International Journal of Machine Learning and Cybernetics (2024) 15:161–176
https://doi.org/10.1007/s13042-023-01817-6

ORIGINAL ARTICLE

Integrated GCN-LSTM stock prices movement prediction based on knowledge-incorporated graphs construction

Yong Shi[1,2,3,5] · Yunong Wang[1,2,3] · Yi Qu[1,2,3] · Zhensong Chen[4]

Received: 31 July 2022 / Accepted: 28 February 2023 / Published online: 16 April 2023

Abstract

Stock prices movement prediction has been a longstanding research topic. Many studies have introduced several kinds of external information like relations of stocks, combined with internal information of trading characteristics to promote forecasting. Different from previous cases, this article proposes a reasonable assumption that major fluctuations of stock prices are mainly triggered by high-volume transactions which usually occur on a group of stocks that share some common features (e.g., stocks in the same industry, region, concept or yield similar volatility), and further develops an integrated GCN-LSTM method to achieve more precise predictions from the perspective of modelling capital flows. First, we construct four kinds of graphs incorporating various relational knowledge (edge) and utilize graph convolutional network (GCN) to extract stock (node) embeddings in multiple time-periods. Then, the obtained temporal sequences of stock embeddings are put into long short-term memory recurrent neural network (LSTM) to discriminate the moving direction of prices. Extensive experiments on major Chinese stock indexes have demonstrated the effectiveness of our model with best accuracy of 57.81% acquired, which is much better than baselines. Moreover, experimental results of GCN-LSTM under different graphs and various node embedding dimensions have been compared and analyzed, indicating the selection of key parameters to achieve optimal performances. Our research findings provide an improved model to forecast stock prices movement directions with a reliable theoretical interpretation, and in depth exhibit insights for further applications of graph neural networks and graph data in business analytics, quantitative finance, and risk management decision-makings.

Keywords Stock prices movement prediction · Knowledge-incorporated graphs · Graph construction · Graph convolutional network (GCN) · Long short-term memory recurrent neural network (LSTM)

✉ Yi Qu
quyi17@mails.ucas.ac.cn

Yong Shi
yshi@ucas.ac.cn

Yunong Wang
wangyunong20@mails.ucas.ac.cn

Zhensong Chen
chenzhensong@cueb.edu.cn

[1] School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, China

[2] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China

[3] Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China

[4] School of Management and Engineering, Capital University of Economics and Business, Beijing 100070, China

[5] College of Information Science and Technology, University of Nebraska at Omaha, Omaha, NE 68182, USA

1 Introduction

Accurate prediction of stock prices movement is of great significance due to its crucial value in business analytics, quantitative finance and risk management decision-makings. It is a temporally-dependent task with high stochasticity and chaotic information [7], which makes it complicated and difficult. Earlier studies normally introduced statistical time-series modeling techniques into experiments, such as Auto-Regressive Integrated Moving Average Model (ARIMA), Generalized Auto-Regressive Conditional Heteroscedasticity Model (GARCH) [2, 4] and their variants that are still being used today [16, 32, 50]. In recent years, along with the expansion of available information and promotion of information technologies, the research paradigm in stock prices prediction has gradually evolved from "model-driven" to "data-driven", leading to various applications of machine learning, deep learning tools, e.g., Random Forest (RF) [21,

31, 38], Artificial Neural Network (ANN) [6, 35, 59], Support Vector Machines (SVM) [24, 41, 48], Convolutional Neural Network (CNN) [11, 22, 25, 27] and Recurrent Neural Network (RNN) [8, 10, 56, 58]. The utilization of these advanced techniques has generated great improvements in stock prices forecasting, and further demonstrated the importance and effectiveness of implementing integrated models based on both structured and unstructured data combined.

In related studies deploying machine learning and deep learning methods into predictions, its performance depends on the ability of models to extract insightful knowledge from simple representations [3], by focusing on characteristics of target stocks and finding patterns of prices fluctuations. It ignores the linkages between stocks in terms of their prices changes, as the fluctuations of prices might exhibit similar trends. This is the phenomenon titled "stock prices synchronicity" that has been proven to be existed [45] and is also consistent with the reality. Thus, it's necessary to consider trading indicators and relations between stocks into prices forecasting and formulate both as graphs. However, different from real graph networks, the links between stocks are not well-defined so that it requires us to incorporate human knowledge into graph construction process. Towards this purpose, we propose a theoretical interpretation from the perspective of capital flows, with the objective of building reasonable edges between stocks. Similar with other kinds of commodities, prices of stocks are also determined by the imbalance between supply and demand, which is the selling and buying orders in real stock markets. Under such circumstances, small amount of buyings/sellings (mostly issued by individual investors) wouldn't impact prices significantly, while only huge capital flows (mostly institutional investors) are capable of that. These large amount of money flows are normally carried out by private funds, public funds, pensions, security companies, etc, and these trades usually occur on a group of stocks that share some common features, like stocks in the same industry, region, concept or yield similar volatility. For instance, some funds are oriented into investing advanced information technologies, such as the leading telecommunications giant Apple Inc., and its related companies including Luxshare-ICT, Han's Laser, Ofilm Group, Goertek and Desay Corporation that are major suppliers of Apple in China and usually called "Apple concept stocks" in Chinese stock market. When making investment decisions, these funds simultaneously buy/sell the stocks in the above group, producing high-volume transactions by its substantial capital flows, and further causing sharp increases or declines on prices of this group of stocks. Consequently, it's reasonable to connect these stocks in the above group and construct graphs by incorporating the prior knowledge of "concept", as well as to capture this kind of similar fluctuating patterns among prices, or called synchronicity. And this is the theoretical assumption guiding us how to build graphs and explaining why our proposed GCN-LSTM is effective, which is also the very source of our research motivation.

As one kind of commonly-seen but sophisticated data, graph is non-Euclidean structured and contains fundamental elements of nodes, node attributes, node labels, edges, edge attributes, graph labels, etc. Among them, the relations of stocks are the basic information to formulate the graphs needed. Given its complexity, it's difficult to process or analyze simply using traditional statistical models or machine learning techniques as they are designed to operate on structured datasets. Therefore, graph representation learning has emerged and evolved quickly in the last 5 years, especially with the advent of graph neural network (GNN) models. The GNN can be employed directly on graph data to perform node-level or graph-level related tasks efficiently. Benefiting from the principle of message passing and information aggregation, GNN has become powerful tools in processing and modeling graph data, further successfully expanded into many other disciplines including medical science [37], electronic commerce [44], chemical engineering [49], computational linguistics [28]. Graph convolutional network (GCN), invented by [29], is actually a landmark model in the development of GNN, of which the primary mechanism is updating node representations layer-wise by encoding global structural information. With appropriate adaptations, GCN could be adopted in many tasks of related fields, like node classification [39], link prediction [13], community detection [51] and graph classification [52]. Nevertheless, there have been very few studies of modeling stock markets by graph representation learning, particularly implementing GNN models. And this inspires us to explore the relationships of stocks, build graphs by incorporating various knowledge, and achieve more precise forecasting of stock prices movement by graph representation learning.

In this article, a reasonable theoretical interpretation regarding stock prices has been proposed to illustrate the mechanism of major fluctuations from capital flows perspective. This provides solid theoretical support and further serves as guidance in constructing knowledge-incorporated graphs needed. Based on that, an integrated two-staged GCN-LSTM methodology has been developed, realizing more productive and efficient performances in forecasting stock prices movement direction. Major contributions of this article are highlighted as follows:

- We construct four kinds of graphs incorporated with various prior knowledge based on a reasonable theoretical interpretation.

- We propose an integrated two-staged methodology of GCN-LSTM to achieve more effective stock prices movement prediction.
- We conduct extensive experiments on several stock indexes, demonstrating the outperformance of GCN-LSTM compared with baselines.
- We compare the predictions of GCN-LSTM under various settings, indicating the optimal selection of key parameters in forecasting.

The remainder of this article is organized as follows: Sect. 2 is a brief review of related work in the filed of stock prices movement prediction. Section 3 specifically illustrates our proposed GCN-LSTM method while dataset, models with parameters, experimental setup are reported in Sect. 4. Experimental results with detailed comparisons and analysis are presented in Sect. 5, and conclusions in Sect. 6.

2 Related work

This section reviews the related work from two aspects. First is machine learning and deep learning techniques utilized in stock prices forecasting, with emphasis on the data usage of external information. Second it presents graph or network related studies in stock prices prediction especially how to formulate graph structures and the modeling process.

2.1 Machine learning and deep learning models

Continuous attention has been gathered in achieving effective forecasting of stock prices movement, as accurate predictions can bring excess returns to investors and lead to better financial risk management. Many previous works have proposed various stock prices forecasting models that can be broadly divided into two categories: the model-driven methods like statistical analysis, and the data-driven techniques represented by machine learning algorithms. Model-driven methods are usually based on strict mathematical assumptions, like assuming the distribution of samples, and this may not be consistent with reality. The latter one, data-driven models, focuses more on the inner structure of the dataset, based on which insightful information could be extracted to support the task. Machine learning-based stock prices movement predictions are mainly carried out on structured data, such as daily trading characteristics (technical indicators), to find patterns from time-varying multivariate datasets and train classifiers to discriminate the moving directions (up and down) of target stocks. Among these powerful methods, the representatives are Decision Tree (DT) and tree-based ensemble learning including Random Forest (RF) [21, 31, 38], AdaBoost [23, 30], Gradient Boosting Decision Tree (GBDT) [40, 60], and Support Vector Machines (SVM) [24, 41, 48] that is good at dealing with small-sampled, non-linear, high-dimensional regression or classification problems. These cases have achieved good performances in predicting stock prices movement, and greatly enriched the research in this field.

