Multimedia Tools and Applications
https://doi.org/10.1007/s11042-022-13231-1

A graph neural network-based stock forecasting method utilizing multi-source heterogeneous data fusion

Xiaohan Li[1] · Jun Wang[1] · Jinghua Tan[1] · Shiyu Ji[1] · Huading Jia[1]

Received: 10 August 2021 / Revised: 12 January 2022 / Accepted: 11 May 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
The study of the prediction of stock market volatility is of great significance to rationally control financial market risks and increase excessive investment returns and has received extensive attention from academic and commercial circles. However, as a dynamic and complex system, the stock market is affected by multiple factors and has a comprehensive capability to include complex financial data. Given that the explanatory variables of influencing factors are diverse, heterogeneous and complex, the existing intelligent algorithms have great limitations for the analysis and processing of multi-source heterogeneous data in the stock market. Therefore, this study adopts the edge weight and information transmission mechanism suitable for subgraph data to complete node screening, the gate recurrent unit (GRU) and long short-term memory (LSTM) to aggregate subgraph nodes. The compiled data contain the metapaths of three types of index data, and the introduction of the association relationship attention dimension effectively mines the implicit meanings of multi-source heterogeneous data. The metapath attention mechanism is combined with a graph neural network to complete the classification of multi-source heterogeneous graph data, by which the prediction of stock market volatility is realized. The results show that the above method is feasible for the fusion of heterogeneous stock market data and the mining of implicit semantic information of association relations. The accuracy of the proposed method for the prediction of stock market volatility in this study is 16.64% higher than that of the dimensional reduction index and 14.48% higher than that of other methods for the fusion and prediction of heterogeneous data using the same model.

Keywords Stock prediction · Graph data · Graph neural network · Multi-source data

✉ Xiaohan Li
365092342@qq.com

[1] School of Economic Information Engineering, Southwestern University of Finance and Economics, 610000 Cheng Du, China

Published online: 27 May 2022

1 Introduction

Stock price trends are nonlinear, unstable time series. In the past 30 years, to make profits in the stock market, investors have continuously studied and forecasted stock prices [15, 25, 44]. Scholars have adopted various transaction data and have derived technical indicators to predict stock market trends [36, 48]. Statistical, economic and other methods have been applied to construct time series predictions [41, 49], and factor pricing models [16, 17] have been used to study stock market fluctuations. For example, Jegadeesh and Titman [23] proposed that a stock price has a trend toward continuing in the original movement direction, and the volume and turnover rate are both momentum factors derived for stock price trend prediction. By using derivative indexes such as the total market value and book-to-market ratio, Fama and French constructed a factor pricing model [18] to interpret the expected rate of return on equity changes in a cross section. However, with the deepening of financial behavior studies, researchers have found that people's financial behaviors in the market are irrational, as investors are affected by cultural, psychological and other factors. Since natural languages such as news events [2, 8, 46, 51], social media [39, 45], and stock bars [24] have arisen, they have become the main indicators in this field of research. In addition, researchers are constantly forming innovative text embedding methods and introducing machine learning algorithms such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for stock market research [12, 13].

With the development of information technology, to obtain more intuitive stock price change trends, graphical indicators have become important quantitative evaluation indicators of the stock market, and graphical indicators such as time-sharing diagrams, average lines, and K-lines have been introduced to evaluate stock price trends. Moving averages, which were proposed by Joseph E. Granville [20], help traders identify existing trends and future trends and detect excessive delays in trends that are about to reverse. K-charts intuitively present the trends of stock prices through patterns, colors, shapes and other elements. Therefore, as an important tool for helping investors in their decision making, K-line charts [6] have been the most widely used approaches. A large number of researchers have adopted K-charts in their studies, and they have mainly applied the time series similarities among K-lines [28] and identified patterns [50] to predict the trends of stock prices. As an important dimension, graphical indicators have been receiving increasing attention. How to integrate graphical indicators with traditional data indicators is also the focus of research.

In recent years, researchers have focused on the processing, integration and analysis of heterogeneous data information in the stock market. For example, Tab et al. [47] proposed using a tensor to replace a series vector for realizing the integration of data information to model market information and applied an event-driven mechanism to balance the heterogeneity of different data types for data prediction. Chai [7] extracted the corresponding features from preprocessed multi-source data and modeled the quantified features by using an extended hidden Markov model to capture the underlying time dependence in the data, which was used for financial time series prediction. Mainstream methods have been used in many studies on the fusion and processing of heterogeneous data in the stock market [26, 31, 52, 53]; however, there are still some limitations.

First, the stock market is a complex dynamic system, and multi-dimensional data sets are the basis for studying stock volatility. The multi-source data in past studies included only trading data and news text data and are not the most widely used graphical indicator data at present. Second, multi-source heterogeneous data embedding quantification methods need to be explored. The multi-source heterogeneous indexes of the stock market ignore the financial characteristics of the indexes. The multiscale problems of transaction data, the financial characteristics of graphical indexes and the traditional quantitative methods of news data mining all have limitations. There is a lack of semantic information in fusion data. In addition, traditional statistical models have limitations for nonlinear and complex data analysis and fail to fully explain the comprehensive impact of multi-source heterogeneous data on the stock market.

To overcome the research limitations mentioned before, this paper proposes a graph fusion neural network method utilizing multi-source heterogeneous data for stock market prediction. This method combines the characteristics of stock evaluation index data, introduces a relationship dimension for data indexes, proposes an embedding method based on graph data, and constructs the subgraph data of the corresponding index data. A convolution operation and graph fusion are carried out on the subgraph to mine semantic information through the weights of the edges and the direction of edge information transmission in the subgraph data, and finally, stock market trend prediction is completed by using fused graph classification. According to the experimental results in terms of the time complexity, accuracy, F1 value and back-test strategy, the proposed MSub-GNN (MultiSubgraph-GNN) method can better predict the trends of stocks through the integration of a graphic index, trading data and stock market news than other approaches. This method is much more advanced than the single-source data index and the traditional heterogeneous data fusion method. The main innovations and contributions of this paper are as follows:

1. This paper builds a more comprehensive data set, including transaction data, news texts and graphical indicators, innovatively introduces relational dimensions for indicator embedding and quantification, fully presents the financial characteristics of indicators, and provides new ideas for the effective information mining of indicators for the stock market.
2. Based on the embedded quantification of a single indicator in the stock market, the study constructs multi-source heterogeneous graph data to achieve data fusion. Different types of indicators for the stock market are represented by heterogeneous nodes, and edges are constructed between heterogeneous nodes to represent the correlation between different types of indicators. It provides a data basis for exploring the interaction between the complex influencing factors of the stock market.
3. The paper effectively realizes the data fusion analysis of the stock market index subgraphs. It creatively proposes that based on multi-source heterogeneous graph data, the graph convolution aggregation operation is applied to realize the vertex aggregation of multi-dimensional subgraph data nodes, and the semantic information of relationships among indicators is fully mined through the metapath, which is combined with the graph neural network. Finally, the paper realizes the classification of complex financial graph data to predict the fluctuation of stock market prices.

This paper is organized as follows. Section 2 reviews the theories involved in this paper and the current research status. Section 3 describes the MSub-GNN stock market prediction model based on multi-source heterogeneous data fusion. Section 4 introduces the experimental process and presents the model training process and experimental results. Finally, Section 5 proposes the next research direction.

2 Relevant work

2.1 Heterogeneous data fusion

Studies on the stock market have already proven that stock price trend prediction is closely related to the characteristics of financial time series [36]. In fact, noisy, nonlinear and random financial time characteristics exist between financial data, and the influencing factors are numerous and complex [32]. However, Edwards R D et al. [14] proved that the trends of financial time series would be repeated, and some special time series trends would appear in the trends of future time series in a very similar way [4]. Historical trading data, such as opening prices, closing prices, maximum prices, minimum prices and trading volumes, directly reflect the changes in the financial market and generate other technical indicators to assist in judging the trends of stocks. The autoregressive (AR), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) models and optimization models use trading time series data for linear analysis [4, 10, 38], and some models, such as RNNs, CNNs and long short-term memory (LSTM), process historical trading data and derivative indexes into tensors for nonlinear analysis and the prediction of stock market fluctuations [19]. Roondiwala et al. [42] used an RNN and LSTM to predict Nifty50 stocks. With the gradually increasing depth of research, text information is being introduced into stock market quantification methods as an evaluation figure, and qualitative text information is being processed from the perspective of news event-driven and sentiment analysis. Atkins et al. [2] built an implicit Dirichlet distribution model and used a naive Bayes method to predict stock market fluctuation directions through financial and economic news. Wei [51] et al. constructed an aggregated news sentiment index of related companies, proving that the level of emotion in news reports can effectively serve as a proxy, which provides valuable support for investment portfolio decisions. Graphical indicators, as important stock market volatility indexes, have also attracted much attention from researchers. Tsai [50] applied image retrieval to extract seven different texture features based on wavelets from candlestick charts, used texture features to retrieve similar historical candlestick charts, and used graphs to retrieve future stock trends for stock prediction. Kusuma [28] et al. input a candlestick graph into a CNN model for training, and the CNN model analyzed and identified the patterns in the candlestick graph and then predicted the future trends of the stock market. The prediction was based on the data of the Taiwan and Indonesia stock markets, and the accuracies were 92.2% and 92.1%, respectively,

In recent years, researchers have studied and explored the integration of multi-source heterogeneous data in the stock market. Zhang [52] proposed a new extended, coupled, and hidden Markov model. This model can effectively integrate news text and historical trading data and predict stock prices. According to their subsequent publication [53], they extracted events and user emotions from network news and social media and realized data integration through a coupling matrix and low-rank matrix decomposed by a tensor framework. Kim [26] et al. proposed a hierarchical graph attention network method that selectively collects information from different relationships, extracts the relationship features of initialized node features, and realizes information fusion for stock market prediction. Li [31] obtained social emotions and professional opinions, created a whole market information space with enterprise characteristics in tensor mode, and applied the tensor learning algorithm to learn the interaction effects of information space on stock trends.

