Applied Soft Computing 145 (2023) 110595

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A knowledge graph–GCN–community detection integrated model for large-scale stock price prediction

Ting Wang [a,*], Jiale Guo [a], Yuehui Shan [a], Yueyao Zhang [b], Bo Peng [a], Zhuang Wu [a]

[a] School of Management and Engineering, Capital University of Economics and Business, 121 Zhangjialukou, Huaxiang Fengtai District, Beijing, 100070, Beijing, China
[b] Viterbi School of Engineering, University of Southern California, 1249 w 36th st, Los Angeles, 90007, CA, USA

ARTICLE INFO

Article history:
Received 11 August 2022
Received in revised form 14 June 2023
Accepted 23 June 2023
Available online 6 July 2023

Dataset link: Source Code and Data (Original data)

Keywords:
Large-scale stock price prediction
Knowledge graph
Graph convolutional neural network
Similar stock
Community detection

ABSTRACT

Owing to uncertainty in the stock market, stock price prediction has always been a challenging research hotspot. In recent years, many stock prediction methods have used stock price series and technical indicators as inputs and the time series algorithm to predict, but they often ignore the influence of deeper factors such as the situation of the stock company and current situation of the stock industry. In addition, most of them predict based on small-scale stock datasets with limited characteristics and have certain defects such as bias, poor stability of prediction results, and lack of statistical significance tests on experimental results. To solve these problems, we propose a new method for stock price prediction based on knowledge graph (KG) and graph convolution neural network (GCN) models. First, stock KG is constructed, and the semantic relationships between stocks are described in the form of triples. Second, the correlations between stocks are quantified by fully utilizing their explicit/implicit relationships in the KG. Third, K-means, community detection (CD), and GCNs are merged to obtain accurate clustering results for similar stocks. Finally, the historical prices of similar stocks are used as the input characteristics of the time series models to predict stock price trends. We collect 4684 A-share market stocks in China from 2013 to 2019 and predicted the stock price trends for 762 of them. The experimental results and significance test show that the proposed method achieve the best accuracy, precision, and F1-measure on large-scale stock datasets and have the best stability, proving that the overall prediction effect outperforms that by state-of-the-art methods.

1. Introduction

Owing to the high noise, dynamics, nonlinearity, non-parametricity, and chaos in financial data, stock market prediction has long been an attractive and challenging task. Various prediction methods (for example, technical indicators analysis, machine learning, and deep learning) and feature selection (for example, market historical data, listed company fundamental information, and financial news information) have been applied to stock price trend predictions [1–3]. The prediction task aims to construct a relationship model between historical financial data and future stock price trends to obtain more accurate time series prediction results. To the best of our knowledge, these traditional methods have become the mainstream methods for academia and industry to solve the problem of stock price trend prediction, but these prediction algorithms typically use the trading data features of individual stocks as model inputs. However, with the improvement of recurrent neural networks (RNNs), the effect of predicting stock price based on individual stock characteristics is approaching its limit of optimization.

In recent years, the development of KGs and GCNs has inspired us to explore and solve the problem of stock price trend prediction by combining these two methods. First, we construct a stock KG based on the fundamental information of listed companies to express the relationships among stocks. Second, inspired by the price linkage mechanism between similar stocks and complex financial networks, we propose a new method based on the constructed stock KG to obtain similar stocks to improve prediction performance. Finally, using GCNs, we extract the relationship features of stocks and guide time series models to predict stock price trends.

At the same time, the current researches are generally conducted on small-scale stock datasets (generally predicting less than 10 stocks) with sparse correlation between stocks and limited characteristics and have certain drawbacks such as poor prediction stability, fluctuation, and bias on experimental results. Therefore, in this study, we use a large-scale dataset of A-share market in China from 2013 to 2019 to make 762 stock price predictions to validate the proposed model. The experimental results show that our model simultaneously achieves

* Corresponding author.
E-mail address: wangting@cueb.edu.cn (T. Wang).

https://doi.org/10.1016/j.asoc.2023.110595
1568-4946/

optimal accuracy, precision, F1 and stability. The prediction results outperform those by state-of-the-art methods. The main contributions of this study are as follows.

(1) Given the price linkage mechanism between similar stocks, we construct a stock KG based on the information about the company's shareholders, industries, and concept stocks. We map the relationship between stocks onto a more intuitive topological graph structure, introduce the concept of contribution to quantify the distance between stocks, and construct a distance adjacency matrix.

(2) To fully utilize the characteristics of similar stocks to predict the price trends of target stocks, we propose a new method that combines CD with GCNs. First, the GCN is embedded as a stock knowledge node, thereby making the representation of stock relationships more complete. Second, the modularity in CD is used as the loss of stock clustering; error backpropagation can make the output clustering results of K-means more accurate.

(3) This study proposes a novel method to predict target stock prices using the price characteristics of similar stocks. We use the historical prices of similar stocks as a feature to input the multivariate time series model to assist it in learning the price fluctuation trend of the target stock to predict stock prices more accurately.

(4) A small dataset can lead to biased, unstable, or fluctuating predictions. Our model addresses this problem and achieves optimal accuracy, precision, and F1 on a large-scale stock dataset. Its robustness and stability are simultaneously verified by a significance test; therefore, it can provide more stable investment returns for investors.

2. Related work

2.1. Prediction of stock prices based on time series models

Earlier, time series prices were input into the long short-term memory (LSTM) [4] model to predict the stock price. However, the accuracy of solely using LSTM for prediction is unsatisfactory. Therefore, Tsai et al. [5] and Ha young et al. [6] proposed an LSTM and CNN integrated model for prediction by extracting the graphic features and time series fluctuations of stock price trends. To improve accuracy, Chen et al. [7] used the LSTM model to encode historical stock prices and then input them into GCNs to predict the price trend by learning the relationship between stocks.

2.2. Prediction of stock prices based on KGs

Google proposed a knowledge graph (KG) concept in 2012. In essence, it is a knowledge base (KBs) with a digraph structure. A KG is a data structure that comprises entities, relationships, and attributes. It represents knowledge using an RDF and attributed graphs. Compared with traditional expert systems and knowledge engineering, KGs eliminate the traditional method of acquiring knowledge by hand, making the data scale considerably larger. Currently, KGs have been applied in many fields, such as the securities investment, intelligent medical treatment, financial risk control, intelligent question answer (QA) system [8], and recommendation system [9]. In the field of stock price prediction, Anil Berk et al. [10] used the KGs and community aware sentiments to predict the trend of stock prices. They first constructed social KGs, collected comments from celebrities in the industry, and extracted comment data through emotion analysis to predict stock prices. Liu et al. [11] constructed a KG of financial news and embedded it in a TransE model to predict stock prices. Using a deep neural network, Long et al. [12] utilized insensitive transaction records and public market information to predict stock prices. Liu et al. [13] predicted stock price trends using the KG of financial news of well-known companies. The method of integrating KGs is indeed of great help in improving the accuracy and precision of stock price prediction. However, the stock price is affected not only by itself, but also by other associated stocks. Therefore, we propose a new method for predicting stock prices using a GCN combined with a KG to obtain similar stocks in industries.

2.3. Prediction of stock prices based on GCNs

Traditional stock price prediction models typically used the information of stock itself to predict stock prices. In fact, connections always remain among several stocks, and the price of one stock is often affected by the prices of others. Typically, the stock prices of similar industries fluctuate together. Therefore, this connection is considered helpful in predicting stock prices. The relationship between stocks must be extracted from the fluctuation data. Ye et al. [14] constructed a cross effect graph among multiple companies, used a GCN to obtain the relevant characteristics among related stocks, and used the GRU model to predict stock prices. Hou et al. [15] proposed a GCN–LSTM joint framework that used spatial dependence or potential interactions between enterprises to predict stock prices. Matsunaga et al. [16] incorporated the relationship between companies into a stock market prediction model using a GCN to simulate the way in which investors make decisions and proved that customer relationship is an effective predictor. Raehyun et al. [17] proposed a hierarchical graph attention model (HATS) that used multiple graphs and hierarchical attention to represent and selectively aggregate information from different types of relationships.

The combination of GCN with time series models can improve the accuracy of stock price predictions, but most of the prediction datasets used in previous studies are small. To prove that the prediction results are generally effective for the entire market, large-scale stock datasets should be used for verification.

2.4. Prediction of stock prices for large-scale stock datasets

In previous studies, small-scale datasets containing only approximately 10 stocks were often selected for evaluation; this might cause a prediction bias in accuracy. In fact, if one attempts to predict hundreds of stocks on a large-scale dataset, the average effectiveness of the model tends to decrease [18]. Chen et al. [7] believed that the relationship information between some companies is relatively sparse, leading to unstable prediction results for small-scale datasets. Moskowitz et al. [19] examined 58 different futures and currency markets globally and found that the time series momentum strategy (MOM) demonstrated statistically and economically significant profitability across these markets. Yan et al. [20] selected a large-scale dataset of 160 stocks in their model to verify this effect. Feng et al. [21] proved the effectiveness of mean regression index-based model (MR) by predicting 88 stocks. To prove the effectiveness of the prediction model for stocks in different industries, Xu et al. [22] selected 88 stocks from nine industries to verify the improvement in prediction accuracy. Liu et al. [23] proposed a GRU model combined with stock quantitative data and news sentiment to predict 357 stocks in China's A-share market, proving that the KG and news sentiment analysis-based model is effective. Lv et al. [24] noted that prediction results for small-scale stock datasets often lacked a statistical significance test. Therefore, they used large-scale datasets containing 424 and 185 stocks, respectively, to conduct