In a recent decade, the emergence and availability of unstructured data, as we defined as external information in this article, has enabled deep learning models to be adopted into stock prices prediction task. These advanced techniques are capable of efficiently analyzing quantities of unstructured data, e.g., texts and images, which are further combined with internal information of trading characteristics to promote forecasting. Li et al. [34] leveraged the ability of LSTM in encoding the context information of textual data, to formulate news headlines as features in stock prices prediction. Chandola et al. [5] proposed a hybrid deep learning model incorporating both Word2Vec and LSTM, to predict the directional movement of stock prices based on time-series financial information and news headlines. By dividing five companies from the same sector into one group and constructing a 3D image sized $15 \times 15 \times 5$ with multiple technical indicators, [47] proposed a 3D CNN-based approach to distinguish the directional trends in stocks' prices. Liu et al. [36] transformed stock prices charts into images and used Deep Learning Neural Networks (DLNN) to conduct image processing, with simulated analysts to predict stock prices movement in the short term.

The utilization of deep learning illustrated above has successfully enhanced the predicting accuracies, benefiting from both the diversity of available information sources and the learning ability of analyzing techniques. Moreover, it has also demonstrated the great value of implementing multi-staged models integrating multiple existing methods, based on both structured and unstructured data combined. Following this direction, many researches have been produced that attempt to introduce relational information into studies and deal with it from graph representation learning, and this will be clarified in detail in the following subsection.

2.2 Graph or network structures utilized

Graph-related studies of stock market prediction could be traced back to Complex Network Analysis, which usually employs specific indicators to evaluate or reflect the situation. For example, the inter-correlation degree is often used to identify interrelationships among different individuals, revealing the risk spillover effect in the stock market or the systemic importance of financial institutions [14, 17]. After years of exploration, scholars have started to investigate the relations of stocks, and further incorporate them into predicting stock prices. Consequently, graph/network structured data and applications of GNN models have emerged in stock prices forecasting, mostly conducted in node-level tasks, in

particular node classification or node embedding representation. Considering that stock prices prediction normally involves time-series dataset, it makes relevant research a temporal modeling task with relations extraction and graph formulation. Feng et al. [18] proposed a temporal graph convolution (TGC) model and contributed a Relational Stock Ranking solution for stock prices prediction, jointly modeling the temporal evolution and relation network of stocks. Li et al. [34] proposed an LSTM-based Relational Graph Convolutional Network (LSTM-RGCN) to analyze the effect of event information of news, achieving relational event-driven stock trend forecasting. Gao et al. [20] invented a Time-aware Relational Attention Network (TRAN) with attention mechanism designed, to capture the time-varying correlation between stocks and realize graph-based stock recommendations. Hou et al. [26] invented a hybrid model integrating GCN and LSTM, similar to our approach, but the adjacency matrix established is actually learned by variational auto encoders (VAE) that is fundamentally different from our knowledge-incorporated graphs. Feng et al. [19] put forward a Relation-aware Dynamic Attributed Graph Attention Network (RA-AGAT) to extract global information, which is further combined with timing characteristics to recommend high-return ratio stocks. Borrowing the idea from relational GCN [46, 53] adopted event information from news and social media, to formulate a relational event-driven stock trend forecasting (REST) framework which has yielded higher returns of investment than baselines. Summarizing the above studies, the important value of implementing integrated/hybrid models has been identified in improving the accuracy of predicting stock prices, as it's essentially a combination of two major phases including relations extraction and temporal modeling, in which the order of multiple modules may be different, or various sections like attention mechanism might be introduced. It can also be seen and noted that GCN is more frequently used in extracting relational information while LSTM is designed for temporal modeling regarding time-series data. Moreover, the combination of multiple techniques in stock prices prediction is actually originated from the utilization of structured and unstructured data, while the latter could be efficiently processed by deep learning models. Unlike existing studies that combine relational-temporal characteristics for forecasting, such as traffic flow [42] and real estate evaluation [43], the linkages between stocks are not well-defined in stock prices prediction. Therefore, it's a vital but challenging task for predicting stock prices movement, which is to construct appropriate graph structures, especially defining effective relations.

Speaking of the relations between stocks, it's crucial to consider and determine what connections are capable of indicating the similarity of price fluctuations, and previous researches have established many kinds of links to build graphs needed from a variety of perspectives. Feng et al. [18] established three relationships that are common-industry, common-customer, common-supplier. Ye et al. [54] designed an industry-based directed-weighted graph, using the ratio of registered capital of companies as weights. Given the assumption that earnings of listed companies might influence their shareholders' stock prices, and in contrast, if this shareholder is also a listed company, its performance would also affect the price of its shareholding company [1, 9] introduced enterprises investments as edges to build graph structures. Similarly, [54] designed an undirected-weighted graph with shareholding ratios as weights. Cheng and Li [12] established connections from the aspect of supply chain, competition, customer, and strategic alliance. In addition, some scholars have concentrated on prices changes themselves, especially the similarity of prices, returns and other attributes, based on which edge relations could be computed and artificially constructed. For instance, [19] calculated the cross-correlation between stocks by the rate of returns, and [55] utilized the correlation coefficient of stock prices sequences to formulate graphs. Generally, there are two ways to define and construct appropriate graph structures needed in stock prices forecasting: one is to explore realistic associations of common features of companies (e.g., in the same industry, the supply chains or stock holdings), while the other is mining the patterns within stock prices or returns by artificially computing similarities or some coefficients, and both have been proven to be positively beneficial.

The literature illustrated above portrays many kinds of relationships between stocks to construct graphs, and they focus on the phenomenon of similar fluctuations of stock prices, such as the well-known stock price synchronicity [45], while fewer researches analyze it from the perspective of essential causes of price changes. As we introduced in Sect. 1, a reasonable assumption has been proposed that major fluctuations of stock prices are mainly triggered by high-volume transactions which usually occur on a group of stocks that share some common features, like stocks in the same industry, region, concept or yield similar volatility. Hence, we connect those stocks sharing the above common features, and formulate four kinds of graphs incorporated with various prior knowledge. This theoretical interpretation has guided us to build graphs needed, and further enhanced the interpretability of our model of GCN-LSTM, with the final results of promoted forecasting achieved in stock prices movement.

3 Methodology

This section presents a comprehensive description of our proposed GCN-LSTM methodology, with necessary mathematical definitions and notations, to indicate the specifics of

three major processes, the graphs construction, the embeddings extraction, and the temporal sequences modeling. The overall architecture of our proposed two-staged integrated GCN-LSTM methodology is shown in Fig. 1, using stock i at time t as an example. We take stocks as nodes, and construct four kinds of knowledge-incorporated graphs, according to the theoretical assumption above. Then we extract node embeddings in multiple time periods using GCN, and get temporal sequences for each stock with labels of prices movement direction, which have been further fed into LSTM for predictions. Our model fully exploits the knowledge-incorporated relations between stocks and considers it into temporal stock embeddings, which encode both relational information and time-varying characteristics of stock prices. Experimental results have demonstrated better forecasting performances in stock prices movements of our model, which will be clarified in the following section.

3.1 Preliminary and graphs construction

Given a dataset containing a group of N stocks from T time periods, for example, a stock index, we denote $\boldsymbol{x}_t^i$