At present, the multi-source heterogeneous data integration method based on stock market indexes transforms data into vectors or tensors, but there is no in-depth exploration of the relationships between index data, and when a machine learning method is used to process the fusion data, it fails to realize the sharing of the trained weight parameters, making the fused heterogeneous data remain in their base form. Hence, this paper builds subgraph data according to the different forms of stock index data and uses different embedding methods to mark the characteristics of index nodes as well as the edges and edge weights between nodes for stock market trading data, graphical indexes and stock market news to realize the efficient fusion of multi-source heterogeneous stock index subgraph data.

2.2 Graph neural networks

To solve the deep learning problem related to graph data, a graph neural network was developed. In merely a few years, Chinese graph neural network technology has developed rapidly and been widely used [21, 40]. In 2013, Bruna et al. [5] first proposed the graph convolutional network (GCN) and defined graph convolution by using the spectral space method. To reduce the space-time complexity, from the perspective of space, Chebnet [11] and the authors of [26] defined a node weight matrix and performed parameterized optimization for the kernel function. With the explosion of the available amount of information and data, data mining has become more complex, and heterogeneity has become more obvious. Graph neural network-related methods have been widely used in biomedical applications, information mining, image processing and other applications and have achieved very brilliant results. In recent years, some researchers in the financial field have also tried to apply relational data to financial research and proposed adopting such methods to predict the stock market. Kim et al. [26] proposed a hierarchical attention network that can be applied to stock market prediction by using relational data. This method can be used to predict individual stock prices and market indexes by selectively aggregating information on different types of relationships and adding this information to the representation of each company. Liu et al. [34, 35] proposed another stock price fluctuation prediction method based on the use of a closed-type regression unit (a gated recurrent unit, GRU) model built from a knowledge graph containing various relationships among the listed companies to predict stock price fluctuations by combining the news sentiments of related stocks and the news sentiments and quantitative characteristics of the stocks of interest. Matsunaga [37] investigated the effectiveness of the cross work between market forecasting and graph neural networks and introduced corporate knowledge graphs into a forecasting model to simulate the decision-making behaviors of investors. The authors also used scrolling windows to backtrack and test the effectiveness of different markets with longer time spreads. Chan et al. [8] included the data of the companies associated with the target company in their analysis by constructing a relationship diagram of the relevant companies and adopted the GCN method for information integration analysis. Although researchers have already investigated and explored stock market prediction using the graph neural network method, this research is still in the initial stage. No other studies have explored the application of the relational dimension of graph data.

3 Model construction of MSub-GNN

3.1 Model introduction

In this paper, researchers build subgraph data with respect to stock market trading data, stock market news, and graphical indicators; consider the characteristics of the subgraph data based on the weights of edges and information transfer directions to adopt the corresponding subgraph convolution process; use the LSTM method to perform node aggregation; and train shared weight parameters during the process of aggregation. A chart is classified to predict the trends of the stock market. As shown in Fig. 1, part ① shows three different types of stock index data for the input model. In part ②, the three types of heterogeneous data are constructed by subgraphs, and node aggregation is carried out by combining the characteristics of various indicators. Specifically, three kinds of edges are set up between the subgraphs to connect the subgraph aggregation nodes and create an edge weight matrix. Model training is used to renew the weight parameter and realize weight sharing for subgraph data fusion, which is employed to characterize the correlations between cross indicators. An embedded vector is generated for the target vertex; this process will be explained in detail in the next section. In addition, part ③ of the figure uses a cross-entropy loss function to train the graph neural network for stock market trend prediction and classification. Figure 1 shows the general framework of the proposed model.

3.2 Construction of graph data

Research on the construction of subgraph data is the key content of this paper. The researchers select trading data, stock market news and graphical indicators to construct subgraph data, and

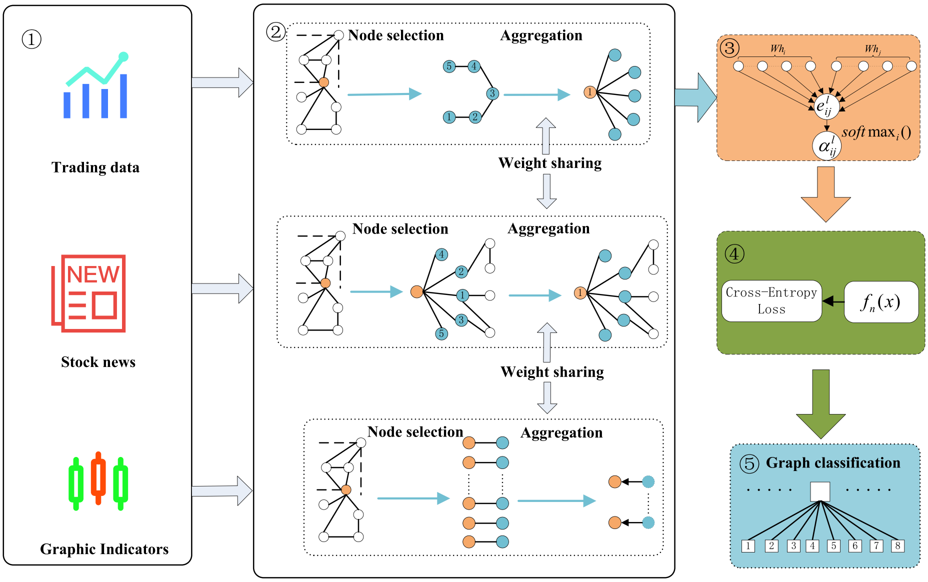


Fig. 1 Model structure diagram (Part ① Input data of three types of stock market indicators. Part ② Three types of subgraph data of the stock market were constructed, and node aggregation was performed by combining the characteristics of various indicators. Parts ③ and ④ A graph neural network was used to predict stock market fluctuation via graph data classification)

these three types of subgraph data compose graph data. As shown in the trading indicator subgraph of Fig. 2(a), five trading days are selected as nodes to construct this subgraph. The node features include six indicators: the opening index, highest index, lowest index, closing index, trading volume, and trading value. According to the continuity of nearby trading days, edges are set between the nodes of nearby trading days. The initial weight of the edge is a random value in [0,1] [27]. The transaction data subgraph node feature is a 5*6 matrix (5 trading day nodes, where each node has 6 features [1]), and the transaction data subgraph edge weight matrix is 5*2 (the number of edges is established between trading days c_5^2). The six node characteristics are the opening index, maximum index, minimum index, closing index, trading volume, and trading quantity of each of the six indicators. In Fig. 2(b) (the stock market news subgraph), the stock market news of each trading day is taken as an indicator subgraph, each news item is taken as a node of the corresponding subgraph, and the associated news text word vector is taken as the node feature. Edges are set between news text, and the news text similarity is taken as the weight of the edge to construct the connection between the news texts. The top 20 groups with the highest edge average weight [22] are chosen as the extracted news subgraph features. The feature is a 100*200 matrix. One hundred is taken as the number of news items in the stock market, the number of news groups and the number of news items in each group, which are set based on the average number of news items in each stock market, and 200 is taken as the 200-dimensional text sentence vector of news headlines. In Fig. 2(c), the node of the graphical index subgraph includes the K-line, a 5-day ecological monetary assessment (EMA) and a 10-day EMA. This paper regresses the graphical indicators to the origin and performs embedding by using the appropriate color and location to effectively

