Pattern Recognition 143 (2023) 109759

Contents lists available at ScienceDirect

Pattern Recognition

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

Temporal-Relational hypergraph tri-Attention networks for stock trend prediction

Chaoran Cui[a,*], Xiaojie Li[a], Chunyun Zhang[a], Weili Guan[b], Meng Wang[c]

[a] School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China
[b] Faculty of Information Technology, Monash University, Clayton, VIC, Australia
[c] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China

ARTICLE INFO

Article history:
Received 1 April 2022
Revised 17 February 2023
Accepted 7 June 2023
Available online 14 June 2023

Keywords:
Stock trend prediction
Stock investment simulation
Hypergraph convolutional networks
Attention mechanism

ABSTRACT

Predicting the future price trends of stocks is a challenging yet intriguing problem given its critical role to help investors make profitable decisions. In this paper, we present a collaborative temporal-relational modeling framework for end-to-end stock trend prediction. Different from existing studies relying on the pairwise correlations between stocks, we argue that stocks are naturally connected as a collective group, and introduce two heterogeneous hypergraphs to separately characterize the stock group-wise relationships of industry-belonging and fund-holding. A novel hypergraph tri-attention network (HGTAN) is proposed to augment the hypergraph convolutional networks with a hierarchical organization of intra-hyperedge, inter-hyperedge, and inter-hypergraph attention modules. In this manner, HGTAN adaptively determines the importance of nodes, hyperedges, and hypergraphs during the information propagation among stocks, so that the potential synergies between stock movements can be fully exploited. Experimental evaluation and investment simulation on real-world stock data demonstrate the effectiveness of our approach.

1. Introduction

For a long time, the stock market has been one of the most important investment options for both individuals and institutions to chase wealth. As recently reported, the overall capitalization of major stock markets worldwide has exceeded 100 trillion U.S. dollars by the first quarter of 2021[1]. Stock trend prediction, which aims to forecast the future price trends of stocks, has received increasing attention due to its potential in helping investors make profitable decisions. Although the famous efficient market hypothesis [1] holds a pessimistic view that the future price of a stock is unpredictable with respect to currently available information, continuous research works [2–4] on stock trend prediction have achieved impressive success in past decades, and provided strong evidence for the predictability of stock markets.

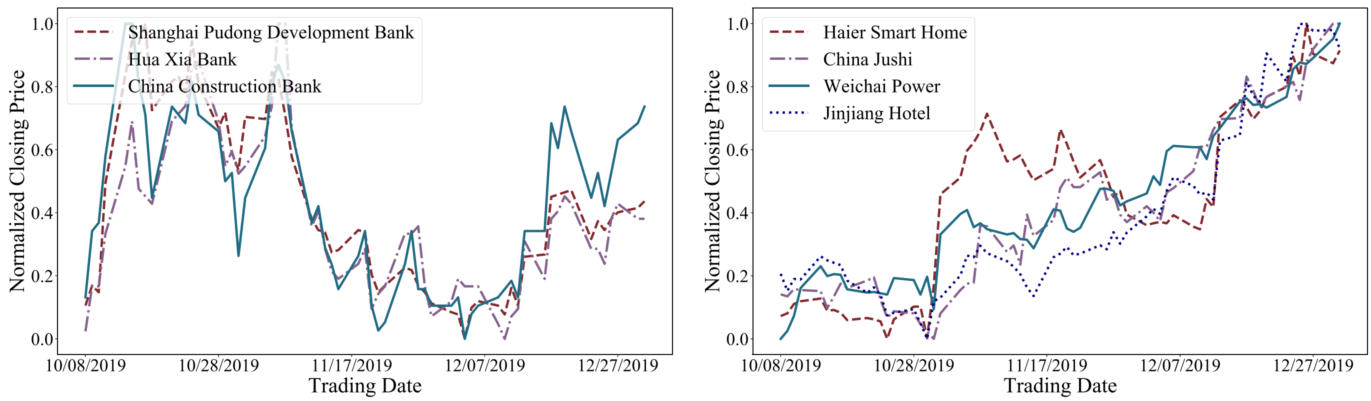
A natural solution to stock trend prediction is to regard it as a time series modeling problem, for which the autoregressive model and its variants [5] were initially applied to fit the stock trends based on the historical price data. Afterwards, classic linear models including logistic regression and support vector machine (SVM) were frequently adopted as the predictive models [6]. However, the inherent non-linear and non-stationary nature of stock prices limits the applicability of these early techniques. With the rise of deep learning, recurrent neural networks (RNNs) [7,8] and transformer networks [9] have shown promising results in stock trend prediction, owing to their powerful abilities to capture the underlying dynamics of the chaotic time series.

In another research line, the relationship information between stocks has proven to be highly valuable in improving stock trend prediction. Especially with the popularity of graph neural networks [10–12], different stocks and their relationships are typically viewed as nodes and edges in a graph, and the influence between stocks is incorporated via the node representation learning applied on the graph. Despite the encouraging progress, existing studies [13,14] make the predictions on future trends depending mainly on the correlations between any pairs of stocks. But in fact, we argue that different stocks are naturally connected as a collective group rather than by pairwise interactions. For example, multiple stocks could belong to the same industry or be held by the same fund, and they may thus share common intrinsic properties. Fig. 1 also displays the price volatility patterns of such two groups of stocks within a certain period of time. Obviously, the stocks in each group exhibit approximately consistent price trends, and the phenomenon suggests the existence of the group-wise relation-

* Corresponding author.
E-mail address: crcui@sdufe.edu.cn (C. Cui).

[1] http://www.businesskorea.co.kr/news/articleView.html?idxno=63985

https://doi.org/10.1016/j.patcog.2023.109759
0031-3203/



(a) Daily closing prices of stocks belonging to the bank industry.

(b) Daily closing prices of constituent stocks of a mutual fund.

Fig. 1. Price volatility patterns of two groups of stocks belonging to the same industry and held by the same fund in China's A-share market, respectively.

ships among stocks. As a result, simply decomposing the group-wise relationships into pairwise ones may inevitably cause the loss of information.

Motivated by the above discussions, we consider the group-wise relationships of industry-belonging and fund-holding among stocks. However, we are not saying that the two types of relationships are interrelated and affected by each other; instead, we believe that they are independent of each other. Therefore, we introduce two heterogeneous hypergraphs [15] to separately characterize the industry-belonging and fund-holding relationships. A hypergraph is a generalization of a simple graph, in which a hyperedge expresses a group-wise relationship that links multiple nodes simultaneously. Accordingly, the recently proposed hypergraph convolutional networks (HGCNs) [16,17] could be easily used for stock representation learning, so that the group-wise relationship information is integrated in stock trend prediction [18]. However, due to the complexity of the influence process between stocks, HGCNs still face three main problems: 1) It equally treats the neighbors of a stock in a hyperedge, and ignores the subtle differences of their impacts on the target stock; 2) When a stock is associated with multiple hyperedges, how to choose proper hyperedge weights remains an open question; and 3) The industry-belonging and fund-holding relationships result in two heterogeneous stock hypergraphs, but it is difficult for HGCNs to effectively coordinate them.

