based on the conceptual sector relationship with R-Vine-Copula method is shown in Fig. 3. Among them, a connecting edge between the nodes indicates that two stocks are related to each other, and the thicker the line, the greater the degree of correlation between the stocks measured by our method, such as Guodian Nanrui stock (600406) and TBEA stock (600089), and Guangyi Innovation stock (603986) and Weier stock (603501). The absence of connecting edges between nodes indicates that our method considers no significant association between stock nodes.

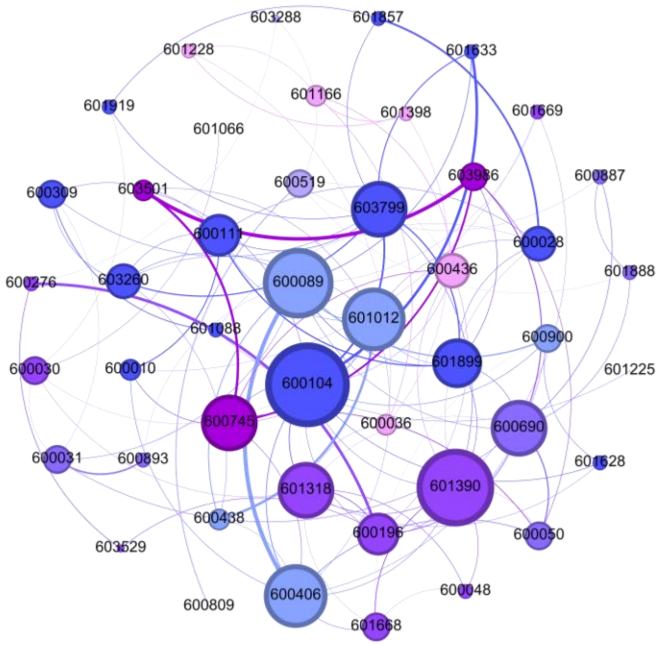

Figure 3. Stock Correlation Chart

According to the results of the stock correlation display, we take Guodian Nanrui stock (600406) as an example, and select TBEA stock (600089), which is connected to it, and Poly Development stock (600048), which is not connected to it, respectively, to plot the time series graph as shown in Fig. 4:

Figure 4(a). Time Series Plot of Stock Prices with Connected Edges

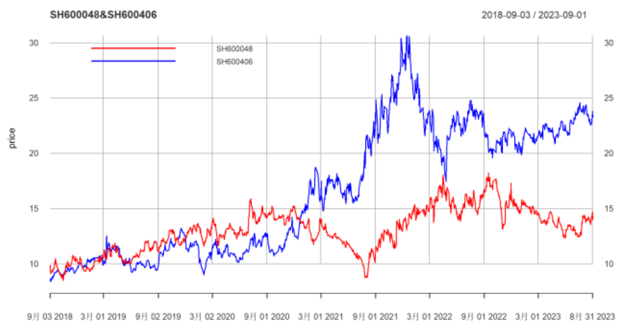

Figure 4(b). Time Series Plot of Stock Prices without Connecting Edges

Fig. 4(a) shows the closing price time series of TBEA and GDNR stocks, while Fig. 4(b) shows the closing price time series of Poly Development and GDNR stocks. It can be seen that the price movements of TBEA and Guodian Nanrui are more similar, while the price movements of Poly Development and Guodian Nanrui are not so much related. It is believed that the closing price trend of TBEA stock can provide some information for the prediction of Guodian Nanrui stock, and at the same time, it verifies that our adjacency matrix construction is more reasonable.

4.2. Baseline Model

To compare and evaluate the models proposed in this paper, we use five models, LSTM, Transformer, GCN, STGCN, and ASTGCN, as comparison models for prediction on the dataset.

4.2.1. LSTM [24]: input stock price series data into LSTM model to predict stock future price.

4.2.2. Transformer [25]: Input stock price series data into the Transformer model to predict the future price of the stock.

4.2.3. GCN [26]: Input the time series with adjacency matrix into the GCN model to predict the future price of the stock.

