Engineering Applications of Artificial Intelligence 126 (2023) 106948

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Stock Price Movement Prediction based on Relation Type guided Graph Convolutional Network☆

Hao Peng [a,*], Ke Dong [b], Jie Yang [c]

[a] School of Computing, Newcastle University, Newcastle, NE1 7RU, UK
[b] School of Computer and Science, Hefei University of Technology, Hefei, 230009, China
[c] Research Institute of Information Technology, Tsinghua University, Beijing, 100084, China

ARTICLE INFO

Keywords:
Stock price movement prediction
Relation type guided graph convolutional network
External attention mechanism
Real-time interaction
Dynamics of relation
Iterative update

ABSTRACT

Stock Price Movement Prediction (SPMP) task tends to predict the future price fluctuation of some stocks, which is challenging because of the volatile nature of financial markets. Recent researches widely introduce Graph Convolutional Network (GCN) and achieve some competitive performances for SPMP. However, their works usually only extract the signal representations of individual stock in the phase of feature extraction whereas they ignore the real-time interactions among companies along the signal channels. In addition, when the financial relations among different entities (i.e., companies and executives) participate in the aggregation of GCN during feature encoding, the recent advances fail to consider the dynamics of the relation type representations and the iterative update between entities and relations. To address this problem, we propose a novel method consisting of an External Attention (EA) module and a Relation Type guided Graph Convolutional Network (RTGCN) for SPMP. In detail, to achieve the real-time interactions among companies along the signal channels, we introduce an external attention mechanism to share the company memory of the price and sentiment signals. Moreover, the proposed RTGCN considers the dynamics of the relation type embeddings and achieves iterative update between entity nodes and relation edges in the graph. The experiments on two open-source datasets (i.e., CSI100E and CSI300E) demonstrate that the proposed SPMP method can achieve state-of-the-art performances and can assuredly bring benefits for the performance improvement of SPMP.

1. Introduction

Stock Price Movement Prediction (SPMP) task aims to predict the price fluctuation of some companies in the stock market. As an important component in the quantitative trading system, an SPMP system can assist in investors automatically achieving high-speed and accurate decisions for higher benefits during long-term or short-swing trading. However, due to the stochastic nature of stocks and confused stock information, it is not easy to capture valuable signals from a bulk of stock information to accomplish accurate stock movement prediction (Cheng and Li, 2021; Wang et al., 2021; Hu et al., 2018). Thus, the researches of SPMP task are very significant for preparing novel SPMP systems to serve real financial transaction scenarios better.

As a binary classification task to predict the “rise” or “fall” of the stock price in a trading day, the early efforts for SPMP mainly depend on the time-series analysis techniques which desire to find a general trading pattern in the historical trading information (e.g., opening price, closing price, and capitalization) (Feng et al., 2019a; Lin et al., 2017; Feng et al., 2019b). Recently, the quick development of deep learning techniques has enlightened some researchers to present a series of competitive methods for SPMP (Feng et al., 2019b; Xu and Cohen, 2018) which can automatically capture effective market signals from the historical trading data and the off-market data, such as news (Li et al., 2021; Liu et al., 2018a; Chen et al., 2019c) and social media (Derakhshan and Beigy, 2019; Shi et al., 2021). However, these works are usually used to mine the fixed trading patterns in historical data and the contextual semantics in the off-market data for a single stock, whereas they ignore the potential connections between the target stock and its related ones. Actually, the stock price movement of the target stock might be influenced by the diffused stock price of its related ones, which is also known as the momentum spillover effect in finance (Ali and Hirshleifer, 2020). The core assumption of the momentum spillover effect in finance is that these stocks are not independent and can interact with each other by binding some specific relations such as shared industry information (Letizia and Lillo, 2019),

☆ This document is the results of the research project funded by the National Science Foundation.
* Corresponding author.
E-mail addresses: hao.peng97@outlook.com (H. Peng), coliapaston@163.com (K. Dong), totoroyang@tsinghua.edu.cn (J. Yang).

https://doi.org/10.1016/j.engappai.2023.106948
Received 27 February 2023; Received in revised form 9 June 2023; Accepted 4 August 2023
Available online 25 August 2023

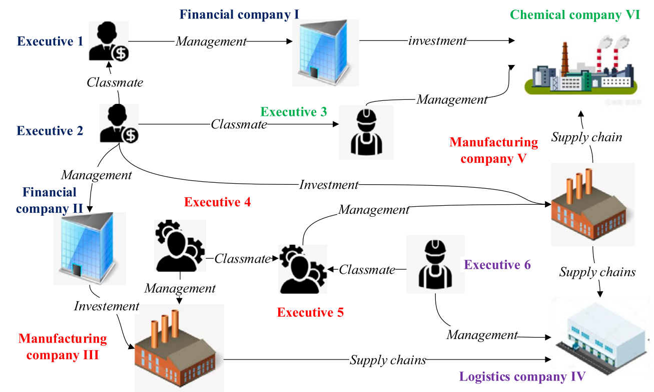

Fig. 1. Example of a market knowledge graph consisting of several kinds of companies and executives. Noted that different kinds of companies or executives have different relational types.

business partnership (Fuschi et al., 2016), and supply chain (Chauhan and Proth, 2005). Inspired by the above, the overall goal of this work is to simultaneously consider the influences of stock signal interaction in the phase of the stock signal feature extraction and the phase of stock signal feature encoding to improve the performance of SPMP.

For the stock signal feature extraction, the recent works have made great efforts to model correlations between stocks (Xu et al., 2022, 2020; Liu and Song, 2017) which mainly focus on the auto-correlation of an individual stock based on its own historical information. However, these works fail to consider the effects of the real-time interaction between stocks and further affect the performances of SPMP (Lo and MacKinlay, 1990). For instance, in a trading day, if the highest prices of the company *I* and *II* are defined as a signal channel, the real-time interactions of the highest price may induce these two stocks to share feel-good factors and further influence the stock price in the future. To avoid this issue, we introduce an external attention mechanism to achieve real-time interactions between stocks along the signal channels for SPMP.

Besides, for the encoding of the stock signal feature, the above-mentioned correlations can also be model as a structural Market Knowledge Graph (MKG) consisting of a series of entity (e.g., company and executive) nodes linked by edges representing specific relations. Graph Convolutional Network (GCN) (Velickovic et al., 2018; Kipf and Welling, 2017), as an important module to learn interactive information from structural data, is naturally suitable to encode the node signal feature representations from MKG. Recently, many GCN-based methods have been proposed and achieved some comparable performances for SPMP (Cheng and Li, 2021; Ye et al., 2020; Feng et al., 2019b; Li et al., 2020; Sawhney et al., 2021). These works usually first constructed a MKG and achieve edge connections according to the fact whether two entities have a certain relation type or not. Then they aggregate the node signal feature of neighbors into the target node and accomplishes the signal feature encoding of the target stock. However, most of these methods usually only consider the simple binary connections (0/1 weight) between entities in MKG and ignore the specific relation type (i.e., one company or executive plays a certain role in another one) between them. Even if the relation type is considered, they build a series of relation-specific graphs which are utilized to aggregate node signal features whereas the semantics of relations are still neglected for the designed models.

As shown in Fig. 1, there are several relations between entities in MKG. Most of the traditional GCN-based models usually ignore these specific relation types and only consider whether there two entities have a connection (Yes for weight 1 and No for weight 0) or not. As a result, the relation types with important semantics are ignored, which leads to unsatisfactory performance for SPMP. In addition, by observing MKG in Fig. 1, we note that the edges with same relation type connected by different entities should have different semantic representations in the GCN-based methods due to the diversity of its connected entities. For instance, in Fig. 1, the financial company *I* and the chemical company *VI* have an *investment* edge, as well as the financial company *II* and manufacturing company *III* has the same edge. However, the same *investment* edges should not represent the absolute same semantics because they connect different entities. This fact implies that the edge representations with semantics should be dynamic according to its connected entities and further incorporated into the node signal feature representations during the node aggregation/update in GCN. To accomplish the above goals, this work proposes a novel relation type guided GCN for the SPMP task.

To sum up, the main contributions of this work are:

- For the better stock signal feature extraction, we introduce an external attention mechanism to achieve the real-time interaction of stocks which can share the memory of the different companies along the stock signal channels such as the stock prices and the sentiments.
- For the encoding of stock signal features, this work proposes a novel relation type guided graph convolutional network. The edge representations with the same relation type are dynamically updated according to their connected entities which can be used to iteratively update node signal features in different GCN layers.
- The proposed method achieves state-of-the-art (i.e., SOTA) performances on two open-source datasets (i.e., CSI100E and CSI300E) compared with the recent advances. These results suggest that the proposed method is very potential to be applied into the practices.

The rest of the paper is organized as follows: Section 2 overviews some related works corresponding to the SPMP task. Section 3 defines the task content of SPMP and the construction of the SPMP dataset. Section 4 describes some details of the proposed SPMP model. Section 5 elaborates the used datasets, evaluation metrics, hyper-parameters setting, and compared benchmarks during experiments. Section 6 reports and discusses the experimental results to demonstrate the effectiveness of the proposed method by comparing it with the recent state-of-the-art baselines. Finally, we conclude this paper in Section 7.

2. Related work

The research of SPMP has gradually become a hotspot which can help investors achieve high-speed decisions for investment allocation and further bring handsome profits (Liu et al., 2018b). According to the Efficient Market Hypotheses (EMH), it is not realistic to accurately predict the stock price because the price always responds earlier for the market information than investors (Fama, 1970; Nguyen et al., 2016). An indirect strategy is to predict the stock price movement instead of the difficult stock price (Feng et al., 2019b; Adebiyi et al., 2014). Some previous researchers based on this idea even suggest that their approaches succeeded in predicting stock movement with more than 50% accuracy (Walczak, 2001; Huang et al., 2005; Qian and Rasheed, 2007). In general, the traditional SPMP methods can be usually summarized as two genres including technical analysis and fundamental analysis. In addition, inspired by the momentum spillover effect in finance, it is important to capture the relation interaction between entities for SPMP (Feng et al., 2019b; Sawhney et al., 2021). In this paper, we will overview them in the following several subsections.

2.1. Technical analysis

Technical analysis methods usually utilize the features in the time-series stock data (e.g., price and volume) to predict the stock price movement (Chen et al., 2019b). The early works conduct SPMP tasks by applying some shallow machine learning methods, such as Support

Vector Machine (SVM), latent dirichlet allocation, and latent space model. These methods usually need some hand-craft stock indicators corresponding to the time-series stock data, which are viewed as inputs for the designed models. For example, by applying some stock indicators such as closed price, volatility, and momentum of the stock market, Nayak et al. (2015) presented a hybridized framework of SVM with the k-nearest neighbor approach for the Indian SPMP. Nguyen and Shirai (2015) proposed a topic sentiment latent dirichlet allocation model to capture topics and sentiment features for SPMP. Pai and Lin (2005) combined the autoregressive integrated moving average model with the SVM model in forecasting stock price movement. Although the above efforts have achieved some comparable performances, the hand-craft feature construction based on stock information usually pays expensive time costs before training. Recent researchers have barely been so hung up about shallow machine learning, especially after the appearance of deep learning.