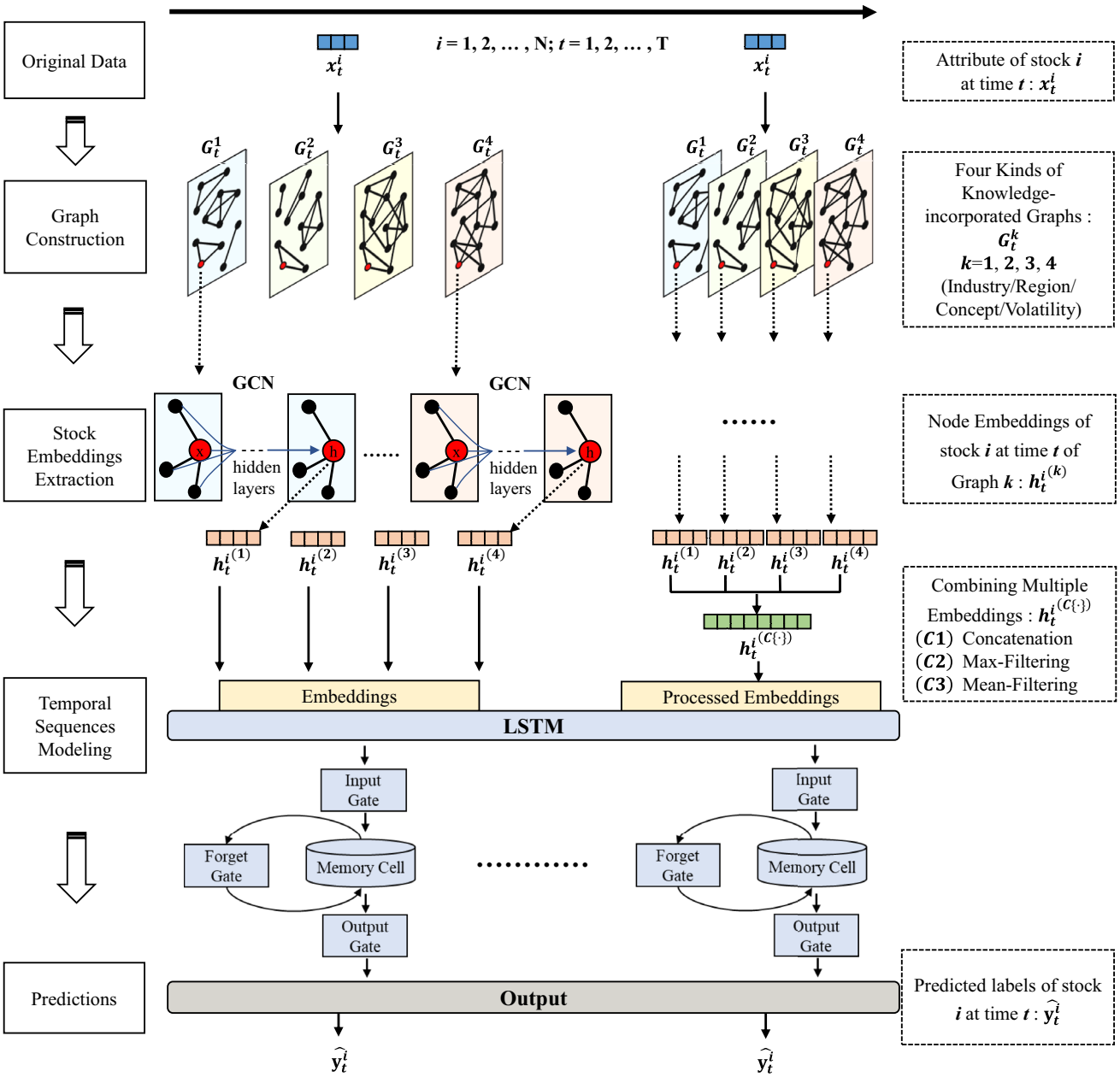

Fig. 1 The overall architecture of our proposed GCN-LSTM methodology

and y_t^i as the attribute vector and its corresponding label of stock $i(i = 1, 2, ..., N)$ at time $t(t = 1, 2, ..., T)$, where the former is usually represented as trading characteristics of stocks, the technical indicators like transactions volume, turnover rate, opening price, highest price, lowest price, closing price, etc. For the label of stocks, given the task of predicting the movement direction of stock prices, we define this prediction task as a binary classification issue, so that the labels depend on the rise or fall of prices in each time period, as shown in formula 1 and the p_t^i indicates the closing price of stock i on day t. Gathering N stocks, the $\mathbf{X}_t = \left[\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, ..., \boldsymbol{x}_t^N\right]^T \in \mathbb{R}^{N\times D}$ and $\boldsymbol{y}_t = [y_t^1, y_t^2, ..., y_t^N]^T \in \mathbb{R}^{N\times 1}$ represent matrixes of stock trading attributes and labels respectively in continuous periods, while D is the dimension of each attribute vector.

$$y_t^i = \begin{cases} 1, \ p_t^i > p_{t-1}^i \\ 0, \ p_t^i \le p_{t-1}^i \end{cases} \tag{1}$$

Considering the simplest form of a graph for node-level tasks, it's commonly composed of nodes, node features, node labels and edges. With the purpose of promoting predictive accuracy of stock prices, the essential part of this study is to formulate the adjacency matrix $\mathbf{A} = \left[a_{ij}\right] \in \mathbb{R}^{N\times N}$ indicating the relationships between stocks (nodes), based on certain prior knowledge. The a_{ij} in $\mathbf{A}$ will be 0 if there's no link between node i and node j, otherwise, it will be 1 in unweighted graphs while in weighted graphs, it's a specific value. Moreover, the symmetry and asymmetry of the adjacency matrix depend on whether the edges in graph are undirected or directed, while the latter is more complicated to analyze.

It's a common sense that the price of a commodity is determined by supply and demand, in particular, the imbalance between supply and demand generate fluctuations of prices. Similar with other commodities, prices of stocks are the same, as the difference in trading volumes between buyers and sellers in a given period of time leads to continuous rising or falling of prices. However, in real stock markets, small amount of buyings/sellings from individual investors have little impact on the prices which will be quickly overwhelmed by competent counterparts. What really causes significant fluctuations is the buying or selling behaviors of institutional investors with large capital flows, like social security funds, pension funds, public funds, etc. And the portfolio allocations of these funds usually tend to focus on stocks with certain commonalities, like in the same industry, registered in the same region, sharing consistent concepts, or yielding similar volatility. Based on this interpretation, we assume the relations of stocks are the following four types of knowledge to better capture and model the hidden fluctuating patterns among stock prices, and to build graph structures that are represented as $\mathbf{G}_t^k = \left\{\mathbf{X}_t, \boldsymbol{y}_t, \mathbf{A}^k\right\}$.

Industry/region graphs: Investment decisions of large funds may be influenced or attracted by some certain policies or events, which are usually issued to a specific sector or region. Thus, if two enterprises belong to the same industry or are registered in the same province, then we add a link between these two nodes (stocks), constructing undirected-unweighted graphs as the values in adjacency matrix are 0 or 1. The industry of each company is identified according to the standards from China Securities Regulatory Commission (CSRC) while the registration information can be obtained from public reports of listed firms.

Concept graphs: "Concept stocks" is a term used as a selecting criterion for portfolios, introducing a group of firms of the same "concept". For example, the suppliers and clients companies related to Apple Inc. can be defined as "Apple Concept Stocks". There remains about 800 concepts in Chinese A Share market, on which trading strategies of many investors are based. Nevertheless, one stock may yield several concepts, so that we adopt a_{ij} as $a_{ij} = \frac{c_{ij}}{c_i}$ to evaluate the relative importance of stock j to stock i, in which c_{ij} refers to the number of concepts that both stock i and j share together, and c_i is the number of concepts that stock i owns. In this way, if two companies share the same concepts, then two edges of different directions are defined between the two nodes, with a_{ij} may not be the same as a_{ji}, finally generating directed-weighted graphs.

Volatility graphs: Considering that companies with similar varying trends of prices may be traded at the same time in high-volume transactions, the volatility of trading characteristics of stocks has been introduced to build graphs. Here we use five widely-used technical indicators as trading attributes and they are Bull And Bear Index (BBI), Commodity Channel Index (CCI), Moving Average Convergence Divergence (MACD), Momentum Index (MTM), and Relative Strength Index (RSI). By computations on sequences of indicators, the edges are established by cosine similarity calculated for each stock under each indicator in a fixed time period, as shown in formula 2, where m indicates the technical indicator. We set a link between stock i and stock j, if the cosine similarity value of one of its technical indicators is in the range of top 10%. Hence, undirected-weighted graphs could be obtained and the edge weight is $a_{ij} = \frac{P}{5}$ with P denoting the number of indicators in the top 10% range.

$$\cos(i, j) = \frac{\sum_t m_t^i * \sum_t m_t^j}{\sqrt{\sum_t m_t^{i^2}} * \sqrt{\sum_t m_t^{j^2}}} \tag{2}$$

Given a collection of N stocks from T periods, graphs with quantity of $4 * T$ could be acquired while each graph consists of N nodes with attributes and labels, and its adjacency matrix is corresponding with one of the definitions of four types of knowledge, providing solid

data foundation for further experimentations. We set the $\mathbf{G}_t^k = \{\mathbf{X}_t, \boldsymbol{y}_t, \mathbf{A}^k\}$ to represent a graph obtained by incorporating relational knowledge k at time t, where $k = [\textbf{Industry}, \textbf{Region}, \textbf{Concept}, \textbf{Volatility}]$. The $\mathbf{A}^k$ won't be changed once the edges have been established while the $\mathbf{X}_t$ and $\boldsymbol{y}_t$ vary over time periods.

3.2 Stock embeddings extraction with GCN

With four kinds of knowledge-incorporated graphs constructed, the first stage of our proposed methodology is to extract node embeddings using GCN. GCN is a pioneering, fundamental and widely-used model in GNN and the field of graph representation learning, of which the basic principle is message passing and information aggregation. By encoding both global graph structures and node attributes, node representations get updated by a layer-propagation rule in formula 3:

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right) \tag{3}$$

in which the adjacency matrix $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, the degree matrix $\tilde{\mathbf{D}} = \text{diag}\left(\sum_j \tilde{\mathbf{A}}_{ij}\right)$, and the $\mathbf{H}^{(l+1)}$, $\mathbf{H}^{(l)}$ are layer-wise updated results. The activation function $\sigma(\cdot)$ often uses *ReLU*, while $\mathbf{W}^l$ is the weight in lth layer. Here we adopt a two-layered GCN, consisting of the input layer, the hidden layer for node embeddings extraction, and the output layer for labels prediction. Utilizing *ReLU* as the activation function and *Softmax* to output binary classification prediction, the general iteration of our developed GCN can be expressed as formula 4:

$$\mathbf{Z}_t^k = \text{Softmax}\left(\hat{\mathbf{A}}^k\text{ReLU}\left(\hat{\mathbf{A}}^k\mathbf{X}_t\mathbf{W}_t^{k(0)}\right)\mathbf{W}_t^{k(1)}\right) \tag{4}$$

in which $\hat{\mathbf{A}}^k = \tilde{\mathbf{D}}^{k^{-\frac{1}{2}}}\tilde{\mathbf{A}}^k\tilde{\mathbf{D}}^{k^{-\frac{1}{2}}}$, and $\tilde{\mathbf{A}}^k = \mathbf{A}^k + \mathbf{I}$ with $\tilde{\mathbf{D}}^k = \text{diag}\left(\sum_j \tilde{\mathbf{A}}_{ij}^k\right)$, according to formula 3 above. The $\mathbf{W}_t^{k(0)}$ and $\mathbf{W}_t^{k(1)}$ are layer-specific trainable weights to be obtained, specifically at time t in knowledge k based graphs.