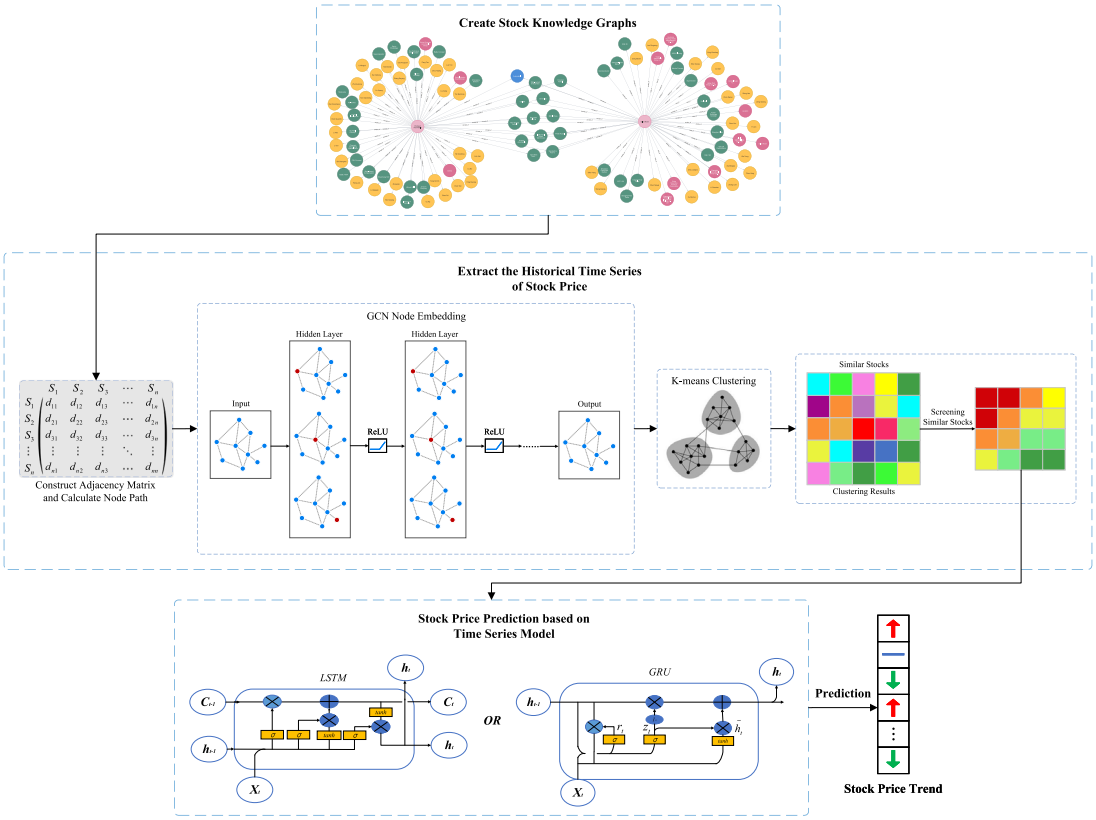

Fig. 1. Overall architecture of our framework.

a significance test for the model effect. Qin et al. [25] proposed a RNN based on dual-stage attention (DARNN) that consisted of an encoder with an input attention mechanism and a decoder with a time attention mechanism. An RNN was used to predict 81 stocks in the NASDAQ-100 stock dataset. Zhang et al. [26] expanded LSTM by decomposing the hidden memory state into multiple components and used multiple frequencies to model potential trading patterns (SFM) to predict the trends of 50 stock prices. Feng et al. [27] designed a new spatio-temporal graph convolutional network (STGCN) model to generate stock relations in a time-sensitive manner (TGC) and verified it on two large-scale datasets containing 3274 stocks on the NASDAQ and 3163 stocks on the New York Stock Exchange. Sawhney et al. [28] used spatiotemporal hypergraph convolutions to learn the temporal evolution in prices and relationships between stocks (STHGCN) and predicted 431 stocks in the S&P500. The results demonstrated the effectiveness of the model.

In conclusion, we select a large-scale dataset containing 4684 stocks to predict 762 stocks to test the significance of performance to prove that our model has better prediction stability than the state-of-the-art methods when facing the large-scale stock price prediction task.

3. System framework

The model proposed in this study is illustrated in Fig. 1. First, a stock KG is constructed that includes the stock, industry, concept stock, shareholder, and stock executive classes. Second, the correlations and distances between stock entities are calculated. Third, an adjacency matrix of the stock distance is constructed. GCNs are used to transfer the adjacency matrix into a vector to complete the node embedding process. Based on an embedded vector with a certain graphic representation ability, K-means is used to cluster the graph-structured stock entities. Through multiple clustering, we obtain stocks that are as accurately similar as possible for a target stock. Finally, the historical prices of the target stock and all its similar stocks in the past 5/10/20 trading days are used as features to input into LSTM and GRU models for training to predict the target stock price trend on the next trading day.

3.1. Construction of the stock KG

We use the Python+Selenium combination crawler to access all stock company pages on the Sina Finance website[1] and extracted stock company information by selecting XPath and element names. The stock company information mainly consists of the following data items: company name, English name, listing market, listing date, stock publishing price, lead underwriter, establishment date, registered capital, secretary of board, company telephone, company fax, company e-mail, company website, postal code, address, office address, company profile, and main business.

We take the relationships of two stocks: Changan Automobile and SAIC Motor, both belong to the industry of Automobile as an example of stock KG in Fig. 2. The stock KG constructed in this paper mainly includes the following stock-related classes and the relationships between their entities: stock, industry, concept stock, shareholder and stock executive.

Entity is described in the form of triples: $\langle Entity, Attribute, Value\rangle$. For example, the stock Changan Automobile entity can be described as $\langle ChanganAutomobile, stockcode, sz000625\rangle$. The attributes of a stock entity include stock ID, stock name, stock code, listing time, industry classification, total market value, total share capital, current market value, and current shares. The attributes of a stock industry include industry ID, index code, and

[1] https://finance.sina.com.cn/stock

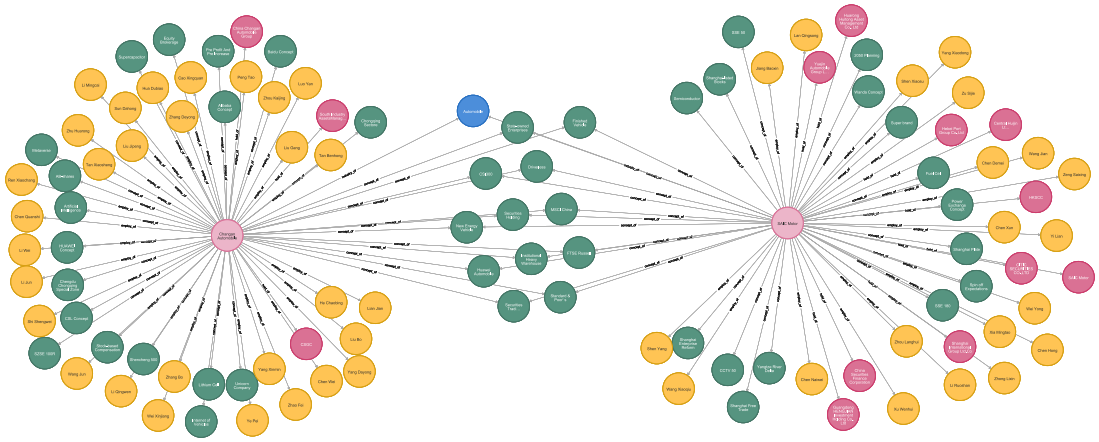

Fig. 2. The stock knowledge graph. Including the following entities: stock (pale pink circle), industry (blue circle), concept stock (green circle), shareholder (pink circle) and stock executive (orange circle).

index names. The attributes of the concept stock include concept stock ID and concept stock name. The attributes of shareholders include the shareholder ID and shareholder name. The attributes of the stock executive include executive ID, name, educational background, age, gender, position, and tenure.

The relationships between entities are described in the form of triples: $\langle head-entity, relationship, tail-entity\rangle$. The relationship between different entities is also different: the relationship between stock entities and stock industry entities is industry_of (belonging to an industry). For example, both SAIC Motor and Changan Automobile belong to Automobile; then, the relationship between stock entity: Changan Automobile and industry entity: Automobile can be expressed as $\langle ChanganAutomobile, industry\_of, Automobile\rangle$. The relationship between stock entities and concept stock entities is concept_of. For example, both SAIC Motor and Changan Automobile belong to CSI 300, SAIC Motor, FTSE Russell, MSCI China, Driverless etc.; then, one of the concept_of relationships between them, such as the concept stock entity: CSI 300 and stock entity: Changan Automobile can be expressed as $\langle CSI300, concept\_of, ChanganAutomobile\rangle$. The relationship between stock entities and shareholder entities is hold_of. For example, HKSCC is one of the shareholders of SAIC Motor; then, the relationship between shareholder entity: HKSCC and stock entity: SAIC Motor can be expressed as $\langle HKSCC, hold\_of, SAICMotor\rangle$. The relationship between stock entities and stock executive entities is employee_of. For example, Wang Jian is the stock executive of SAIC Motor; then, the relationship between stock executive entity: Wang Jian and stock entity: SAIC Motor can be expressed as $\langle WangJian, employee\_of, SAICMotor\rangle$. The attributes of the entity and entity relationships are listed in Table 1 and 2 respectively.

In Table 2, the weight refers to the proportion of the free-float market value of a stock in the total free-float market value of the industry to which it belongs; that is, the greater the proportion, the greater the weight. The weight value reflects the impact of the rise and fall of a stock on its industry index. The weight of each stock is published on the Sina Finance website, and we collect this value using a web crawler. For example, the weight of each stock in CSI 300 is released on a subpage of Sina Finance.[2]

3.2. Refining the similar stocks based on community detection

The relationship in the stock market is complex, and numerous factors affect stock prices. This study builds a KG to further explore the potential relationships between stocks. Based on the KG, we map the deep relationship to a more intuitive topology structure and then define and calculate the correlation value and semantic distance between stocks. Community detection (CD) and GCN are combined to refine the similar stocks of the target stock through multiple K-means clustering that is used to predict the stock price trend using a multivariate time series model. Among them, CD is also called community discovery, which is used to reveal the network clustering behavior, as well as a type of clustering method. A "community" can be defined as a set of nodes with the same characteristics.

3.2.1. Computing the correlation between stocks

Whether it is a common clustering task or a CD task for graph-clustering, the degree of clustering between nodes must be quantified. We call this the correlation between nodes.