To address the issues, we propose a hypergraph tri-attention network (HGTAN), which augments HGCNs with a triple attention mechanism. HGTAN is equipped with the hierarchical intra-hyperedge, inter-hyperedge, and inter-hypergraph attention modules, which measure the importance of different nodes, hyperedges, and hypergraphs, respectively. In this way, HGTAN selectively aggregates the information from different sources, and fully exploits the potential synergies between stock movements. Note that our approach provides a collaborative temporal-relational modeling capacity for end-to-end stock trend prediction, because we deploy HGTAN to model the group-wise relationships among stocks after capturing the temporal dynamics of stocks with an attention-based gated recurrent unit (GRU) model. HGTAN is also compatible with all the other sequence models for stock trend prediction.

Extensive experiments are carried out on real-world data collected from China's A-share market as well as two public datasets collected from NASDAQ and NYSE markets. The results show the superiority of our approach over state-of-the-art methods for stock trend prediction. In addition, we simulate the stock investment using the trading strategies based on different methods, and the results show that our approach earns significantly higher returns with limited downside risk. Finally, detailed ablation studies are performed to investigate the efficacy of the key components in our approach. The data and codes of our work have been released at https://github.com/lixiaojieff/HGTAN.

In summary, the main contributions of our work are:

- We introduce two heterogeneous hypergraphs to separately characterize the group-wise relationships of industry-belonging and fund-holding among stocks. To the best of our knowledge, we are the first to leverage both the group-wise relationships of industry-belonging and fund-holding relationships for stock trend prediction.
- We propose a novel HGTAN consisting of hierarchical attention modules to consider the importance of different nodes, hyperedges, and hypergraphs when guiding the information propagation in stock hypergraphs.
- We conduct both experimental evaluation and investment simulation on real-world data, and the results demonstrate the validity and rationality of our approach.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 details the proposed framework for stock trend prediction. Experimental setups are described in Section 4, and the results and analysis are reported in Section 5. Section 6 concludes our work and outlines the directions of future research.

2. Related work

In this paper, we first review the existing literature on stock trend prediction. Then, we present a brief overview of the topic of hypergraph learning, which is closely related to our work.

2.1. Stock trend prediction

In the early stage, many statistical models such as autoregressive integrated moving average (ARIMA) [5] and Kalman filters [19] were widely adopted as solutions to stock trend prediction. Besides, some technical indicators were designed based on stocks' historical prices and volumes to provide insights about the future trends [20]. Machine learning techniques like logistic regression and SVM have also shown promise for stock trend prediction [6]. The major limitation of these research efforts lies in that they make the premise that the input signals are linear and stationary, regardless of the fact that the stock market is a highly volatile dynamic system. Meanwhile, they may have to manually extract useful features from the stock time series, which requires a considerable amount of domain knowledge and engineering skills.

With the huge surge of deep learning, RNNs have become a popular alternative to substitute the traditional time series models for stock trend prediction. For example, Nelson et al. [21] used the long short-term memory (LSTM) network to predict the future trends of stocks based on the price history alongside with technical indicators. Akita et al. [22] converted newspaper articles into event representations and modeled the temporal effects of past events on opening prices about multiple companies with the LSTM network. Zhang et al. [8] proposed a state frequency memory (SFM) network to decompose the hidden states of the LSTM memory cells into multiple components, each of which captures a particular frequency of latent trading pattern underlying the fluctuation of stock prices. Qin et al. [7] presented a dual-stage attention-based recurrent neural network (DARNN), which integrates two attention modules within a LSTM network to adaptively extract relevant input features at each time step and select relevant encoder hidden states across all time steps. Most recently, the transformer architecture [9], which relies solely on the self-attention mechanism to model temporal context information, has been reported to achieve remarkable results in stock trend prediction.

In real financial markets, stocks are broadly correlated with each other via a variety of relationships. It is natural to believe that the price change of a stock would be significantly affected by other related stocks. Typically, the stock relationships are viewed as a graph, and some graph-based learning methods are applied to model stock representations more effectively. For example, Chen et al. [23] established a stock graph according to the shareholding information and adopted the graph convolutional networks (GCNs) to forecast the rising or falling of stock prices. Feng et al. [13] introduced a new component in neural network modeling, named temporal graph convolution (TGC), which handles the impact between different stocks by encoding stock relationships in a time-sensitive way. Kim et al. [14] proposed a hierarchical graph attention network for stock trend prediction (HATS), which selectively aggregates information on different relationship types to learn stock representations. Bai et al. [24,25] developed new kernel measures between time-varying financial networks for multiple co-evolving financial time series analysis. Cheng et al. [4] constructed financial knowledge graphs by extracting entity relations from raw news and proposed a heterogeneous graph attention network, which leverages a two-stage attention mechanism to study internal sequential patterns and inter-source lead-lag relations.

Despite the impressive progress, most existing works simply assume stocks to be correlated in a pairwise manner. However, as previously mentioned, stocks are usually connected to each other as a group, and decomposing the group-wise relationships into pairwise ones may result in the loss of information. In this paper, we directly characterize the group-wise relationships among stocks by hypergraph modeling.

2.2. Hypergraph learning

In a simple graph, an edge can only connect two nodes, which is limited to represent pairwise connections. However, in many real-world applications, objects go beyond pairwise relationships. Hypergraph is designed to describe such a topological structure beyond pairwise interactions between nodes, in which a hyperedge can connect any number of nodes [26]. In fact, a hypergraph will degenerate into an ordinary graph if each hyperedge is associated with only two nodes.

Table 1
Summary of key notations and definitions.

Notation	Definition
$\mathcal{S}$	Set of stocks
n	Number of stocks, i.e., $n = \lvert\mathcal{S}\rvert$
m	Size of lookback window
s	Stock index, i.e., $s \in \mathcal{S}$
t	Index of trading days, i.e., $1 \le t \le m$
$\mathcal{X}$	Historical price records of stocks
$\boldsymbol{x}_{s,t}$	Price attributes of s at the t-th trading day
$\mathcal{G}_i = (\mathcal{S}, \mathcal{E}_i, \boldsymbol{w}_i)$	Industry-belonging hypergraph
$\mathcal{G}_f = (\mathcal{S}, \mathcal{E}_f, \boldsymbol{w}_f)$	Fund-holding hypergraph
e	Hyperedge index, i.e., $e \in \mathcal{E}_i$ or $e \in \mathcal{E}_f$
$w_{i,e}, w_{f,e}$	Importance of e in $\mathcal{G}_i$ and $\mathcal{G}_f$
$\widehat{\boldsymbol{y}}_s, \boldsymbol{y}_s$	Predicted and ground-truth trends of s
$\boldsymbol{h}_t$	Hidden state at the t-th trading day
$\boldsymbol{g}_s$	Temporal dynamics representation of s
$\boldsymbol{r}_s, \bar{\boldsymbol{r}}_s$	Embedding of s before and after the update of HGTAN
$\mathcal{H}_s$	Subset of hyperedges containing s
$\mathcal{N}_e$	Subset of nodes forming hyperedge e
$\boldsymbol{r}_s^e$	Hyperedge-specific embedding of s with respect to e
$\boldsymbol{r}_s^i, \boldsymbol{r}_s^f$	Hypergraph-specific embeddings of s regarding $\mathcal{G}_i$ and $\mathcal{G}_f$

In the era of deep learning, the application of graph neural networks for hypergraphs has received increasing attention. Feng et al. [16] were the first to perform spectral convolution on hypergraphs, and developed a hypergraph neural network framework. Yadati et al. [27] proposed a new way of training a GCN on hypergraphs, where a hypergraph is transformed into a simple graph and each hyperedge is represented by a simple edge whose weight is proportional to the maximum distance between two nodes in the hyperedge. Bai et al. [17] introduced two end-to-end trainable operators to the family of graph neural networks, i.e., hypergraph convolution and hypergraph attention, in order to deal with the non-pairwise relationships among nodes.