Based on the specifically designed GCN model above, each graph $\mathbf{G}_t^k$ has been subsequently deployed into full-training process with Cross Entropy loss function utilized to optimize between the output predicted labels $\mathbf{Z}_t^k$ and true labels of stock prices movement. After training is completed, we derive the weight matrix in the first hidden layer of GCN and employ it to generate sequential embeddings of each stocks trained by GCN under various knowledge-incorporated graphs, which can be described as $\mathbf{H}_t^k$ in formula 5:

$$\mathbf{H}_t^k = \hat{\mathbf{A}}^k\mathbf{X}_t\mathbf{W}_t^{k(0)} \tag{5}$$

In this way, temporal sequences of embeddings with length of T have been collected as $\mathbf{H}_t^k$, and the $\mathbf{h}_t^{i(k)}$ denotes the embedding for stock i in graph k at time t, which then will be fed into LSTM for stock prices movement prediction. In here, the dimension of embeddings extracted (the units in hidden layer of GCN) may affect the final forecasting performances, so that it has been regarded as key parameters to be further discussed.

3.3 Temporal sequences modeling by LSTM

RNN is quite good at processing dataset containing sequential information, for example, the time-series data, based on which there have been numerous studies and applications regarding stock markets. However, if the continuous time periods of sequences input into RNN are too long, problems like gradient vanishing or gradient explosion may occur, and that is a major shortcoming of traditional RNN. To overcome this, many well-defined and widely-used variants of RNN have been invented, among which the gate control mechanism performs more effectively, producing typical representatives of LSTM and Gated Recurrent Unit (GRU). Here we choose LSTM as the main classifier in our methodology, also is the second stage of labels prediction.

Yielding a similar structure of neural network, LSTM contains an input layer, one or many hidden layers with memory cells, and an output layer. Each memory cell has three gates including the input gate, the forget gate, and the output gate, deciding which part of information introduced is to be input, remembered or output respectively. Given the trained temporal sequences of node representations incorporating both relational knowledge and time-varying information, we separately feed the four sequences of stock embeddings into LSTM to predict the moving direction of stock prices as a binary classification task. The major process of LSTM is formulated as follows:

$$\boldsymbol{i}_t^i = \sigma(\mathbf{W}_i^i\boldsymbol{h}_t^i + \mathbf{U}_i^i\boldsymbol{h'}_{t-1}^i + \boldsymbol{b}_i^i) \tag{6}$$

$$\boldsymbol{f}_t^i = \sigma(\mathbf{W}_f^i\boldsymbol{h}_t^i + \mathbf{U}_f^i\boldsymbol{h'}_{t-1}^i + \boldsymbol{b}_f^i) \tag{7}$$

$$\boldsymbol{o}_t^i = \sigma(\mathbf{W}_o^i\boldsymbol{h}_t^i + \mathbf{U}_o^i\boldsymbol{h'}_{t-1}^i + \boldsymbol{b}_o^i) \tag{8}$$

$$\boldsymbol{z}_t^i = tanh(\mathbf{W}_z^i\boldsymbol{h}_t^i + \mathbf{U}_z^i\boldsymbol{h'}_{t-1}^i + \boldsymbol{b}_z^i) \tag{9}$$

$$\boldsymbol{c}_t^i = \boldsymbol{i}_t^i \odot \boldsymbol{z}_t^i + \boldsymbol{f}_t^i \odot \boldsymbol{c}_{t-1}^i \tag{10}$$

$$\boldsymbol{h'}_t^i = \boldsymbol{o}_t^i \odot tanh(\boldsymbol{c}_t^i) \tag{11}$$

in which the $\boldsymbol{h}_t^i$ and $\boldsymbol{h'}_t^i$ are input and output sequences indicating stock i at time t, while the $\boldsymbol{i}_t^i$, $\boldsymbol{f}_t^i$, $\boldsymbol{o}_t^i$ denote input gate, forget gate and output gate respectively, with learnable weights of $\mathbf{W}$, $\mathbf{U}$ and bias term of $\boldsymbol{b}$. The predicted label of i at time t obtained by LSTM are represented as $\hat{\boldsymbol{y}}_t^i$ in the following:

$$\hat{\boldsymbol{y}}_t^i = Softmax(\boldsymbol{h'}_t^i \mathbf{W}_{fc}^i) \tag{12}$$

The final output of $\hat{\boldsymbol{y}}_t^i$ is the predicted labels of moving direction of stock i at time t, after getting through a fully-connected layer of *Softmax* while its optimization process uses Cross-Entropy loss function.

Moreover, considering the possible enhancement by fusing various prior knowledge, we have also introduced three means of combinations concerning the four kinds of relations defined in this study, of which the primary thought is consistent with ensemble learning. Particularly, we take three processing methods, the Concatenation, the Mean-Filtering, and the Max-Filtering to integrate the four embeddings extracted from four knowledge-incorporated graphs, combining multiple relations into a processed embedding for each stock at a certain time. The four kinds of original embeddings and three processed embeddings (combinations) are further used as inputs in temporal sequences modeling stage, also the labels prediction by LSTM, to help improve predictive ability of GCN-LSTM and also compare the different impacts on forecasting precisions. The three means of combinations are basically feature aggregation operations, and the Concatenation will expand the dimension of input embeddings, while the Mean-Filtering and the Max-Filtering are similar with the Mean-Pooling and the Max-Pooling, both are dimension reduction techniques in image processing which compress information with important features preserved. In addition, the two filterings don't change the original dimension of embeddings, but enable efficient integration with computational efficiency promoted. Detailed experimentation results and analysis will be reported in the following Sect. 5.

4 Experiments

In this section, we first give a comprehensive overview of dataset used in experiments and how it's been processed, while then is arrangement of baselines with key parameters settings, and finally the evaluating metrics and experimental setup will be presented.

4.1 Data acquisition and processing

In experiments of this study, real stock prices data from Chinese A Share market has been introduced, and we take four well-known and important stock indexes as they are SSE 50, CSI 100, CSI 300, and CSI 500, which respectively contain 50, 100, 300, and 500 stocks and cover the most actively traded group of stocks in China. The continuing time periods have been arranged from 2017-Jan-3 to 2021-Sep-30, totally 1155 labelled transaction days. After delaminating discontinuity of some listed firms and removing incomplete indicators in raw data, it has resulted in slight changes on the quantity of stocks embodied in each index, also the number of nodes formulated in each graph.

Given the collected data above, the time-varying sequences of trading characteristics in four indexes include six primary indicators, and they are trading volume, turnover rate, opening price, lowest price, highest price, closing price, while the labels of prices movement direction are acquired according to formula 1. The prior knowledge information of relations for building graphs, such as the registered places of enterprises, the concepts of stocks, the sectors, and the five technical indicators of volatility, have been obtained from a comprehensive financial information provider with most could be found at public reports. Table 1 is the description of knowledge-incorporated graphs constructed in the first stage of GCN-LSTM, reporting the nodes, the number of edges in four kinds of graphs, and the proportion of rise/fall labels of instances. Table 2 is the description of temporal stock embeddings sequences extracted for the second stage of GCN-LSTM, introducing the number of sequences for training/testing process.

Table 1 Description of knowledge-incorporated graphs

Stock indexes	Number of nodes	Number of edges in four graphs (industry, region, concept, volatility)	Number of total labels (rise/fall)	Time periods (number of graphs)
SSE 50	43	84, 91, 1806, 300	22,519/27,146	1155
CSI 100	85	218, 507, 7140, 1238	46,780/51,395	1155
CSI 300	250	2448, 5686, 62,250, 14,896	136,791/151,959	1155
CSI 500	439	28,209, 6783, 192,282, 57,472	224,308/282,737	1155

Table 2 Description of extracted stock embeddings sequences

Stock indexes	Number of stocks	Number of sequences for training	Number of labels for training (rise/fall)	Number of sequences for testing	Number of labels for testing (rise/fall)
SSE 50	43	34,013	15,122/18,891	14,878	7003/7875
CSI 100	85	67,235	32,131/35,104	29,410	13,817/15,593
CSI 300	250	197,750	93,806/103,944	86,500	41,441/45,059
CSI 500	439	347,249	149,872/197,377	151,894	70,445/81,449

4.2 Baselines selection and parameters

In order to perform comparative experiments and verify the effectiveness of our proposed GCN-LSTM regarding stock prices prediction, various conventional methods have been selected as baselines into experiments, especially the widely-used machine learning techniques. Here we take several representatives including tree-based models of DT, ensemble learning of RF, AdaBoost, XGBoost, GBDT, Multi-Layer Perceptron (MLP) in artificial neural network, SVM, with the data usage of structured tabular vectors and experiments of simple classifications. Additionally, LSTM has also been introduced as a baseline, based on the temporal sequences of multivariate trading indicators, to reveal the critical value and improving effect of relations inclusion. Key parameters settings of GCN-LSTM and baselines are presented in Table 3 while machine learning models are carried out by a Python-based repository called Scikit-learn with LSTM, GCN-LSTM implemented in a deep learning framework of PyTorch.