A stock KG includes five entities: stock, industry, concept stock, shareholder, and stock executive. Stock executive entities typically refer to people. The situation of an entity with ambiguity is serious, and rarely does one person simultaneously serves as a stock executive for multiple stocks. If multiple stock entities are path-connected through stock executive entities, their corresponding matrices are too sparse to be effectively expressed. Therefore, the relationship between entities of stock executives and stocks is ignored when constructing an adjacency matrix. Every two stock entities may be connected through the industry, concept stock, or shareholders and are not unique; that is, every two stocks can belong to the same concept stock, the same industry, or be held by the same shareholders simultaneously. Therefore, first, the concept of contribution degree for the intermediate connected nodes must be introduced to comprehensively measure the semantic distance between every two stocks. The contribution of the intermediate connected nodes is defined as: By examining the industries, concepts, and shareholder nodes that any two stocks are jointly connected with, the less the number of the stocks are connected with these three types of intermediate nodes, the greater are their contribution and the closer is the relationship between the two stocks.

$$Contribution_{Industry} = \sum (Weight_A + Weight_B) \tag{1}$$

$$Contribution_{Concept} = \frac{2}{degree_{Concept}} \tag{2}$$

$$Contribution_{Shareholder} = \frac{2}{degree_{Shareholder}} \tag{3}$$

When two stocks belong to a same industry, its contribution is calculated using Eq. (1). $Weight_A$ and $Weight_B$ represent the

[2] http://finance.sina.com.cn/qizhi/hs300.html#qz

Table 1
Attributes of an entity in a stock KG.

Entity type	Number of entity	General attributes
Stock	4684	Stock ID, Stock Name, Stock Code, Listing Time, Industry Classification, Total Market Value, Total Share Capital, Current Market Value, and Current Shares
Industry	31	Industry ID, Index Code and Index Name
Concept Stock	456	Concept Stock ID and Concept Stock Name
Shareholder	30227	Shareholder ID and Shareholder Name

Table 2
Entity relationships.

Entity type		Name of relationship	Number of relations	Attribute of relations
Stock	Industry	Industry_of	4156	ID, Weight
Stock	Concept Stock	Concept_of	46022	ID
Stock	Shareholder	Hold_of	40537	ID, Share Proportion

weights of stocks A and B, respectively, in their common industry index. The weight values of the stocks are crawled from the Sina Finance website, as mentioned in Section 3.1. When two stocks share a common concept or are connected to a common shareholder, their contributions are calculated using Eqs. (2) and (3). For any two stocks, $degree_{concept}$ represents the number of concept stocks to which both belong, and $degree_{shareholder}$ represents the number of shareholders holding both stocks. Finally, by adding the contribution values of the three types of intermediate-connected nodes, we obtain the correlation value of every two directly connected stocks. The correlation value between every two of all the 4684 stocks in the KG is calculated using Eq. (4).

$$Correlation\,(A,B) = Contribution_{Industry} + Contribution_{Concept} + Contribution_{Shareholder} \tag{4}$$

In contrast, the path distance is inversely proportional to the correlation value; the closer the path distance, the more likely the stocks will be grouped with the target stock and regarded as similar stocks.

$$Distance\,(A,B) = \frac{1}{Correlation(A,B)} \tag{5}$$

Meanwhile, because the path of two stocks contains other stocks, the distances of each pair of adjacent stocks on the path should first be calculated, and then the distances of all pairs of adjacent stocks should be added to obtain the final path distance.

For example, the fragment information of stocks A and B in the KG is shown in Fig. 3, and their attributes are listed in Table 3. The value between stock A and Industry(Q) indicates that its weight in the Q industry index is 0.1 and the same in H; the weight of stock B in the Q industry index is 0.05 and that in H is 0.12. Shareholder(100) indicates that 100 shareholders hold stocks A and B, and ConceptStock(10) indicates that A and B belong to the same 10 concept stocks. The correlation and distance value between A and B can be calculated as follows.

step 1:

$$Contribution_{Industry} = \sum (Weight_A + Weight_B) = 0.1 + 0.1 + 0.05 + 0.12 = 0.37$$

step 2:

$$Contribution_{Concept} = \frac{2}{degree_{Concept}} = \frac{2}{10} = 0.2$$

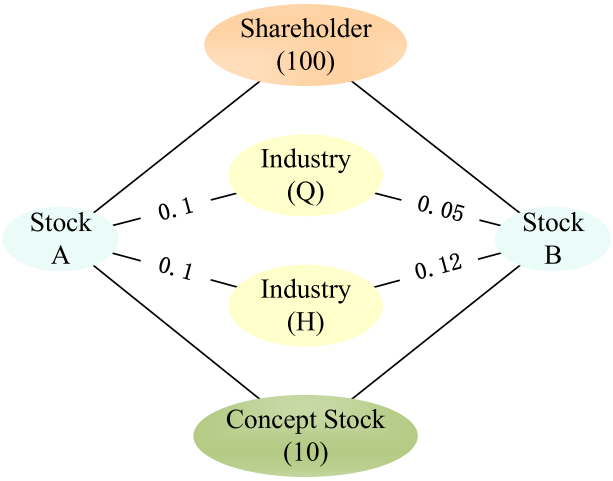

Fig. 3. Calculation process of the correlation value.

step 3:

$$Contribution_{Shareholder} = \frac{2}{degree_{Shareholder}} = \frac{2}{100} = 0.02$$

step 4:

$$Correlation\,(A,B) = Contribution_{Industry} + Contribution_{Concept} + Contribution_{Shareholder} = 0.37 + 0.2 + 0.02 = 0.59$$

step 5:

$$Distance\,(A,B) = \frac{1}{Correlation(A,B)} = \frac{1}{0.59} \approx 1.69$$

3.2.2. Construction of the adjacency matrix for GCN node embedding

A GCN cannot directly understand topological graph structures; it typically recognizes the relationship between the nodes of the topological structure in the form of an adjacency matrix corresponding to the graph. Therefore, the construction of an adjacency matrix is indispensable to help the GCN learn the structure of the stock KG.

The distance between all the stocks can form an adjacency matrix A. The stock is represented as S, and the distance is represented as d. First, we construct an adjacency matrix of all

Table 3
Entity relationships.

Stock	Weight of industry	Shareholder	Concept stock
A	Weight of A in industry Q is 0.1; Weight of A in industry H is 0.1	There are 100 same shareholders as B	Belong to the same 10 concept stocks as B
B	Weight of B in industry Q is 0.05; Weight of B in industry H is 0.12	There are 100 same shareholders as A	Belong to the same 10 concept stocks as A

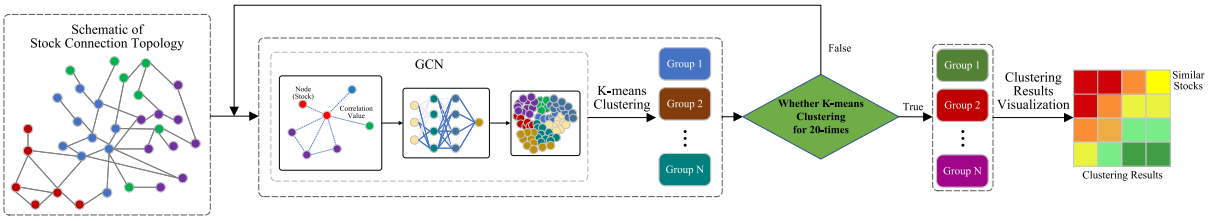

Fig. 4. Community detection model.

4684 stocks in China's A-share market as a complete set. Then, to compare the effect with previous studies, 762 stocks are selected to form the sub-matrix of A for subsequent experiments.

$$A = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array} \begin{array}{c} \begin{array}{ccccc} S_1 & S_2 & S_3 & \cdots & S_n \end{array} \\ \begin{bmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & d_{nn} \end{bmatrix} \end{array} \tag{6}$$

Because the links between stocks in the adjacency matrix are extremely complex and dense, to achieve the desired clustering effect, we ignore the links between stocks that are far away from each other; that is, we delete the links whose distance between two stocks is greater than the threshold. Experiments revealed that when the threshold of distance is set to five, a sparse adjacency matrix and better clustering effect can be obtained.

3.2.3. Community detection model

The community detection model in this paper is divided into two parts:

(1) GCN node embedding
(2) K-means clustering

Our community detection model is illustrated in Fig. 4. First, a GCN is used to convert the adjacency matrix into an embedding of nodes represented by vectors. The distance feature is extracted using the GCN to obtain the embedding of each node, where each node represents a stock. Finally, K-means is used to cluster the embedding results obtained from the GCN. All stocks similar to the target stock can be obtained by integrating K-means and the GCN.

We use a GCN to transform the adjacency matrix into an embedding of vector nodes. Based on the embedding vectors, combined with the K-means algorithm, the goal of node clustering in the graph structure is obtained. GCN can be formally expressed as:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^l W^l) \tag{7}$$

where α denotes the activation function, H denotes the feature of each layer, $H^{(l)}$ represents the feature vector of the node at the lth layer, $H^{(l+1)}$ represents the feature vector of the node at the $(l+1)$-th layer after convolution, and $W^{(l)}$ represents the parameters of convolution at the lth layer. The input is stock distance adjacency matrix A, $\widetilde{A} = A + I$, where I represents the identity matrix, and $\widetilde{D}$ represents the degree matrix of $\widetilde{A}$:

$$\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij} \tag{8}$$

The definition and selection of errors in a model are extremely important. Here, the modularity of the CD task is used as the backpropagation of our model.

$$Modularity = \sum_i (e_{ii} - a_i^2) \tag{9}$$

e_{ii} represents the proportion of edges within community i to the total number of edges in the graph, and a_i represents the proportion of the number of endpoints of the edges connected to community i (equivalent to the sum of the degrees of all endpoints in community i) to the total number of endpoints. The stop condition of the GCN is set to a maximum of 300 iterations, or when the error is no longer reduced during 10 continuous iterations, the model iteration training is stopped.