4.2.4. STGCN [27]: A spatial approach based spatio-temporal graph convolution model, a variant of the GCN model. The time series with the adjacency matrix are fed into STGCN to predict future stock prices.

4.2.5. ASTGCN [22]: Spatio-temporal graph convolutional network based on spatio-temporal attention, a variant of the GCN model. The time series and neighbor matrix are input into the ASTGCN model to predict future stock prices.

4.3. Experimental Parameter Setting

We normalize the closing price of each stock separately so that it lies between 0 and 1. The experimental environment is Pytorch and the Adam optimizer is used for optimization. We set the learning rate to 0.001, the batch size to 32 for 200 rounds of training, and the length of the sliding time frame h to 10 to predict the closing price of a stock for $m = [1,2,3,4,5]$ days, respectively. The Chebyshev polynomial order is set to 3, the number of hidden layers of the GCN module and the number of hidden layers of the Bi-LSTM module are both set to 64, the random seed is set to 42, and the model loss function is set to the L1 loss.

4.4. Assessment of Indicators

In order to accurately predict the prices of stocks, we choose the Mean Absolute Error (MAE), the Mean Square Error $(RMSE)$, and the Mean Absolute Percentage Error $(MAPE)$ to evaluate the efficiency of the model.

MAE which is the average of the absolute value of the difference between the predicted and true values of the closing prices of all stocks, is used to assess the absolute difference between the true and predicted values of all stocks on each trading day. The smaller its value, the better the model performance.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{21}$$

$RMSE$, which is the open-root mean square error of the predicted and true values of the closing prices of all stocks. The smaller its value, the better the model performance.

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

$MAPE$, which is the average of the absolute value of the difference between the predicted and true closing price of all stocks as a proportion of the true value. The smaller its value, the better the model performance.

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

5. EXPERIMENTAL RESULTS AND ANALYSIS

5.1. Overall Performance

The order of the Chebyshev kernel of the AGLSTM model is set to $K = [1,2,3,4,5]$, and the model is predicted one step forward, and the results are shown in Table 3:

Table 3. Chebyshev kernel one step forward prediction results for different orders

	K=1	K=2	K=3	K=4	K=5
MAE	1.7151	1.9332	1.6522	2.0241	2.0174
RMSE	4.8375	6.0218	4.5489	6.3214	7.1525
MAPE	0.0369	0.0377	0.0296	0.0369	0.0357

It can be seen that the model performs optimally on all three evaluation metrics when the order of the Chebyshev kernel is 3. Therefore, the Chebyshev order chosen for the graph convolution layer of the AGLSTM model is $K = 3$.

For GCN, STGCN, ASTGCN, LSTM, Transformer with our proposed model AGLSTM, 1-step forward prediction and multiple-step forward prediction of stock prices are performed respectively. The performance of the model is measured in terms of three evaluation metrics, MAE, $RMSE$ and $MAPE$, and the results are shown in Fig. 5 and Table 4.