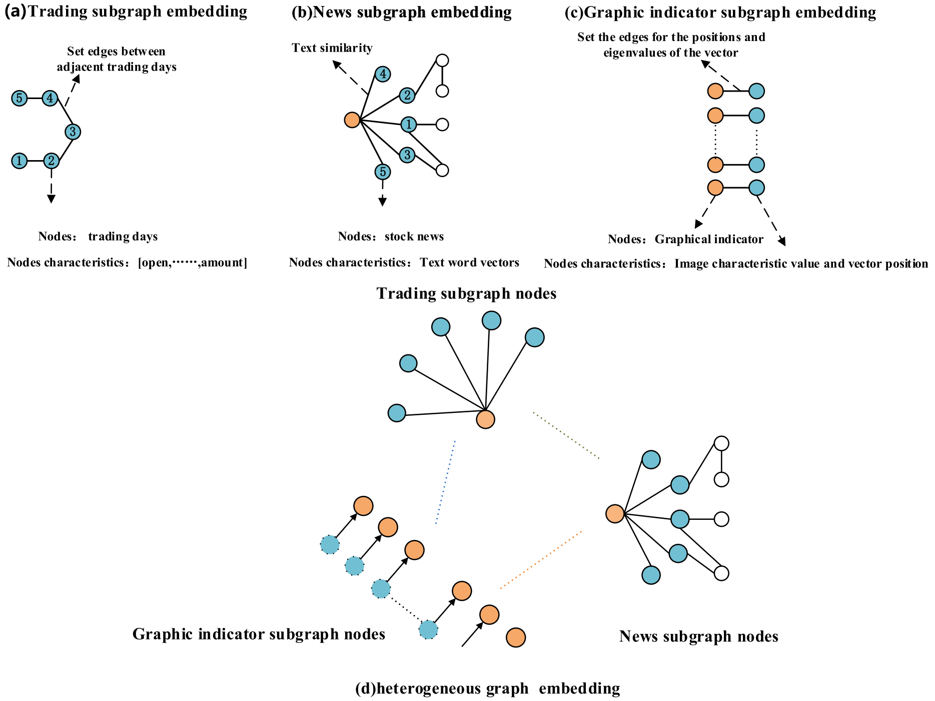

(a)Trading subgraph embedding
Set edges between adjacent trading days
5
4
3
1
2
Nodes: trading days
Nodes characteristics: [open,······,amount]
(b)News subgraph embedding
Text similarity
4
2
1
3
5
Nodes: stock news
Nodes characteristics: Text word vectors
(c)Graphic indicator subgraph embedding
Set the edges for the positions and eigenvalues of the vector
Nodes: Graphical indicator
Nodes characteristics: Image characteristic value and vector position
Trading subgraph nodes
Graphic indicator subgraph nodes
News subgraph nodes
(d)heterogeneous graph embedding

Fig. 2 Graphical data embedding method ((**a**) trading subgraph embedding, (**b**) news subgraph embedding, (**c**) graphical indicator subgraph embedding, and (**d**) heterogeneous graph embedding)

prevent dimensional overlap between the transaction data and the graphical indicator data. The graphical indicators are sliced by following the bit generation rate (BGR) channel, redundant information in the graphical indicators is filtered to remove a larger amount of blank space in the graphical indicators, and then, the indicators are stitched into vectors. The graphical index subgraph is constructed as follows. In reference [3], the MIP solver is called once every 200 nodes. Considering the number of news subgraph nodes, to ensure the balance of heterogeneous index nodes and improve the calculation efficiency, the subgraph is composed of 200 nodes, and each node has 300 dimensional characteristics. The node features in the first half are the feature values of the picture, and the second half of the node features are the positions of the feature values in the vector. An edge is established between each node with a feature value and the node with the corresponding feature value position. The initial weight of the edge is a random value between [0,1]. Due to page limitations, the method for constructing subgraphs and a subgraph data sample are described in this paper in detail in the Appendix 1. The abovementioned three different types of heterogeneous nodes construct corresponding types of edges. The first edge type refers to the construction of trading indicator nodes and stock market news nodes; the second edge type refers to the construction of trading indicator nodes and graphical indicator node edges; and the third edge type refers to the construction of stock market news nodes and graphical indicator nodes. Multisource heterogeneous graph data are constructed for the stock market to realize the integration of heterogeneous subgraphs.

3.3 Subgraph convolution aggregation

To combine each index characteristic and the embedding method, an appropriate node filleting and aggregation method is adopted to preprocess the nodes, as this can effectively improve the efficiency of the subsequent subgraph data fusion and analysis processes. First, the vertex field nodes of each type of subgraph are sampled. To ensure computational efficiency, the number of vertex neighborhood nodes of each type of subgraph is the same. As described in Section 3.2, the vertex neighborhood node of the news text subgraph is the 5 nodes with the largest edge weight. The top 20 groups with the highest average weight of edges [22] are selected as the extracted news subgraph features. The neighboring nodes of the vertices of the graphical indicator subgraph are image feature nodes and corresponding position nodes. The neighbor nodes of the vertices of the trading indicator subgraph are 5 working day nodes. Second, for the sampled neighborhood, the LSTM and GRU methods are mainly used to complete the aggregation process. There are differences in the structures of the two. The simpler structure of the GRU more easily converges, so the smaller data set uses the GRU structure. In contrast, the LSTM unit is used for corresponding data processing. Combined with the node characteristics of the constructed transaction subgraph, the data volume set is not a large-scale volume. In view of the complexity of sentence vector data, the news subgraph adopts the LSTM method for neighborhood node aggregation, while the graphical index subgraph and transaction data index subgraph adopt the GRU method to realize neighborhood node aggregation. Finally, for the aggregated vertices, the attention mechanism graph neural network based on nodes and edges completes the classification and prediction of multi-source heterogeneous graph data. This paper uses edge weights to carry out vertex screening for stock market news index subgraphs and realizes an attention mechanism based on the edge weights. Standard stochastic gradient descent and back propagation techniques are adopted to update the model parameters. The initial weight of each edge is established according to the similarity of the compared news texts. During the model iteration process, the weight of the stock market news graph data edge is updated, as shown in Eq. (1), and the updated edge weight e^l_{ij} is obtained. According to Eq. (2), the edge weights are scored to optimize edge weights so that the model can obtain more accurate neighborhood nodes for vertex embedding. Equation (1) regularizes the edge weight matrix e^l_{ij} to obtain the regularized edge weight matrix α^l_{ij} so that the edge weight matrix can be easily calculated and compared. In Formula (2), the parameterized vector $\overrightarrow{a}^T$ explains the edge-based attention mechanism.W is the weight matrix of node training, and node features are aggregated.

$$\alpha^l_{ij} = \mathrm{softmax_i}\left(\mathrm{e^l_{ij}}\right) \tag{1}$$

$$e^l_{ij} = \mathrm{LeakyReLU}\left(\overrightarrow{\mathrm{a}}^{\mathrm{T}}\left[\mathrm{Wh_i} \parallel \mathrm{Wh_j}\right]\right) \tag{2}$$

The neighborhood nodes selected by edge weights are embedded vertices, and $N(i)$ denotes the largest edge weight among the five connected nodes. The embedding process for the vertex features of the neighborhood nodes is shown in the following equations:

$$h^{t+1}_{xN(i)} = \mathrm{LSTM}\left(\left\{\mathrm{h^t_{xj}}, \forall_\mathrm{j} \in \mathrm{N(i)}\right\}\right) \tag{3}$$

$$h^{t+1}_{xi} = \sigma\left(w \cdot \mathrm{concat}\left(\mathrm{h^t_{xj}}, \mathrm{h^{t+1}_{xN(i)}}\right)\right) \tag{4}$$

$$h^{t+1}_{xi} = \mathrm{norm}\left(\mathrm{h^{t+1}_{xi}}\right) \tag{5}$$

As shown in Eq. (3), the LSTM method is used to embed the neighborhood nodes of graph data to generate $h^{t+1}_{xN(i)}$. For node feature $h^{t+1}_{xN(i)}$ after aggregation, Eq. (4) concatenates the generated embedded layer and vertex feature $h^{t+1}_{xN(i)}$ with node feature h^t_{xi} of the original layer. Equation (4) splices the generated embedded layers and vertex features with the original layer and then implements the embedding operation. Finally, the whole process of convolution embedding is completed by the normalization of Eq. (5).

For the graphical index and the transaction data index, a message passing mode is used to update and embed the nodes of the corresponding index subgraph. A_{eij} in Eqs. (6) and (8) denotes the edge weight matrix, which can be updated continuously according to the training process. During the process of updating nodes, a GRU-based gate control unit is adopted to perform node embedding and updating. The updated information of graphical indicator $m^t_{ki}$is obtained by using the hidden graphical indicator state h^t_{ki} in Eq. (6). The updated information and hidden state FF of the graphical indicators in Eq. (7) are used to obtain the FF by updating the hidden states of the nodes.

$$m^t_{ki} = M_t\left(h^t_{ki}, h_{kj}, e_{ij}\right) = A_{eij}h^t_{ki} \tag{6}$$

$$h^{t+1}_{ki} = U_t\left(h^t_{ki}, m^t_{ki}\right) = \mathrm{GRU}\left(\mathrm{h^t_{ki}}, \mathrm{m^t_{ki}}\right) \tag{7}$$

Through the transaction metric subgraph node obtained during the process of acquiring the hidden state h^t_{di} in Eq. (8), the historical trading data indicators yield the updated information

m^t_{di}. As shown in Eq. (9), the hidden state of the gate control unit input node h^t_{di} and the updated information m^t_{di} are used to obtain the updated node hidden state h^{t+1}_{di}.

$$m^t_{di} = M_t\left(h^t_{di}, h^t_{dj}, e_{ij}\right) = A_{eij} h^t_{dj} \tag{8}$$

$$h^{t+1}_{di} = U_t\left(h^t_{di}, m^t_{di}\right) = GRU\left(h^t_{di}, m^t_{di}\right) \tag{9}$$

By setting three types of edge connection subgraphs, building up a three-dimensional tensor of edge weights A_{aij} to represent the degrees of correlation among different indexes, and training the three-dimensional tensor weight value, the updated information m^t_{mi} is finally obtained via the node hidden states h^{t+1}_{xi}, h^{t+1}_{ki}, and h^{t+1}_{di}after the subgraph convolutions are aggregated, and the three-dimensional tensor A_{aij} is used to carry out message transmission m^t_{mi}. The information is updated with the hidden state of the aggregation graph node by Eq. (11), and the GRU updates the hidden state of the aggregation graph node to realize the data fusion of the subgraph. Finally, Eq. (12) inputs the updated node hidden state into the fully connected layer to complete the classification and prediction steps for the whole model.