4.3 Evaluation and experimental setup

Considering that the distribution of labels is relatively well-balanced in our collected stock dataset, as shown in Tables 1 and 2, conventional measures in classification tasks like the Accuracy (Acc.) and the F1-Score (F1) could be utilized as evaluating metrics for our experiments. All values of metrics reported in the following evaluations are arithmetic means and standard deviations calculated based on five times of experiments. Before getting into the formulation of knowledge-incorporated graphs, all six primary indicators for each stock have been normalized sequentially along the timeline of transaction days. Given the temporal sequences of node embeddings extracted, we divide all sequences into two parts, of which the first 70% is training set and the rest 30% is for the use of testing. For transparent and fair comparisons, in constructing volatility based graphs, we employ the first 70% of total trading days included in sequences to measure the similarity of technical indicators, as only the information in training set has been used. Moreover, it's been adopted that the paradigm of implementing 10 historically consecutive days to predict the next day's stock prices movement. For conventional machine learning techniques, the inputs are 10-days-stacked vectors sized $60 * 1$ and each contains 6 trading attributes, while the inputs of LSTM and GCN-LSTM are 10-days sequences. In our proposed GCN-LSTM, the dimension of each embedding in temporal sequences is determined by the number of units in hidden layer of GCN, and its influences on forecasting performances will be discussed in the following.

Table 3 Key parameters settings of models in experiments

Models in experiments	Key parameters settings
DT	Splitting criterion = gini, splitter = best
RF	Number of estimators = 100, splitting criterion = gini
AdaBoost	Number of estimators = 100, learning rate = 0.5, algorithm = samme
XGBoost	Number of estimators = 100, learning rate = 0.1, max_depth=5
GBDT	Number of estimators = 100, learning rate = 0.1, max_depth=5
MLP	Activation function = relu, optimization solver = adam, size of hidden layer = (30, 10)
SVM	Kernel function = rbf, gamma = 10
LSTM	Seq = 10, batchsize = 50, total layers = 2, hidden units = 32, optimization solver = adam, learning rate = 0.01, dropout = 0.3
GCN-LSTM	For GCN: total layers = 3, input units = 6, output units = 2; For LSTM: the same as above arrangements.

5 Results and analysis

This section comprehensively illustrates the results of experimentations and first we compare GCN-LSTM with baselines regarding the best predictions achieved. To investigate the optimal selection of key parameters, the comparison of GCN-LSTM using different graph structures including four kinds of original embeddings and three processed embeddings, and sensitivity analysis of dimension of extracted node embeddings are presented. Furthermore, two statistical tests as well as computational efficiency analysis have been performed to ensure the reliability and adaptability of our model.

5.1 Predictions of our proposed GCN-LSTM with baselines

Table 4 is the general comparison of predictions achieved by our proposed GCN-LSTM and baselines, in which the bolded highlight the best one of that column. All values of GCN-LSTM here are the optimal results realized under various graphs with dimension of node embeddings set as 16, in particular the four kinds of original embeddings and the three types of processed embeddings. Obviously, the GCN-LSTM method has outperformed all eight baselines by a marginal promotion of 2–8% in terms of Accuracy and F1-score on four stock indexes, with good efficiency in dealing with node-level tasks of stock prices prediction obtained. In addition, the direct implementation of LSTM with usage of temporal sequences of multivariate trading indicators, has shown better forecasting compared with the rest seven machine learning models, benefiting from the inclusion of time-varying information and the advantage of LSTM in processing time-series. This implies that the introduction of temporal information is capable of improving the predictive ability, compared with simple classifications of structured tabular vectors. Also, by the prior knowledge of relations extracted by GCN and the strength of LSTM in sequential modelling, our proposed model has successfully led to more superior classification performances regarding stock prices movement prediction, and further justified the theoretical assumption put forward earlier.

5.2 Predictions of GCN-LSTM under different graph structures

Based on the graph construction process of our proposed methodology, four kinds of relational knowledge-incorporated graphs including Industry, Region, Concept, Volatility have been built with three kinds of combinations processed by Concatenation, Mean-Filtering, Max-Filtering. The total seven types of embedding sequences have been separately fed into GCN-LSTM, to conduct comparative experiments and reveal the differences. Table 5 presents the predictions of GCN-LSTM under various graphs and their combinations, while the bolded values highlight the best of that column. It can be seen that, among four kinds of original embeddings, the results obtained by Concept and Volatility graphs are slightly better than the Region and Industry graphs, due to the different graph properties. The Concept graph is directed-weighted, and the Volatility graph is undirected-weighted, of which both provide more informative gains, while the latter two are

Table 4 Predictions comparison of GCN-LSTM and baselines

	Stock indexes	SSE 50		CSI 100		CSI 300		CSI 500	
	Model	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Structured tabular vectors	DT	49.84 ± 0.29	48.01 ± 1.17	50.18 ± 0.30	47.34 ± 0.46	49.89 ± 0.09	48.39 ± 0.19	50.14 ± 0.09	47.37 ± 0.12
	RF	49.67 ± 0.36	44.93 ± 1.20	50.53 ± 0.19	45.46 ± 0.40	50.21 ± 0.14	44.40 ± 0.26	50.61 ± 0.20	44.91 ± 1.14
	AdaBoost	50.12 ± 0.03	47.16 ± 0.07	50.09 ± 0.03	49.18 ± 0.06	49.93 ± 0.02	46.06 ± 0.11	50.43 ± 0.02	45.39 ± 0.05
	XGBoost	49.86 ± 0.20	44.90 ± 0.28	50.04 ± 0.05	43.80 ± 0.34	50.05 ± 0.11	43.81 ± 0.12	50.61 ± 0.04	44.67 ± 0.15
	GBDT	50.04 ± 0.15	47.45 ± 0.27	50.35 ± 0.15	46.50 ± 0.51	50.03 ± 0.05	47.09 ± 0.10	50.46 ± 0.06	46.25 ± 0.19
	MLP	52.30 ± 0.17	40.90 ± 0.50	52.57 ± 0.09	35.59 ± 0.82	51.45 ± 0.05	35.87 ± 0.34	52.41 ± 0.06	29.48 ± 0.26
	SVM	51.76 ± 0.03	45.17 ± 0.07	51.08 ± 0.45	42.38 ± 1.39	51.28 ± 0.21	43.00 ± 0.10	51.89 ± 0.01	35.74 ± 0.05
Temporal sequences of multivariate trading indicators	LSTM	53.51 ± 0.33	62.31 ± 0.80	54.24 ± 0.71	61.48 ± 1.59	51.45 ± 0.29	56.23 ± 0.56	53.21 ± 0.31	61.91 ± 0.53
Temporal sequences of extracted stock embeddings	GCN-LSTM	**57.81 ± 0.24**	**67.92 ± 0.86**	**56.94 ± 0.12**	**67.16 ± 0.09**	**55.05 ± 0.37**	**64.21 ± 0.45**	**57.32 ± 0.28**	**69.16 ± 0.32**

Table 5 Various graphs and combinations

	Stock indexes	SSE 50		CSI 100		CSI 300		CSI 500	
	Graph	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Original embeddings	Industry	54.95 ± 0.44	65.11 ± 0.81	55.02 ± 0.43	66.70 ± 0.61	53.49 ± 0.28	62.22 ± 0.28	54.21 ± 0.64	67.45 ± 0.55
	Region	55.05 ± 0.46	64.55 ± 0.83	54.42 ± 0.23	66.64 ± 0.44	53.35 ± 0.12	61.69 ± 0.37	54.88 ± 0.14	**70.10 ± 0.19**
	Concept	55.20 ± 0.50	64.78 ± 0.35	55.66 ± 0.54	67.01 ± 0.43	54.36 ± 0.18	63.04 ± 0.37	55.33 ± 0.15	67.28 ± 0.13
	Volatility	56.03 ± 0.48	65.59 ± 0.87	55.06 ± 0.31	66.61 ± 0.52	54.28 ± 0.15	62.15 ± 0.22	55.74 ± 0.11	66.92 ± 0.12
Processed/combined embeddings	Concatenation	56.46 ± 0.38	66.78 ± 1.00	55.49 ± 0.31	66.20 ± 0.14	54.43 ± 0.28	62.64 ± 0.31	56.49 ± 0.21	67.20 ± 0.53
	Mean-filtering	**57.81 ± 0.24**	**67.92 ± 0.86**	56.31 ± 0.12	67.11 ± 0.46	54.53 ± 0.18	63.49 ± 0.61	57.32 ± 0.28	69.16 ± 0.32
	Max-filtering	56.61 ± 0.21	67.28 ± 0.86	**56.94 ± 0.12**	**67.16 ± 0.09**	**55.05 ± 0.37**	**64.21 ± 0.45**	**57.32 ± 0.20**	68.82 ± 0.16

undirected-unweighted. Furthermore, from the view of combinations, the integration of multiple relations presented as processed embeddings has generated enhancing effects on stock prices forecasting, with promotions of 1–2% in Accuracy and F1 compared with single relation based graphs. It has exhibited the reasonability of our formulated graph structures under the guidance of the theoretical assumption introduced above, and also provided a feasible direction for further improvements by appropriate fusions. Comparing the three combining means, the Mean-Filtering and the Max-Filtering operations have performed better than the Concatenation. And it has to be noted that all values acquired in here are based on the setting of node embeddings dimension as 16, thus, the sensitivity analysis of this key parameter is also necessary to be carried out.