Building on the success of hypergraph learning, Sawhney et al. [18] proposed a spatiotemporal hypergraph convolutional network (STHGCN) to model the temporal evolution in stock prices and the industry-belonging relationships of stocks for predicting the future trends. However, STHGCN equally treats different nodes in the stock hypergraph, and simply mixes up all relationship information hidden in different hyperedges. By contrast, both the group-wise relationships of industry-belonging and fund-holding are leveraged in our work, and a triple attention mechanism is designed to hierarchically quantify the importance of nodes, hyperedges, and hypergraphs for better guiding the information propagation among stocks.

3. Framework

In this section, we present a collaborative temporal-relational modeling framework for end-to-end stock trend prediction. Firstly, we formulate the problem of stock trend prediction. Then, we illustrate the process of temporal dynamics modeling for stocks. Finally, we detail the hypergraph tri-attention network (HGTAN) that models the group-wise relationships among stocks. Fig. 2 displays the overall architecture of our framework.

3.1. Problem formulation

To formulate our problem, we declare some notations in advance. Throughout this paper, we use calligraphic capital letters (e.g., $\mathcal{X}$), bold capital letters (e.g., $\boldsymbol{X}$), bold lowercase letters (e.g., $\boldsymbol{x}$), and non-bold letters (e.g., x) to represent sets, matrices, vectors, and scalars, respectively. If not clarified, all vectors are in column form. Table 1 summarizes some key notations and definitions used throughout this paper.

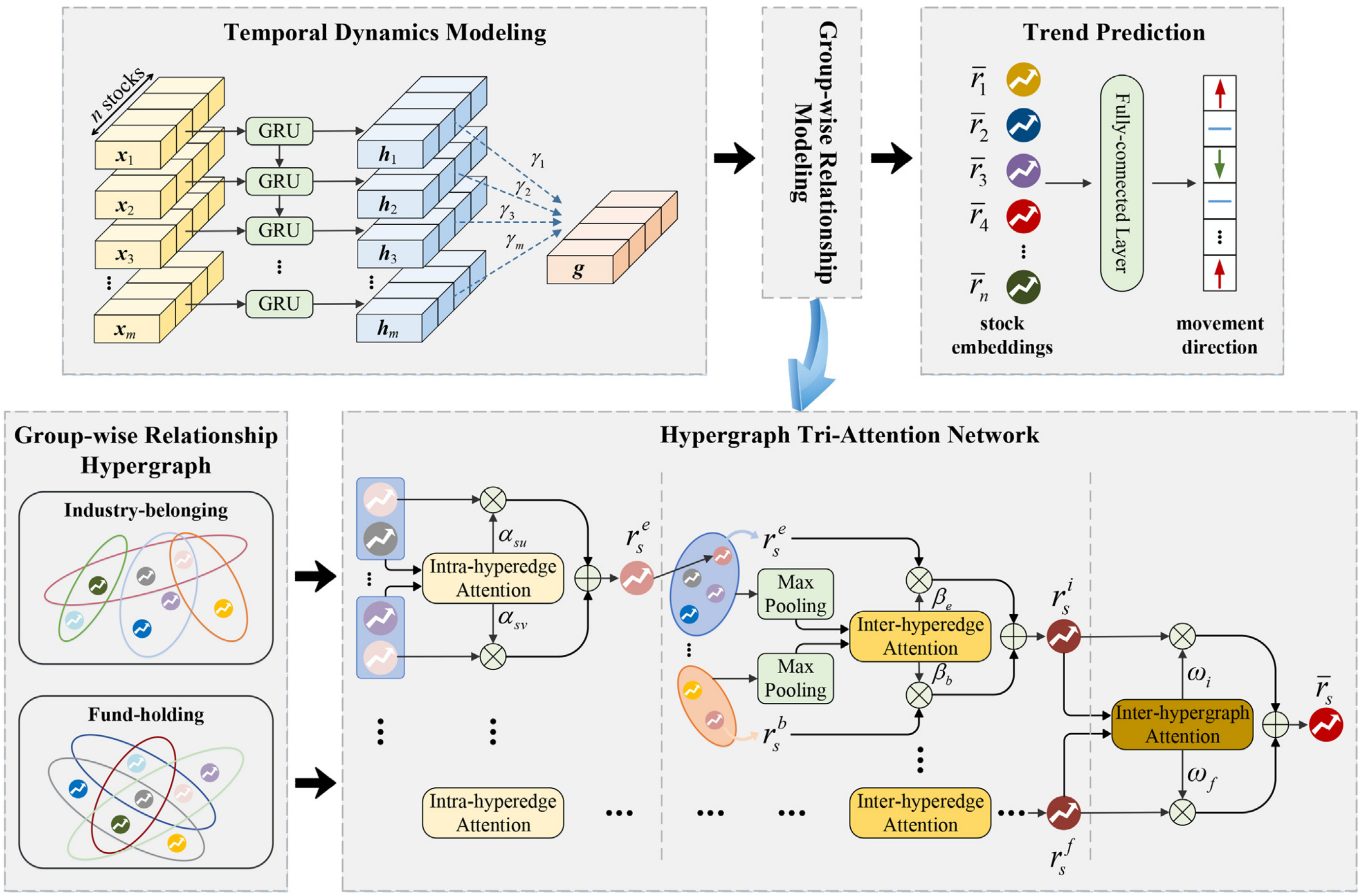


Fig. 2. The overall architecture of our framework.

Given a set of n stocks $\mathcal{S}$, we collect the historical price records over a sliding lookback window of m trading days for each stock, i.e., $\mathcal{X} = \{[\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, \ldots, \boldsymbol{x}_{s,m}], s \in \mathcal{S}\}$, where $\boldsymbol{x}_{s,t}$ is a set of price attributes of stock s at the t-th trading day, and $1 \le t \le m$. Besides, we introduce two hypergraphs to model the group-wise relationships among stocks, i.e., the industry-belonging and fund-holding relationships, respectively. A hypergraph is an extension of a simple graph, in which a set of nodes are defined as a weighted hyperedge. Let $\mathcal{G}_i = (\mathcal{S}, \mathcal{E}_i, \boldsymbol{w}_i)$ be the industry-belonging hypergraph, where $\mathcal{S}$ is taken as the node set, $\mathcal{E}_i$ is the set of hyperedges connecting different stocks belonging to the same industry, and $\boldsymbol{w}_i$ is a weight vector with the element $w_{i,e}$ representing the importance of hyperedge $e \in \mathcal{E}_i$. In a similar way, we denote by $\mathcal{G}_f = (\mathcal{S}, \mathcal{E}_f, \boldsymbol{w}_f)$ the fund-holding hypergraph, which contains the hyperedges connecting the stocks held by the same fund.