Recently, the development of deep learning has enlightened some researchers to present a good deal of methods to discover the hidden trading patterns for SPMP (Bao et al., 2017; Nelson et al., 2017). As we all know, stock trading data is a kind of time-series data, the researcher applies some commonly used modules of deep learning such as Recurrent Neural Networks (RNN) and Convolutional Neural Network (CNN) to automatically capture the global or local trading signal features in the time series (Lin et al., 2017; Hsieh et al., 2011). For instance, Nelson et al. (2017) exploited the usage of Long Short-Term Memory (LSTM) networks for SPMP by applying the price historical data. Lin et al. (2017) presented an end-to-end hybrid model that combines CNNs with LSTM to automatically learn local and global time-series signal features for SPMP. Based on LSTM and the attention mechanism, Feng et al. (2019a) employed adversarial training and added perturbances to reduce the stochasticity of the stock price variable and optimized the model to work well for SPMP. Although these technical analysis methods have achieved progress, technical analysis suffer that they are incapable of consider the influences of off-market data.

2.2. Fundamental analysis

Compared with the above technical analysis, fundamental analysis also exploits the off-market data for SPMP (Hu et al., 2018; Xu and Cohen, 2018; Zhao et al., 2022). Recently, with the maturity of Natural Language Processing (NLP), some NLP technologies such as event extraction (Ding et al., 2014), sentiment analysis (Shi et al., 2021), attention mechanism (Hu et al., 2018), and pre-trained language model (Yang et al., 2019) have been utilized to automatically capture market signal features for the SPMP models. For instance, Ding et al. (2015) extracted numerous events from the financial news and employed a CNN to model the influences of events for SPMP. Hajek and Barushka (2018) analyzed sentiment and topic of the stock information to model complex stock market relations for the SPMP of the USA companies. Hu et al. (2018) introduced a hierarchical attention mechanism and designed a Hybrid Attention Network (HAN) by using news texts to predict the stock price trend. Considering the redundancy of financial texts, Chen et al. (2019a) proposed a hierarchical aggregation method to automatically group, extract, and aggregate signal features mined from the news texts for SPMP. In addition, for the SPMP task, it is necessary to consider the high stochasticity of the market caused by random-walk patterns during model training. To reduce this problem, Xu and Cohen (2018) exploited text and price signals to imitate the generative process from market information and introduce randomness. Subsequently, they presented a deep generative model for the SPMP task. Feng et al. (2019a) introduced an adversarial attack strategy to improve the generalization of the model for the stock price movement prediction task. In short, fundamental analysis is usually better at mining the semantic information from the off-market data than the technical analysis, whereas it usually lacks the ability to model correlations between stocks in the market.

2.3. Graph Convolutional Network for stock relation modeling

Recent researchers have attempted to apply the stock relations for SPMP. Graph Convolutional Networks (GCN) (Velickovic et al., 2018; Kipf and Welling, 2017), as a high-performance neural network for structural data learning, is naturally suitable to model the complex relation information between companies for SPMP. For example, Chen et al. (2018) built a graph including all involved companies based on investment facts from the real market. Subsequently, they applied GCN to respective design a pipeline and joint models for the stock prediction. Li et al. (2020) considered the influence of overnight financial news and proposed a LSTM relational GCN model which constructs relation-specific graphs to aggregate node semantics in text for SPMP. Sawhney et al. (2021) deemed that the traditional graph-based approaches fail to consider the temporal evolution of stock prices and inter-stock relations. Thus, they presented a spatio-temporal hypergraph convolutional network to learn price evolution over a relation hypergraph matrix filled with 0/1 weights. To capture the relation importance varying with time, Cheng and Li (2021) proposed an attribute-driven graph attention network to obtain relation embeddings by attention mechanism and further aggregate attributes by employing GCN for SPMP. Zhao et al. (2022) considered implicit and explicit relations between entities and presented a dual attention network to aggregate stock information by modeling the mutual and inner influences among entities for SPMP. Although the above methods have incorporated the relation information into their designed models and achieved some competitive performances, they usually ignore two facts: (I) The real-time interactions of different companies along stock signal channels always occurs in trading days according to the momentum spillover effect in finance. (II) The GCN-based methods should consider specific relations in MKG whose semantics should be dynamic to achieve iterative update between relation edge and entity node during aggregation. To reduce these issues, we presented a novel model based on an External Attention mechanism and a Relation Type Guided GCN (EA-RTGCN) to improve the performance of SPMP.

The above-mentioned advances usually apply a static Market Knowledge Graph (MKG) in GCN to predict the stock price movement. This trait means that the weights between nodes are fixed and keep unchanged in different trading days during the aggregation of the stock node. Recently, some researchers have introduced the concept of temporal graphs in the GCN-based method for SPMP. This idea deems that MKG in different trading days is dynamically changed. They established a temporal graph consisting of market knowledge graphs linked to a single trading day and further served SPMP. For instance, Jafari and Haratizadeh (2022) proposed a GCNET model which constructs snapshots of the temporal graphs by calculating the influence scores between nodes and applying quadratic discriminant analysis algorithm (James et al., 2013) to generate weighted edges in the graph. Liu and Paterlini (2023) presented a LSTM-GCN model for SPMP, which inserts GCN into LSTM cells and constructs a temporal graph using value chain data to achieve the update of the input gate in the cascaded LSTM cells. Our work is different from the above methods. First, compared with the above temporal graph, the given MKG in our work is a static one and cannot be changed in different trading days. Then, most adjacent matrices derived from the temporal graphs in the above-mentioned works usually belong to the 0/1 (whether two nodes have a connection or not) or weighted (0∼1) adjacent ones. This fact suggests that the semantics of the specific relations between entities fails to be considered during aggregation. Our work considers the semantics of the specific relations and inserts the mapped relation index into an adjacent matrix which can be converted into an adjacent tensor consisting of a series of relation semantic embeddings. This trait means that the proposed model can provide operating space to fuse semantics of relations during aggregation.

Table 1
The collected signal features of the released datasets in Zhao et al. (2022).

Data Meta	Features
Financial entity	Companies, executives.
Relation type	Industry category, supply chain, business partnership, investment⋯
Historical price	Opening price, closing price, highest price, lowest price, trade volume.
News sentiment	Positive sentiment $Q(i)^+$, negative sentiment $Q(i)^-$, sentiment divergence $D(i)$.

3. Problem formulation and market signals input construction

3.1. Task description

In the stock mark, there are many effective stock signals including a series of historical price information, social media information, and the complex MKG. Mathematically, the price and social medial information can be converted as signal vectors to supply the basic information for SPMP. MKG contains the correlations between entities which are utilized to capture the dependencies between the target company and the related entities. In this work, based on these three kinds of market signals, we aim to predict the stock price movement of a certain company in the tth trading day. Therefore, we convert the stock price movement prediction task into a binary classification problem which can be formally written as:

$$y = Indicator\left(p_t > p_{t-1}\right) = sign\left(\frac{p_t - p_{t-1}}{p_{t-1}}\right) = \begin{cases} 0, & p_t < p_{t-1}, \\ 1, & p_t \geq p_{t-1}. \end{cases}, \tag{1}$$

where 0 denotes the "fall" of stock price, 1 means the "rise" of stock price, p_t denotes the closed price of the stock for the target stock in trading day t, and $sign(\cdot)$ is a sign function that inputs 1 if $p_t \geq p_{t-1}$ otherwise 0.

3.2. Market signal input construction

In this paper, we refer to the work in Zhao et al. (2022) and apply their released datasets (i.e., CSI100E and CSI300E) to confirm the effectiveness of EA-RTGCN for SPMP. Thus, it is necessary to detail the data organization, especially the market signal input construction. In the following subsections, we describe some key information of them including the bi-typed hybrid-relational MKG and several market indicators whose basic signal features used in these two datasets are shown in Table 1.

3.2.1. Bi-typed hybrid-relational market knowledge graph

In Zhao et al. (2022), a bi-typed hybrid-relational MKG was constructed for the predicted stocks of companies. There are two kinds of knowledge nodes in MKG consisting of the company- and executive-typed nodes. Different from the traditional methods that only learn relations between companies in the market (Ye et al., 2020; Li et al., 2020), most of the listed companies can be connected by numerous cross-appointment executives. Therefore, the released MKG also collects a large number of executives as the intermediary executive-typed nodes to build more abundant connections between companies. As shown in Fig. 1, because of the existence of executive-type nodes, many meta-relation connections involved in companies and executives can be built in the constructed MKG. (e.g., *financial company I* $\xrightarrow{Management}$ *executive 1* $\xrightarrow{Classmate}$ *executive 2* $\xrightarrow{Management}$ *financial company II* and *financial company II* $\xrightarrow{Management}$ *executive 2* $\xrightarrow{Classmate}$ *executive 3* $\xrightarrow{Management}$ *che-mical company VI*).

3.2.2. Market signals consisting of historical price and media news

Technical Indicators. According to the released datasets in Zhao et al. (2022), we utilize daily stock price and volume data, including opening price (op), closing price (cp), highest price (hp), lowest price (lp), and trading volume (tv) to represent the stock trading signals for a certain company. For the ith company, the representation of technical indicators consisting of stock prices and trading volume is defined as $\boldsymbol{p}_i = [op(i), co(i), hp(i), lp(i), tv(i)]^T \in \mathbb{R}^5$.

Sentiment Signals. In this work, we also build sentiment signals by referring to Zhao et al. (2022) and identifying sentiment polarities from about 1.1G news texts, including positive media sentiment, negative media sentiment, and media sentiment divergence. By referring to the work in Li et al. (2021), these sentiment polarities can be calculated as:

$$Q(i)^+ = \frac{N(i)^+}{N(i)^+ + N(i)^-}, Q(i)^- = \frac{N(i)^-}{N(i)^+ + N(i)^-}, \quad D(i) = \frac{N(i)^+ - N(i)^-}{N(i)^+ + N(i)^-}, \tag{2}$$

where $N(i)^+$ and $N(i)^-$ respectively denote the sum of the frequency of each positive and negative sentiment word found in the news corresponding to the ith company, $D(i)$ is the sentiment divergence of ith company. Thus, sentiment signals of ith company computed from the financial news are expressed as $\boldsymbol{q}_i = [Q(i)^+, Q(i)^-, D(i)^+]^T \in \mathbb{R}^3$.