5.3 Predictions of GCN-LSTM under various embedding dimensions

The above experimental results have already confirmed the effectiveness of our proposed graph structures and GCN-LSTM method. Nevertheless, considering the possible influence of node embeddings dimension in GCN-LSTM, sensitivity analysis of this key parameter is necessary as the informative gains embodied in stock embeddings extracted may be different in various dimensions settings. For example, it's insufficiently expressed when the dimension is too small, and large ones are too sparse, making it hard to distinguish from others. Based on that, the value settings of node embeddings dimensions has become a hyperparameter in our methodology of GCN-LSTM. We take a group of selections into experiments, including 4, 8, 16, 32 and 64, also the hidden units in GCN, to analyze its effect with seven types of embeddings. Figure 2 indicates the predictions comparison of GCN-LSTM under various node embedding dimensions in four stock indexes, with mean and standard deviation of both Accuracy and F1 Score reported by curves. It can be concluded that GCN-LSTM is sensitive to the changes of embeddings size, while its variations have produced dramatic impacts on forecasting performances of stock prices movement. Too small or too large values is not advantageous to the prediction capability of GCN-LSTM, while the optimal selection of node embedding is 16 that has yielded more superior results in most cases, and is also the same in all previous experimentations.

5.4 Complexity analysis and statistical significance

To ensure the reliability and adaptability of our model GCN-LSTM, complexity analysis and statistical significance tests have been conducted in this subsection. Figure 3 shows the computational efficiency comparison of all experimented models in the training process. We display it in the form of time consumption in seconds averaged by the number of stocks totally, for more clear presentation. It can be seen obviously that our proposed GCN-LSTM, even the three combinations, yield less computations compared with direct usage of LSTM, demonstrating both outstanding and efficient predictions. The converging speed of GCN-LSTM is much faster than LSTM, because the input of GCN-LSTM in this study is relational knowledge-incorporated node embeddings that could be processed more efficiently than the original trading indicators sequences.

Moreover, two well-known statistical tests have been carried out as indicated by [15] and widely-applied in many researches, the Wilcoxon signed-rank test and the Friedman test. Both are designed for pairwise comparisons of multiple objectives, and we have performed these two tests under the one-tail-test with a significance of $\alpha = 0.005$ (see [33, 57] for more details). Three dimensions are introduced to identify whether there remains a statistically significant difference between the predictive results achieved by the two pair models, and Table 6

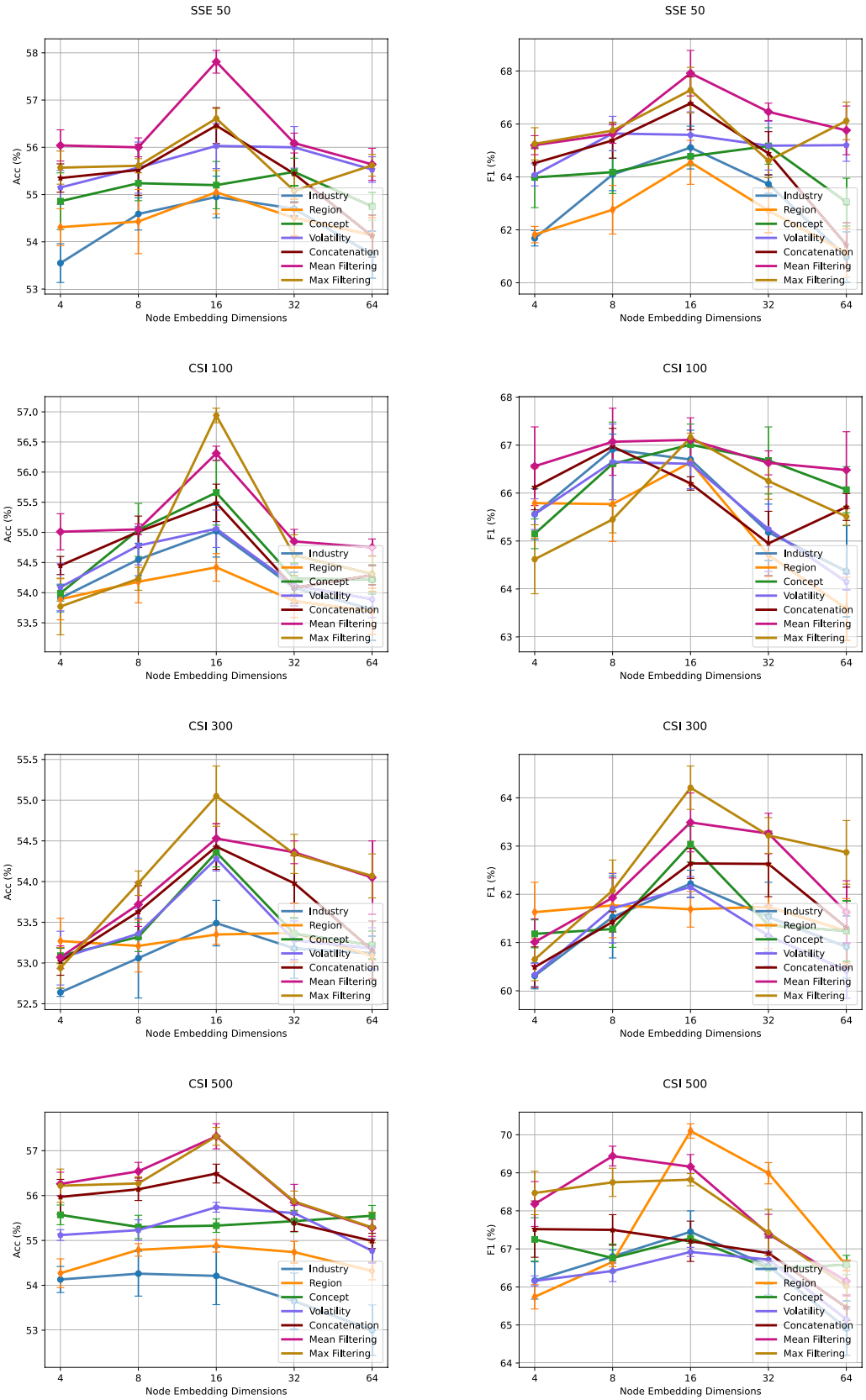

Fig. 2 Predictions comparison of GCN-LSTM under various node embedding dimensions

presents the overall results. In the first dimension, we have implemented statistical tests on GCN-LSTM (Max-filtering) with 8 baseline models, and it's concluded that GCN-LSTM (Max-filtering) is significantly different from baselines, again its enhancement on forecasting performances has been verified. In the second dimension, we have compared the results of GCN-LSTM (Max-filtering) with the other four kinds of single relation based graphs, with the findings obtained that multiple relations fusion based graphs outperform the original one relation based graphs. Thirdly is the two-by-two comparison of three combinations of GCN-LSTM, and it reveals that there's no critical difference between GCN-LSTM (Max-filtering) and GCN-LSTM (Mean-filtering) while both have realized better predictions than the concatenation.

6 Conclusion

In this article, we have developed a two-staged integrated GCN-LSTM methodology for improved stock prices movement prediction, with stock embeddings extraction using GCN and temporal sequences modeling by LSTM. Based on the theoretical interpretation of huge stock prices fluctuations, we have introduced four kinds of prior knowledge of relations between stocks, combined with the internal information of trading characteristics, to establish the graphs needed. The stock embeddings sequences extracted encode both relational information and time-varying characteristics of stock prices, making it easier to distinguish or discriminate. Extensive experiments on major Chinese stock indexes have not only proved the reasonability of our assumption but further enhanced the interpretability of our model, with more effective forecasting acquired by the best accuracy of 57.81%, compared with other conventional machine learning approaches and deep learning method. Additionally, the predictive ability of GCN-LSTM under different graphs and various node embedding dimensions has been analyzed, to indicate the optimal arrangement of key parameters, with complexity and statistical tests conducted to ensure the reliability and adaptability of GCN-LSTM. Our findings provide a promising direction in forecasting stock prices movement and exhibit insights for further applications of graph neural networks and graph data in business analytics, quantitative finance, and risk management decision-makings.