Because K-means is sensitive to the initial centers and can easily get stuck at a locally optimal value, we set the number of random $seeds$ = [10, 20, 100, 200, 1000] and the range of k = [2,3,4,5]; thus, a total of 20 clustering operations are conducted. The parameters of each cluster are as follows: [$k = 2$, $seed = 10$], [$k = 2$, $seed = 20$], …, [$k = 5$, $seed = 200$], [$k = 5$, $seed = 1000$]. After 20-times K-means secondary clustering for the clustering results of the GCN, we obtain many stocks that belong to the same class as the target stock; this is called the similar stock of the target stock in this study. The purpose of multiple K-means clustering is to obtain similar stocks as accurately as possible because more accurate characteristics of similar stocks can enable our model to achieve better prediction effect.

For example, as shown in Fig. 5, stock sh600004 obtains 80 similar stocks after 20-times clustering: [sz300383, sz300168, sz000100, sz002745, …, sz000089, sh603885, sh600757, and sh600452]. The total number of times that [sz300383, sz300168, sz002987, sz002439, …, sh600060] and sh600004 are grouped together in 20-times clustering is 20. However, the total number of times that [sh603885, sh600757, sz000089, …, sh600452] and sh600004 are grouped in 20-times clustering is only once.

In this study, we select similar stocks that are grouped together with the target stocks for 20-times in 20 clustering results as the feature input for the following time series model because ablation experiments have shown that this strategy can enable our model to achieve the best prediction effect.

Among the 20 clustering results, the daily candlestick chart of five similar stocks grouped with sh600004 for 20-times is shown in Fig. 6, and the daily candlestick chart of five similar stocks grouped with sh600004 only once is shown in Fig. 7. A comparison reveals that the similarity extent of the price trend between similar stocks grouped with sh600004 for 20-times is significantly higher than that of stocks grouped only once. Notably, the more

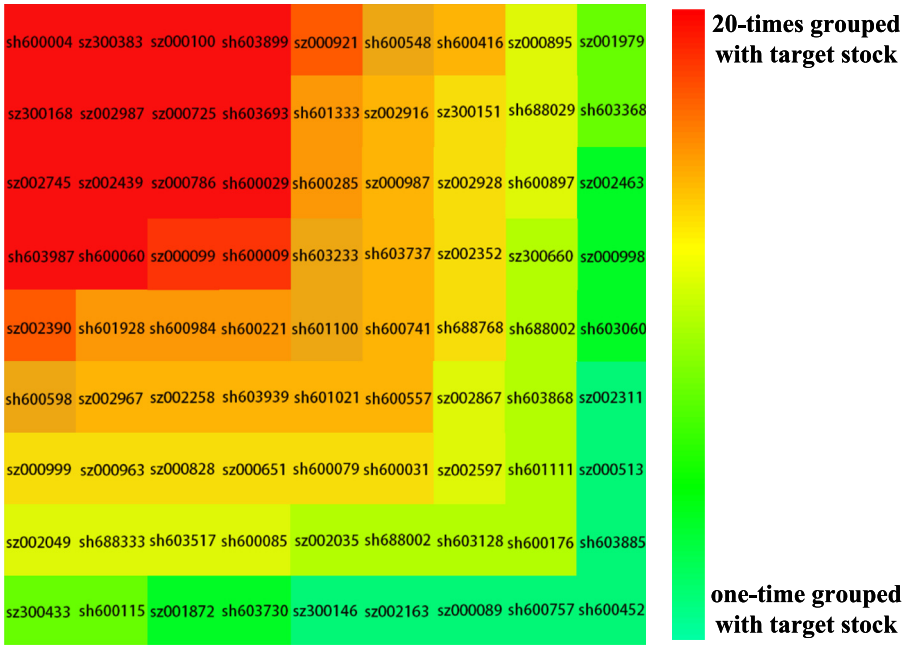

Fig. 5. Heat map—Demonstrating the number of times that similar stocks are grouped with sh600004.

times similar stocks are grouped with the target stock, the higher the similarity between them, and the characteristics provided by similar stocks will be more accurate in the future. Therefore, if only one-time clustering is performed, we may mistakenly select some similar stocks owing to the selection deviation of the initial clustering center of K-means; this will interfere with the subsequent stock price prediction. Therefore, performing multiple K-means clustering to obtain similar stocks more accurately is reasonable.

3.3. Predict the stock price trend based on multivariate time series models

We select the historical prices of different lengths of similar stocks, that is, for the past 5/10/20 trading days, as the multivariate time series characteristics and simultaneously input them into the LSTM and GRU models. The specific information of input historical prices is as follows:

[
target stock: [open price, high price, low price, close price],
similar $stock_1$: [open price, high price, low price, close price],
similar $stock_2$: [open price, high price, low price, close price],
……,
similar $stock_n$: [open price, high price, low price, close price]
],

n refers to the number of selected similar stocks corresponding to the target stock.

We use GRU and LSTM simultaneously because they have their own characteristics and can complement advantages of each other; thus, different stocks may have different prediction effects in these two models. Therefore, the selection of these two models is used as a hyperparameter for tuning, and the overall performance of prediction results can be improved.

The LSTM model can be formally expressed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{10}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{11}$$

$$\widetilde{c_t} = \tanh(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{12}$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \tag{13}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{14}$$

$$h_t = o_t * \tanh(c_t) \tag{15}$$

where h_{t-1} denotes a hidden state summarizing all past information up to the $(t-1)$-th trading day. x_t represents the current input, that is, the historical price data of similar stocks obtained in the previous step. x_t and h_{t-1} are connected to a forget gate f_t and an input gate i_t, respectively, f_t determines the amount of the past information to forget, whereas i_t controls the amount of information that must be brought into current hidden state h_t. $\widetilde{c}_t$ represents the update value of the cell state that is determined by x_t and h_{t-1} obtained through a neural network layer. c_t represents an update gate, and i_t controls the features of $\widetilde{c}_t$ that are used to update c_t. Finally, h_t is computed by output gate o_t and cell state c_t, where the calculation of o_t is the same as those of f_t and i_t. W_f, b_f, W_i, b_i, W_c, and b_c are the parameters to be learned. $*$ represents the Hadamard product. $\sigma(\cdot)$ and $\tanh(\cdot)$ represent sigmoid and *tanh* activation functions, respectively.

The GRU model can be formally expressed as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{16}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{17}$$

$$\widetilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{18}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t \tag{19}$$

where h_{t-1} denotes a hidden state summarizing all past information up to the $(t-1)$-th trading day. x_t represents the current input, that is, the historical price data of similar stocks obtained in the previous step. x_t and h_{t-1} are connected to an update gate z_t and a reset gate r_t, respectively, z_t controls the amount of information to be brought to the current hidden state, whereas r_t determines the amount of the past information to forget. Then, h_{t-1} is reset by r_t and generates a memory cell $\widetilde{h}_t$ that adds new information to h_t. Finally, h_t is the memory updated at

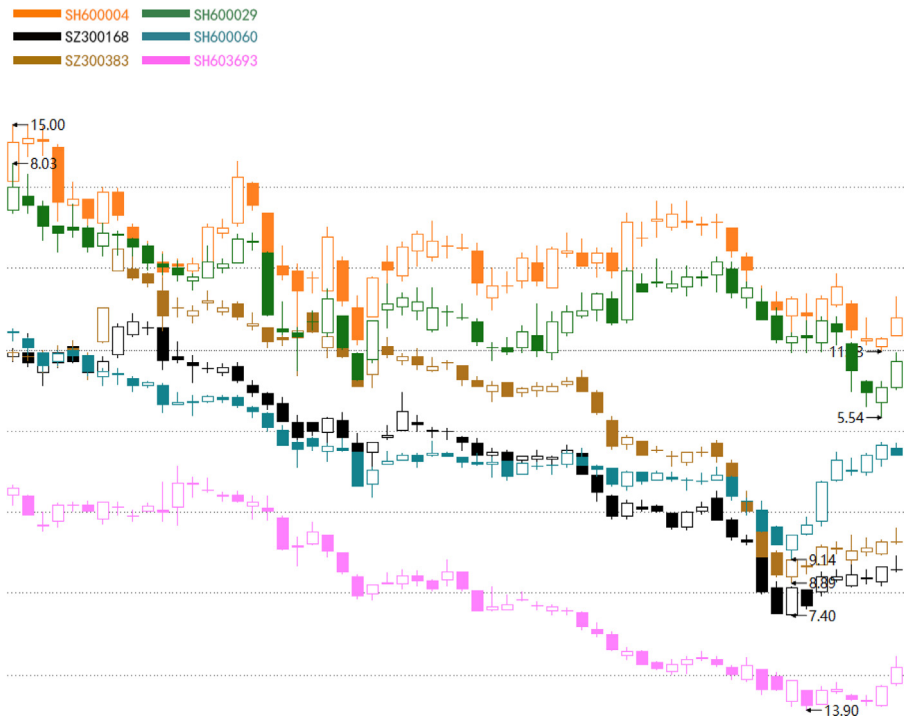

Fig. 6. Daily candlestick chart of similar stocks grouped with sh600004 for 20-times.