4. Methodology

In this section, as shown in Fig. 2, we detail the structure of the proposed EA-RTGCN consisting of the following four parts: (I) A tensor incorporated module that can extract signal features based on the real-time interactions along the signal channels of the technical indicators $\boldsymbol{p}_i$ and sentiment signals $\boldsymbol{q}_i$. (II) A temporal embedding learning module is employed to capture the temporal information by a unidirectional Gated Recurrent Unit (GRU) network. (III) A Relation Type guided GCN (RTGCN) module converts the relation type indexes in MKG into a series of dynamical relation type embeddings, and subsequently achieves iterative update between nodes and relations during aggregation. (IV) A stock price movement prediction module conducts a binary classification task to predict the "rise" and "fall" of the stock price for the target company in the future.

4.1. Tensor incorporated module

To automatically capture the local potential features respectively in the technical indicator and the sentiment vectors, we introduce two 1-dimension CNNs (i.e., Conv1D) to extract the useful price and sentiment features. If the number of the companies is defined as N, the stock price and the sentiment tensors are represented respectively as $\boldsymbol{p} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_N] \in \mathbb{R}^{N\times5}$ and $\boldsymbol{q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_N] \in \mathbb{R}^{N\times3}$. As a result, the stock price and sentiment feature tensors can be respectively written as:

$$\boldsymbol{x}_p = \mathrm{Conv1D}(\boldsymbol{p}) \in \mathbb{R}^{N\times d_p}, \quad \boldsymbol{x}_q = \mathrm{Conv1D}(\boldsymbol{q}) \in \mathbb{R}^{N\times d_q}, \tag{3}$$

where $\mathrm{Conv1D}(\cdot)$ denotes a 1-dimension CNN operator, $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ respectively mean the extracted feature tensors from the stock price and sentiment representations.

In essence, the application of 1-dimension CNNs aims to capture the local features of stock prices and the sentiment polarities for the target company. In addition, the real-time interactions of different companies in the same trading day are useful for the SPMP task. In this work, along the signal channel of price and sentiment, we introduce

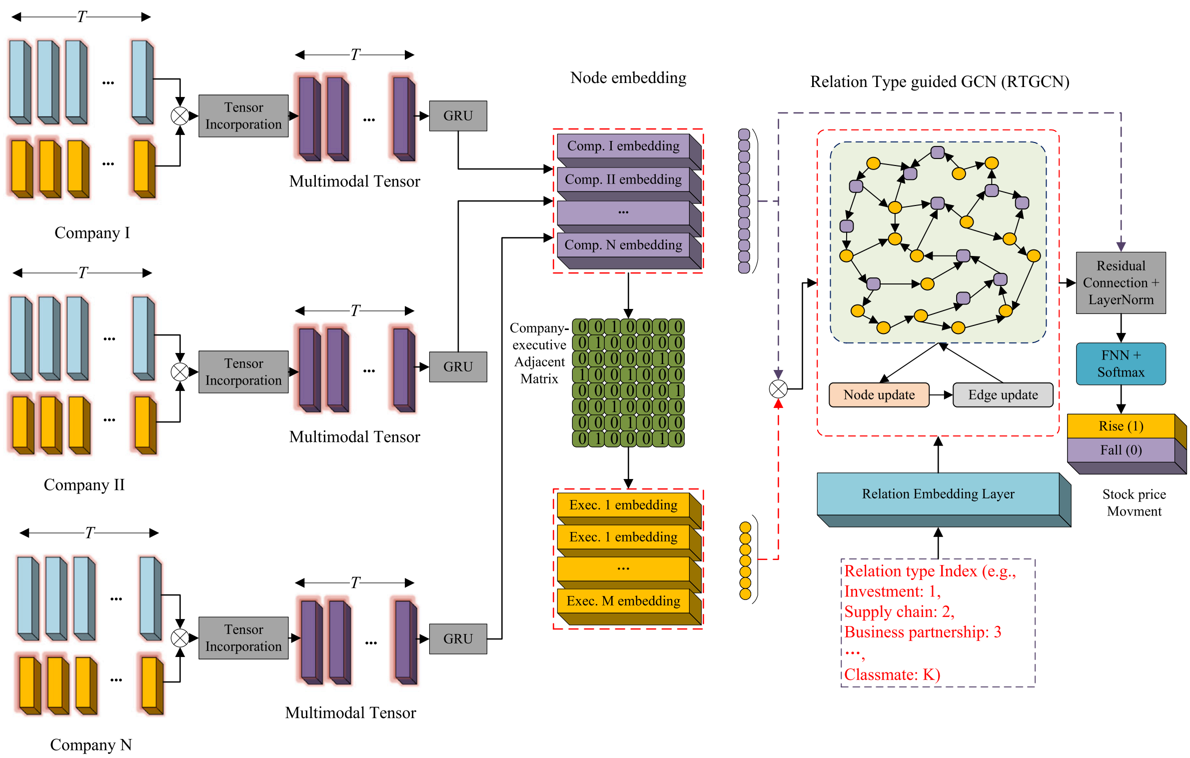

Fig. 2. The basic components of the proposed method for the stock price movement prediction task.

an external attention mechanism to compute the updated price and sentiment representation

$$
\begin{aligned}
x^{e-att}_{p,:,i} &= W_{p,2} \cdot Norm(softmax(W_{p,1} \cdot x_{p,:,i})) \in \mathbb{R}^N, \\
x^{e-att}_{q,:,j} &= W_{q,2} \cdot Norm(softmax(W_{q,1} \cdot x_{q,:,j})) \in \mathbb{R}^N,
\end{aligned} \tag{4}
$$

where $x^{e-att}_p = [x^{e-att}_{p,:,1}, x^{e-att}_{p,:,2}, \dots, x^{e-att}_{p,:,5}]^T \in \mathbb{R}^{N\times 5}$ and $x^{e-att}_q = [x^{e-att}_{q,:,1}, x^{e-att}_{q,:,2}, x^{e-att}_{q,:,3}]^T \in \mathbb{R}^{N\times 3}$ mean the updated company tensor respectively split along the dimension of the price and sentiment, $W_{p,1} \in \mathbb{R}^{S\times N}$, $W_{p,2} \in \mathbb{R}^{N\times S}$, $W_{q,1} \in \mathbb{R}^{S\times N}$, and $W_{q,2} \in \mathbb{R}^{N\times S}$ denote the corresponding transformation matrices, S is the channel number in the external attention mechanism for the technical indicator and sentiment signal, and the $Norm(\cdot)$ defines an operation of the normalization along the 0 axis. Meanwhile, we apply a residual connection to avoid the problem of the network degradation, which can be formulated as:

$$
\begin{aligned}
x^{e-att'}_p &= ReLU(x^{e-att}_p + x_p), \\
x^{e-att'}_q &= ReLU(x^{e-att}_q + x_p),
\end{aligned} \tag{5}
$$

where $ReLU(\cdot)$ denotes an activation function.

To achieve the fusion of the price and sentiment tensors, a nonlinear layer with an activation function is utilized to generate the signal feature representation of ith company in a certain trading day:

$$
x_{i,:} = tanh(W_f \cdot concat([x_{p,i,:}; x_{q,i,:}, x^{e-att'}_{p,i,:}; x^{e-att'}_{q,i,:}]) + b_f), \tag{6}
$$

where $W_f \in \mathbb{R}^{d_x\times(d_p+d_q+8)}$ and $b_f \in \mathbb{R}^{d_x}$ respectively denote a transformation matrix and a bias vector, $concat(\cdot)$ means a concatenation operation, $tanh(\cdot)$ defines an activation function, and the overall signal feature tensor for the ith company can be written as $x = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{N\times d_x}$. Noted that the computed x only represents the market signal of one trading day.

4.2. Temporal embedding learning module

In the stock market, for the predicted tth trading day of ith company, we assume that the market signals (computed in Section 4.1) in past T trading days are denoted as $\overline{x}^t_i = [x^{t-T}_i, x^{t-T+1}_i, \dots, x^{t-1}_i] \in \mathbb{R}^{T\times d_x}$. To fuse the market signals in the T trading days, we apply a unidirectional GRU neural network to learn its temporal embedding:

$$
\hat{x}^t_i = \overrightarrow{\mathrm{GRU}}(x^{t-T}_i, x^{t-T+1}_i, \dots, x^{t-1}_i) \in \mathbb{R}^{d_g}, \tag{7}
$$

where d_g denotes the size of the hidden state in GRU and $\hat{x}^t_i$ is the computed temporal embedding for ith company in tth trading day.

4.3. Relation type guided Graph Convolutional Network

4.3.1. The generation of executive representations

In this section, we apply the proposed RTGCN to aggregate the company and executive information in MKG for SPMP. To achieve the aggregated operation, we first construct a company-executive adjacent matrix derived from the provided MKG in Zhao et al. (2022). Then, as mentioned in Section 4.1, we have already obtained the temporal embedding $\hat{x}^t_i$ for the ith company in tth trading day which is viewed as a company node embedding in MKG. Also, assuming that the number of the executives is M, the simple binary relation (0/1) adjacent matrix $A^{c\to e} \in \mathbb{R}^{M\times N}$ (i.e., whether one executive is related to a certain company) can be acquired from MKG. As a result, the necessary executive node representation u^t_j is calculated by summing their corresponding company representations as (see Fig. 3):

$$
u^t_j = \sum_{i=1}^{N} A^{c\to e}_{j,i} \cdot \hat{x}^t_i \tag{8}
$$

4.3.2. The node update of the proposed RTGCN

As mentioned in Sections 4.1 and 4.3.1, we have already obtained the company and executive node representations. Therefore, the initial node representations in the proposed RTGCN can be written as:

$$
s^{(t,0)}_i = W_s \cdot s^t_i + b_s, \tag{9}
$$

where $s^t = [x^t_1, x^t_2, \dots, x^t_N, u^t_1, u^t_2, x^t_M] \in \mathbb{R}^{(M+N)\times d_g}$, and $W_s \in \mathbb{R}^{d_g\times d_g}$ and $b_s \in \mathbb{R}^{d_g}$ respectively denote the transformation matrix and the bias vector.

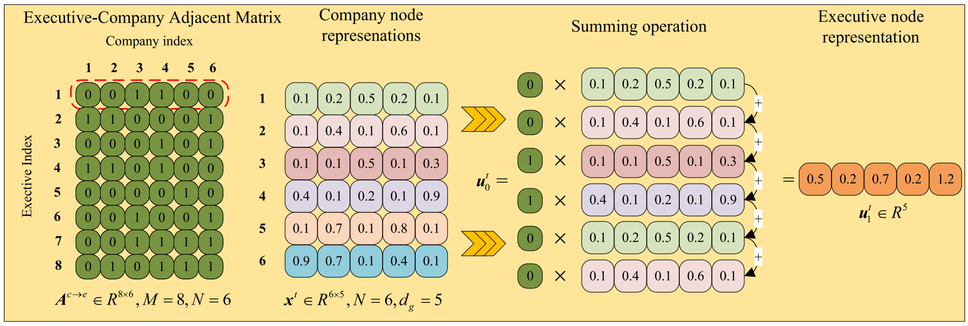

Fig. 3. An example of the computation of the executive representations.