Though the GCN-LSTM demonstrates good results by integrating relational knowledge with temporal information, it could also be further promoted by incorporating more kinds of external relations. In the process of constructing graphs, instead of using all stocks as nodes, maybe it's better to only include those most-closely-related companies, which would reduce the computations in graph formulation and modeling. Furthermore, as widely recognized in many studies, finding the linkages between stocks with more meaningful implications is beneficial to more reliable enhancement, because the utilization of real relationships is capable of greatly promoting the models' representation and learning capability of graph data. Given that our research only builds four kinds of knowledge-incorporated graphs, many other

Table 6 Results of Wilcoxon signed-rank test and Friedman test

		Wilcoxon signed-rank test $\alpha = 0.005$ p value	Friedman test
Dimension 1			
GCN-LSTM (Max-Filtering) vs.	DT	0.0001	$H_0 : e_1 = e_2 = e_3 = e_4 = e_5 = e_6 = e_7 = e_8 = e_9$ $F = 142.235$ $p = 0.000$ (Reject H_0)
	RF	0.0001	
	AdaBoost	0.0001	
	XGBoost	0.0001	
	GBDT	0.0001	
	MLP	0.0001	
	SVM	0.0001	
	LSTM	0.0001	
Dimension 2			
GCN-LSTM (Max-Filtering) vs.	GCN-LSTM (Industry)	0.0001	$H_0 : e_1 = e_2 = e_3 = e_4 = e_5$ $F = 64.780$ $p = 0.000$ (Reject H_0)
	GCN-LSTM (Region)	0.0001	
	GCN-LSTM (Concept)	0.0001	
	GCN-LSTM (Volatility)	0.0001	
Dimension 3			
GCN-LSTM (Max-Filtering)	vs. GCN-LSTM (Mean-Filtering)	0.3684	
GCN-LSTM (Mean-Filtering)	vs. GCN-LSTM (Concatenation)	0.0001	–
GCN-LSTM (Max-Filtering)	vs. GCN-LSTM (Concatenation)	0.0001	

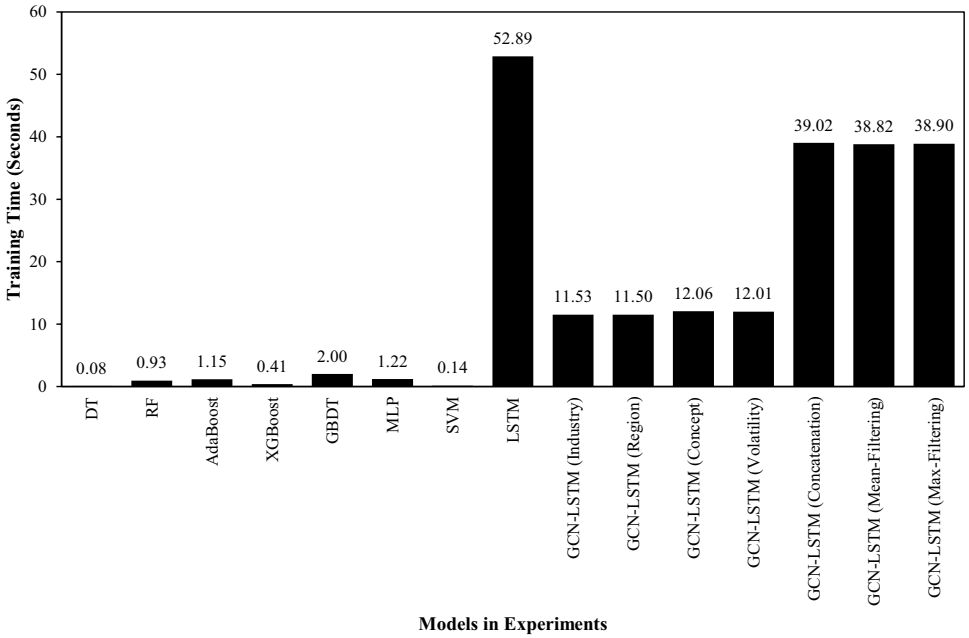

Fig. 3 Computational efficiency comparison of GCN-LSTM and baselines (averaged by the number of stocks)

types of relations to be introduced still need to be investigated in depth. This is also constrained by the availability of data as some connections between companies (e.g., supply chain) can't be accessed publicly. Thus, perhaps it could be tackled by collecting, constructing a knowledge repository of stock relationships, and keeping it constantly updated, with the final aim of providing solid data foundations for further applications of graph neural networks and graph data in related areas and tasks.

Acknowledgements All authors express sincere gratitude to reviewers and editors of International Journal of Machine Learning and Cybernetics for their valuable comments and careful work, with special thanks to Dr. Yunlong Mi from Central South University for his insights and supports that helped this work improved substantially.

Funding This work has been supported by Key Projects (Grants number 71932008, 72231010) and Youth Project (Grant number 71901155) of National Natural Science Foundation of China.

Data availability statement All stock prices and indicators utilized in experiments are collected from a major financial information provider in China, Wind Information Co., Ltd (https://www.wind.com.cn/Default.html).

Declarations

Conflict of interest All authors have no relevant conflict of interests to declare.

References

1. Attig N, Fong WM, Gadhoum Y, Lang LHP (2006) Effects of large shareholding on information asymmetry and stock liquidity. J Bank Finance 30(10):2875–2892. https://doi.org/10.1016/j.jbankfin.2005.12.002
2. Babu CN, Reddy BE (2014) A moving-average filter based hybrid arima-ann model for forecasting time series data. Appl Soft Comput 23:27–38. https://doi.org/10.1016/j.asoc.2014.05.028
3. Bhosale YH, Patnaik KS (2022) Application of deep learning techniques in diagnosis of COVID-19 (coronavirus): a systematic review. Neural Process Lett. https://doi.org/10.1007/s11063-022-11023-0
4. Bildirici M, Ersin Özgür Ömer (2009) Improving forecasts of garch family models with the artificial neural networks: an application to the daily returns in Istanbul stock exchange. Expert Syst Appl 36(4):7355–7362. https://doi.org/10.1016/j.eswa.2008.09.051
5. Chandola D, Mehta A, Singh S, Tikkiwal VA, Agrawal H (2022) Forecasting directional movement of stock prices using deep learning. Ann Data Sci. https://doi.org/10.1007/s40745-022-00432-6
6. Chandar S K (2021) Hybrid models for intraday stock price forecasting based on artificial neural networks and metaheuristic algorithms. Pattern Recognit Lett 147:124–133. https://doi.org/10.1016/j.patrec.2021.03.030
7. Chen C, Zhao L, Bian J, Liu TY (2019) Investment behaviors can tell what inside: exploring stock intrinsic properties for stock trend prediction. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2019), Association for Computing Machinery, New York, NY, USA, pp 2376–2384. https://doi.org/10.1145/3292500.3330663
8. Chen W, Yeo CK, Lau CT, Lee BS (2018) Leveraging social media news to predict stock index movement using RNN-boost. Data Knowl Eng 118:14–24. https://doi.org/10.1016/j.datak.2018.08.003
9. Chen Y, Wei Z, Huang X (2018) Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In: Proceedings of the 27th ACM international conference on information and knowledge management (CIKM 2018),