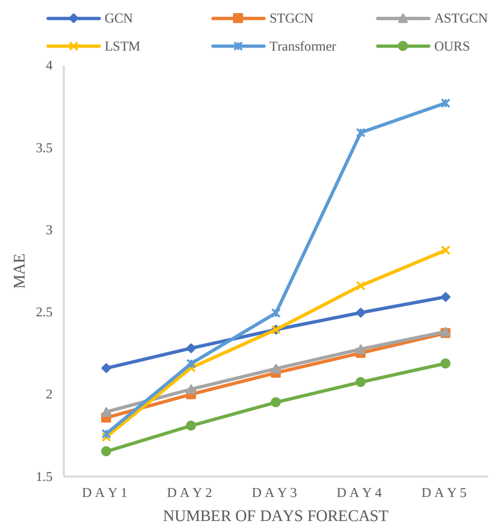

Figure 5(a). MAE Results

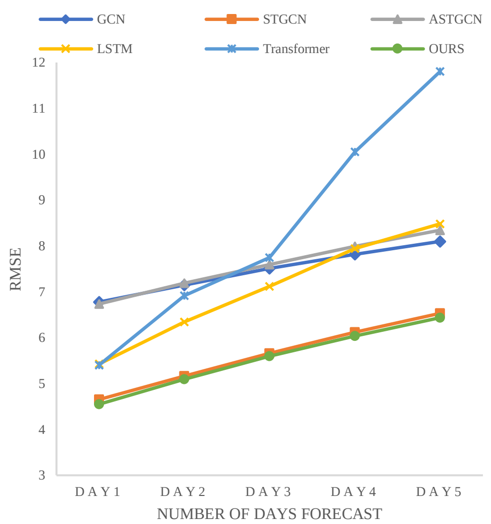

Figure 5(b). RMSE Results

In Fig. 5, the green solid line shows the AGLSTM model. In terms of the average absolute error of MAE, our model achieves the minimum estimation error in all 5 steps forward, and the average absolute error of the model's prediction one step forward is only 1.6522; the STGCN model and the ASTGCN model also have better prediction results, and the model with the worst prediction results one step forward is the GCN model. The LSTM model, which is the most popular model used in time series analysis, and the Transformer model, which is the most novel model, both achieve good accuracy in the prediction of the first step forward, but the prediction effect decreases faster than the other models in the prediction of the third step and the subsequent ones, especially the Transformer model, the average absolute error of the prediction of the fourth day and the fifth day is more than 3.5 percent.

In terms of $RMSE$ root-mean-square error, our proposed AGLSTM model and STGCN model both achieve the best prediction in the prediction of the five steps forward. In the prediction of one step forward, the $RMSE$ error of our proposed AGLSTM model is 4.5489, which is reduced by 0.1025 compared to the STGCN model and 2.2288 compared to the GCN model. The predictions of the GCN model and the ASTGCN model are closer overall, and the LSTM model is significantly better than both of them for the prediction of 1 to 3 steps forward, and is only slightly less effective than the GCN and the ASTGCN model for the prediction of 5 steps forward. The root-mean-square error of prediction of the Transformer model is only 5.4029 at 1 step forward, but reaches 11.8069 at 5 steps forward, indicating that the model performance decreases rapidly with the increase of the number of prediction steps, and the effect is poorer in the long-term prediction.

Since the closing prices of the 46 stocks we selected have large differences, the MAE and $RMSE$ indicators of different stocks also have large differences, and the prediction results are not stable, so we also considered the MAPE indicator, as shown in Table 4:

Table 4. MAPE results for different models

	GCN	STGCN	ASTGCN	LSTM	Transformer	OURS
DAY1	0.0558	0.0511	0.0270	0.0234	0.0294	0.0296
DAY2	0.0570	0.0513	0.0288	0.0283	0.0338	0.0316
DAY3	0.0581	0.0514	0.0306	0.0329	0.0393	0.0333
DAY4	0.0589	0.0512	0.0322	0.0348	0.0694	0.0347
DAY5	0.0590	0.0527	0.0337	0.0393	0.0736	0.0362

From the results of the Mean Absolute Percentage Error ($MAPE$) metric, it can be seen that in the prediction of 5 consecutive steps forward, the ASTGCN model, the LSTM model and our proposed AGLSTM model perform better. In the prediction of 1 step forward, the $MAPE$ of our proposed model is only 0.0296, which means that the difference between the predicted value and the true value of the stock price is 2.96%, which is only 0.0061 higher than that of the LSTM model and 0.0026 higher than that of the ASTGCN model, which are both smaller differences. The worst prediction performance was for the GCN model and the STGCN model, both of which had an average absolute percentage error of more than 0.05.In the forward multi-step prediction, the LSTM model starts to underperform our model in the 4-step forward and 5-step forward predictions, while the Transformer model shows a large increase in the $MAPE$ results from the 4-step forward prediction, suggesting that the Transformer model has a lower performance in the long term prediction.