The forecast of stock market is usually cast as a regression or classification problem [13,28], which seeks to predict the price value changes or movement directions of stocks in the future, respectively. As it is much more difficult to estimate the exact price values, we focus on classifying the movement directions of stock prices. Specifically, we consider three movement directions, i.e., the rising, falling, and steady trends. Given the price records of the past m trading days and the group-wise relationships among stocks, our goal is to learn a mapping function $[\widehat{\boldsymbol{y}}_1, \widehat{\boldsymbol{y}}_2, \ldots, \widehat{\boldsymbol{y}}_n] = f(\mathcal{X}, \mathcal{G}_i, \mathcal{G}_f)$, where $\widehat{\boldsymbol{y}}_s$ is the predicted probability distribution over the rising, falling, and steady trends of stock s on the following trading day $m+1$. The future movement direction is subsequently chosen as the one with the largest probability. $\boldsymbol{y}_s$ is the ground-truth label given as a one-hot vector indicating the real movement direction of s. We use the cross entropy as the loss function to penalize the deviation of $\widehat{\boldsymbol{y}}_s$ from $\boldsymbol{y}_s$, i.e.,

$$l(\widehat{\boldsymbol{y}}_s, \boldsymbol{y}_s) = -\sum_c y_{s,c} \ln \widehat{y}_{s,c}. \tag{1}$$

where $\widehat{y}_{s,c}$ and $y_{s,c}$ are the c-th elements of $\widehat{\boldsymbol{y}}_s$ and $\boldsymbol{y}_s$, respectively. The desired mapping function $f(\cdot)$ can be determined by minimizing the loss over all stocks across different lookback windows.

3.2. Temporal dynamics modeling

Due to the volatility dependencies over time in financial markets, the historical price records of a stock play a critical role in predicting its future trend. In this paper, we use the GRU model to capture the temporal dynamics of each stock from its time series price data. Generally, the vanilla RNN suffers from the gradient vanishing and explosion problems, so its variants like LSTM and GRU are commonly used in real applications. Compared to LSTM, the GRU model has a relatively simpler structure, leading to approximately 75% of the parameters of LSTM[2] and faster training ability [29]. In our experiments, we actually tried both LSTM and GRU models, and there is no obvious difference in performance and efficiency between them.

At the t-th trading day, we individually feed the price attributes $\boldsymbol{x}_{s,t}$ of stock s to the GRU model. In the following, we shall drop the stock index s for notational simplicity. Formally, the GRU model

[2] Theoretically, the space complexity of a LSTM unit is $4(ld + l^2 + l)$, while that of GRU is $3(ld + l^2 + l)$, where d and l are the size of input and hidden state, respectively.

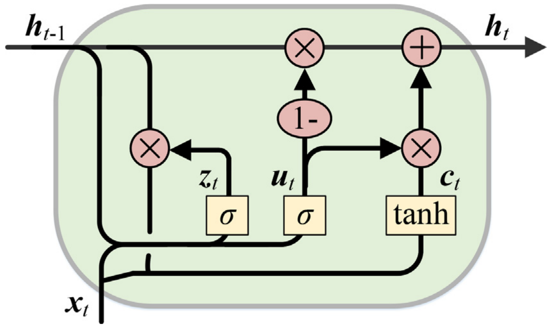


Fig. 3. Structure diagram of the GRU model.

executes the calculations as follows:

$$\begin{aligned} \boldsymbol{z}_t &= \sigma(\boldsymbol{W}_z[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]), \\ \boldsymbol{u}_t &= \sigma(\boldsymbol{W}_u[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]), \\ \boldsymbol{c}_t &= \tanh(\boldsymbol{W}_c[\boldsymbol{z}_t \odot \boldsymbol{h}_{t-1}, \boldsymbol{x}_t]), \\ \boldsymbol{h}_t &= (1-\boldsymbol{u}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{u}_t \odot \boldsymbol{c}_t, \end{aligned} \tag{2}$$

where $\boldsymbol{h}_{t-1}$ denotes a hidden state summarizing all past information up to the $(t-1)$-th trading day. $\boldsymbol{h}_{t-1}$ and $\boldsymbol{x}_t$ are firstly concatenated and transformed into a reset gate $\boldsymbol{z}_t$ and a update gate $\boldsymbol{u}_t$, respectively. The former determines how much of the past information to forget, while the latter controls how much of that needs to be brought into the current hidden state $\boldsymbol{h}_t$. Then, $\boldsymbol{h}_{t-1}$ is reset with $\boldsymbol{z}_t$, and it is again concatenated with $\boldsymbol{x}_t$ to generate a memory cell $\boldsymbol{c}_t$. $\boldsymbol{c}_t$ represents the new information to be added to $\boldsymbol{h}_t$. Finally, $\boldsymbol{h}_t$ is computed as the combination of $\boldsymbol{h}_{t-1}$ and $\boldsymbol{c}_t$, and $\boldsymbol{u}_t$ serves as a balance factor in this procedure. In Eq. (2), $\mathbf{W}_z$, $\mathbf{W}_u$, and $\mathbf{W}_c$ are the parameters to be learned, $\odot$ denotes the Hadamard product, and $\sigma(\cdot)$ and $\tanh(\cdot)$ denote the sigmoid and tanh activation functions, respectively. For a more intuitive understanding, the structure diagram of the GRU model is displayed in Fig. 3.

The GRU model successively outputs the hidden states across all m trading days, i.e., $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_m$. However, the price volatility of stocks may not be sequentially dependent. For example, stock prices frequently exhibit periodic changes over long time intervals, a typical representative of which is the phenomena of calendar effects [30]. Therefore, it is necessary to discriminate the importance of historical contexts at different moments for predicting the future trends of stocks. To this end, we further introduce a temporal attention layer to selectively emphasize informative hidden states of past trading days and suppress less useful ones [7]. Specifically, the attention weight γ_t of the hidden state $\boldsymbol{h}_t$ is measured by examining how well it is compatible with a query reference. Based on the recency bias hypothesis [31] that the future trend of a stock has a stronger correlation with its recent volatility, we choose the latest hidden state $\boldsymbol{h}_m$ as the query reference. γ_t is thus defined as follows:

$$\gamma_t = \frac{\exp(s(\boldsymbol{h}_t, \boldsymbol{h}_m))}{\sum_{j=1}^{m} \exp\left(s\left(\boldsymbol{h}_j, \boldsymbol{h}_m\right)\right)}, \tag{3}$$

where

$$s(\boldsymbol{h}_t, \boldsymbol{h}_m) = (\boldsymbol{U}_k \boldsymbol{h}_t)^T (\boldsymbol{U}_q \boldsymbol{h}_m) \tag{4}$$

is a compatibility function that transforms $\boldsymbol{h}_t$ and $\boldsymbol{h}_m$ into a latent space, and computes their dot product in the space. We generate an unified embedding $\boldsymbol{g}$ to describe the global temporal dynamics of a stock, which is computed as the weighted sum of the transformed hidden states, i.e.,

$$\boldsymbol{g} = \sum_{t=1}^{m} \gamma_t \boldsymbol{U}_v \boldsymbol{h}_t. \tag{5}$$

In the temporal attention layer, $\boldsymbol{U}_q$, $\boldsymbol{U}_k$, and $\boldsymbol{U}_v$ are three transformation matrices to be learned. Following the same steps, we can yield the temporal dynamics representations of all stocks, which are denoted as $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_n$.