Three types of edge-connected subgraphs are setup, the hidden states h^{t+1}_{xi}, h^{t+1}_{ki}, and h^{t+1}_{di} of the nodes after the subgraph convolution operation are aggregated, *tanh* nonlinear transformation is performed on node $h^{\phi t}_{mi}$ after grouping and aggregation according to the metapath, and the inner product is calculated with metapath weight vector a^T, as shown in formula (10). The importance $m_{\phi i}$ of each metapath is calculated, A_{aij} is the aggregate node feature weight matrix, h is the offset, and $\frac{1}{|V|}$ represents the normalization of the importance of the metapath. Multi-source heterogeneous graph data consist of three types of index node sets and connection sets. The metapath is a sequence of complex network graph data constructed in this paper traversing a set of heterogeneous nodes, and the nodes of each metapath are defined as neighboring nodes.

$$m_{\phi i} = \frac{1}{|V|} \sum\nolimits_{i \in V} a^T \cdot \tanh\left(\mathrm{A_{aij}} \cdot \mathrm{h^{\phi t}_{mi}} + \mathrm{h}\right) \tag{10}$$

Formula (11) is based on obtaining the importance of metapath $m_{\phi i}$ and *softmax* normalization processing to obtain the weight coefficient of each metapath. The weight coefficient represents the importance of the metapath for the classification and prediction of the complex network of the stock market.

$$\beta_{\phi i} = \frac{\exp\left(\mathrm{m}_{\phi \mathrm{i}}\right)}{\sum_{i=1}^{P} \exp\left(\mathrm{m}_{\phi \mathrm{i}}\right)} \tag{11}$$

Different metapath weight coefficients ϕ_i and node-level attention mechanisms $N_{\phi i}$ are combined to obtain the final aggregation node $N_{\phi i}$, which is aggregated according to metapath ϕ_i, and the final aggregation node N uses two types of attention mechanisms based on the classical features of nodes. The important information contained in the node layer and the semantic layer is constructed.

$$N = \sum\nolimits_{i=1}^{P} \beta_{\phi_i} N_{\phi_i} \tag{12}$$

The updated information and the hidden state N of the aggregated graph nodes are updated by the gate control unit of Eq. (13) to update the hidden state h^{t+1}_{mi} of the node to complete subgraph data fusion. Finally, Eq. (14) inputs the updated node hidden state h_{mi}^{t+1} in the fully connected layer to complete the classification prediction of the entire model.

$$h_{mi}^{t+1} = GRU\left(h_{mi}^{t}, m_{mi}^{t}\right) \tag{13}$$

$$y = \left(\sum\nolimits_{i,t} soft\ max\left(W_t h_{mi}^{t+1}\right)\right) \tag{14}$$

4 Experimental simulation and results analysis

4.1 Data set introduction

This paper adopts the China Stock Market & Accounting Research (CSMAR) database to conduct experiments. The data table ranging from 2013-01-11 to 2019-11-25 in this database is used to construct the stock market chart data for the Shanghai Composite Index, China Securities Index (CSI) 300 and Shenzhen Composite Index. Dgl is used to construct graph data and carry out the corresponding convolution operation. The news index subgraph uses each news item as a graph node, the word vector of the news text as the corresponding node feature, and the similarity of news text as the edge weight to construct the connections between news texts. Each trading day is a subgraph. The genism library is used to convert the news texts into 200-dimensional word vectors as node features. The similarities between news texts are calculated by a similarity model as the weights of the news subgraph edges.

The subplan for the trading data indicators selects five working days as nodes and takes six indexes, the opening index, maximum index, minimum index, closing index, trading volume and number of transactions during each trading day, as node features. An edge is set between adjacent trading days, and the initial weight of each subgraph edge is a random value in [0,1]. The graphical index subplot includes a K plot, a 5-day moving average and a 20-day moving average. The graphical index image is spliced into the BGR channel by using the cv2 library and stitched into a vector. Redundant values of 190, 191 and 51 in the vector are deleted, and 30,000 values are randomly selected as the training data. Two hundred node plots are constructed, and each node has a 300-dimensional feature. Node features 0–99 are the eigenvalues of the graphical index eigenvectors, and node features 99–199 are the position values of the graphical index eigenvectors. An edge is set for the corresponding node. The initial weight of each edge is a random value in [0,1]. Figure 3 shows the news subgraph data.

4.2 Model comparison

This paper mainly performs comparative experiments from two aspects. First, the advantages of multi-source data are demonstrated from the viewpoint of data indicators. On this basis, multisource data fusion and processing methods are compared.

4.2.1 Validation of the index dimensionality

Combined with the inputs of different indexes, the stock market news text is obtained using Grat-Conv, and the historical trading data and graphical indexes are obtained using Gated-Graph-Conv [29]. Convolution and fully connected layer networks are also adopted. The basic network parameters are shown in Table 1.

Fig. 3 Indicator subgraph data (50 nodes are randomly selected for display). The edge thickness is related to the text similarity, and the node diameter and color depth depend on the number of connected nodes

Reference [30] proved the effectiveness of multidimensional data in the pioneering multiple kernel k-means (MKKM) [33] and TeSIA [30] data fusion methods through experiments, showing the explanatory ability of data fusions for stock market fluctuations and that the multidimensional data index was better than the dimensionality reduction index. To further verify the effectiveness of the multidimensional indicators utilized by the method of this paper, the experiment employs the data of the Shanghai Composite Index from 2019-01-01 to 2019-09-30 (trading days only) for training and the data from 2019-10-01 to 2019-09-30 (trading days only) for testing. We combine the T + 1 closing price with the T closing price to mark the fluctuation of the stock market. First, the model is verified according to different index inputs, and the prediction effects of different index subgraphs are presented as input values. Combined with the method in this paper and in accordance with the network settings in Table 1, after the convolution operation with the subgraphs of different indicators, the fully connected layer is connected to complete the prediction. In the seventh group of experiments in Table 2, the final prediction is completed by docking the fully connected layer after the state update of the convolutional aggregation graph for each subgraph. The experimental results are presented in Table 2. The introduction of multi-source index data effectively improves the prediction accuracy of the model proposed in this paper.

4.2.2 Comparison of integration and analysis methods

To further verify the superiority of the constructed model, this paper introduces multi-source index methods such as a support vector machine (SVM) [9], a random forest (RF) [43], MKKM [33], LSTM [30], and TeSIA [30] for model comparison. The embedding methods include the vector method and tensor method. The vector method combines the three types of features into vectors. The vector method is embedded into the corresponding prediction method. The authors of a study on tensor methods [47] used second-order tensors to simulate complex market information, and convolution LSTM (ConvLSTM) was used to process the tensor and generate the tensor data for analysis purposes. The model parameter settings

Table 1 Network parameter settings

Index Subgraph	Methods	Parameter Settings			
		Input Size	Output Size	Drop Rate	Aggregator Type
Stock news	GratConv1	200	128	0.1	LSTM
	GratConv2	128	64	0	LSTM
Trading data	GatedGraphConv	6	6	0	GRU, steps=5
Graphical indicator	GatedGraphConv1	300	256	0.1	GRU, steps=3
	GatedGraphConv2	256	128	0	GRU, steps=3

Table 2 Comparison of the prediction performances of multi-source indicators

	Trade	News	Graphics	Acc	F1-score
1	√			49.29%	0.5631
2		√		55.38%	0.6221
3			√	52.37%	0.5826
4	√	√		56.92%	0.6232
5	√		√	56.93%	0.6381
6		√	√	53.84%	0.5914
7	√	√	√	70.76%	0.8241

introduced for the comparison are shown in Table 3. The model proposed in this paper is set according to the relevant parameters in Table 1.

SVM [9] Support vector machines (SVMs) are widely used machine learning classification algorithms. Through the constructed kernel function, the points that need to be classified are mapped from low latitudes to high latitudes to achieve the classification task. According to the processing of vectors and tensors, the features are input. C is the penalty coefficient, the linear kernel function is adopted, and the maximum number of iterations is set to 5000.

Random Forest [43] The RF exhibits excellent performance in the multi-source feature classification task. The RF method is used to make predictions by combining the vector and tensor methods. The minimum number of samples required for segmentation is set to 6, and the maximum number of decision trees is set to 3.

MKKM [33] The goal of MKKM clustering is to find the optimal combination from a set of predetermined kernels; thus, the feature vectors can be accurately divided into several classes, and the input tensor mode data are directly classified according to their dimensionality. In this study, we connect multiple information source features and explore the best clustering results in the kernel space for the judgment of stock market trends. The k-means kernel number is set to 3, the kernel function adopts the Gaussian function, gamma in the Gaussian kernel function is set to 1/3, and the cluster category K is set to 2.