To achieve the node aggregation between the target node and its neighbors, we first collect all the relation types such as *Investment*, *Industry_category*, *Supply_chain*, and *Business_partnership* and construct a mapped dictionary from relation types to indexes.[1] Then, we transform the specific relation types between entities into a series of indexes which are inserted into the constructed company-executive adjacent matrix $\boldsymbol{A} \in \mathbb{R}^{(M+N)\times(M+N)}$. Noted that the constructed adjacent matrix is different from the traditional binary one, which consists of multiple relational (called as "edges") type indexes in this work.

In the proposed RTGCN, we first randomly initialize a trainable embedding lookup table in the designed relation embedding layer (see Fig. 4(a)). Then, the relation type guided company-executive adjacent matrix $\boldsymbol{A}$ can be transformed into an adjacent tensor $\overline{\boldsymbol{A}} \in \mathbb{R}^{(M+N)\times(M+N)\times p}$ consisting of edges embeddings with semantics. Here the $\overline{\boldsymbol{A}}_{i,j,:} \in \mathbb{R}^p$ denotes the relation type embedding between node i and node j and p is the dimension of the relation type embedding. Also, p can be understood as the number of signal channels in the adjacency tensor. This idea implies that the obtained adjacent tensor $\overline{\boldsymbol{A}}$ can be split as $\overline{\boldsymbol{A}}^* = [\overline{\boldsymbol{A}}_{:,:,1}, \overline{\boldsymbol{A}}_{:,:,2}, \overline{\boldsymbol{A}}_{:,:,p}]$ where $\overline{\boldsymbol{A}}_{:,:,k} \in \mathbb{R}^{(M+N)\times(M+N)}$ and $k = 1, 2, 3, \ldots, p$. These split adjacent matrices can be used to respectively conduct node aggregation which incorporates all the aggregated results along p channels to generate the updated node representations.

Thus, we assume that the proposed RTGCN is stacked L layers in the model. If the node representations (for ith company in tth trading day) from $(l-1)$-layer is represented as $\boldsymbol{s}^{(t,l-1)} \in \mathbb{R}^{d_g}$, as shown in Fig. 4(b), the updated node representation in lth layer can be obtained as:

$$\boldsymbol{s}_{i,k}^{(t,l)} = \sum_{j=1}^{N+M} \overline{\boldsymbol{A}}_{i,j,k} \cdot \overline{\boldsymbol{W}}^{(l)} \cdot \boldsymbol{s}_j^{(t,l-1)} \tag{10}$$

where $\overline{\boldsymbol{W}} \in \mathbb{R}^{d_g\times d_g}$ is a transformation matrix. Subsequently, a pooling operation and a nonlinear transformation layer are applied to acquire the updated node representation:

$$\begin{aligned} \overline{\boldsymbol{s}}_i^{(t,l)} &= Pool(\boldsymbol{s}_{i,1}^{t,l}, \boldsymbol{s}_{i,2}^{(i,1)}, \ldots \boldsymbol{s}_{i,d_p}^{(i,1)}) \in \mathbb{R}^{d_g}, \\ \boldsymbol{s}_i^{(t,l)} &= ReLU(\boldsymbol{W}_p \cdot \overline{\boldsymbol{s}}_i^{(t,l)} + \boldsymbol{b}_p) \in \mathbb{R}^{d_g}, \end{aligned} \tag{11}$$

where $Pool(\cdot)$ denotes a pooling operator, $\boldsymbol{W}_p \in \mathbb{R}^{d_g\times d_g}$ and $\boldsymbol{b}_p \in \mathbb{R}^{d_g}$ respectively are the projected matrix and the bias vector. Although we have utilized the semantic information of the relation types between entities, the same relation type should have different semantics according to its connected different entities. Therefore, it is also necessary to employ node representations in the $(l-1)$th layer to update the edge type embeddings in the lth layer, which is detailed in the following subsection.

[1] The format of the mapped dictionary is: {"*Investment*": 0, "*Industry_category*": 1, "*Supply_chain*": 2, ⋯}

4.3.3. The dynamics of relation and iterative update for the proposed RTGCN

As mentioned above, the adjacent matrix $\boldsymbol{A}$ consisting of different edge indexes can be initialized to an adjacent tensor $\overline{\boldsymbol{A}} \in \mathbb{R}^{(M+N)\times(M+N)\times p}$. However, the discussion in Section 1 reminds us that assigning a single node-independent representation for each edge label is not enough to express the complex relation semantics between entities. To solve this issue, we apply a self-attention mechanism (see Fig. 4(c)) to dynamically update edge representations in $\overline{\boldsymbol{A}}$ which can be expressed as:

$$\begin{aligned} &\boldsymbol{s}_{query,i}^{(t,l)} = \boldsymbol{W}_q \cdot \boldsymbol{s}_i^{(t,l)}, \boldsymbol{s}_{key,j}^{(t,l)} = \boldsymbol{W}_k \cdot \boldsymbol{s}_j^{(t,l)}, \\ &\overline{\boldsymbol{A}}_{value,i,j}^{(l-1)} = \boldsymbol{W}_v \cdot \overline{\boldsymbol{A}}_{i,j,:}^{(l-1)}, \\ &\boldsymbol{m}_{i,j}^{(t,l)} = softmax\left(\frac{\boldsymbol{s}_{query,i}^{T,(t,l)} \cdot \boldsymbol{s}_{key,j}^{(t,l)}}{\sqrt{p}}\right) \in \mathbb{R}^{p\times p}, \\ &\overline{\boldsymbol{A}}_{i,j,:}^{(l)} = \boldsymbol{m}^{(t,l)} \cdot \overline{\boldsymbol{A}}_{value,i,j}^{(l-1)}, \end{aligned} \tag{12}$$

where $\boldsymbol{W}_q \in \mathbb{R}^{p\times d_g}$, $\boldsymbol{W}_k \in \mathbb{R}^{p\times d_g}$, and $\boldsymbol{W}_v \in \mathbb{R}^{p\times p}$ denote the transformation matrices. In detail, the self-attention mechanism in lth layer can first respectively encode the representations of connected entity node i and node j (see the first line in Eq. (12)). These two representations are respectively viewed as the query and key ones serving the subsequent query operation for the encoding of the edge representation in the $(l-1)$th layer. Then, we also encode the edge representation $\overline{\boldsymbol{A}}_{i,j,:}^{(l-1)}$ to obtain the retrieval value representation $\overline{\boldsymbol{A}}_{value,i,j}^{(l-1)}$ (see the second line in Eq. (12)). Finally, by generating a retrieval matrix $\boldsymbol{m}_{i,j}^{(t,l)}$, we can dynamically update adjacent tensor in different layers (see the last two lines in Eq. (12)). Thus, the edge representation for each edge label is node-dependent and can effectively express the complex semantics of relations between entities.

Overall, in a single RTGCN layer, the entity node representations are first updated by the aggregation based on the adjacent tensor (see Eqs. (10) and (11)). Then the updated entity node representations are subsequently utilized to facilitate the update of the edge embedding by combining with the original adjacent tensor in the last layer (see Eq. (12)). Further, from the global perspective of RTGCN with multiple layers, as shown in Fig. 5, one can observe a phenomenon of interactive update between entity node and adjacent tensor. We call this process as iterative update which can fully achieve the relation type guided entity node aggregation and dynamics of the semantic edge embedding in adjacent tensor.

4.4. Stock price movement prediction module

After passing through stacked L layer RTGCN layers, we collect the initial node embeddings of N companies and all their corresponding node representation for each RTGCN layer. Subsequently, we concatenate these embeddings and apply a linear layer to acquire the final representation $\boldsymbol{s}_{final,i}^t$ for the ith company in the tth trading day:

$$\boldsymbol{s}_{final,i}^t = \boldsymbol{W}_{fin} \cdot concat([\boldsymbol{s}^{(t,0)}; \boldsymbol{s}^{(t,1)}; \cdots; \boldsymbol{s}^{(t,L)}]) + \boldsymbol{b}_{fin}, \tag{13}$$

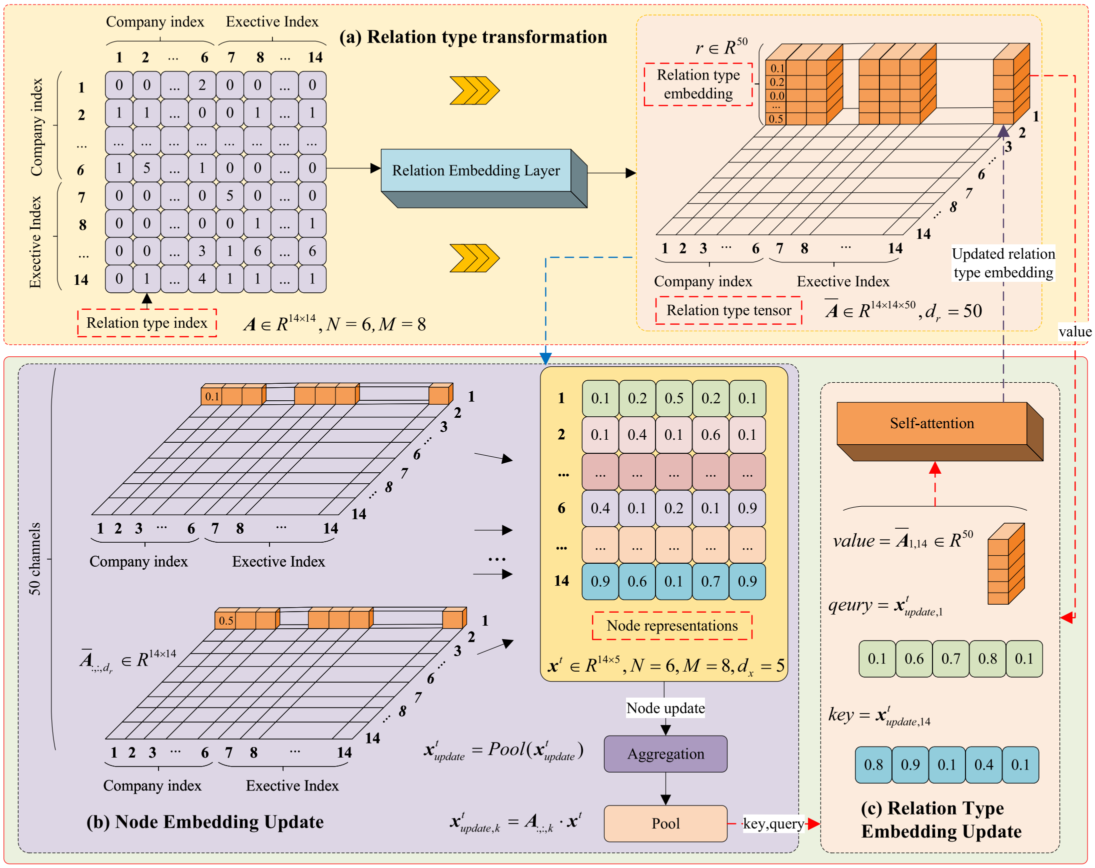

Fig. 4. An instance of the node and relation representation update in the proposed graph neural network.