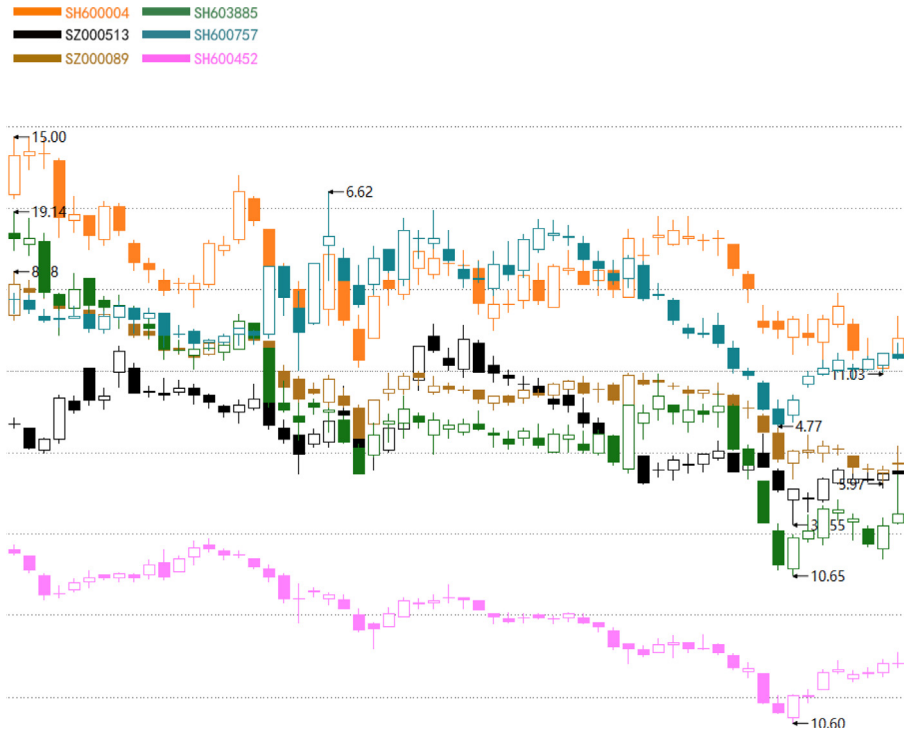

Fig. 7. Daily candlestick chart of similar stocks grouped with sh600004 only once.

time t and is equal to the sum of h_{t-1} and $\widetilde{h}_t$. z_t is used as the balance factor here, where $(1-z_t)*h_{t-1}$ indicates selective "forget" for the hidden state at the previous time, forgetting some unimportant information in h_{t-1} and also discarding the irrelevant information. $z_t*\widetilde{h}_t$ indicates further selective "memory" of candidate hidden states and will forget some unimportant information in $\widetilde{h}_t$, that is, to further select $\widetilde{h}_t$. In summary, h_t is the final memory that will forget some information passed by h_{t-1} and add some information input by the current node. W_z, W_r, and W are the parameters to be learned. $*$ represents the Hadamard product. σ and tanh represent sigmoid and *tanh* activation functions, respectively.

For instance, for all stock prices from 01/04/2013 to 12/31/2019 (containing a total of 1513 trading days), the situation of

the closing price of the next trading day being higher than that of the current trading day is regarded as an upward trend and is marked with tag [1]; otherwise, it is a downward trend, marked with tag [0]. The tag sequence of all real stock price trends during this period can be obtained as [10111... 11111] that has a length of 1513.

Assuming that we want to predict the price trend of a target stock on a certain trading day, we can input the price trend data of the past 5/10/20 trading days of similar stocks as features into the multivariate time series model and then perform training and prediction. In this study, the historical stock price is split into a 9:1 ratio for training and validation, respectively. The prediction result sequence of the stock price trend is represented by tags [0] or [1].

4. Experimental setup

4.1. Data collection

To conduct experiments on a large-scale stock dataset, we adopt the same methodology as in Raehyun et al. [17]. We use the open-source financial tool AkShare and obtained historical prices from 762 A-share market stock prices in China from 01/04/2013 to 12/31/2019 (filtering out newly listed stocks and deleting delisted stocks). Five numerical features are extracted from each stock: open price, high price, low price, close price, and trading volume.

4.2. Evaluation metrics

Stock price prediction is considered a binary classification problem, and we select a few metrics to evaluate the performance of our model, including accuracy, precision, recall, F1-measure, and AUC. In this study, positive classes with upward stock trend are marked as "1" and negative classes with downward stock trend are marked as "0".

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{20}$$

$$Precision = \frac{t_p}{tp + f_p} \tag{21}$$

$$Recall = \frac{t_p}{t_p + f_n} \tag{22}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{23}$$

t_p represents the true positive class, t_n represents the true negative class, f_p represents the false positive class, and f_n represents the false negative class.

$$AUC = \frac{\sum(p_j, n_j) p_i > n_j}{P * N} \tag{24}$$

where P denotes the number of positive classes, n denotes the number of negative classes, p_i denotes the prediction score of positive classes, and n_j denotes the prediction score of negative class.

4.3. Baseline

In this study, the following state-of-the-art methods are used as baselines to compare and evaluate the performance of our model.

- MOM [19]: Uses time series momentum effects to predict stock price trends.
- MR [21]: The mean regression index is used to predict the current stock price trend in the direction opposite to that of the past average price.
- LSTM [4]: Input the time series prices into LSTM model to predict stock price.
- DARNN [25]: An improved RNN model with the dual-stage attention that consisted of an encoder and a decoder mechanisms.
- SFM [26]: An extension of an LSTM that decomposes hidden memory states into multiple components.
- GCN [7]: Uses an LSTM to encode historical stock prices and input the results into a GCN.
- TGC [27]: A new STGCN model is proposed to generate stock relations in a time-sensitive manner.
- HATS [17]: A new type of hierarchical attention network is proposed to selectively aggregate different types of relational data.
- STHGCN [28]: Uses hypergraphs to model industry affiliation relationships of stocks and imposes gated temporal convolution to capture the time series of stock price features.

4.4. Primary parameters of our model

For the baseline, we set up the optimal hyperparameter based on the advice provided by the authors in the original paper to reach the optimal results. The source code and data for our model have been published on GitHub.[3]

We set the random seed in the model as [10, 20, 100, 200, 1000] and the hidden size from the time series model as [32, 64, 128] with the Adam optimizer [29] in the neural network training process. We set the batch_size to [8, 16, 32, 64, 128], learning rate to [10^{-2}, 10^{-3}, 10^{-4}], weight_decay to [10^{-2}, 10^{-3}, 10^{-4}], and the number of iterations to 100.

The experimental environment is configured with an NVIDIA RTX 3080Ti GPU*4, Core i9-12900K, and 128 GB of memory for model training and testing.

To predict the prices of large-scale stocks, we parallelize the model and improve its efficiency by increasing the number of GPUs. For example, in the following experiment of prediction performance evaluation on large-scale stock datasets; that is Section 5.1, we adopt a cluster consisting of 32 such servers (totaling 128 GPUs) using historical price of past 5/10/20 trading days of stocks to trial. In these three groups of experiments, the average total time spent on training, tuning and predicting for each stock is about 53 s. Finally, in the current cluster environment, it may take approximately 11 h to train, tune and predict all the 762 stocks. In this paper, the execution time of other experiments is close to that of this experiment.

5. Evaluation results and analysis

For every baseline compared with the model proposed in this study, we repeat the training and testing many times and obtain the average performance to reduce the fluctuation caused by randomness. Through these experiments, we address the following research questions.

RQ1: On large-scale stock datasets, does the prediction performance of our model outperform that of the baselines (state-of-the-art methods)?

RQ2: Is the prediction performance of our model stable over a series of trading days?

[3] https://github.com/Gjl12321/KG_GCN_Stock_Price_Trend_Prediction_System

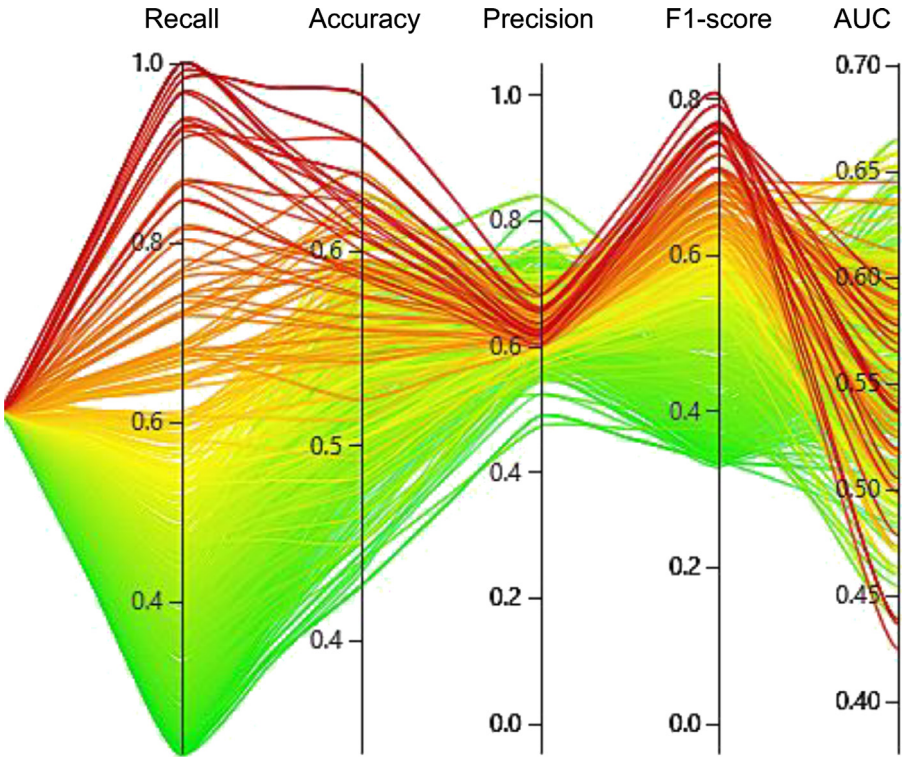

Fig. 8. NNI parameters tuning procedure.

RQ3: What impact do the different k-values of K-means clustering have on our model?

RQ4: What impact will the different choice from similar stocks have on our model?

RQ5: What impact will the choice of two typical time series algorithms have on our model?

5.1. Prediction performance evaluation on large-scale stock datasets

We analyze the performance of various methods considering different prices in different time ranges, including the past 5/10/20 trading days. The stock trend depends on the historical data within the past one or two weeks, or within the past month.

The parameter tuning procedure is illustrated in Fig. 8. For the time series model, hyperparameters must be tuned to the optimal level to generate better performance. In addition, the input features of the time series algorithms differed because we select different similar stocks for each target stock. Therefore, we adopt Neural Network Intelligence (NNI), an automatic tuning tool, to individually tune each target stock. Notably, the tuning process stabilized the prediction of our model. This is because non-optimal hyperparameters might cause exceptions in the time series algorithm (indicated by the green and red lines in the graph), whereas a more precise prediction can be generated by tuning the hyperparameters (indicated by the yellow line in the graph).