Overall, the STGCN model performs better on the $RMSE$ metrics but poorly on the $MAPE$ metrics. The ASTGCN model performs better on the $MAPE$ metrics but significantly underperforms our model on the MAE and $RMSE$ metrics. The LSTM model and the Transformer model have better performance in the forward 1-step prediction, but their prediction effectiveness decreases rapidly with the increase of the prediction step, especially for the Transformer model. While the MAE and $RMSE$ metrics of our proposed AGLSTM model are optimal among the performance of all the models, the $MAPE$ metric is slightly insufficient, but the gap is very small with the better-performing ASTGCN model and LSTM model. Considering the three metrics together, our model has the best performance.

5.2. Ablation Experiments

We conducted ablation experiments to verify the validity of the different components of the AGLSTM model, and the results of the ablation experiments when the AGLSTM model is predicted one step forward are given in Table 5.

Table 5. Experimental results of the AGLSTM model for one step forward ablation

	w/o_GCN	w/o_BiLSTM	w/o_Attention	AGLSTM
MAE	1.6790	2.0282	1.8275	1.6522
RMSE	4.7678	8.2656	6.0022	4.5489
MAPE	0.0301	0.0272	0.0276	0.0296

From the results of the ablation experiments, when removing the GCN, our model has better results on MAE metrics and $RMSE$ metrics, and poor results on $MAPE$. When removing Bi-LSTM, our model has better results on $MAPE$ metrics and poorer results on MAE metrics and $RMSE$ metrics. When removing the attention mechanism, our model performs better on the MAE and $RMSE$ metrics than when removing Bi-LSTM, and better on the $MAPE$ metric than when removing GCN. The AGLSTM model has the best overall performance, which shows that the setup of our model is reasonable.

5.3. Tests of Results

In order to compare the significance of the experimental results of the different models, we performed the Friedman test on the predictions of the different models predicting 1 step forward. The Friedman test is used for the comparison of multiple models and allows for an assessment of the statistical significance of the mean rank difference for each method. We get a p-value of 0.000 < 0.05 for the Friedman test statistic, which indicates that there is a statistically significant difference in the performance of the different models.

Meanwhile, disregarding the trading price, we calculate the log price return earned by buying a stock based on the predicted increase or decrease in the stock price on the next business day. If the price is predicted to rise on the next trading day, the stock is bought and sold the following day. Calculating the average log returns predicted by the different models 1 step forward is shown in Table 6:

Table 6. Average logarithmic rate of return

	True	OURS	GCN	STGCN	ASTGCN	LSTM	Transformer
Average logarithmic rate of return	-0.0467%	-0.0342%	-0.0989%	-0.1079%	-0.0498%	-0.1073%	-0.0513%

The first column in Table 6 represents the average logarithmic return on all stocks bought and sold on the next trading day for all trading days in the forecast period. As can be seen, our proposed model improves the average logarithmic return per trade by 0.0125 percentage points and is ahead of all the compared models.

6. CONCLUSION

Developing new stock relationships and deriving useful information from them is a very rewarding task. In this article, we proposed a new idea for measuring the correlations among stocks in the application of deep learning for stock prediction, which is to use tail correlation coefficients as an aid to construct the adjacency matrix, since stocks with high tail correlation coefficients are more likely to have similar price movements. Our proposed AGLSTM model introduces the attention mechanism and combines GCN with Bi-LSTM, which integrates the correlations in both temporal and spatial dimensions to achieve the prediction of stock prices. The experimental results on real datasets show that the AGLSTM model performs well in both short-term prediction and long-term prediction, which can provide a reference for investors' investment decisions. In future research, stock price prediction can be improved in two ways to achieve better prediction results. First, since our study only considered stock closing price data and conceptual sector information, but not textual information, we can introduce textual information such as news headlines and policy documents into the model based on this study to improve the model performance. Secondly, our construction of the adjacency matrix is static, and in future research, we can dynamically portray the correlations between stocks to adapt to the changes in stock relationships in different periods, so as to improve the prediction accuracy of the model.