3.3. Group-wise relationship modeling

In temporal dynamics modeling, different stocks are regarded as independent of each other. However, there exist many potential relationships between stocks, and the stocks with closer relationships are more likely to have similar price trends. More importantly, we notice that stocks are naturally connected as a collective group rather than by pairwise interactions. Therefore, as opposed to prior works [13,14,23] that depend on stock pairwise correlations, we leverage the hypergraph structures to characterize the inherent group-wise relationships among stocks, including the industry-belonging and fund-holding relationships.

Recently, hypergraph convolutional networks (HGCNs) [16] have emerged for modeling hypergraph-structured data and achieved state-of-the-art performance. HGCNs essentially define an information propagation rule in hypergraphs for data representation learning via a convolution operator [17]. Taking the industry-belonging hypergraph $\mathcal{G}_i = (\mathcal{S}, \mathcal{E}_i, \boldsymbol{w}_i)$ as an example, the convolution operator updates the embedding of stock $s \in \mathcal{S}$ by aggregating the information from its local neighbors in each hyperedge:

$$\bar{\boldsymbol{r}}_s = \delta\left(\sum_{e \in \mathcal{H}_s} w_{i,e} \sum_{u \in \mathcal{N}_e} \boldsymbol{P}\boldsymbol{r}_u\right), \tag{6}$$

where $\boldsymbol{r}_s$ and $\bar{\boldsymbol{r}}_s$ are the embeddings of s before and after the update, respectively. $\mathcal{H}_s \subseteq \mathcal{E}_i$ is the subset of hyperedges containing s, $\mathcal{N}_e \subseteq \mathcal{S}$ is the subset of nodes forming hyperedge e, and $\boldsymbol{P}$ is a projection matrix to be learned. $\delta(\cdot)$ denotes a nonlinear activation function like LeakyReLU. In practice, the hyperedge weight $w_{i,e}$ is usually normalized to avoid numerical instabilities. We omit this step here for expression simplicity.

By accepting the temporal dynamics representation of a stock as the input embedding (e.g., $\boldsymbol{r}_s = \boldsymbol{g}_s$), HGCNs can further integrate the group-wise relationship information in stock trend prediction [18]. However, as mentioned in the introduction, it may be inappropriate to directly apply HGCNs to the stock hypergraphs due to three main reasons: 1) HGCNs equally treat the neighbors of a stock in a hyperedge, and ignore the subtle differences of their impacts on the target stock; 2) When a stock is associated with multiple hyperedges, how to choose proper hyperedge weights remains an open question; and 3) The industry-belonging and fund-holding relationships result in two heterogeneous stock hypergraphs, but it is difficult for HGCNs to effectively coordinate them. Therefore, we propose a hypergraph tri-attention network (HGTAN), which augments HGCNs with a triple attention mechanism [32]. In particular, HGTAN consists of intra-hyperedge, inter-hyperedge, and inter-hypergraph attentions modules. Benefiting from the three hierarchically designed modules, HGTAN simultaneously takes account of the importance of different nodes, hyperedges and hypergraphs, and adaptively determines the optimal way of information propagation in stock hypergraphs. In the following, we elaborate the attention module at each level in HGTAN.

Intra-Hyperedge Attention The intra-hyperedge attention module aims to learn the importance of local neighbors of a stock within the same hyperedge. Using the same notations as given in Eq. (6), for stock s and its neighbor u in hyperedge e, we quantify the degree to which s is close to u by

$$d(\boldsymbol{r}_s, \boldsymbol{r}_u) = \delta\left(\boldsymbol{a}_d^T[\boldsymbol{P}\boldsymbol{r}_s, \boldsymbol{P}\boldsymbol{r}_u]\right), \tag{7}$$

where $\boldsymbol{a}_d$ is a shared attention vector when computing the degree between any pair of stocks. The neighbor weight α_{su} is further computed indicating how s should attend to u in e:

$$\alpha_{su} = \frac{\exp\left(d(\boldsymbol{r}_s, \boldsymbol{r}_u)\right)}{\sum_{v\in\mathcal{N}_e}\exp\left(d(\boldsymbol{r}_s, \boldsymbol{r}_v)\right)}. \tag{8}$$

In the intra-hyperedge attention module, the embedding of s with respect to e is updated via the weighted aggregation of the information provided by its neighbors:

$$\boldsymbol{r}_s^e = \delta\left(\sum_{u\in\mathcal{N}_e}\alpha_{su}\boldsymbol{P}\boldsymbol{r}_u\right). \tag{9}$$

Inter-Hyperedge Attention Intuitively, a stock may belong to multiple industries or be held by multiple funds at the same time. As a result, a node can be covered by multiple hyperedges in each stock hypergraph, and the hyperedge-specific node embedding as shown in Eq. (9) only reflects a partial view of the node. To gain a more comprehensive picture, we develop the inter-hyperedge attention module to further fuse all hyperedge-specific embeddings of an individual stock. As aforementioned, denote by $\mathcal{H}_s$ the subset of hyperedges containing s in $\mathcal{G}_i$, for hyperedge $e\in\mathcal{H}_s$, the module generates the hyperedge embedding $\boldsymbol{q}_e$ for e by

$$\boldsymbol{q}_e = \text{pool}(\{\boldsymbol{r}_s^e, s\in\mathcal{N}_e\}), \tag{10}$$

where pool$(\cdot)$ denotes the element-wise max-pooling operation. In graph theory [33], the importance of e can be measured by its closeness centrality, i.e., how close it is to all other hyperedges:

$$c(\boldsymbol{q}_e) = \sum_{b\in\mathcal{H}_s}\delta\left(\boldsymbol{a}_c^T[\boldsymbol{Q}\boldsymbol{q}_e, \boldsymbol{Q}\boldsymbol{q}_b]\right), \tag{11}$$

where $\boldsymbol{a}_c$ and $\boldsymbol{Q}$ are the trainable attention vector and projection matrix in the inter-hyperedge attention module, respectively. Similar to (8), the normalized version of $c(\boldsymbol{q}_e)$ is used to indicate the weight of e regarding s:

$$\beta_e = \frac{\exp\left(c(\boldsymbol{q}_e)\right)}{\sum_{b\in\mathcal{H}_s}\exp\left(c(\boldsymbol{q}_b)\right)}. \tag{12}$$

Furthermore, all hyperedge-specific embeddings of s are aggregated to globally characterize s in $\mathcal{G}_i$:

$$\boldsymbol{r}_s^i = \sum_{e\in\mathcal{H}_s}\beta_e\boldsymbol{r}_s^e. \tag{13}$$

Analogously, the overall representation of s in the fund-holding hypergraph $\mathcal{G}_f$ can be obtained and denoted as $\boldsymbol{r}_s^f$.