In this experiment, we select similar stocks that are grouped with the target stock for 20-times out of 20 clustering tests and input them as features for the time series model. For comparison with the baseline methods, we use multiple experiments to take means of the results for summarize. For example, predicting the prices of 762 stocks for the following trading day using the prices of the past five days; the results of this are shown in Table 4, sorted in reverse order by accuracy indicator.

Accuracy and precision are the two most important metrics for stock price prediction because they can directly determine whether investors can make optimal decisions and eventually make a profit. As shown in Table 5, our method significantly outperforms the state-of-the-art models in terms of accuracy and precision in predicting the 762 stock price trends for the following trading day on large-scale stock datasets. Therefore, we also obtain the highest F1 value in terms of overall performance. To be more precise with numerical data, our model significantly outperforms the state-of-the-art models, with accuracies of 13.21%, 12.57%, and 12.53% higher than those of the second place, respectively when considering the previous 5/10/20 trading days. In terms of precision, our model also significantly outperforms the second place by 26.36%, 25.12%, and 26.44%, respectively, along with excellent F1-measure of 3.09%, 1.17%, and 0.67%, higher than those of the second place. However, our model has more of a downtrend in the recall indicator; this is affected by the bear market phase in China from 2013 to 2019, with the number of falling stocks generally being greater than the number of rising stocks. As aforestated, our model has the highest accuracy and precision and a high probability of correctly predicting rising stock trends. In conclusion, compared with the state-of-the-art methods, our model generates the best overall performance when predicting large-scale stock price trends.

5.2. Stability verification on some trading days

From late April to early May 2018, the Chinese stock market was a bear market. We select the prediction accuracy of seven trading days for comparison with the state-of-the-art models and use the historical price series of similar stocks in the first five trading days as the characteristics of the time series algorithms for prediction. In this experiment, we select similar stocks that are all 20-times grouped with the target stock out of the 20 clustering results as the input for the time series models.

As presented in Table 6, our model performs well on some trading days; although the accuracy on one trading day is not the highest, the overall prediction accuracy of our model over the

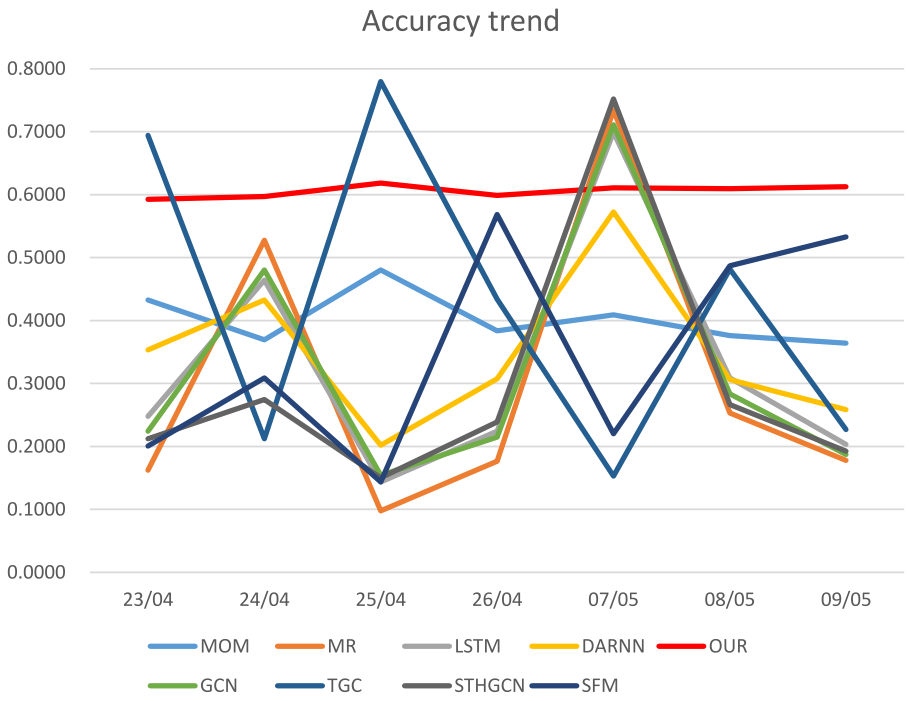

Fig. 9. Compared with the baseline, the fluctuation of accuracy on some trading days.

Table 4
Predicting performance of each 762 stocks using the prices of the past five trading days.

Ranking	Stock code	Accuracy	Precision	Recall	F1	AUC
1	sh600391	0.649	0.6775	0.8021	0.7336	0.6053
2	sz300205	0.64	0.625	0.7692	0.6896	0.6449
3	sz002674	0.6225	0.6526	0.7209	0.6850	0.6343
4	sh600403	0.6184	0.6597	0.7191	0.6881	0.6120
.....	.....	.....	.....	.....	.....	.....
759	sz002613	0.4379	0.6111	0.1222	0.2037	0.5989
760	sh600287	0.4333	0.6666	0.1318	0.2201	0.5703
761	sz000589	0.4313	0.5769	0.1648	0.2564	0.5533
762	sh600731	0.4295	0.5882	0.2197	0.32	0.5445

Table 5
Performance of stock price prediction over different number of past trading days.

Model	5-days accuracy				10-days accuracy				20-days accuracy			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
MOM [19]	34.52%	34.79%	31.88%	33.27%	34.50%	34.94%	32.07%	33.44%	35.73%	35.19%	32.82%	33.96%
MR [21]	35.59%	*39.37%*	33.77%	36.36%	34.73%	29.34%	31.79%	30.52%	35.32%	38.03%	33.60%	35.68%
LSTM [4]	34.92%	35.34%	33.91%	34.27%	35.09%	38.09%	34.37%	35.90%	35.03%	36.43%	34.23%	35.20%
DARNN [25]	37.68%	37.81%	**35.17%**	*36.43%*	*38.89%*	38.59%	*35.22%*	36.82%	38.41%	37.99%	**39.24%**	*38.60%*
SFM [26]	33.29%	26.83%	33.35%	29.52%	34.95%	24.82%	33.34%	28.22%	34.54%	26.93%	33.32%	29.49%
GCN [7]	37.24%	37.23%	33.54%	35.22%	37.44%	39.07%	34.49%	36.62%	37.30%	*39.28%*	34.16%	36.54%
TGC [27]	37.43%	38.28%	34.05%	36.01%	38.42%	*39.35%*	**35.72%**	*37.44%*	37.81%	36.96%	34.49%	35.67%
HATS [17]	38.74%	36.92%	34.29%	35.52%	38.05%	39.23%	34.52%	36.67%	*38.85%*	38.70%	*35.06%*	36.78%
STHGCN [28]	*38.53%*	37.35%	*34.65%*	35.89%	38.81%	36.57%	35.11%	35.75%	38.45%	37.22%	32.82%	34.87%
Our	**51.74%**	**65.73%**	31.29%	**39.52%**	**51.46%**	**64.47%**	31.02%	**38.61%**	**51.38%**	**65.72%**	31.45%	**39.27%**

Table 6
Compared with the baseline, the accuracy on some trading days.

Date Model	04/23	04/24	04/25	04/26	05/07	05/08	05/09
MOM [19]	43.27%	36.94%	48.02%	38.39%	40.90%	37.60%	36.41%
MR [21]	16.23%	*52.77%*	9.76%	17.68%	*73.48%*	25.33%	17.81%
LSTM [4]	24.80%	46.44%	14.38%	22.43%	69.92%	30.87%	20.32%
DARNN [25]	35.36%	43.27%	20.19%	30.74%	57.26%	30.61%	25.86%
SFM [26]	20.05%	30.87%	14.38%	*56.86%*	22.03%	*48.68%*	*53.30%*
GCN [7]	22.43%	48.02%	15.44%	21.50%	71.11%	28.36%	18.73%
TGC [27]	**69.39%**	21.24%	**77.97%**	43.40%	15.30%	48.15%	22.69%
HATS [17]	37.86%	38.26%	31.40%	38.13%	50.26%	44.59%	33.25%
STHGCN [28]	21.24%	27.44%	15.04%	23.88%	**75.20%**	26.65%	19.26%
Our	*59.24%*	**59.68%**	*61.84%*	**59.89%**	61.09%	**60.95%**	**61.24%**

Table 7
Paired samples statistics.

		Mean	N	Std. deviation	Std. error mean
Pair 1	M1-MOM [19]	0.402186	7	0.0420501	0.0158934
	M10-Our	0.605614	7	0.0095738	0.0036186
Pair 2	M2-MR [21]	0.304371	7	0.2355527	0.0890305
	M10-Our	0.605614	7	0.0095738	0.0036186
Pair 3	M3-LSTM [4]	0.327371	7	0.1929691	0.0729355
	M10-Our	0.605614	7	0.0095738	0.0036186
Pair 4	M4-DARNN [25]	0.347557	7	0.1227401	0.0463914
	M10-Our	0.605614	7	0.0095738	0.0036186
Pair 5	M5-SFM [26]	0.351671	7	0.1748171	0.0660747
	M10-Our	0.605614	7	0.0095738	0.0036186
Pair 6	M6-GCN [7]	0.322271	7	0.2021255	0.0763962
	M10-Our	0.605614	7	0.0095738	0.0036186
Pair 7	M7-TGC [27]	0.425914	7	0.2448854	0.092558
	M10-Our	0.605614	7	0.0095738	0.0036186
Pair 8	M8-HATS [17]	0.391071	7	0.0647279	0.0244648
	M10-Our	0.605614	7	0.0095738	0.0036186
Pair 9	M9-STHGCN [28]	0.298157	7	0.2046916	0.0773662
	M10-Our	0.605614	7	0.0095738	0.0036186

Table 8
Paired samples test.