LSTM [30] As a special cyclic neural network, LSTM introduces a gate structure to efficiently overcome the long dependence problem of traditional cyclic neural networks. Its good characteristics make LSTM widely used in quantitative stock research. In this paper, vector

Table 3 Model parameter settings

Method	Parameters
SVM	C=0.8; kernel=linear; max_iter=5000
Random Forest	Max_feature=none; min_samples_split=10; n_estimators=3
MKKM	View=3; kernel=RBF; gamma=1/3; k=2
LSTM	Dropout=0.2; return_sequences=True
TeSIA	Tensor_order=3; tensor_size(i=5, j=1, k=10); Max_iter=5000
MHDA	n_components=5;n_iter=1000;tol=0.01; covariance_type=full

and tensor data are applied as feature inputs for the LSTM model to observe the experimental results. Dropout is set to 0.2 to prevent overfitting, and the hidden state of the output contains the result of all time steps.

TeSIA [30] After combining the characteristics of the three indexes extracted in Section 3 to construct a third-order tensor, the redundancy is removed by tensor decomposition, and a TensorFlow model is established according to the direction of price movement. TeSIA integrates multi-source information to establish a baseline for prediction, and it has achieved good results in some multi-source data prediction projects. Because this method is based on the tensor data processing method, there is no vector method-based data verification step. Third-order tensors were adopted with dimensions i = 5, j = 1, and k = 10, and the maximum number of iterations was set to 5000.

MHDA [30] By integrating multi-source information such as transaction data, news event data, and investor comments and using domain-specific emotional dictionaries and relationship diagrams, features are extracted from transaction data, user discussion forum comments, and news events. A multivariate Gaussian mixture model is used to represent features, and the features are integrated into a hidden Markov model to capture time-dependent information to predict stock market price fluctuations.

The algorithm in Table 4 is run in Python 3.7.3 software, and the hardware contains an Intel i7 processor and 8 GB of DDR4 RAM. The parameter configuration of the algorithm is shown in Table 3. We list the time complexity, accuracy, F1 and Matthews correlation coefficient (MCC) values yielded by each method in Table 4.

From Table 4, the accuracy of the SVM using multi-source index data in terms of stock market prediction only reaches 48.69%. Although the prediction accuracy of the RF is slightly higher than that of the SVM algorithm, it is basically maintained around the level of random probability. By considering the relationships between the three indexes, MKKM and TESIA exhibit improved prediction performance, and the obtained accuracy, F1 and MCC values are better than those of the RF and SVM methods. The MSub-GNN method proposed in this research paper embeds the relationships between the three types of indexes and the relationships between the trading days into graph data edges and subgraphs and then uses a multi-index subgraph convolution operation to forecast the stock market. The accuracy, F1 value and Matthews correlation coefficient of the prediction method in this paper reach 70.76%, 0.8241, and 0.421, respectively, which are all higher than those of the other baseline models. From the perspective of model prediction performance, the method proposed in this paper is the best.

4.3 Back testing strategy

As shown in Tables 2 and 4, the method adopted in this paper uses multi-source and heterogeneous stock market indicators to predict the stock market, as previously mentioned. Such a method is much better than not only single-source index prediction but also other multi-source index prediction methods. In this paper, we choose the data of the Shanghai Composite Index, CSI 300 and Shenzhen Composite Index for the trading days from 2013-01-11 to 2019-11-24 for training and the data for 100 trading days from 2019-11-25 to 2020-04-22 for back testing. When the model predicts an increase, it generates a buy signal; when it predicts a fall, it generates a sell signal, but a trade is not triggered by the same continuous signals. The initial

Table 4 Comparison of the accuracies of the prediction methods

Methods	Time Complexity	Vector-based			Tensor-based			Graph-based		
		Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	$O(d^3 + n \times d^2)$	48.69%	0.5326	0.007	47.56%	0.5423	0.003	- -	- -	- -
RandomForest	$O(n \times d \times m)$	52.31%	0.5627	0.012	48.23%	0.5689	0.005	- -	- -	- -
MKKM	$O(v \times k \times n^3)$	61.54%	0.6512	0.150	60.45%	0.6357	0.137	- -	- -	- -
LSTM	$O(nm + n^2 + n)$	63.45%	0.6823	0.352	61.23%	0.6512	0.326	- -	- -	- -
TeSIA	$O(nd + d^2)$	- -	- -	- -	63.07%	0.6821	0.392	- -	- -	- -
MHDA	$O(n^2 \times d \times I)$	- -	- -	- -	63.87%	0.7361	0.386	- -	- -	- -
Our Methods methodmethod	$O(n^2 \times d \times I)$	- -	- -	- -	- -	- -	- -	70.76%	0.8241	0.421

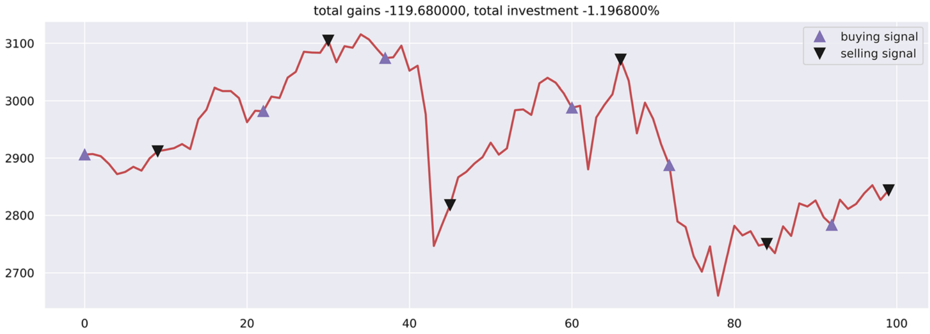


Fig. 4 Trading data indicator strategy

strategy capital totals 10,000, and the income settlement standard is the closing price on the given trading day. Due to page limitations, this paper presents and analyzes the experimental results of the Shanghai Composite Index in detail. In Figs. 4, 5 and 6, Methods 1, 2 and 3 in Table 1 are used to conduct trade back testing. In Figs. 7 and 8, the TeSIA and MHDA methods, which have a better effect in terms of multi-source index prediction, are adopted for comparison purposes. Figure 9 shows the MSub-GNN method proposed in this paper.

Figures 4, 5, 6, 7, 8 and 9 show an investment simulation of buying and selling signals and returns, respectively. The figures do not more intuitively show the benefits of several strategies. Figure 10 is the comparison chart regarding the benefits of the different strategies. As shown in

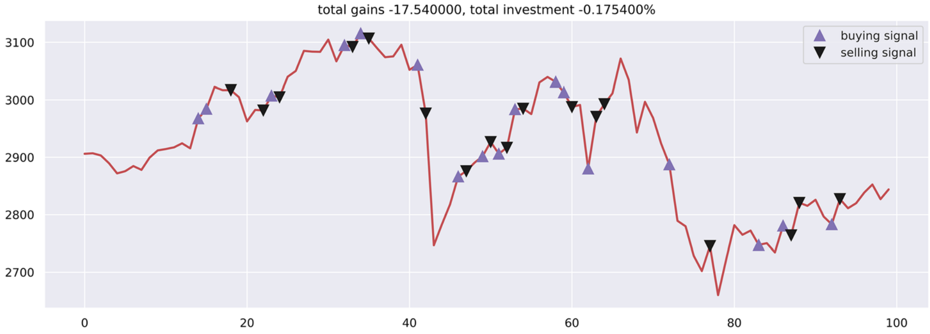


Fig. 5 Trading strategies for news indicators

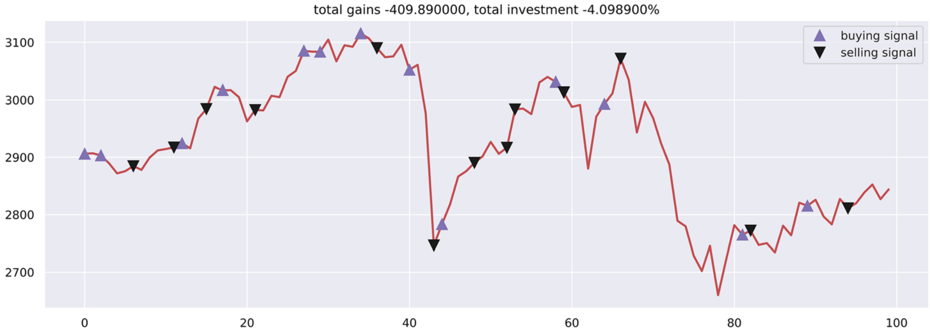


Fig. 6 Graphical indicator trading strategy

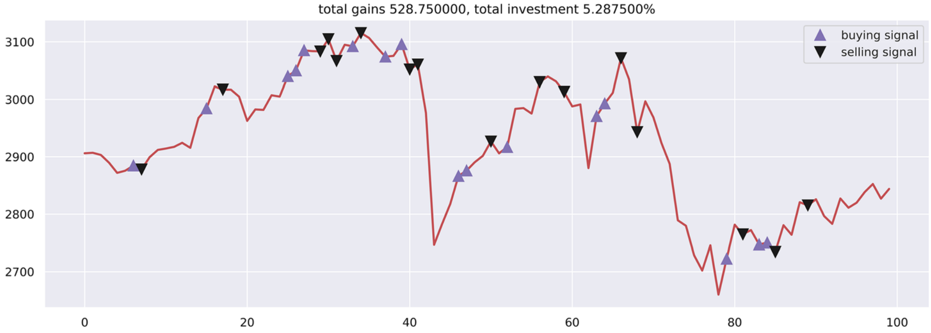


Fig. 7 TeSIA trading strategy

Figs. 4, 5, 6, 7, 8, 9 and 10, due to the novel coronavirus epidemic in March 2020, the Shanghai Stock Exchange Index rose by 4.73%, the Shenzhen Stock Exchange Index rose by 3.17% and the Growth Enterprise Index decreased by 1.18% over the same period. The investment yield of the method proposed in this paper is 17.32%, which is the highest yield. The yields of the TeSIA and MHDA trading strategies are 5.28% and 7.32%, respectively, which are higher than the index in the same period, while other strategies produce yields that are lower than that of the index. In this article, since the short-term return in the stock market forecast is not considered part of the strategy, the back-test return strategy in this paper is closely correlated to the trend of the predicted index. In the situation with only two sharp drops, the strategy in this paper achieves good returns. From the red