Inter-Hypergraph Attention For stock s, the inter-hypergraph attention module works on combining its dual representations $\boldsymbol{r}_s^i$ and $\boldsymbol{r}_s^f$ obtained from the heterogenous industry-belonging and fund-holding hypergraphs, respectively. Here, the attention mechanism is applied to balance the trade-off between $\boldsymbol{r}_s^i$ and $\boldsymbol{r}_s^f$ in the combination. $\boldsymbol{r}_s^i$ and $\boldsymbol{r}_s^f$ are firstly compared with each other by

$$\begin{aligned} o\left(\boldsymbol{r}_s^i\right) &= \delta\left({\boldsymbol{a}_o}^T\left[\boldsymbol{V}\boldsymbol{r}_s^i, \boldsymbol{V}\boldsymbol{r}_s^f\right]\right), \\ o\left(\boldsymbol{r}_s^f\right) &= \delta\left({\boldsymbol{a}_o}^T\left[\boldsymbol{V}\boldsymbol{r}_s^f, \boldsymbol{V}\boldsymbol{r}_s^i\right]\right), \end{aligned} \tag{14}$$

where $\boldsymbol{a}_o$ and $\boldsymbol{V}$ are the model parameters in the hypergraph-level attention. The relative weights of $\boldsymbol{r}_s^i$ and $\boldsymbol{r}_s^f$ in the combination are then computed by

$$\begin{aligned} \omega_i &= \frac{\exp(o(\boldsymbol{r}_s^i))}{\exp(o(\boldsymbol{r}_s^i)) + \exp(o(\boldsymbol{r}_s^f))}, \\ \omega_f &= \frac{\exp(o(\boldsymbol{r}_s^f))}{\exp(o(\boldsymbol{r}_s^i)) + \exp(o(\boldsymbol{r}_s^f))}. \end{aligned} \tag{15}$$

Finally, HGTAN completes the update of the embedding of s by

$$\bar{\boldsymbol{r}}_s = \omega_i\boldsymbol{r}_s^i + \omega_f\boldsymbol{r}_s^f, \tag{16}$$

and predicts the probability distribution $\hat{\boldsymbol{y}}_s$ over the future movement directions of s by feeding $\bar{\boldsymbol{r}}_s$ into a fully-connected layer with softmax activations. The entire update process of HGTAN is shown in Algorithm 1.

Algorithm 1: The update process of HGTAN.

Input: Industry-belonging hypergraph $\mathcal{G}_i = (\mathcal{S}, \mathcal{E}_i, \boldsymbol{w}_i)$,
Fund-holding hypergraph $\mathcal{G}_f = (\mathcal{S}, \mathcal{E}_f, \boldsymbol{w}_f)$,
Temporal dynamics representations of stocks $\{\boldsymbol{g}_s, \forall s\in\mathcal{S}\}$.
Output: Final embeddings of stocks $\{\bar{\boldsymbol{r}}_s, \forall s\in\mathcal{S}\}$.

for $s\in\mathcal{S}$ **do**
  $\boldsymbol{r}_s = \boldsymbol{g}_s$;**for** $\mathcal{G}\in\left\{\mathcal{G}_i, \mathcal{G}_f\right\}$ **do**
    Find the subset of hyperedges $\mathcal{H}_s\subseteq\mathcal{E}$ containing s;**for** $e\in\mathcal{H}_s$ **do**
      Find the subset of nodes $\mathcal{N}_e\subseteq\mathcal{S}$ forming e;**for** $u\in\mathcal{N}_e$ **do**
        Compute the neighbor weight α_{su}; \\ Eq. (8)
      end
      Generate the hyperedge-specific stock embedding $\boldsymbol{r}_s^e$; \\ Eq. (9)
    end
    for $e\in\mathcal{H}_s$ **do**
      Generate the hyperedge embedding $\boldsymbol{q}_e$; \\ Eq. (10)Compute the hyperedge weight β_e; \\ Eq. (12)
    end
    Generate the hypergraph-specific stock embedding $\boldsymbol{r}_s^i$ or $\boldsymbol{r}_s^f$; \\ Eq. (13)
  end
  Compute the hypergraph weights ω_i and ω_f; \\ Eq. (15)Generate the final stock embedding $\bar{\boldsymbol{r}}_s$; \\ Eq. (16)
end
return $\bar{\boldsymbol{r}}_s$;

4. Experimental setup and implementation

In this section, we describe the experimental setup and implementation details for our performance evaluation.

4.1. Data collection

Recently, there emerges several public datasets [8] for stock trend prediction, but most of them lack the stock relational data, especially the group-wise relationships required in this study. As such, we used a financial data API [3] to collect the historical price and relational data of stocks from China's A-share market by ourselves. Specifically, we chose the stocks that have price records between 01/04/2013 and 12/31/2019, obtaining 2433 stocks. Similar to the setting in [13], we further performed a filtering step to eliminate those stocks that were traded on less than 98% of all trading days. This finally results in 758 stocks. The main statistics of the data collection are summarized in Table 2.

For each stock, we extracted its six price attributes per day, including the open price, high price, low price, close price, trading volume, and trading value. If a stock experienced a temporary trading suspension, we used the price attributes of the last day before the suspension as an alternative. Additionally, we introduced four technical indicators, i.e., 5-, 10-, 20-, and 30-day moving averages, to capture the past weekly and monthly trends. Note that each attribute of a stock was separately normalized by dividing by its

[3] https://tushare.pro/document/2

Table 2
Main statistics about our collected dataset from China's A-share market.

Statistics	
Number of stocks	758
Number of training days	1021
Number of validation days	340
Number of testing days	331
Number of industry-belonging relationships	104
Number of fund-holding relationships	61
Percentage of rising days	38.0%
Percentage of falling days	37.3%
Percentage of steady days	24.7%

maximum value over the entire trading horizon. Following the previous studies [8,13], the price data were chronologically split into three time periods in the ratio of 6:2:2 for training (i.e., the first 1021 days), validation (i.e., the following 340 days), and testing (i.e., the last 331 days). Given that the goal is to predict the future price trends of stocks, most existing research [7,8,13,20] on stock trend prediction constructs an independent validation set covering the trading days after the days in the training set, instead of adopting the traditional cross-validation, in order to avoid the data leakage problem that uses the information from the future to forecast the data in the past. Besides, the training, validation, and testing sets are usually divided into three consecutive time periods in prior works, because continuous time series data are less prone to the distribution shift issue.

We also considered the industry-belonging and fund-holding relationships of stocks. For the former, we grouped all stocks into 104 industry categories according to the Shenwan Industry Classification Standard[4]; for the latter, we selected 61 mutual funds established before 2013 in the A-share market, and acquired the constituent stocks of each fund from the quarterly portfolio reports.

To ensure the comparability and fairness of the empirical results, we further carried out the experiments on two public stock datasets constructed in [13], where the authors collected the price records of 1026 NASDAQ and 1737 NYSE stocks ranging from 01/02/2013 and 12/08/2017. On both datasets, only the information of stock industry-belonging relationships is provided, without the fund-holding relationships. Details of the two public datasets can be found in [13].

4.2. Evaluation methodology and metric

In our study, the stock trend for the next trading day is defined as one of the directions of rising $(+1)$, falling (-1), and steady (0). The ground-truth label is determined based on the change ratio of the closing price, i.e.,

$$y = \begin{cases} +1, & \text{if } \frac{p_{m+1}-p_m}{p_m} \geq \beta_{\text{rising}}; \\ -1, & \text{if } \frac{p_{m+1}-p_m}{p_m} \leq \beta_{\text{falling}}; \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

where p_m and p_{m+1} are the closing prices at the m-th and $(m+1)$-th trading days, respectively. β_{rising} and β_{falling} are two threshold parameters. To balance the number of samples in different categories, we set $\beta_{\text{rising}} = 0.55\%$ and $\beta_{\text{falling}} = -0.50\%$ in line with the previous works [9,28].