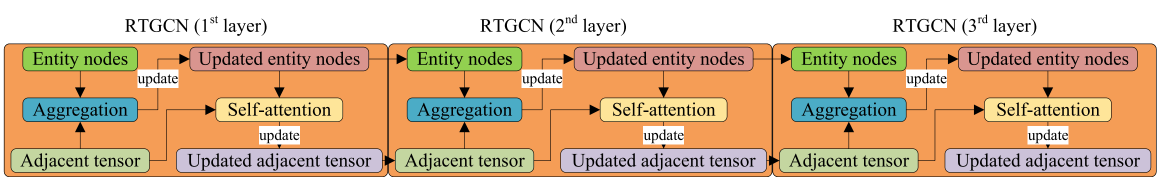

Fig. 5. An illustration of the iterative update between entity and relation in the proposed RTGCN.

where $concat(\cdot)$ means a concatenation operator, $\boldsymbol{W}_{fin}$ denotes a projected matrix and $\boldsymbol{b}_{fin}$ is a bias vector. To avoid the problem of vanishing gradient and network degradation, we introduce a residual connection with normalization, which is followed as:

$$\bar{\boldsymbol{s}}^t_{final,i} = Norm(\boldsymbol{s}^t_{final,i} + \boldsymbol{x}^t_i). \tag{14}$$

As a result, the stock price movement can be predicted as:

$$\hat{\boldsymbol{y}}^t_i = softmax(\boldsymbol{W} \cdot \bar{\boldsymbol{s}}^t_{final,i} + \boldsymbol{b}), \tag{15}$$

where $\hat{\boldsymbol{y}}^t_i \in \mathbb{R}^2$ is the predicted probability distribution of stock price movement for the ith company in the tth trading day, $\boldsymbol{W} \in \mathbb{R}^{d_g\times 2}$ and $\boldsymbol{b} \in \mathbb{R}^2$ respectively denote the transformation matrix and the bias vector. Assuming that the provided training instances are defined as $\{\{\boldsymbol{x}^t_i, \boldsymbol{y}^t_i\}^N_{i=1}\}^D_{t=1}$ where $\boldsymbol{y}^t_i$ is the golden label of stock price movement in the tth trading day for the ith company, and D denotes the number of the predicted trading days. Therefore, the cross-entropy loss can be written as:

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \sum_{t=1}^{D}\sum_{i=1}^{N} \hat{\boldsymbol{y}}^t_i \log \boldsymbol{y}^t_i \tag{16}$$

5. Experiment preparation

5.1. Datasets

In this work, we conduct experiments based on two open-source datasets called CSI100E and CSI300E which are released in Zhao et al. (2022) and have different numbers of companies and executives.[2] In detail, the stock data in datasets are collected from the famous China Securities Index (CSI) and the news data for generating sentiment signals is collected from four mainstream financial websites including Sina, Hexun, Sohu, and Eastmoney.[3] Besides, many relations including four kinds of relations between companies, two relations between executives, and two kinds of relations between company and executive[4]

[2] The datasets for the experimental evaluation are available at: https://github.com/trytodoit227/DANSMP.

[3] The websites of Sina, Hexun, Sohu and Eastmoney are respectively available at: http://www.sina.com, http://www.hexun.com, http://www.sohu.com and http://www.eastmoney.com.

[4] The four kinds of relations between companies include *Investment*, *Industry category*, *Supply chain*, and *Business partnership*. The relations between executives contain *Classmate* and *Colleague*. And the relations between company and executive are *Management* and *Investment*.

Table 2
Statistics of the CSI100E and CSI300E datasets.

	CSI100E	CSI300E
#Companies(Nodes)	73	185
#Executives(Nodes)	163	275
#Investment(Edges)	7	44
#Industry category(Edges)	272	1043
#Supply chain(Edges)	27	23
#Business partnership(Edges)	98	328
#meta-relation CEC	18	42
#meta-relation CEEC	134	252
#Classmate(Edges)	338	592
#Colleague(Edges)	953	2224
#Management(Edges)	166	275
#Investment(Edges)	1	8
#Train period	21/11/2017–05/08/2019	21/11/2017–05/08/2019
#Valid period	06/08/2019–22/10/2019	06/08/2019–22/10/2019
#Test period	23/10/2019–31/12/2019	23/10/2019–31/12/2019

are collected to construct the MKG for CSI100E and CSI300E.[5] The overall statistics of datasets are illustrated in Table 2. The usage details of the covered market signals are depicted in Section 3.2.

5.2. Evaluation protocols

For the evaluated metrics, referring to the works in Cheng and Li (2021), Li et al. (2021) and Zhao et al. (2022), we mainly apply the Directional Accuracy (DACC) and AUC (the area under the precision–recall curve) to evaluate the overall performance of the proposed method in the following experiments. The DACC can be written as $DACC = \frac{n}{N}$, where n denotes the number of predictions which witness the same direction of stock movements between the predicted trend and the actual stock trend, and N is the total number of predictions. In addition, considering the comparison between our model and the method in Multi-GCGRU (Ye et al., 2020), we also introduce the metrics including F_1 score and the Matthews Correlation Coefficient (MCC) which can be expressed as:

$$\begin{aligned} \mathrm{F}_1 &= \frac{2\times \mathrm{P}\times \mathrm{R}}{(\mathrm{P}+\mathrm{R})}, \mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \\ \mathrm{MCC} &= \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}, \end{aligned} \tag{17}$$

where TP, TN, FP, and FN are respectively the shorts of true positive, true negative, false positive, and false negative. P and R respectively denote Precision and Recall metrics.[6]

5.3. Hyper-parameter settings

In the experiments, the best performances of the proposed models are obtained by a strict grid search according to the setting of the hyper-parameters. The best group of hyper-parameters is shown in Table 3.[7]

5.4. Compared baselines

To confirm the effectiveness of the proposed approaches, we list some recent popular methods and compare our model with them in the following:

Table 3
The hyper-parameter settings on two datasets.

Hyper-parameter	CSI100E	CSI300E
Lookback window size T	20	30
The kernel size of Conv1D in Eq. (3)	1	1
The number of the channel S in Eq. (4)	64	64
The outputted dimension d_x in Eq. (6)	10	10
The GRU hidden size d_g in Eq. (7)	78	44
The dim of the relation embedding p	100	20
The layer number L of RTGCN	2	2
The way of pooling in Eq. (11)	Sum	Sum
Learning rate	0.0008	0.00085
Maximum number of epochs	400	400
Step of early stopping	30	30

- **LSTM** (Hochreiter and Schmidhuber, 1997) and **GRU** (Cho et al., 2014). The LSTM and GRU neural networks belong to the scope of the RNN model that can achieve promising performance on time-series data. In the evaluation, we apply two-layer LSTM or GRU networks for the stock price movement prediction task.
- **GCN** (Kipf and Welling, 2017): The graph convolutional network performs aggregated operations for the attributes of the neighbor nodes on the constructed binary company-executive graph. In this study, we apply a two-layer GCN network to achieve aggregated operation for the neighboring companies and executives.
- **GAT** (Velickovic et al., 2018). The graph attention neural network introduces an attention mechanism for the process of aggregation which considers different importances of the neighbors adaptively. In this work, a two-layer GAT network is utilized to encode the company and the executive information.
- **RGCN** (Schlichtkrull et al., 2018). Relational graph convolutional network considers the heterogeneity of the constructed market knowledge graph and conducts the specialized mapping matrices for each relation. In this work, two-layer RGCN is employed to achieve information aggregation according to the different company-executive relations.
- **HGT** (Hu et al., 2020): The heterogeneous graph transformer network applies a popular transformer architecture to capture features of different entity nodes based on relation-specific transformation matrices on the provided MKG.
- **Multi-GCGRU** (Ye et al., 2020) consists of a Multi-GCN and GRU which are respectively used to fuse multiple pre-defined or data-driven stock graph structures and learn the temporal information for SPMP. In this work, according to the provided relations in MKG, we create 8 graphs to conduct multi-graph and dynamical graph convolution operations for SPMP.
- **MAN-SF** (Sawhney et al., 2020). By applying a GRU module, an attention mechanism, and a graph attention neural network,

[5] According to Zhao et al. (2022), the relations corresponding to companies and executives are collected from a publicly available API tushare https://tushare.pro/ and a website http://www.51ifind.com/.

[6] Considering that the simplification of the listed data, we only compare the proposed method with Multi-GCGRU based on F_1 score and MCC metrics.

[7] The proposed methods are implemented with PyTorch on a GTX 3090 GPU. To prevent the issue of overfitting, we employ an early-stopping strategy based on AUC metric over the validation dataset.

Table 4
The performance comparison between the proposed model and recent advances for the stock price movement prediction.

Genre	Methods	CSI100E		CSI300E	
		DACC	AUC	DACC	AUC
RNN-based methods	LSTM (Hochreiter and Schmidhuber, 1997)	51.14	51.33	51.78	52.24
	GRU (Cho et al., 2014)	51.66	51.46	51.11	52.30
GCN-based methods	GCN (Kipf and Welling, 2017)	51.58	52.18	51.68	51.81
	GAT (Velickovic et al., 2018)	52.17	52.78	51.40	52.24
	RGCN (Schlichtkrull et al., 2018)	52.33	52.69	51.79	52.59
	HGT (Hu et al., 2020)	53.01	52.51	51.70	52.19
	MAN-SF (Sawhney et al., 2020)	52.86	52.23	51.91	52.48
	Multi-GCGRU (Ye et al., 2020)	53.65	54.28	52.77	54.54
	STHAN-SR (Sawhney et al., 2021)	52.78	53.05	52.89	53.48
	AD-GAT (Cheng and Li, 2021)	54.56	55.46	52.63	54.29
	DANSMP (Zhao et al., 2022)	57.75	60.78	55.79	59.36
	EA-RTGCN (ours)	**60.66**	**64.62**	**58.28**	**61.31**
	Methods	F_1	MCC	F_1	MCC
GCN-based methods	Multi-GCGRU (Ye et al., 2020)	56.64	0.1442	55.82	0.1398
	EA-RTGCN (ours)	60.09	0.2193	58.21	0.1677

this approach fuses chaotic temporal signals from financial data, social media indicators, and stock relation information in a hierarchical fashion to predict future stock price movement.

- **STHAN-SR** (Sawhney et al., 2021): This method utilizes hypergraph and temporal Hawkes attention mechanism to rank stocks which only contain historical price data and explicit company relations. The compared baseline in this work slightly modifies the objective function and introduces its main framework for SPMP.
- **AD-GAT** (Cheng and Li, 2021). This approach presents an attribute-driven graph attention network to automatically capture attribute-sensitive momentum spillover of stocks. It can model market information space that has feature interaction and effectively improve the performance of the stock movement prediction.
- **DANSMP** (Zhao et al., 2022). This method constructs a MKG consisting of companies & executives nodes and abundant company-executive edges. In addition, a dual attention network is presented to learn the momentum spillover signals by the provided MKG for the SPMP tasks.