Association for Computing Machinery, New York, NY, USA, pp 1655–1658. https://doi.org/10.1145/3269206.3269269
10. Chen Y, Wu J, Wu Z (2022) China's commercial bank stock price prediction using a novel k-means-lstm hybrid approach. Expert Syst Appl 202:117370. https://doi.org/10.1016/j.eswa.2022.117370
11. Chen YC, Huang WC (2021) Constructing a stock-price forecast CNN model with gold and crude oil indicators. Appl Soft Comput 112:107760. https://doi.org/10.1016/j.asoc.2021.107760
12. Cheng R, Li Q (2021) Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In: Proceedings of the AAAI conference on artificial intelligence (AAAI 2021), Palo Alto, CA, USA, pp 55–62. https://doi.org/10.1609/aaai.v35i1.16077
13. Coşkun M, Koyutürk M (2021) Node similarity-based graph convolution for link prediction in biological networks. Bioinformatics 37(23):4501–4508. https://doi.org/10.1093/bioinformatics/btab464
14. De Pontes LS, Rêgo LC (2022) Impact of macroeconomic variables on the topological structure of the Brazilian stock market: a complex network approach. Phys A Stat Mech Appl 604:127660. https://doi.org/10.1016/j.physa.2022.127660
15. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18. https://doi.org/10.1016/j.swevo.2011.02.002
16. Emenogu NG, Adenomon MO, Nweze NO (2020) On the volatility of daily stock returns of total Nigeria plc: evidence from garch models, value-at-risk and backtesting. Financ Innov 6(1):1–25. https://doi.org/10.1186/s40854-020-00178-1
17. Esmaeilpour Moghadam HE, Mohammadi T, Kashani MF, Shakeri A (2019) Complex networks analysis in Iran stock market: the application of centrality. Phys A Stat Mech Appl 531:121800. https://doi.org/10.1016/j.physa.2019.121800
18. Feng F, He X, Wang X, Luo C, Liu Y, Chua TS (2019) Temporal relational ranking for stock prediction. ACM Trans Inf Syst 37(2):1–30. https://doi.org/10.1145/3309547
19. Feng S, Xu C, Zuo Y et al (2022) Relation-aware dynamic attributed graph attention network for stocks recommendation. Pattern Recognit 121:108119. https://doi.org/10.1016/j.patcog.2021.108119
20. Gao J, Ying X, Xu C et al (2021) Graph-based stock recommendation by time-aware relational attention network. ACM Trans Knowl Discov Data 16(1):1–21. https://doi.org/10.1145/3451397
21. Ghosh P, Neufeld A, Sahoo JK (2022) Forecasting directional movements of stock prices for intraday trading using lstm and random forests. Finance Res Lett 46:102280. https://doi.org/10.1016/j.frl.2021.102280
22. Gunduz H, Yaslan Y, Cataltepe Z (2017) Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. Knowl Based Syst 137:138–148. https://doi.org/10.1016/j.knosys.2017.09.023
23. Guoying Z, Ping C (2017) Forecast of yearly stock returns based on adaboost integration algorithm. In: 2017 IEEE international conference on smart cloud, New York, NY, USA, pp 263–267. https://doi.org/10.1109/SmartCloud.2017.49
24. Hao PY, Kung CF, Chang CY et al (2021) Predicting stock price trends based on financial news articles and using a novel twin support vector machine with fuzzy hyperplane. Appl Soft Comput 98:106806. https://doi.org/10.1016/j.asoc.2020.106806
25. Hoseinzade E, Haratizadeh S (2019) CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Syst Appl 129:273–285. https://doi.org/10.1016/j.eswa.2019.03.029
26. Hou X, Wang K, Zhong C, Wei Z (2021) ST-Trader: a spatial-temporal deep neural network for modeling stock market movement. IEEE/CAA J Autom Sin 8(5):1015–1024. https://doi.org/10.1109/JAS.2021.1003976
27. Kanwal A, Lau MF, Ng SP et al (2022) BiCuDNNLSTM-1dCNN-a hybrid deep learning-based predictive model for stock price prediction. Expert Syst Appl 202: 117123. https://doi.org/10.1016/j.eswa.2022.117123
28. Karnyoto AS, Sun C, Liu B et al (2022) Augmentation and heterogeneous graph neural network for AAAI2021-COVID-19 fake news detection. Int J Mach Learn Cybern 13:2033–2043. https://doi.org/10.1007/s13042-021-01503-5
29. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: 5th international conference on learning representations (ICLR 2017), Toulon, France. https://openreview.net/pdf?id=SJU4ayYgl
30. Kohli PPS, Zargar S, Arora S et al (2019) Stock prediction using machine learning algorithms. In: Applications of Artificial Intelligence Techniques in Engineering, Advances in Intelligent Systems and Computing, vol 698, Springer, Singapore, pp 405–414. https://doi.org/10.1007/978-981-13-1819-1_38
31. Kong A, Zhu H, Azencott R (2021) Predicting intraday jumps in stock prices using liquidity measures and technical indicators. J Forecast 40(3):416–438. https://doi.org/10.1002/for.2721
32. Kumar R, Kumar P, Kumar Y (2022) Three stage fusion for effective time series forecasting using Bi-LSTM-ARIMA and improved DE-ABC algorithm. Neural Comput Appl 34:18421–18437. https://doi.org/10.1007/s00521-022-07431-x
33. Li MW, Xu DY, Geng J, Hong WC (2022) A hybrid approach for forecasting ship motion using CNN-GRU-AM and GCWOA. Appl Soft Comput 114:108084. https://doi.org/10.1016/j.asoc.2021.108084
34. Li W, Bao R, Harimoto K, Chen D, Xu J, Su Q (2020) Modeling the stock relation with graph network for overnight stock movement prediction. In: Proceedings of the 29th international joint conference on artificial intelligence (IJCAI 2020), pp 4541–4547. https://doi.org/10.24963/ijcai.2020/626
35. Liu G, Ma W (2022) A quantum artificial neural network for stock closing price prediction. Inf Sci 598:75–85. https://doi.org/10.1016/j.ins.2022.03.064
36. Liu Q, Tao Z, Tse Y et al (2022) Stock market prediction with deep learning: the case of china. Finance Res Lett 46:102209. https://doi.org/10.1016/j.frl.2021.102209
37. Liu S, Li T, Ding H et al (2020) A hybrid method of recurrent neural network and graph neural network for next-period prescription prediction. Int J Mach Learn Cybern 11(12):2849–2856. https://doi.org/10.1007/s13042-020-01155-x
38. Lohrmann C, Luukka P (2019) Classification of intraday S&P500 returns with a random forest. Int J Forecast 35(1):390–407. https://doi.org/10.1016/j.ijforecast.2018.08.004
39. Manessi F, Rozza A (2021) Graph-based neural network models with multiple self-supervised auxiliary tasks. Pattern Recognit Lett 148:15–21. https://doi.org/10.1016/j.patrec.2021.04.021
40. Nakagawa K, Yoshida K (2022) Time-series gradient boosting tree for stock price prediction. Int J Data Min Model Manag 14(2):110–125. https://doi.org/10.1504/IJDMMM.2022.123357
41. Pan Y, Xiao Z, Wang X et al (2017) A multiple support vector machine approach to stock index forecasting with mixed frequency sampling. Knowl Based Syst 122:90–102. https://doi.org/10.1016/j.knosys.2017.01.033
42. Peng H, Du B, Liu M et al (2021) Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. Inf Sci 578:401–416. https://doi.org/10.1016/j.ins.2021.07.007

43. Peng H, Li J, Wang Z et al (2023) Lifelong property price prediction: a case study for the Toronto real estate market. IEEE Trans Knowl Data Eng 35(3):2765–2780. https://doi.org/10.1109/TKDE.2021.3112749
44. Qiao J, Wang L, Duan L (2021) Sequence and graph structure co-awareness via gating mechanism and self-attention for session-based recommendation. Int J Mach Learn Cybern 12(9):2591–2605. https://doi.org/10.1007/s13042-021-01343-3
45. Roll R (1988) R2. J Finance 43(3):541–566. https://doi.org/10.1111/j.1540-6261.1988.tb04591.x
46. Schlichtkrull M, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M (2018) Modeling relational data with graph convolutional networks. In: The semantic web: European semantic web conference (ESWC 2018), Lecture Notes in Computer Science, vol 10843, Springer, Cham, pp 593–607. https://doi.org/10.1007/978-3-319-93417-4_38
47. Sezer OB, Ozbayoglu AM (2018) Algorithmic financial trading with deep convolutional neural networks: time series to image conversion approach. Appl Soft Comput 70:525–538. https://doi.org/10.1016/j.asoc.2018.04.024
48. Tang H, Dong P, Shi Y (2019) A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points. Appl Soft Comput 78:685–696. https://doi.org/10.1016/j.asoc.2019.02.039
49. Wan X, Cen L, Chen X et al (2022) A novel multiple temporal-spatial convolution network for anode current signals classification. Int J Mach Learn Cybern 13:3299–3310. https://doi.org/10.1007/s13042-022-01595-7
50. Wang L, Ma F, Liu J et al (2020) Forecasting stock price volatility: new evidence from the GARCH-MIDAS model. Int J Forecast 36(2):684–694. https://doi.org/10.1016/j.ijforecast.2019.08.005
51. Wang X, Li J, Yang L et al (2021) Weakly-supervised learning for community detection based on graph convolution in attributed networks. Int J Mach Learn Cybern 12(12):3529–3539. https://doi.org/10.1007/s13042-021-01400-x
52. Xie Y, Yao C, Gong M et al (2020) Graph convolutional networks with multi-level coarsening for graph classification. Knowl Based Syst 194:105578. https://doi.org/10.1016/j.knosys.2020.105578
53. Xu W, Liu W, Xu C, Bian J, Yin J, Liu TY (2021) Rest: relational event-driven stock trend forecasting. In: Proceedings of the Web Conference 2021 (WWW 21), Association for Computing Machinery, New York, NY, USA, pp 1–10. https://doi.org/10.1145/3442381.3450032
54. Ye J, Zhao J, Ye K, Xu C (2021) Multi-graph convolutional network for relationship-driven stock movement prediction. In: 25th international conference on pattern recognition (ICPR), Milan, Italy, pp 6702–6709. https://doi.org/10.1109/ICPR48806.2021.941269
55. Yin X, Yan D, Almudaifer A, Yan S, Zhou Y (2021) Forecasting stock prices using stock correlation graph: a graph convolutional network approach. In: 2021 international joint conference on neural networks (IJCNN), Shenzhen, China, pp 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533510
56. Yujun Y, Yimei Y, Wang Z (2021) Research on a hybrid prediction model for stock price based on long short-term memory and variational mode decomposition. Soft Comput 25(21):13513–13531. https://doi.org/10.1007/s00500-021-06122-4
57. Zhang Z, Hong WC (2021) Application of variational mode decomposition and chaotic grey wolf optimizer with support vector regression for forecasting electric loads. Knowl Based Syst 228:107297. https://doi.org/10.1016/j.knosys.2021.107297
58. Zhao J, Zeng D, Liang S, Kang H, Liu Q (2021) Prediction model for stock price trend based on recurrent neural network. J Ambient Intell Humaniz Comput 12(1):745–753. https://doi.org/10.1007/s12652-020-02057-0
59. Zhong X, Enke D (2017) Forecasting daily stock market return using dimensionality reduction. Expert Syst Appl 67:126–139. https://doi.org/10.1016/j.eswa.2016.09.027
60. Zhou F, Zhang Q, Sornette D, Jiang L (2019) Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices. Appl Soft Comput 84:105747. https://doi.org/10.1016/j.asoc.2019.105747