We evaluated the performance of an algorithm in terms of the classification accuracy, precision, recall, and F_1 score on stock trends. These metrics can be calculated according to the number of predictions correctly made for positive classes (tp) and negative classes (tn), as well as that of predictions made falsely for both classes $(fp$ and $fn)$, i.e.,

$$\text{accuracy} = \frac{tp+tn}{tp+tn+fp+fn}, \tag{18}$$

$$\text{precision} = \frac{tp}{tp+fp}, \tag{19}$$

$$\text{recall} = \frac{tp}{tp+fn}, \tag{20}$$

$$F_1 = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{21}$$

4.3. Investment simulation

The classification performance only reflects the ability of models to forecast future stock trends, but what ultimately matters in stock markets is the profitability. To test whether the predictions made by a model can make a profit, we set up a back-testing by simulating the stock investment in China's A-Share market. In particular, the model is regarded as a market-timing tool, which generates the trading signals of buying, holding, or selling stocks based on its predictions about future trends. At the beginning of the back-testing, we evenly allocate the investment budget to each stock. The trading strategy is executed as follows: if the model predicts that a stock is expected to have a rising trend the next day, the trader will invest in the stock at the closing price. After a purchase, if the model predicts that the stock price continues to rise or keeps steady, the trader will hold the stock on that day; on the other hand, if the model predicts that the stock may show a falling trend, the trader will sell it at the closing price. In the back-testing, we only take long positions (aka. buy-hold-sell) on stocks and ignore short positions (aka. borrow-sell-buy).

Note that there is no restriction on the number of stock transactions in our trading strategy. But in fact, if the more accurately the model predicts the stock trends, the more transactions in our trading strategy will bring more profits. Intuitively, if the model can accurately predict the stock trends on the next trading day every time, we can always buy or hold in advance the stocks that will rise to earn more returns, and can always sell in advance the stocks that will fall to reduce potential losses. Therefore, it is not necessary to avoid frequent stock transactions in our trading strategy, and the better trend predictions can lead to the better profitability.

The back-testing was conducted on the trading days covered by the test set, i.e., from 08/22/2018 to 12/31/2019. At the end of the back-testing, we counted the cumulative investment return rate (IRR), which is calculated by summing over the return rates of all stocks. Transaction costs are expenses incurred when buying or selling a stock in the market. Once each transaction (i.e., buying or selling) is executed, transaction costs need to be subtracted from investment returns. Most of previous works [13,20] performed back-testing without consideration of transaction costs. However, as widely reported in literature [34], many trading strategies fail to yield the excess return once transaction costs are included. In our back-testing, we take into account a transaction cost rate of 0.03% when calculating the investment return rate, which is in accordance with the stock market practice in China. As a result, our back-testing provides a more realistic evaluation of the profitability of trading strategies.

In addition, we care about the risk exposure of trading strategies, which can be assessed by the metrics of maximum drawdown (MDD) and Sharpe ratio (SR). A maximum drawdown is the maximum observed losses from a peak of trading strategies during the back-testing. Obviously, a smaller maximum drawdown indicates a

[4] http://www.swsindex.com/idx0530.aspx

lower downside risk of trading strategies. The Sharpe ratio helps investors understand the return of trading strategies to the risk. It is defined as the ratio of the average return earned in excess of the risk-free rate to the volatility of excess return, i.e.,

$$SR = \frac{IRR_a - R_f}{\sigma}, \tag{22}$$

where IRR_a denotes the annual investment return rate, R_f denotes the risk-free rate, and σ denotes the standard deviation of excess return during the back-testing. In this case, we set the risk-free rate to the interest rate of one year time deposit announced by the People's Bank of China in 2019[5], i.e., $R_f = 1.5\%$. Intuitively, the greater the value of the Sharpe ratio, the more attractive the risk-adjusted return.

4.4. Baseline

We compared our proposed HGTAN against different algorithms for stock trend prediction. Specifically, we first chose two traditional stock technical indicators as baselines:

- **MOM** [35]: The momentum technical indicator suggests that the future direction of stock price is consistent with that in the recent period.
- **MR** [28]: The mean reversion indicator predicts the movement of stock price as the opposite direction of current price towards the past average price.

Three deep sequence models were further introduced to the comparison:

- **LSTM** [21]: This method is the LSTM model that sequentially accepts the time series price data to make a prediction on future stock trend.
- **DARNN** [7]: This method utilizes a dual-stage attention-based recurrent neural network, which adaptively extracts relevant input features at each time step and selects relevant encoder hidden states across all time steps.
- **SFM** [8]: This method extends LSTM by decomposing the hidden memory states into multiple components, and models the latent trading patterns with multiple frequencies to predict the trend of stock prices.

Besides, we experimented with several recently proposed algorithms that are also based on stock relationships:

- **GCN** [23]: This method uses a LSTM network to encode the historical price data of stocks, and the results are then fed into a GCN to learn based on the relationships between stocks.
- **TGC** [13]: This method devises a new component of neural network modeling, named temporal graph convolution, which generates the relational embeddings of stocks in a time-sensitive way.
- **HATS** [14]: This method presents a hierarchical graph attention network that selectively aggregates different types of relational data to learn stock representations.
- **STHGCN** [18]: This method models the industry-belonging relationships of stocks via a hypergraph, and introduces a gated temporal convolution to capture the temporal dependencies in stock price features.

Note that most of the baseline methods adopt LSTM or GRU as the backbone model to capture the temporal dynamics of stocks. Therefore, we choose an attention-based GRU model in HGTAN to provide a fair comparison with them. More complicated sequence models like the transformer network [9] can also be adopted, but we leave them for future exploration.

[5] https://cn.investing.com/economic-calendar/pboc-deposit-rate-1082

4.5. Implementation details

We implemented all methods evaluated in this study with Pytorch[6] except SFM, for which the original Keras implementation[7] was directly used. For the sake of reproducibility, the data and codes of our work have been released at https://github.com/lixiaojieff/HGTAN.

In our implementation, to ensure the fairness of the empirical results, the price data of stocks at each trading day is embedded in a 16-dimensional space before fed into different methods. For baseline methods, we determined the optimal hyperparameters either following the suggested settings in the original papers or based on the accuracy obtained on the validation set. In our model, all hyperparameters were also optimized with the validation set. Specifically, we tuned most hyperparameters over the set $\{8, 16, 32, 64\}$. Finally, we chose the number of hidden layers and the number of cells per layer in the GRU model to be 2 and 32, respectively. The size of stock embeddings before and after the update of HGTAN was set to 16 and 8, respectively. Our network was trained with the mini-batch Adam optimizer [36]. During training, we empirically set the batch size to 64, the dropout rate to 0.5, the initial learning rate to 10^{-3}, and the maximum number of epochs to 600, respectively.

5. Experimental results and analysis

In this section, we report a series of experimental results to validate the effectiveness of the proposed HGTAN method. Note that for each method compared in our experiments, we repeat the training and testing procedures five times, and report the average performance to alleviate the fluctuations caused by random initializations. Through these experiments, we try to address the following research questions:

- **RQ1**: Does our method achieve superior performance in stock trend prediction?
- **RQ2**: Does our method help investors earn excess returns from stock investment in real market?
- **RQ3**: Does our method work better with the group-wise relationships of industry-belonging and fund-holding among stocks?
- **RQ4**: Does our method benefit from the triple attention mechanism in hypergraph modeling?