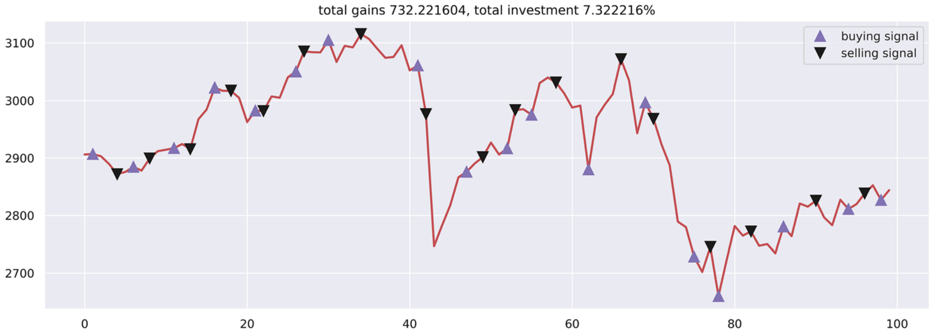


Fig. 8 MHDA trading strategy

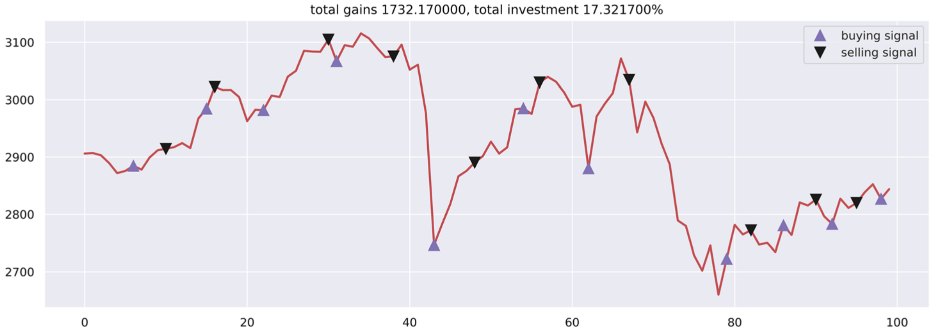


Fig. 9 MSub-GNN trading strategy

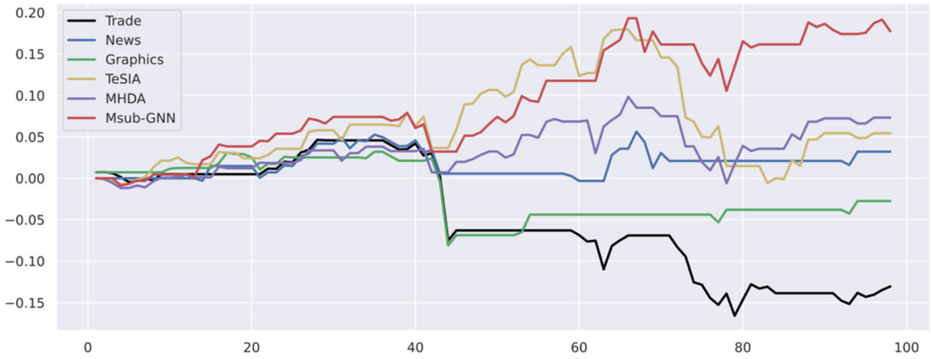


Fig. 10 Comparison of strategy benefits

rectangle in the figure, the news indicators utilized in this paper play an important role in stopping losses due to the black swan event (the novel coronavirus outbreak). Moreover, the graphic and transaction record indexes can better predict the overall trend of the stock market. Table 5 provides an overall description and evaluation of the trading strategy through the Sharpe ratio, Sortino ratio, information ratio, and max drawdown. The promise ratio and the information ratio are 0.978, 1.325, and 1.413. Compared with the baseline model, the proposed method has higher risks and returns, and the excess return brought by active investment is the highest, with a maximum drawdown of 21.53%, which is in a larger downtrend. It also has a better stop loss performance than the baseline model. The Sharpe ,ratio Sotino ratio, and information ratio of the method proposed in this article are 0.978, 1.325, and 1.413, respectively. Compared with the baseline model, the method proposed in this article has a higher risk return, and the excess return brought by active investment is the highest, with a maximum drawdown of 21.53%. In a larger downtrend, it has better stop loss performance. The multi-source heterogeneous data index effectively realizes the complementarity of the prediction function, and the convolution method applicable to the subgraph used in this paper effectively processes and analyzes the subgraph index.

5 Conclusion and outlook

This paper proposed a graph neural network for the fusion of multi-source heterogeneous subgraphs for application in stock market trend prediction. This new network redefined the three kinds of embedded representation methods for stock market indexes and built a trading data index subgraph, a stock market news index subgraph, and graphical indicators. This network adopted different convolution methods for node pair data aggregation in graphs. The hidden states of the aggregated

Table 5 Trading strategy back test benefits overview

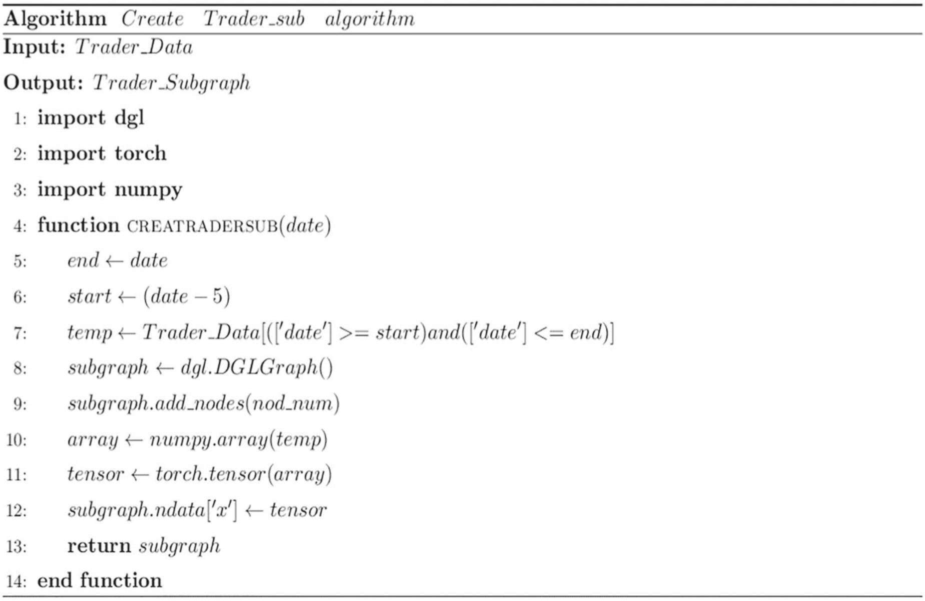
	Methods	Sharpe Ratio	Sortino Ratio	Information Ratio	Max Drawdown
1	Trade	0.437	0.562	0372	38.29%
2	News	0.416	0.534	0.346	33.58%
3	Graphics	0.451	0.685	0.426	36.28%
4	TeSIA	0.847	1.126	1.132	28.71%
5	MHDA	0.876	1.116	1.332	30.12%
7	Msub-GNN	0.978	1.325	1.413	21.53%

nodes were updated by using the weights of the heterogeneous edges obtained after the convolution operation to complete subgraph fusion. Finally, a fully connected classification layer classification was used to make predictions. The method in this paper realized the semantic mining and expression of the correlation relationships among the prediction indexes, which effectively improved the accuracy of stock market trend prediction and yielded good results in the back-test experiment. However, the model proposed in this article is not sufficient for mining the implicit semantics of different types of indicator nodes and the interrelationship between indicator nodes. Combined with the characteristics of the multi-source heterogeneous graph data of the stock market constructed in this article, especially the relationship between indicator subgraphs, the relationship between the subgraph nodes still needs to be explored in depth, and the construction of a multiple attention mechanism to fully mine the rich semantic information of the fusion of multi-source heterogeneous indicators will be the direction of future research. The model can be applied to high-frequency quantitative investment and generalized to predict the trends of other financial products.

Appendix 1

Trading subgraph data construction

The trading subgraph data construction process is illustrated in the following table. First, the program defines the interval of the determined trading day and builds a subgraph DGLGraph(). Then, based on the trading day, subgraph nodes are set up with the add_nodes() function. Finally, the trading day data index is taken as the node feature via ndata().