6. Experiment analysis

6.1. Main results

In this work, we conduct experiments on CSI100E and CSI300E. The overall experimental results are reported in Table 4

In Table 4, it can be observed that the proposed model outperforms all the recent advances, including the RNN-based methods and the GNN-based methods. Specifically, compared with the recent advances, it respectively achieves performance improvements with at least 2.91%↑ DACC & 3.84%↑ AUC on CSI100E, and 2.49%↑ DACC & 1.95%↑ AUC on CSI300E (EA-RTGCN vs DANSMP). These results demonstrate the effectiveness of the proposed model that introduces an external attention mechanism and applies relation type guided graph convolutional network. In detail, AD-GAT & DANSMP apply the neural tensor network to achieve tensor fusion and STHAN-SR designs five temporal return ratio features which can be concatenated to learn signal features for each trading day. Compared with the above manners for signal feature extraction, the proposed tensor incorporated module consisting of a Conv1D and an external attention modules in Eqs. (3) and (4) performs better and can effectively learn signal features from the technical indicators and the sentiment vectors by real-time interaction along different signal channels (detailed evidence can be seen in Section 6.2). This bottom fine-grained signal feature extraction/fusion manner is very necessary to mine stock information and further release the potentials of the upper RTGCN module to enhance the performance of SPMP.

Besides, for the signal feature encoding, the proposed RTGCN is another booster to enhance the performance of SPMP. Considering the traditional GCN-based methods including GCN, GAT, AD-GAT, MAN-SF, and DANSMP, they usually apply 0/1 or weighted adjacent matrices to aggregate the neighbor nodes while they may ignore the influences of edges connected to different entities. Thus, it is logical for our model to achieve better performances than the above-analyzed benchmarks. In addition, although RGCN, HGT, Multi-GCCGRU, and STHAN-SR have introduced the specific relations that can guide the model to generate specific-relation graphs or hypergraphs, the semantics of specific relations are still not considered and incorporated into the aggregated entity representations. As a result, on the one hand, compared with the RNN-based methods, the introduction of GCN architecture can effectively enhance the performance of SPMP. On the other hand, among the multitudinous GCN-based methods, the semantics of relation is also beneficial to enhance the representational ability of GCNs during aggregation for the SPMP task.

Moreover, we also test the other two metrics (i.e., F_1 and MCC) to compare the proposed methods with the recent typical Multi-GCCGRU which is researched on a similar dataset. By transferring Multi-GCCGRU to our used datasets, Multi-GCCGRU creates 8 relation-specific graphs which are employed to conduct multiple and dynamical graph convolution operations during aggregation. However, because the specific semantics of relation are also ignored in this model and the influences of noise during multi-graph fusion, this method obviously performs worse than our approach according to the gaps of F_1 and MCC scores.

6.2. The effectiveness analysis of the applied tensor incorporated module

To verify the effectiveness of the proposed tensor incorporated module, we first consider the Neural Tensor Network (NTN) to fuse features in Zhao et al. (2022). Then, we respectively apply a 1-dimension CNN module (Conv1D), an external attention module EA($\mathcal{S}$ =∗),[8] the combination of Conv1D and EA($\mathcal{S}$ =∗), and the compared MTN as the tensor incorporated module to fuse the multimodal features in the phase of the signal feature extraction. The experimental results are listed in Table 5.

In Table 5, it can be observed that the proposed model achieves the best performance when it utilizes the combined tensor incorporated module consisting of Conv1D and EA($\mathcal{S}$ =∗) modules to conduct the multimodal feature fusion. Also, if we only consider the single module

[8] $\mathcal{S}$ denotes the channel number in the external attention module.

Table 5
The performance comparison of the proposed model with different tensor incorporated modules on two datasets.

Methods	CSI100E		CSI300E	
	DACC	AUC	DACC	AUC
SPMP w/MTN	58.73	63.71	58.07	61.10
SPMP w/Conv1D	57.48	61.96	56.73	58.23
SPMP w/EA(S = 64)	58.45	63.55	57.79	60.52
SPMP w/Conv1D + EA(S = 64)	**60.66**	**64.62**	**58.28**	**61.31**

such as NTN, Conv1D, and EA(S =∗), NTN achieves the best performance because it supplies a multi-dimensional bilinear tensor layer that obtains two directly related features across multiple dimensions. However, Conv1D only scans features in a limited receptive field for a single input stock price and sentiment vector and performs worse than others. Although EA(S =∗) achieves real-time interactions among different entities along the signal channels, the low-dimensional transformations for the technical indicators ($\boldsymbol{p} \in \mathbb{R}^{N\times 5}$) and the sentiment signals ($\boldsymbol{q} \in \mathbb{R}^{N\times 3}$) limit the potential of the external attention. By utilizing Conv1D and EA(S =∗), the proposed tensor incorporated module (please see last line in Table 5) accomplishes the combination of the price & sentiment features in the receptive field and the real-time interactions in a shared company memory. Therefore, compared with NTM, the proposed method with tensor incorporated module consisting of Conv1D and EA(S =∗) can achieve SOTA performance for SPMP.

6.3. The performance analysis of the proposed model with different pooling mechanism

In RTGCN, we define three manners of pooling including Max, SUM, and AVG during aggregation on MKG. To validate the influence of the pooling manners, we conduct a series of experiments with different setting of pooling manner on two datasets. Noted that the other hyper-parameters are same with the settings in Table 3. The experimental results are shown in Table 6.

In Table 6, it is clear that the model with SUM pooling manner can achieve the best performance for SPMP compared with models with other pooling manner including AVG and Max. Because the pooling process means that the model requires to compute p node embeddings (see Eq. (10)) and generates the updated node representations. The SUM pooling manner can effectively keep the information of the stock indicators (i.e., technical indicator and the sentiment signal), instead of synchronously scaling the node representation in AVG pooling manner or only selecting the most distinct feature values in Max pooling manner. Actually, compared with other operations such as AVG and MAX, the SUM pooling operation is a more widespread pooling manner in some currently popular graph convolutional networks. In this work, we also adopt this pooling manner for the proposed RTGCN model to guide the relation type embeddings to update the node representations.

6.4. The performance analysis of the proposed model with different RTGCN layers and dynamics of relation

In this subsection, we discuss the influences of the layer number and the dynamics of relations in RTGCN. We fix the manner of pooling and the dimension of the relation type embedding and subsequently set the layer number of RTGCN as $L \in \{1, 2, 3, 4\}$. The experimental results are reported in Table 7.[9]

In Table 7, one can observe that the best performances are simultaneously obtained on the CSI100E and CSI300E datasets when the layer number of the RTGCN is set as $L = 2$. If the layer number of RTGCN is $L \le 2$, the dynamically updated edge representations can guide the model to aggregate the signal feature embeddings of neighbors within 2 hops to the target entity. Thus, the performance of the proposed model becomes better with the increase of the layer number. However, for effective connections between two entities in MKG, a few nodes that are more than 2 hops can have straightforward semantic relations. Therefore, a large number of RTGCN layers mean that more invalid information viewed as noises participate in the process of aggregation. This invalid information is actually not beneficial for the performance enhancement of SPMP. In addition, with the increase of the RTGCN layers, the over-smoothing problem makes the aggregated node and relation type representations in different layers might tend to be consistent, which leads to the deterioration of the overall performance of the proposed model. Meanwhile, based on the best hyper-parameter group, we remove the attention mechanism (see Eq. (12)) and cancel the dynamics of relation representations. The experimental results (i.e., RTGCN vs RTGCN*) demonstrate that the dynamics of relations in adjacent tensor can effectively facilitate RTGCN to achieve iterative update between relation and entity in different RTGCN layers and further enhance the performance of SPMP.

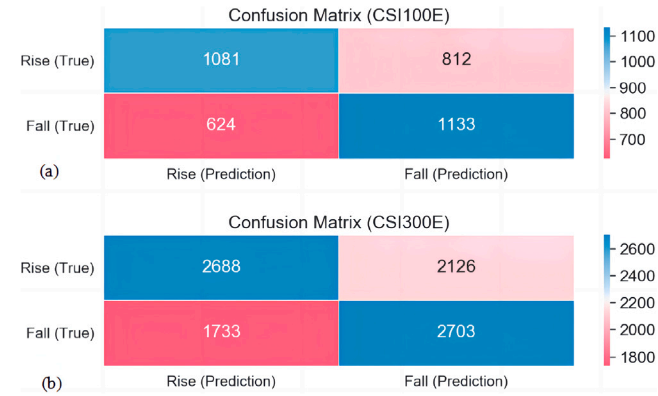

Fig. 6. The confusion matrix of the proposed method on the test data of CSI100E and CSI300E.

6.5. Case study for the predicted instances

To further display the predicted instances and evaluate the preferences of the proposed model, we construct the confusion matrices consisting of TP, TN, FP, and FN (see Eq. (17)) on CSI100E and CSI300E datasets.[10] The visual results are shown in Fig. 6.

From Fig. 6, it can be found that the proposed model has a basic ability to predict the instances of stocks (more than 58% DACC performances in Table 4). Also, the predicted stock movement trajectory is fluctuant because both the instances with "rise" and "fall" labels can be predicted in the given interval of trading days. Furthermore, noted that the proposed model on datasets displays the differentiated abilities to predict "rise" and "fall" trends in trading days. In detail, the model is better at correctly predicting "fall" labels instead of the "rise" label in the given instances. We argue that the "fall" prediction in a trading day may have early warnings in the past which can be captured from the sharp historical fluctuation of price and sentiment signals in a fixed window. Actually, except for the technological breakthroughs and favorable policies, the "rise" of most stocks is mild in trading days. This fact means that it is not easy to collect full market signals to predict the "rise" of stocks.

[9] RTGCN(SUM, 2, 50) denotes the proposed RTGCN that respectively sets pooling manner, the number of RTGCN layer, and the dimension of relation type embeddings as SUM, 2, and 50. RTGCN*(SUM, 2, 50) means that the dynamic update of relation representations is canceled (i.e., remove Eq. (12)).

[10] Because there are too many companies in the provided dataset, it is difficult to directly visualize predicted instances of each stock. As a result, we collect all the predicted instances and only constructed a single confuse matrix for each dataset.

Table 6
The performance comparison of the proposed model with different pooling manners on the two datasets.

Methods	CSI100E		Methods	CSI300E	
	DACC	AUC		DACC	AUC
SPMP w/RTGCN(SUM, 2, 100)	**60.66**	**64.62**	SPMP w/RTGCN(SUM, 2, 20)	**58.28**	**61.31**
SPMP w/RTGCN(AVG, 2, 100)	55.89	57.58	SPMP w/RTGCN(AVG, 2, 20)	55.63	58.29
SPMP w/RTGCN(Max, 2, 100)	58.06	61.45	SPMP w/RTGCN(MAX, 2, 20)	56.79	59.36

Table 7
The performance comparison of the proposed model with different RTGCN layers on the two datasets.