		Paired differences					t	df	Significance	
		Mean	Std. deviation	Std. error mean	95% confidence interval of the difference				One-Sided p	Two-Sided p
					Lower	Upper				
Pair 1	M1-M10	−0.2034286	0.0404054	0.0152718	−0.2407973	−0.1660599	−13.321	6	0	0
Pair 2	M2-M10	−0.3012429	0.2362122	0.0892798	−0.5197027	−0.082783	−3.374	6	0.007	0.015
Pair 3	M3-M10	−0.2782429	0.1938642	0.0732738	−0.4575373	−0.0989484	−3.797	6	0.004	0.009
Pair 4	M4-M10	−0.2580571	0.125789	0.0475438	−0.3743926	−0.1417217	−5.428	6	0.001	0.002
Pair 5	M5-M10	−0.2539429	0.1757222	0.0664167	−0.4164587	−0.091427	−3.823	6	0.004	0.009
Pair 6	M6-M10	−0.2833429	0.2027918	0.0766481	−0.470894	−0.0957917	−3.697	6	0.005	0.01
Pair 7	M7-M10	−0.1797	0.244914	0.0925688	−0.4062077	0.0468077	−1.941	6	0.05	0.1
Pair 8	M8-M10	−0.2145429	0.0661294	0.0249946	−0.2757024	−0.1533833	−8.584	6	0	0
Pair 9	M9-M10	−0.3074571	0.2037641	0.0770156	−0.4959075	−0.1190068	−3.992	6	0.004	0.007

seven trading days is the least fluctuating and the most steady. Furthermore, the standard deviation of our model is the lowest among the state-of-the-art models, as shown in Table 7, proving that its stability is the best. As shown in Fig. 9, the accuracy curve of our model has the smallest fluctuation compared with that of the other models. If the result of the prediction model is unstable, the risk increases, and the investors incur unnecessary losses in stock investment. This is not desired. Hence, the stability of prediction is particularly important. In practice, our model can provide an important reference for investors to make better judgments and decisions by virtue of the stable prediction accuracy and provide more stable return on investment and income to investors. Our model can predict more accurately and steadily in the bear market period such that investors can respond and decide to avoid losses in time.

At the same time, to prove that the prediction performance of our model is significant, we conduct a significance test, that is, paired samples t-test, on the accuracy of all models, as shown in Table 8. According to the results, the one-sided p obtained by the t-test between M1-MOM [19], M2-MR [21], M3-LSTM [4], M4-DARNN [25], M5-SFM [26], M6-GCN [7], M7-TGC [27] M8-HATS [17], M9-STHGCN [28], and our model M10 is less than 0.05. This indicates a significant difference between the accuracy of our model and those of the state-of-the-art models.

5.3. Impact of different k-value of K-means clustering on our model

The characteristic input of the multivariate time series model is provided by the historical prices of many similar stocks; however, K-means clustering relies on the random selection of the initial clustering center. To avoid defects in the K-means algorithm, we use multiple clustering. We randomly select 10 target stocks for the ablation experiment and set k = [2], [3], [4], [5], [2,3,4,5]. Each value of k corresponds to five random seeds: $seed$ = [10, 20, 100, 200, 1000], as presented in Table 9.

In this experiment, we obtain similar stocks when k = [2], [3], [4], [5], and [2,3,4,5] using K-means. Specifically, when k = [2], [3], [4], [5], we select stocks that are grouped with the target stock for five-times in the five clustering results as similar stocks. When k = [2,3,4,5], we select stocks that are grouped with the target stocks for all 20-times out of the 20 clustering results as similar stocks.

Through experiments, we find that when k = [2], [3], [4], [5], and clustering is performed five-times respectively, the final prediction results are relatively poor. This is because using a single value of k and fewer clustering times will lead to extremely few similar stocks or the wrong selection of some similar stocks, resulting in the model being unable to obtain sufficient characteristic information. Therefore, we set k = [2, 3, 4, 5] and $seed$ = [10, 20, 100, 200, 1000] to conduct 20-times clustering to obtain more accurate similar stocks, such that the model can obtain more effective stock prices for training, to improve prediction performance and reduce the negative impact caused by the defects of the K-means algorithm.

5.4. Impact of different choice from similar stocks on our model

As shown in Table 10, we randomly select 10 target stocks and conduct an ablation experiment to select similar stocks by setting the clustering times. We use the following four strategies to select similar stocks:

Table 9
Impact of different k-values of K-means.

The value of k	5-days accuracy				
	Accuracy	Precision	Recall	F1	AUC
k = 2, seed = [10,20,100,200,1000] 5-times clustering	54.38%	64.42%	38.77%	46.83%	0.5966
k = 3, seed = [10,20,100,200,1000] 5-times clustering	54.16%	63.51%	45.02%	51.35%	0.5915
k = 4, seed = [10,20,100,200,1000] 5-times clustering	54.04%	64.20%	40.60%	48.36%	0.6018
k = 5, seed = [10,20,100,200,1000] 5-times clustering	54.01%	64.85%	38.54%	45.28%	0.5964
k = [2,3,4,5], seed = [10,20,100,200,1000] 20-times clustering	55.47%	65.81%	44.87%	51.15%	0.5981

Table 10
Impact of different choice from similar stocks.

Parameters and selection of similar stocks	5-days accuracy				
	Accuracy	Precision	Recall	F1	AUC
(1-time clustering) [k = 3, seed = 20], Select TOP-5 of similar stocks	44.31%	63.64%	9.70%	15.71%	0.5123
(1-time clustering) [k = 3, seed = 20], Select all similar stocks	45.17%	91.66%	5.71%	4.20%	0.5139
(20-times clustering) k = [2,3,4,5], seed = [10,20,100,200,1000], Select TOP-5 of similar stocks	46.51%	77.14%	9.91%	15.64%	0.5479
(20-times clustering) k = [2,3,4,5], seed = [10,20,100,200,1000], Select the similar stocks 20-times clustered with target stock	55.47%	65.81%	44.87%	51.15%	0.5981

Table 11
Impact of choice from two typical time series algorithms.

Time series algorithms /Stock code	GRU 5-days accuracy					LSTM 5-days accuracy				
	Accuracy	Precision	Recall	F1	AUC	Accuracy	Precision	Recall	F1	AUC
sh600132	**60.92%**	66.05%	76.59%	70.93%	0.5796	51.65%	66.66%	21.95%	33.02%	0.4989
sh600391	54.30%	70%	18.18%	28.65%	0.6384	**64.90%**	67.75%	80.21%	73.36%	0.6053
sh600491	**57.61%**	61.73%	78.02%	68.93%	0.5463	45.03%	75%	13.18%	22.42%	0.5135
sh600513	43.70%	67.64%	23.71%	35.11%	0.532	**54.96%**	70.42%	51.54%	59.52%	0.5782
sh600662	**52.63%**	81.48%	24.71%	37.93%	0.6345	49.34%	75%	20.22%	31.85%	0.5585
sh600808	46.71%	72.22%	14.60%	24.29%	0.5996	**55.92%**	59.01%	46.15%	51.79%	0.6058
sh600851	**58.66%**	71.42%	50.56%	59.21%	0.6356	54%	64.10%	31.25%	42.01%	0.5903
sh601939	**61.18%**	65.67%	55%	59.86%	0.6055	55.92%	74%	40.65%	52.48%	0.6083
sz000702	46.71%	57.69%	17.64%	27.02%	0.5187	**57.89%**	66.66%	49.41%	56.75%	0.621
sz000731	**54.60%**	68.11%	50%	57.66%	0.5876	39.47%	52.63%	21.27%	30.30%	0.4559
sz002397	54.30%	71.42%	43.01%	53.69%	0.6056	**61.58%**	64.86%	60%	62.33%	0.6413
sz300246	44.37%	60%	10.34	17.64%	0.5666	**58.94%**	68.88%	39.24%	49.99%	0.6346

(1) Only one-time clustering is performed, and the stocks are ranked according to the number of times they are grouped with the target stock. Because the number of times that all similar stocks grouped with the target stock is all the once, they are sorted alphabetically by the stock code, and the TOP-5 similar stocks are used as the input of time series algorithms.

(2) Only one-time clustering is performed, and all similar stocks grouped with the target stock are selected as the input for the time series algorithms.

(3) In the 20-times of clustering results, the ranking is performed according to the number of times similar stocks are grouped with the target stock, and the TOP-5 similar stocks are used as the input of the time series algorithms. When the times are the same, they are sorted alphabetically by the stock code.

(4) The time series algorithms are input by selecting all similar stocks that are grouped with the target stocks for 20-times out of the 20 clustering results.

The results show that if only one-time clustering is performed, the overall performance of the model was poor. This is because only one-time clustering cause K-means to deviate from the selection of the initial clustering center, resulting in the wrong selection of some similar stocks and thus interfering with the subsequent stock prediction. In addition, if only the TOP-5 similar stocks are selected as the input of time series algorithms, the prediction results still cannot meet expectations. This is because considerably few features are present to satisfy the training requirements of the model. However, the model achieves the best effect by selecting all similar stocks grouped with the target stocks for 20-times as the input of the time series algorithms. This shows that similar stocks with more accurate characteristics can be obtained through multiple clustering. Accordingly, we use this strategy to predict stock price trends.

5.5. Impact of choice from two typical time series algorithms on our model

Our model uses two different time series algorithms, LSTM and GRU, for simultaneous prediction because different stocks will be offered different prediction performances by these two time series models that have their own characteristics and can complement each other. A LSTM has three different gates with many parameters; therefore, training it is difficult. However, a GRU contains only two gates, and when the hyperparameters are all optimized, their performances are equivalent. The structure of the GRU is simpler, requires fewer training samples, and is easy to implement. Therefore, we debug the selection of these two algorithms as a hyperparameter and select one of them with relatively good performance for each stock to predict to achieve better overall performance.

In this experiment, we select the historical prices of similar stocks from the 20 clustering results that are all 20-times grouped with the target stock as the input for the two time series algorithms. Table 11 presents the results of the comparison. Differences can be observed in the accuracy obtained by selecting different time series algorithms for the same stock. For example, the accuracy of stock sh600132 on the GRU is higher than that of

the LSTM algorithm, whereas the accuracy of stock sh600391 on the LSTM model is higher than that of the GRU.