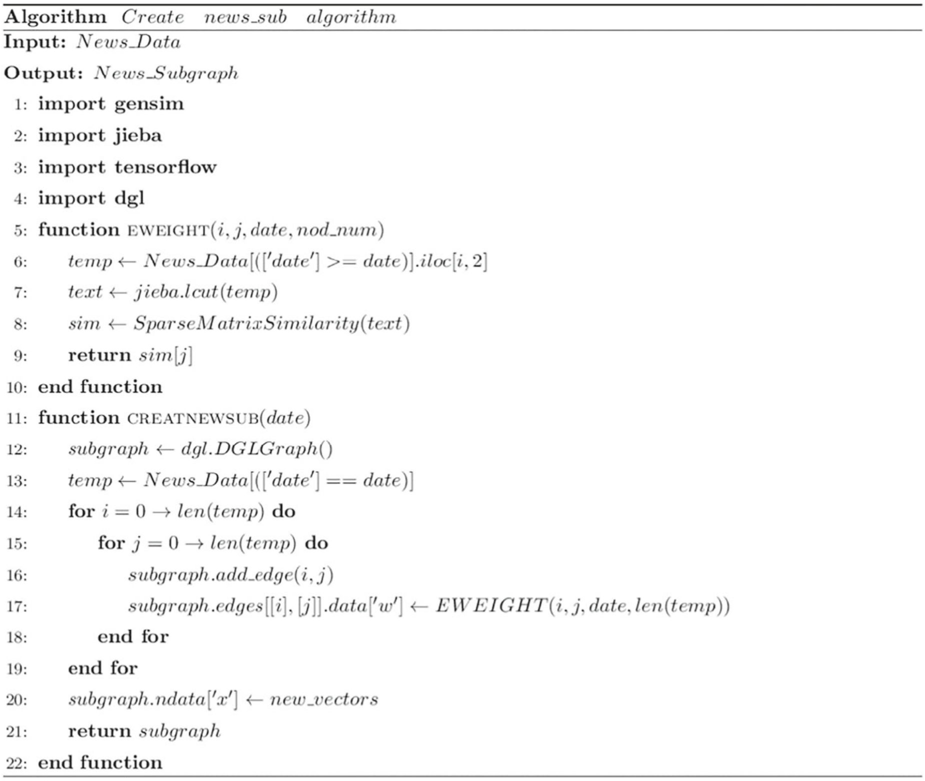


```
Algorithm Create Trader_sub algorithm
Input: Trader_Data
Output: Trader_Subgraph
import dgl
import torch
import numpy
function CREATRADERSUB(date)
    end ← date
    start ← (date − 5)
    temp ← Trader_Data[(['date'] >= start)and(['date'] <= end)]
    subgraph ← dgl.DGLGraph()
    subgraph.add_nodes(nod_num)
    array ← numpy.array(temp)
    tensor ← torch.tensor(array)
    subgraph.ndata['x'] ← tensor
    return subgraph
end function
```

Subgraph data construction for stock market news

Subgraph data construction for stock market news

To construct the stock market news subgraph, two core functions are used. The first function, EWEIGHT(), is mainly used to calculate the similarities between news events and the edge weight values for the construction of the subgraph data. Jieba () is used for word classification, and SparseMatrixSimilarity () is used to calculate the similarity between texts. The second function is CREATENEWSUB (). It acts as a subgraph builder function, where each piece of news text is used as a node, Add_edge () increases the edges between the nodes, and EWEIGHT() uses the edge weights to assign values. The news text vector contains node characteristics.



Algorithm *Create news_sub algorithm*

Input: *News_Data*

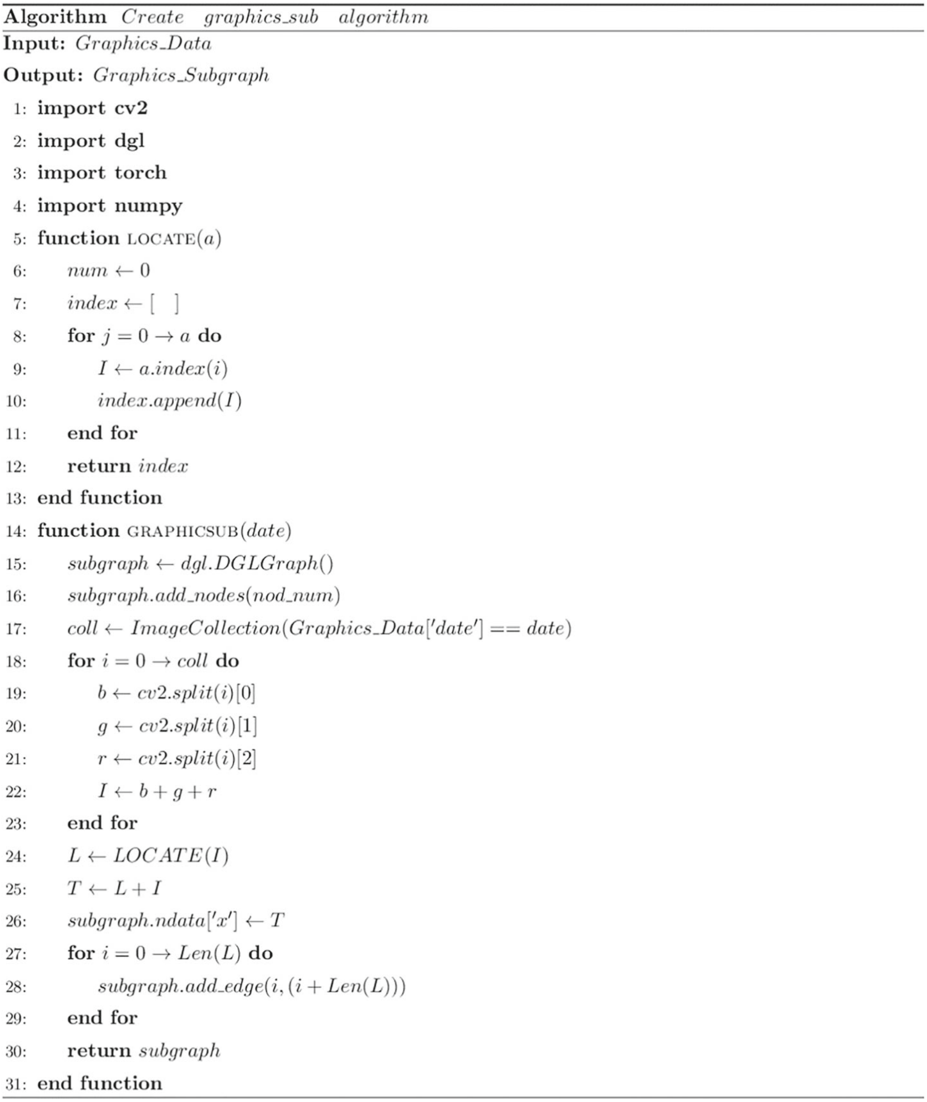
Output: *News_Subgraph*

```
import gensim
import jieba
import tensorflow
import dgl
function EWEIGHT($i, j, date, nod\_num$)
    $temp \leftarrow News\_Data[(['date'] >= date)].iloc[i, 2]$
    $text \leftarrow jieba.lcut(temp)$
    $sim \leftarrow SparseMatrixSimilarity(text)$
    return $sim[j]$
end function
function CREATNEWSUB($date$)
    $subgraph \leftarrow dgl.DGLGraph()$
    $temp \leftarrow News\_Data[(['date'] == date)]$
    for $i = 0 \rightarrow len(temp)$ do
        for $j = 0 \rightarrow len(temp)$ do
            $subgraph.add\_edge(i, j)$
            $subgraph.edges[[i], [j]].data['w'] \leftarrow EWEIGHT(i, j, date, len(temp))$
        end for
    end for
    $subgraph.ndata['x'] \leftarrow new\_vectors$
    return $subgraph$
end function
```

Construction of graphical index subgraph data

Construction of graphical index subgraph data

Two core functions are applied to construct graphical index subgraph data. The first function, LOCATE(), is mainly used to obtain the position of a number in a vector. The second function, GRAPHICSUB(), is used as the construction function for the graphical index subgraph. A ‘for’ loop and cv2.split are used to extract the three primary color features of the graphical index, and the characteristic value is used as the corresponding node feature. Then, the second ‘for’ loop of the function is used to establish edges between the characteristic value nodes and the characteristic value position nodes.



```
Algorithm $Create\ \ graphics\_sub\ \ algorithm$
Input: $Graphics\_Data$
Output: $Graphics\_Subgraph$
import cv2
import dgl
import torch
import numpy
function LOCATE($a$)
    $num \leftarrow 0$
    $index \leftarrow [\ \ ]$
    for $j = 0 \rightarrow a$ do
        $I \leftarrow a.index(i)$
        $index.append(I)$
    end for
    return $index$
end function
function GRAPHICSUB($date$)
    $subgraph \leftarrow dgl.DGLGraph()$
    $subgraph.add\_nodes(nod\_num)$
    $coll \leftarrow ImageCollection(Graphics\_Data['date'] == date)$
    for $i = 0 \rightarrow coll$ do
        $b \leftarrow cv2.split(i)[0]$
        $g \leftarrow cv2.split(i)[1]$
        $r \leftarrow cv2.split(i)[2]$
        $I \leftarrow b + g + r$
    end for
    $L \leftarrow LOCATE(I)$
    $T \leftarrow L + I$
    $subgraph.ndata['x'] \leftarrow T$
    for $i = 0 \rightarrow Len(L)$ do
        $subgraph.add\_edge(i, (i + Len(L)))$
    end for
    return $subgraph$
end function
```

The shared link details the procedures in this article[1].