ACKNOWLEDGEMENTS

This work was supported by Stock Price Prediction Based on Attention GCN-BiLSTM Modeling, Anhui University of Finance and Economics, Graduate Student Innovation Fund Project". (No. ACYC2023198)

REFERENCES

[1] Roll R. R2 [J]. Journal of finance, 1988, 43(3): 541-566.

[2] Wang Y, Qu Y, Chen Z. Review of graph construction and graph learning in stock price prediction [J]. Procedia computer science, 2022, 214: 771-778.

[3] Ghosh P, Neufeld A, Sahoo J K. Forecasting directional movements of stock prices for intraday trading using LSTM and random forests [J]. Finance research letters, 2022, 46: 102280.

[4] Lohrmann C, Luukka P. Classification of intraday S&P500 returns with a Random Forest [J]. International journal of forecasting, 2019, 35(1): 390-407.

[5] Yun K K, Yoon S W, Won D. Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process [J]. Expert systems with applications, 2021, 186: 115716.

[6] Tang H, Dong P, Shi Y. A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points [J]. Applied soft computing, 2019, 78: 685-696.

[7] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions [J]. European journal of operational research, 2018, 270(2): 654-669.

[8] Wang J, Sun T, Liu B, et al. CLVSA: A convolutional LSTM based variational sequence-to-sequence model with attention for predicting trends of financial markets [J]. arXiv preprint arXiv:2104.04041, 2021.

[9] Wu J, Xu K, Chen X, et al. Price graphs: Utilizing the structural information of financial time series for stock prediction [J]. Information sciences, 2022, 588: 405-424.

[10] Wang T, Guo J, Shan Y, et al. A knowledge graph–GCN–community detection integrated model for large-scale stock price prediction [J]. Applied soft computing, 2023, 145: 110595.

[11] Shi Y, Wang Y, Qu Y, et al. Integrated gcn-lstm stock prices movement prediction based on knowledge-incorporated graphs construction [J]. International journal of machine learning and cybernetics, 2024, 15(1): 161-176.

[12] Ma Y, Mao R, Lin Q, et al. Multi-source aggregated classification for stock price movement prediction [J]. Information fusion, 2023, 91: 515-528.

[13] Xu C, Huang H, Ying X, et al. HGNN: Hierarchical graph neural network for predicting the classification of price-limit-hitting stocks [J]. Information sciences, 2022, 607: 783-798.

[14] Jafari A, Haratizadeh S. GCNET: graph-based prediction of stock price movement using graph convolutional network [J]. Engineering applications of artificial intelligence, 2022, 116: 105452.

[15] Song G, Zhao T, Wang S, et al. Stock ranking prediction using a graph aggregation network based on stock price and stock relationship information [J]. Information sciences, 2023, 643: 119236.

[16] Feng S, Xu C, Zuo Y, et al. Relation-aware dynamic attributed graph attention network for stocks recommendation [J]. Pattern recognition, 2022, 121: 108119.

[17] Aas K, Czado C, Frigessi A, et al. Pair-copula constructions of multiple dependence [J]. Insurance: mathematics and economics, 2009, 44(2): 182-198.

[18] Bedford T, Cooke R M. Probability density decomposition for conditionally dependent random variables modeled by vines [J]. Annals of mathematics and artificial intelligence, 2001, 32: 245-268.

[19] Guo S, Lin Y, Feng N, et al. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting [C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 922-929.

[20] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering [J]. Advances in neural information processing systems, 2016, 29.

[21] Chen K, Zhou Y, Dai F. A LSTM-based method for stock returns prediction: A case study of China stock market [C]//2015 IEEE international conference on big data (big data). IEEE, 2015: 2823-2824.

[22] Li S, Jin X, Xuan Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting [J]. Advances in neural information processing systems, 2019, 32.

[23] Chen Y, Wei Z, Huang X. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction[C]//Proceedings of the 27th ACM international conference on information and knowledge management. 2018: 1655-1658.

[24] Chaolong L, Zhen C, Wenming Z, et al. Spatio-temporal graph convolution for skeleton based action recognition[C]//Thirty-second AAAI conference on artificial intelligence. 2018, 2.