6. Conclusion and future work

Prices of similar stocks affect each other. In this study, we argue that using a KG combined with a GCN to predict stock prices will improve prediction accuracy. We contribute to the construction of a stock KG to map the connections among stocks into a more intuitive topological graph structure and define the distances among stock nodes based on these connections. In addition, we implant CD with a GCN, which uses the modularity in CD as the loss of stock clustering, and reverse error transfer to output more accurate clustering results. Finally, similar stock price features corresponding to the target stock obtain by the clustering algorithm are input to a multivariate time series prediction model to assist the learning process of the fluctuation trend of the target stock and predict the stock price trend more accurately.

We select 762 A-share market stocks in China for the prediction, and the results show that our proposed model yield the highest accuracy and precision. Because we eliminate the bias of the prediction results caused by the size of the stock dataset, the stability is also optimal among the baseline algorithms. In summary, our model outperforms the state-of-the-art models in predicting large-scale stocks.

Stock prices can be affected by various factors and we will continue to further improve our prediction algorithm, such as considering the combination of positive and negative company news, to achieve higher accuracy and confidence in the future.

CRediT authorship contribution statement

Ting Wang: Supervision, Methodology, Writing – review & editing. **Jiale Guo:** Methodology, Writing – original draft. **Yuehui Shan:** Writing – review & editing. **Yueyao Zhang:** Validation. **Bo Peng:** Investigation. **Zhuang Wu:** Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my source code and data at the Attach File step

Source Code and Data (Original data) (Mendeley Data)

Acknowledgments

This work was supported in part by the National Social Science Fund of China (No. 19BXW120); Scientific Research Project of Beijing Municipal Education Commission (General Social Science Project) (No. SM201910038010); Backup Academic Leaders Grant of Capital University of Economics and Business; and Special Fund of Fundamental Research Expenses of Beijing Municipal University of Capital University of Economics and Business (No. ZD202105).

References

[1] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, G. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2017, pp. 2627–2633, http://dx.doi.org/10.24963/ijcai.2017/366.

[2] Y. Wang, L. Wang, F. Yang, W. Di, Q. Chang, Advantages of direct input-to-output connections in neural networks: The elman network for stock index forecasting, Inform. Sci. 547 (2021) 1066–1079, http://dx.doi.org/10.1016/j.ins.2020.09.031.

[3] X. Wu, H. Chen, J. Wang, L. Troiano, V. Loia, H. Fujita, Adaptive stock trading strategies with deep reinforcement learning methods, Inform. Sci. 538 (2020) 142–158, http://dx.doi.org/10.1016/j.ins.2020.05.066.

[4] D.M.Q. Nelson, A.C.M. Pereira, R.A. de Oliveira, Stock market's price movement prediction with LSTM neural networks, in: 2017 International Joint Conference on Neural Networks, IJCNN, IEEE, ISBN: 978-1-5090-6182-2, 2017, pp. 1419–1426, http://dx.doi.org/10.1109/IJCNN.2017.7966019.

[5] Y.-C. Tsai, C.-Y. Chen, S.-L. Ma, P.-C. Wang, Y.-J. Chen, Y.-C. Chang, C.-T. Li, Finenet: A joint convolutional and recurrent neural network model to forecast and recommend anomalous financial items, in: T. Bogers, A. Said, P. Brusilovsky, D. Tikk (Eds.), Proceedings of the 13th ACM Conference on Recommender Systems, ACM, New York, NY, USA, ISBN: 9781450362436, 2019, pp. 536–537, http://dx.doi.org/10.1145/3298689.3346968.

[6] H.Y. Kim, C.H. Won, Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models, Expert Syst. Appl. (ISSN: 09574174) 103 (2018) 25–37, http://dx.doi.org/10.1016/j.eswa.2018.03.002.

[7] Y. Chen, Z. Wei, X. Huang, Incorporating corporation relationship via graph convolutional neural networks for stock price prediction, in: A. Cuzzocrea, J. Allan, N. Paton, D. Srivastava, R. Agrawal, A. Broder, M. Zaki, S. Candan, A. Labrinidis, A. Schuster, H. Wang (Eds.), Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, New York, NY, USA, ISBN: 9781450360142, 2018, pp. 1655–1658, http://dx.doi.org/10.1145/3269206.3269269.

[8] A. Bahrammirzaee, A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems, Neural Comput. Appl. (ISSN: 0941-0643) 19 (8) (2010) 1165–1195, http://dx.doi.org/10.1007/s00521-010-0362-z.

[9] P. Paranjape-Voditel, U. Deshpande, An association rule mining based stock market recommender system, in: 2011 Second International Conference on Emerging Applications of Information Technology, IEEE, ISBN: 978-1-4244-9683-9, 2011, pp. 21–24, http://dx.doi.org/10.1109/EAIT.2011.90.

[10] A.B. Altuner, Z.H. Kilimci, A novel deep reinforcement learning based stock direction prediction using knowledge graph and community aware sentiments, 2021, http://dx.doi.org/10.48550/arXiv.2107.00931.

[11] Y. Liu, Q. Zeng, H. Yang, A. Carrio, Stock price movement prediction from financial news with deep learning and knowledge graph embedding, in: K. Yoshida, M. Lee (Eds.), Knowledge Management and Acquisition for Intelligent Systems, in: Lecture Notes in Computer Science, vol. 11016, Springer International Publishing, Cham, ISBN: 978-3-319-97288-6, 2018, pp. 102–113, http://dx.doi.org/10.1007/978-3-319-97289-3_8.

[12] J. Long, Z. Chen, W. He, T. Wu, J. Ren, An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market, Appl. Soft Comput. (ISSN: 15684946) 91 (2020) 106205, http://dx.doi.org/10.1016/j.asoc.2020.106205.

[13] Y. Liu, Q. Zeng, J. Ordieres Meré, H. Yang, Anticipating stock market of the renowned companies: A knowledge graph approach, Complexity (ISSN: 1076-2787) 2019 (2019) 1–15, http://dx.doi.org/10.1155/2019/9202457.

[14] J. Ye, J. Zhao, K. Ye, C. Xu, Multi-graph convolutional network for relationship-driven stock movement prediction, in: 2020 25th International Conference on Pattern Recognition, ICPR, IEEE, ISBN: 978-1-7281-8808-9, 2021, pp. 6702–6709, http://dx.doi.org/10.1109/ICPR48806.2021.9412695.

[15] X. Hou, K. Wang, C. Zhong, Z. Wei, ST-trader: A spatial-temporal deep neural network for modeling stock market movement, IEEE/CAA J. Autom. Sin. (ISSN: 2329-9266) 8 (5) (2021) 1015–1024, http://dx.doi.org/10.1109/JAS.2021.1003976.

[16] D. Matsunaga, T. Suzumura, T. Takahashi, Exploring graph neural networks for stock market predictions with rolling window analysis, 2019, http://dx.doi.org/10.48550/arXiv.1909.10660.

[17] R. Kim, C.H. So, M. Jeong, S. Lee, J. Kim, J. Kang, HATS: A hierarchical graph attention network for stock movement prediction, 2019, http://dx.doi.org/10.48550/arXiv.1908.07999.

[18] C. Cui, X. Li, J. Du, C. Zhang, X. Nie, M. Wang, Y. Yin, Temporal-relational hypergraph tri-attention networks for stock trend prediction, 2021, http://dx.doi.org/10.48550/arXiv.2107.14033.

[19] T.J. Moskowitz, Y.H. Ooi, L.H. Pedersen, Time series momentum, J. Financ. Econ. (ISSN: 0304405X) 104 (2) (2012) 228–250, http://dx.doi.org/10.1016/j.jfineco.2011.11.003.

[20] Y. Yan, D. Yang, A stock trend forecast algorithm based on deep neural networks, Sci. Program. (ISSN: 1058-9244) 2021 (2021) 1–7, http://dx.doi.org/10.1155/2021/7510641.

[21] F. Feng, H. Chen, X. He, J. Ding, M. Sun, T.-S. Chua, Enhancing stock movement prediction with adversarial training, in: T. Eiter, S. Kraus (Eds.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, California, ISBN: 978-0-9992411-4-1, 2019, pp. 5843–5849, http://dx.doi.org/10.24963/ijcai.2019/810.

[22] Y. Xu, S.B. Cohen, Stock movement prediction from tweets and historical prices, in: I. Gurevych, Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Stroudsburg, PA, USA, 2018, pp. 1970–1979, http://dx.doi.org/10.18653/v1/P18-1183.

[23] J. Liu, Z. Lu, W. DU, Combining enterprise knowledge graph and news sentiment analysis for stock price prediction, in: T. Bui (Ed.), Proceedings of the 52nd Hawaii International Conference on System Sciences, in: Proceedings of the Annual Hawaii International Conference on System Sciences, Hawaii International Conference on System Sciences, 2019, http://dx.doi.org/10.24251/HICSS.2019.153.

[24] D. Lv, S. Yuan, M. Li, Y. Xiang, An empirical study of machine learning algorithms for stock daily trading strategy, Math. Probl. Eng. (ISSN: 1024-123X) 2019 (2019) 1–30, http://dx.doi.org/10.1155/2019/7816154.

[25] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, G.W. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI '17, AAAI Press, ISBN: 9780999241103, 2017, pp. 2627–2633.

[26] L. Zhang, C. Aggarwal, G.-J. Qi, Stock price prediction via discovering multi-frequency trading patterns, in: S. Matwin, S. Yu, F. Farooq (Eds.), Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, ISBN: 9781450348874, 2017, pp. 2141–2149, http://dx.doi.org/10.1145/3097983.3098117.

[27] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, T.-S. Chua, Temporal relational ranking for stock prediction, ACM Trans. Inf. Syst. (ISSN: 1046-8188) 37 (2) (2019) 1–30, http://dx.doi.org/10.1145/3309547.

[28] R. Sawhney, S. Agarwal, A. Wadhwa, R.R. Shah, Spatiotemporal hypergraph convolution network for stock movement forecasting, in: 2020 IEEE International Conference on Data Mining, ICDM, IEEE, ISBN: 978-1-7281-8316-9, 2020, pp. 482–491, http://dx.doi.org/10.1109/ICDM50108.2020.00057.

[29] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, http://dx.doi.org/10.48550/arXiv.1412.6980.