The shared link details the procedures in this article[1]

[1] https://pan.baidu.com/s/1Wlj7FewoUDcwMWBET58VNg.

Acknowledgements The research work is supported by the National Natural Science Foundation of China (NSFC) (71873108 and 62072379), Fundamental Research Funds for the Central Universities kjcx20210103), Financial Intelligence and Financial Engineering Key Lab of Sichuan Province, Jiaozi Institute of Financial Technology Innovation, Southwest University of Finance and Economics (cgzh20210204), Research Program of Science and Technology at Universities of Inner Mongolia Autonomous Region (2021GG0164) and Financial Innovation Center of the Southwestern University of Finance and Economics.

References

1. Arasu A, Widom J. Resource sharing in continuous sliding-window aggregates[EB/OL]. [2019-10-02]. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/sharing.pdf
2. Atkins A, Niranjan M, Gerding E (2018) Financial news predicts stock market volatility better than close price[J]. J Financ Data Sci 4(2):120–137
3. Belov G, Scheithauer G (2006) A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting[J]. Eur J Oper Res 171(1):85–106
4. Box GEP, Jenkins GM, Reinsel GC et al (2015) Time series analysis: forecasting and control[M]. Wiley, Hoboken
5. Bruna J, Zaremba W, Szlam A et al (2013) Spectral networks and locally connected networks on graphs[J]. arXiv preprint arXiv:1312.6203
6. Bulkowski TN (2012) Encyclopedia of Canlestick charts[M]. Wiley,&nbsp;Hoboken
7. Chai L, Xu H, Luo Z et al (2020) A multi-source heterogeneous data analytic method for future price fluctuation prediction[J]. Neurocomputing 418:11–20
8. Chan WS (2003) Stock price reaction to news and no-news: drift and reversal after headlines[J]. J Financial Econ 70(2):223–260
9. Chen Y, Hao Y (2017) A feature weighted support vector machine and k-nearest neighbor algorithm for stock market indices prediction [J]. Expert Syst Appl 80:340–355
10. De Gooijer JG, Hyndman RJ (2006) 25 years of time series forecasting[J]. Int J Forecast 22(3):443–473
11. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering[C]. In: Proceedings of Advances in Neural Information Processing Systems, 3844–3852
12. Ding X, Zhang Y, Liu T et al (2014) Using structured events to predict stock price movement: An empirical investigation[C]. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1415–1425
13. Ding X, Zhang Y, Liu T et al (2015) Deep learning for event-driven stock prediction[C]. In: Proceedings of the Twenty-fourth International Joint Conference on Artificial Intelligence
14. Edwards RD, Magee J, Bassetti WHC (2018) Technical analysis of stock trends[M]. CRC Press, &nbsp;Boca Raton
15. Fama EF (1970) Efficient capital markets: A review of theory and empirical work[J]. J Finance 25(2):383–417
16. Fama EF, French KR (1992) The cross-section of expected stock returns[J]. J Financ 47(2):427–465
17. Fama EF, French KR (1993) Common risk factors in the returns on stocks and bonds[J]. J Financ Econ 33(1):3–56
18. French FKR (1996) Multifactor explanations of asset pricing anomalies[J]. J Financ 51(1):55–84
19. Goodfellow I, Bengio Y, Courville A et al (2016) Deep learning[M]. MIT Press, Cambridge
20. Granville JE (1960) A strategy of daily stock market timing for maximum profit[M]. Prentice-Hall, &nbsp;Hoboken
21. Guo J-Y, Li R-H(2020) Graph neural network based anomaly detection in dynamic networks[J]. J Softw 31(03):156–170
22. Huang TL (2018) The puzzling media effect in the Chinese stock market[J]. Pac-Basin Financ J 49:129–146
23. Jegadeesh N, Titman S (1993) Returns to buying winners and selling losers: Implications for stock market efficiency[J]. J Financ 48(1):65–91
24. Jiao G, Zhang Y(2019) Research on user participation behavior of online stock community[J]. J Inf Syst 1
25. Kahneman D (2003) Maps of bounded rationality: Psychology for behavioral economics[J]. Am Econ Rev 93(5):1449–1475
26. Kim R, So CH, Jeong M et al. (2019) Hats: A hierarchical graph attention network for stock movement prediction[J]. arXiv preprint arXiv:1908.07999
27. Kipf TN, Welling M (2016)Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907
28. Kusuma RM, I, Ho TT, Kao WC et al (2019) Using deep learning neural networks and candlestick chart representation to predict stock market[J]. arXiv preprint arXiv:1903.12258

29. Li Y, Tarlow D, Brockschmidt M et al (2015) Gated graph sequence neural networks[J]. arXiv preprint arXiv:1511.05493
30. Li Q, Jiang LL, Li P et al (2015)Tensor-based learning for predicting stock movements[C]. In: Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence
31. Li Q, Wang J, Wang F et al (2017) The role of social sentiment in stock markets: a view from joint effects of multiple information sources[J]. Multimed Tools Appl 76(10):12315–12345
32. Li Lihui T, Xiang Y, Haidong et al (2005) Financial time series forecasting based on SVR[J]. Comput Eng Appl 41(30):221–224
33. Liu X, Dou Y, Yin J et al (2016) Multiple kernel k-means clustering with matrix-induced regularization[C]. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence,&nbsp;1888–1894
34. Liu Y, Zeng Q, Yang H et al (2018) Stock price movement prediction from financial news with deep learning and knowledge graph embedding[C]. In: Proceedings of the Pacific Rim Knowledge Acquisition Workshop. Springer, Cham, 102–113
35. Liu J, Lu Z, Du W (2019) Combining enterprise knowledge graph and news sentiment analysis for stock price prediction[C]. In: Proceedings of the 52nd Hawaii International Conference on System Sciences
36. Lo AW, MacKinlay AC (1988) Stock market prices do not follow random walks: Evidence from a simple specification test[J]. Rev Financ Stud 1(1):41–66
37. Matsunaga D, Suzumura T, Takahashi T (2019) Exploring graph neural networks for stock market predictions with rolling window analysis[J]. arXiv preprint arXiv:1909.10660
38. Menon VK, Vasireddy NC, Jami SA et al (2016) Bulk price forecasting using spark over nse data set[C]. In: Proceedings of International Conference on Data Mining and Big Data. Springer, Cham, 137–146
39. Mittal A, Goel A (2012) Stock prediction using twitter sentiment analysis[J]. Standford University, 15
40. Qu Q, Yu H, Huang R (2018) Spammer detection technology of social network based on graph convolution network[J]. J Netw Inform Secur 004(005):39–46
41. Rojas I, Valenzuela O, Rojas F et al (2008)Soft-computing techniques and ARMA model for time series prediction[J]. Neurocomputing 71(4–6):519–537
42. Roondiwala M, Patel H, Varma S (2017) Predicting stock prices using LSTM[J]. Int J Sci Res (IJSR) 6(4): 1754–1756
43. Shihavuddin A, Ambia MN, Arefin M et al (2010) Prediction of stock price analyzing the online financial news using Naive Bayes classifier and local economic trends [C]. In: Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering. Piscataway, IEEE, 22–26
44. Shiller RJ (2015) Irrational exuberance: Revised and expanded third edition[M]. Princeton University Press, &nbsp;Princeton
45. Si J, Mukherjee A, Liu B et al (2013) Exploiting topic based twitter sentiment for stock prediction[C]. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, vol 2: Short Papers, 24–29
46. Simon HA (1996) Designing organizations for an information-rich world[J]. Int Libr Crit Writ Econ 70: 187–202
47. Tan J, Wang J, Rinprasertmeechai D et al (2019) A Tensor-based eLSTM model to predict stock price using financial news[C]. In: Proceedings of the 52nd Hawaii International Conference on System Sciences
48. Tanaka-Yamawaki M, Tokuoka S (2007) Adaptive use of technical indicators for the prediction of intra-day stock prices[J]. Phys A 383(1):125–133
49. Tang H, Chiu KC, Xu L (2003) Finite mixture of ARMA-GARCH model for stock price prediction[C]. In: Proceedings of the Third International Workshop on Computational Intelligence in Economics and Finance (CIEF’2003), North Carolina, USA, 1112–1119
50. Tsai CF, Quan ZY (2014) Stock prediction by searching for similarities in Canlestick charts[J]. ACM Trans Manage Inform Syst (TMIS) 5(2):9
51. Wei YC, Lu YC, Chen JN et al (2017) Informativeness of the market news sentiment in the Taiwan stock market[J]. North Am J Econ Financ 39:158–181
52. Zhang X, Li Y, Wang S et al (2018) Enhancing stock market prediction with extended coupled hidden markov model over multi-sourced data[J]. Knowl Inf Syst
53. Zhang X, Zhang Y, Wang S et al (2018) Improving stock market prediction via heterogeneous information fusion[J]. Knowl Based Syst 143:236–247

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.