Methods	CSI100E		Methods	CSI300E	
	DACC	AUC		DACC	AUC
SPMP w/RTGCN(SUM, 1, 100)	60.41	63.25	SPMP w/RTGCN(SUM, 1, 20)	57.63	60.29
SPMP w/RTGCN(SUM, 2, 100)	**60.66**	**64.62**	SPMP w/RTGCN(SUM, 2, 20)	**58.28**	**61.31**
SPMP w/RTGCN ∗ (SUM, 2, 100)	58.27	59.06	SPMP w/RTGCN ∗ (SUM, 2, 20)	57.02	60.01
SPMP w/RTGCN(SUM, 3, 100)	59.56	61.45	SPMP w/RTGCN(SUM, 3, 20)	58.13	60.98
SPMP w/RTGCN(SUM, 4, 100)	58.46	60.12	SPMP w/RTGCN(SUM, 4, 20)	56.79	59.36

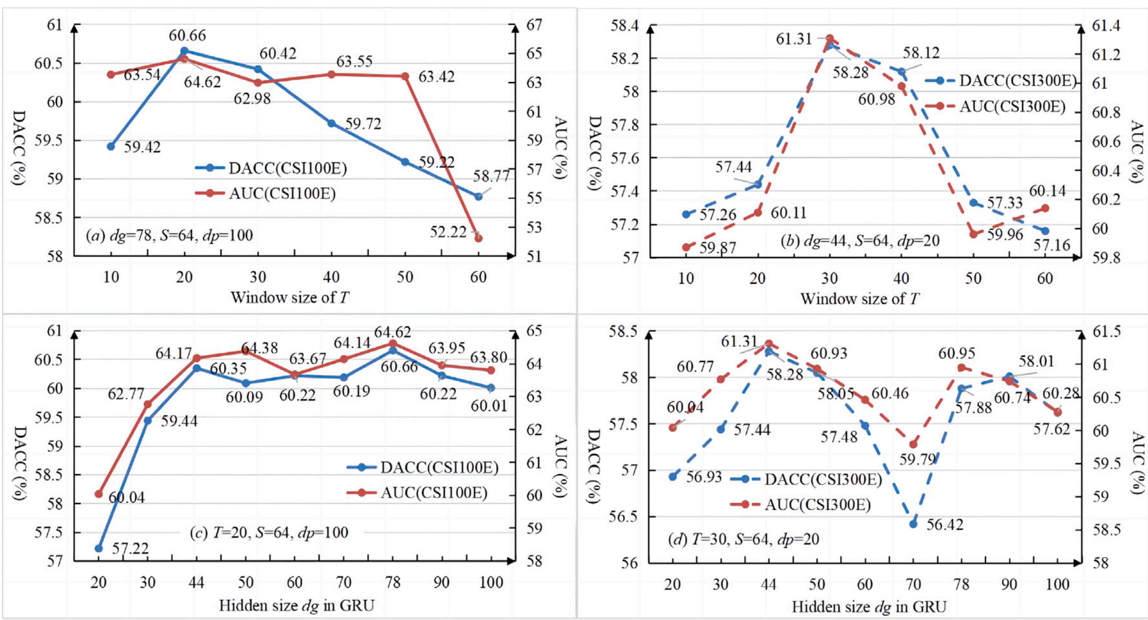

Fig. 7. The sensitivity of the window size T and the hidden size d_g in the proposed model.

6.6. The sensitivity for the hyper-parameters in the proposed model

This subsection focuses on the discussion of hyper-parameter sensitivity. In detail, we consider the window size of the trading day interval T in the temporal embedding learning module and the hidden size d_g in GRU (see Eq. (7)) and respectively set them as $d_g \in \{20, 30, 40, 44, 50, 60, 70, 78, 90, 100\}$ and $T \in \{10, 20, 30, 40, 50, 60\}$ on CSI100E and CSI300E. The experimental results are illustrated in Fig. 7. Meanwhile, we also discuss the sensitivity of the channel number S in the external attention module and the relation embedding size p in RTGCN. In detail, we fix the best hyper-parameters $T\&d_g\&p$ and set the channel number S as $S \in \{8, 16, 32, 64, 128, 256\}$. Similarly, we fix $T\&d_g\&S$ and set the relation embedding size p as a series of discrete integers. The experimental results are shown in Fig. 8.

In Fig. 7, it can be found that the sensitivities of the hyper-parameters of the proposed model on the two datasets. In detail, on CSI100E, the proposed model is sensitive to the window size T. It is not beneficial for the improvement of the proposed model when the model has a larger or smaller window size (see Fig. 7(a)). If the window size is smaller ($T \leq 20$), we deem that less trading information (i.e., stock price and new sentiment) is considered during training and evaluation. Meanwhile, a larger window size might lead that the model is easier to be confused by the earlier trading history and makes the performance of prediction worse. Therefore, when the window size $T = 20$, the proposed model can achieve the best performance (DACC: 60.66% and AUC: 64.62%) on CSI100E. Furthermore, when the hidden size d_g in GRU is less than a threshold (i.e., $d_g = 78$, see Fig. 7(c)), the performance of the proposed model is overall increased and further reaches peak value. When the hidden size d_g of GRU is greater than this threshold, the performance of the proposed model is saturated and even slightly reduced.

In addition, on CSI300E, we can find a similar conclusion to the case on CSI100E by observing the performance curve in Fig. 7(b). For the hyper-parameter d_g on the CSI300E, we notice that a larger d_g is not beneficial for the performance improvement of the proposed model. When the hyper-parameter d_g is greater than a threshold (i.e., $d_g = 44$), the performance of the proposed model is deteriorating rapidly. We suggest that a larger hidden size in GRU may introduce stochasticity and noises respectively from the stock prices and the news sentiments during learning stock temporal information. Specifically, in Fig. 6(d), some local details exhibit that the model performances have an upward trend when d_g distributes in an interval [70, 78]. We argue that this interval is the most sensitive and unstable region for d_g which makes agent in optimizer easier fall into a suboptimal critical point during optimization and leads to dramatic performance fluctuations though the change of hyper-parameter constraints (i.e., hidden size

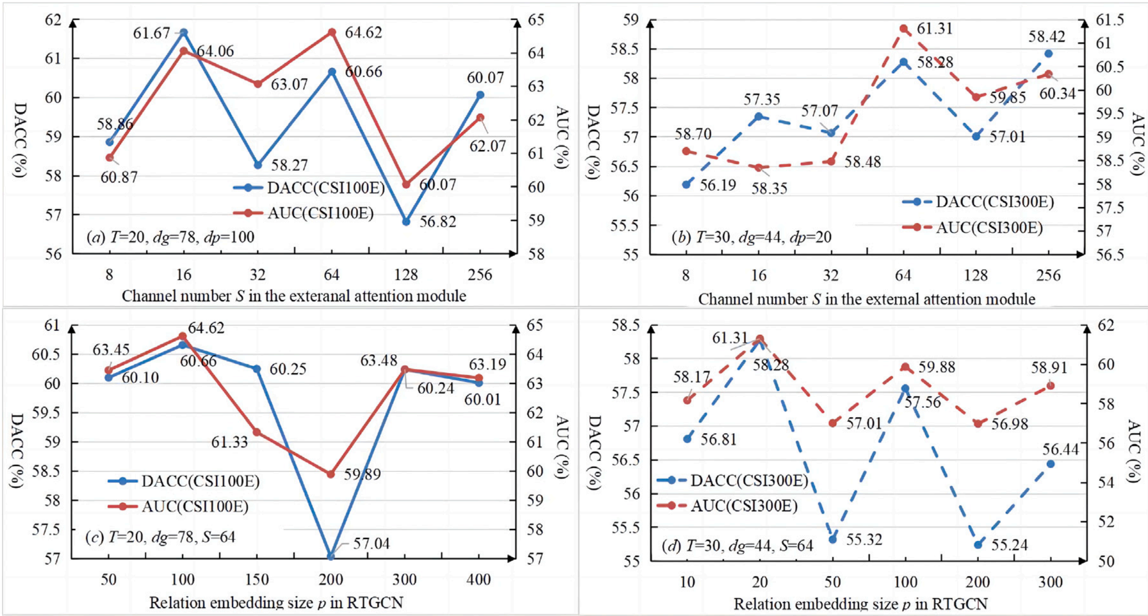

Fig. 8. The sensitivity of the channel number S and the relation embedding size p in the proposed model.

d_g) is slight. Thus, the model with $d_g = 70$ performs worst among all the settings of d_g. When d_g exceeds 78, the sensitivity of the hyper-parameter is attenuated and the performance decay becomes unhurried. This fact implies that the strict grid research is necessary to ensure the best hyper-parameter setting (i.e., $d_g = 44$ in CSI300E) and SOTA performance for SPMP.

Furthermore, in Fig. 8, it should be noted that the performances of the proposed model are also sensitive to the hyper-parameters S and p. In detail, for channel number S in Fig. 8(a) and (b), the best performances of the proposed method on CSI100E and CSI300E are accomplished when $S = 64$. A larger channel number that is more than a threshold (i.e., $S = 64$ on CSI100E and CSI300E) might not bring continuous performance improvements for the proposed approach. In the original work of external attention derived from the computational vision, the defaulted channel number is set as $S = 64$. We follow this empirical setting during the phase of the pre-research and conduct a strict grid search for model optimization. The experimental results also verify that the best settings of the channel number distribute near the default value for the external attention. Similarly, for the hyper-parameter p shown in Fig. 8(c) and (d), a large p might not be beneficial for the performance improvement of the proposed methods. We argue that the number of the relation types in the market knowledge graph is very small (see Table 2). The limited scale of relation types suggests that the proposed model usually does not require to construct edge embeddings with larger dimensions to distinguish and represent the complex semantics of relations. Besides, when RTGCN conducts node update based on the dynamics of edge embeddings, each dimension of edge embedding is viewed as a channel during pooling operation in Eq. (11). Thus, a larger dimension ($p > 100$ for CSI100E and $p > 20$ for CSI300E) of the edge embeddings can increase the computing costs and is easy to introduce noises to reduce the performance of the proposed method.

7. Conclusion

This work proposes a SPMP model via an External Attention (EA) module and a Relation Type guided Graph Convolutional Network (RTGCN). The recent efforts mainly introduce Graph Convolutional Network (GCN) to aggregate information among entities and achieve some competitive performances for SPMP. However, by analyzing these advances, we argue that two flaws may need to be concerned: (i) Real-time interaction along the signal channels. Most existing methods usually extract local signal features of an individual company while they lack the global interaction among different companies along different signal dimensions in a certain trading day. (ii) Dynamics of relation and the iterative update. Intuitively, the semantic representations of relations should be considered and dynamically updated according to the specific entities and relations, which can facilitate iterative update between entities and relations in different GCN layers.

Actually, a few recent works ever focus on these traits in GCN for the SPMP task. Considering the benefits (i.e., contributions) of the model, for the first issue, the introduced EA module can learn the global features of different companies by setting global weights in specific signal channels. For the second problem, the presented RTGCN can achieve the dynamics of relation representation using linked entities and relations to iteratively guide the model to aggregate entity information from multiple semantic dimensions better. The experiments demonstrate that the proposed model outperforms the recent advances and achieves SOTA performances.

For the implication of the proposed method, we argue that the above EA and RTGCN modules are two common modules and have the potential for application in other domains. Besides, EA and RTGCN modules in the model are hot-pluggable and can be flexibly substituted by other advanced modules. However, we find that the overfitting problem occurs in the training phase caused by the complex structures of networks. In the future, we plan to design a matched adversarial training algorithm to reduce overfitting issue and further improve the performance of SPMP.

CRediT authorship contribution statement

Hao Peng: Conceptualization, Methodology, Software, Visualization, Investigation, Writing – original draft. **Ke Dong:** Supervision, Writing – review & editing. **Jie Yang:** Data curation, Validation.

Declaration of competing interest

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

Data availability

Data will be made available on request

References

Adebiyi, A.A., Adewumi, A.O., Ayo, C.K., 2014. Comparison of ARIMA and artificial neural networks models for stock price prediction. J. Appl. Math. 2014, 614342:1–614342:7.

Ali, U., Hirshleifer, D., 2020. Shared analyst coverage: Unifying momentum spillover effects. J. Financ. Econ. 136 (3), 649–675.

Bao, W., Yue, J., Rao, Y., 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS One 12 (7).

Chauhan, S.S., Proth, J.-M., 2005. Analysis of a supply chain partnership with revenue sharing. Int. J. Prod. Econ. 97 (1), 44–51.

Chen, D., Ma, S., Harimoto, K., Bao, R., Su, Q., Sun, X., 2019a. Group, extract and aggregate: Summarizing a large amount of finance news for forex movement prediction. In: Proceedings of the Second Workshop on Economics and Natural Language Processing. Association for Computational Linguistics, Hong Kong, pp. 41–50.

Chen, Y., Wei, Z., Huang, X., 2018. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In: Cuzzocrea, A., Allan, J., Paton, N.W., Srivastava, D., Agrawal, R., Broder, A.Z., Zaki, M.J., Candan, K.S., Labrinidis, A., Schuster, A., Wang, H. (Eds.), Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018. ACM, pp. 1655–1658.

Chen, C., Zhao, L., Bian, J., Xing, C., Liu, T., 2019b. Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction. In: Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., Karypis, G. (Eds.), Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019. ACM, pp. 2376–2384.

Chen, D., Zou, Y., Harimoto, K., Bao, R., Ren, X., Sun, X., 2019c. Incorporating fine-grained events in stock movement prediction. CoRR abs/1910.05078.

Cheng, R., Li, Q., 2021. Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. AAAI Press, pp. 55–62.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, a Meeting of SIGDAT, a Special Interest Group of the ACL. ACL, pp. 1724–1734.

Derakhshan, A., Beigy, H., 2019. Sentiment analysis on stock social media for stock price movement prediction. Eng. Appl. Artif. Intell. 85, 569–578.

Ding, X., Zhang, Y., Liu, T., Duan, J., 2014. Using structured events to predict stock price movement: An empirical investigation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp. 1415–1425.

Ding, X., Zhang, Y., Liu, T., Duan, J., 2015. Deep learning for event-driven stock prediction. In: Yang, Q., Wooldridge, M.J. (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015. AAAI Press, pp. 2327–2333.

Fama, E.F., 1970. Efficient capital markets: A review of theory and empirical work. J. Finance 25 (2), 383–417.

Feng, F., Chen, H., He, X., Ding, J., Sun, M., Chua, T., 2019a. Enhancing stock movement prediction with adversarial training. In: Kraus, S. (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019. ijcai.org, pp. 5843–5849.

Feng, F., He, X., Wang, X., Luo, C., Liu, Y., Chua, T., 2019b. Temporal relational ranking for stock prediction. ACM Trans. Inf. Syst. 37 (2), 27:1–27:30.

Fuschi, D.L., Tvaronavičienė, M., et al., 2016. A network based business partnership model for SMEs management. Entrep. Sustain. Issues 3 (3), 282–289.

Hajek, P., Barushka, A., 2018. Integrating sentiment analysis and topic detection in financial news for stock movement prediction. In: Proceedings of the 2nd International Conference on Business and Information Management. ICBIM '18, Association for Computing Machinery, New York, NY, USA, pp. 158–162.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Hsieh, T., Hsiao, H., Yeh, W., 2011. Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. Appl. Soft Comput. 11 (2), 2510–2525.

Hu, Z., Dong, Y., Wang, K., Sun, Y., 2020. Heterogeneous graph transformer. In: Huang, Y., King, I., Liu, T., van Steen, M. (Eds.), WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020. ACM / IW3C2, pp. 2704–2710.

Hu, Z., Liu, W., Bian, J., Liu, X., Liu, T., 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In: Chang, Y., Zhai, C., Liu, Y., Maarek, Y. (Eds.), Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018. ACM, pp. 261–269.

Huang, W., Nakamori, Y., Wang, S.-Y., 2005. Forecasting stock market movement direction with support vector machine. Comput. Oper. Res. 32 (10), 2513–2522.

Jafari, A., Haratizadeh, S., 2022. GCNET: Graph-based prediction of stock price movement using graph convolutional network. Eng. Appl. Artif. Intell. 116, 105452.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An Introduction to Statistical Learning, Vol. 112. Springer.

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net.

Letizia, E., Lillo, F., 2019. Corporate payments networks and credit risk rating. EPJ Data Sci. 8 (1), 21:1–21:29.

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: Bessiere, C. (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020. ijcai.org, pp. 4541–4547.

Li, Q., Tan, J., Wang, J., Chen, H., 2021. A multimodal event-driven LSTM model for stock prediction using online news. IEEE Trans. Knowl. Data Eng. 33 (10), 3323–3337.

Lin, T., Guo, T., Aberer, K., 2017. Hybrid neural networks for learning the trend in time series. In: Sierra, C. (Ed.), Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. ijcai.org, pp. 2273–2279.

Liu, Q., Cheng, X., Su, S., Zhu, S., 2018a. Hierarchical complementary attention network for predicting stock price movements with news. In: Cuzzocrea, A., Allan, J., Paton, N.W., Srivastava, D., Agrawal, R., Broder, A.Z., Zaki, M.J., Candan, K.S., Labrinidis, A., Schuster, A., Wang, H. (Eds.), Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018. ACM, pp. 1603–1606.

Liu, C., Paterlini, S., 2023. Stock price prediction using temporal graph model with value chain data. arXiv preprint arXiv:2303.09406.

Liu, H., Song, B., 2017. Stock trends forecasting by multi-layer stochastic ANN bagging. In: 29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA, November 6-8, 2017. IEEE Computer Society, pp. 322–329.

Liu, Y., Zeng, Q., Yang, H., Carrio, A., 2018b. Stock price movement prediction from financial news with deep learning and knowledge graph embedding. In: Yoshida, K., Lee, M. (Eds.), Knowledge Management and Acquisition for Intelligent Systems. Springer International Publishing, Cham, pp. 102–113.

Lo, A.W., MacKinlay, A.C., 1990. When are contrarian profits due to stock market overreaction? Rev. Financ. Stud. 3 (2), 175–205.

Nayak, R.K., Mishra, D., Rath, A.K., 2015. A naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices. Appl. Soft Comput. 35, 670–680.

Nelson, D.M.Q., Pereira, A.C.M., de Oliveira, R.A., 2017. Stock market's price movement prediction with LSTM neural networks. In: 2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017. IEEE, pp. 1419–1426.

Nguyen, T.H., Shirai, K., 2015. Topic modeling based sentiment analysis on social media for stock market prediction. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Beijing, China, pp. 1354–1364.

Nguyen, T.H., Shirai, K., Velcin, J., 2016. Sentiment analysis on social media for stock movement prediction. Expert Syst. Appl. 42 (24), 9603–9611.

Pai, P.-F., Lin, C.-S., 2005. A hybrid ARIMA and support vector machines model in stock price forecasting. Omega 33 (6), 497–505.

Qian, B., Rasheed, K., 2007. Stock market prediction with multiple classifiers. Appl. Intell. 26 (1), 25–33.

Sawhney, R., Agarwal, S., Wadhwa, A., Derr, T., Shah, R.R., 2021. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. AAAI Press, pp. 497–504.

Sawhney, R., Agarwal, S., Wadhwa, A., Shah, R.R., 2020. Deep attentive learning for stock movement prediction from social media text and company correlations. In: Webber, B., Cohn, T., He, Y., Liu, Y. (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. Association for Computational Linguistics, pp. 8415–8426.

Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M., 2018. Modeling relational data with graph convolutional networks. In: Gangemi, A., Navigli, R., Vidal, M., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (Eds.), The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings. In: Lecture Notes in Computer Science, vol. 10843, Springer, pp. 593–607.

Shi, Y., Zheng, Y., Guo, K., Ren, X., 2021. Stock movement prediction with sentiment analysis based on deep learning networks. Concurr. Comput. Pract. Exp. 33 (6).

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net.

Walczak, S., 2001. An empirical analysis of data requirements for financial forecasting with neural networks. J. Manage. Inf. Syst. 17 (4), 203–222.

Wang, G., Cao, L., Zhao, H., Liu, Q., Chen, E., 2021. Coupling macro-sector-micro financial indicators for learning stock representations with less uncertainty. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. AAAI Press, pp. 4418–4426.

Xu, H., Cao, D., Li, S., 2022. A self-regulated generative adversarial network for stock price movement prediction based on the historical price and tweets. Knowl.-Based Syst. 247, 108712.

Xu, H., Chai, L., Luo, Z., Li, S., 2020. Stock movement predictive network via incorporative attention mechanisms based on tweet and historical prices. Neurocomputing 418, 326–339.

Xu, Y., Cohen, S.B., 2018. Stock movement prediction from tweets and historical prices. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Melbourne, Australia, pp. 1970–1979.

Yang, L., Dong, R., Ng, T.L.J., Xu, Y., 2019. Leveraging BERT to improve the FEARS index for stock forecasting. In: Proceedings of the First Workshop on Financial Technology and Natural Language Processing. Macao, China, pp. 54–60.

Ye, J., Zhao, J., Ye, K., Xu, C., 2020. Multi-graph convolutional network for relationship-driven stock movement prediction. In: 25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021. IEEE, pp. 6702–6709.

Zhao, Y., Du, H., Liu, Y., Wei, S., Chen, X., Zhuang, F., Li, Q., Liu, J., Kou, G., 2022. Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks. CoRR abs/2201.04965.