5.1. Classification performance

For stock trend prediction on the China's A-share dataset, we studied how different methods perform when considering different lengths of price records, including the past 5 trading days, 10 trading days, and 20 trading days (i.e., $m = 5$, 10, and 20 in Eq. (3)-(5)). The settings simulate the scenario that we predict the future stock price trend depending on the historical data from the past week, two weeks, or month. Table 3 lists the performance comparison between different methods, from which we can make the following observations:

- As a traditional technical indicator, MR exhibits acceptable performance with respect to the other complex deep learning based models. For example, in the case of 5-trading day records, MR achieves the best result in terms of precision. The observation is in accordance with previous financial empirical studies [37], which have proven that simple forms of technical analysis contain significant forecasting power, especially with the short-term technical indicators.

[6] https://pytorch.org

[7] https://github.com/z331565360/State-Frequency-Memory-stock-prediction

Table 3
Classification performance of different methods with the transaction records over different number of past trading days on the China's A-share dataset.

Method	5 trading days				10 trading days				20 trading days			
	Accuracy	Precision	Recall	F_1	Accuracy	Precision	Recall	F_1	Accuracy	Precision	Recall	F_1
MOM [35]	34.52%	34.79%	31.88%	33.27%	34.50%	34.94%	32.07%	33.44%	35.73%	35.19%	32.82%	33.96%
MR [28]	35.59%	**39.37%**	33.77%	36.36%	34.73%	29.34%	31.79%	30.52%	35.32%	38.03%	33.60%	35.68%
LSTM [21]	34.92%	35.34%	33.91%	34.27%	35.09%	38.09%	34.37%	35.90%	35.03%	36.43%	34.23%	35.20%
DARNN [7]	37.68%	37.81%	35.17%	36.43%	38.89%	38.59%	35.22%	36.82%	38.41%	37.99%	**39.24%**	38.60%
SFM [8]	33.29%	26.83%	33.35%	29.52%	34.95%	24.82%	33.34%	28.22%	34.54%	26.93%	33.32%	29.49%
GCN [23]	37.24%	37.23%	33.54%	35.22%	37.44%	39.07%	34.49%	36.62%	37.30%	39.28%	34.16%	36.54%
TGC [13]	37.43%	38.28%	34.05%	36.01%	38.42%	39.35%	35.72%	37.44%	37.81%	36.96%	34.49%	35.67%
HATS [14]	38.74%	36.92%	34.29%	35.52%	38.05%	39.23%	34.52%	36.67%	38.85%	38.70%	35.06%	36.78%
STHGCN [18]	38.53%	37.35%	34.65%	35.89%	38.81%	36.57%	35.11%	35.75%	38.45%	37.22%	32.82%	34.87%
HGTAN	**39.51%**	38.90%	**36.96%**	**37.89%**	**39.83%**	**41.72%**	**37.32%**	**39.37%**	**40.02%**	**41.77%**	39.03%	**40.32%**

The best result in terms of each metric is indicated in bold, and the second best one is underlined. This convention is also adopted in the following tables.

Table 4
Classification performance of different methods on the NASDAQ and NYSE datasets when considering 10 trading days of price records.

Method	NASDAQ				NYSE			
	Accuracy	Precision	Recall	F_1	Accuracy	Precision	Recall	F_1
MOM [35]	31.26%	33.91%	32.26%	33.07%	26.83%	37.44%	32.53%	34.81%
MR [28]	32.75%	35.04%	34.19%	34.61%	27.78%	34.90%	34.34%	34.62%
LSTM [21]	37.22%	34.64%	36.56%	35.52%	45.73%	36.22%	38.04%	37.08%
DARNN [7]	40.46%	37.05%	37.76%	37.40%	47.98%	41.41%	39.53%	**40.44%**
SFM [8]	33.41%	11.13%	33.33%	16.68%	45.73%	15.24%	34.48%	21.13%
GCN [23]	39.75%	**40.82%**	38.78%	**39.76%**	45.99%	35.89%	37.38%	36.31%
TGC [13]	39.98%	38.24%	38.08%	38.16%	47.95%	**41.94%**	38.54%	40.15%
STHGCN [18]	40.11%	39.84%	**39.09%**	39.46%	47.08%	39.48%	37.57%	38.47%
HGTAN	**40.67%**	38.11%	38.86%	38.48%	**48.25%**	41.02%	**39.84%**	40.42%

- As deep sequence models, LSTM and SFM considerably fall behind those stock relationship based contenders, including GCN, TGC, HATS, STHGCN, and HGTAN. This highlights the importance of relationship information for stock trend prediction. On the other hand, DARNN frequently achieves competitive performance merely using the historical price data. Contrasting to LSTM and SFM, DARNN additionally incorporates the attention mechanism to adaptively select relevant time series features. The performance gap between them suggests the benefit of introducing the attention mechanism to capture stock temporal dynamics.
- The proposed HGTAN consistently outperforms the other competitors, leading to the best or runner-up performance in all cases. More precisely, it exceeds the second place by an average of nearly 1.0% and 1.70% in terms of accuracy and F_1 score, respectively. The results clearly demonstrate the effectiveness of HGTAN for stock trend prediction, and provide the evidence that the research question **RQ1** can be positively answered.
- HGTAN offers a steady improvement in performance with the increase of the lookback window of past trading days. For instance, the F_1 score obtained by HGTAN gradually arises from 37.89% in the case of 5-trading day records to 39.37% and 40.32% in those of 10- and 20-trading day records, respectively. Note that such a phenomenon is not observed for most of the other methods, which could experience a performance degradation when considering a longer period of records. Therefore, we believe that HGTAN has a higher capability of making full use of historical price data.

Furthermore, we continued our experiments on the NASDAQ and NYSE stock datasets when considering 10 trading days of price records. Table 4 shows the classification results of different competitors. As aforementioned, on the NASDAQ and NYSE datasets, only the industry-belonging relationships between stocks are available. It can be found that the performance of graph-based models is significantly improved by leveraging the relationship information. Due to the lack of fund-holding relationships of stocks, our HGTAN can only implemented without the inter-hypergraph module. However, even in such a limited setting, HGTAN achieves promising prediction results, consistently standing at the top two positions in most evaluation metrics. This observation again verifies the effectiveness of HGTAN for stock trend prediction, and underpins our belief that the research question **RQ1** can be positively supported.

Table 5
Profitability of different methods during the back-testing.

Method	IRR	MDD	SR
Buy-and-Hold	3.84%	19.57%	0.076
MOM [35]	4.01%	8.74%	0.179
MR [28]	4.89%	18.13%	0.123
LSTM [21]	4.73%	13.87%	0.147
DARNN [7]	3.23%	12.34%	0.083
SFM [8]	5.85%	5.26%	0.513
GCN [23]	6.51%	14.44%	0.217
TGC [13]	8.23%	8.60%	0.513
HATS [14]	11.55%	7.31%	0.697
STHGCN [18]	6.23%	10.43%	0.248
HGTAN	**25.17%**	**4.23%**	**1.792**

5.2. Profitability

Table 5 reports the profitability comparison of different methods during the back-testing. To observe the overall volatility of the stock market, a passive buy-and-hold trading strategy is also included as a benchmark for the comparison. In the buy-and-hold strategy, investors are assumed to purchase all stocks at the beginning of the back-testing, and hold them until they are sold at the end of the back-testing. For a more intuitive understanding, Fig. 4

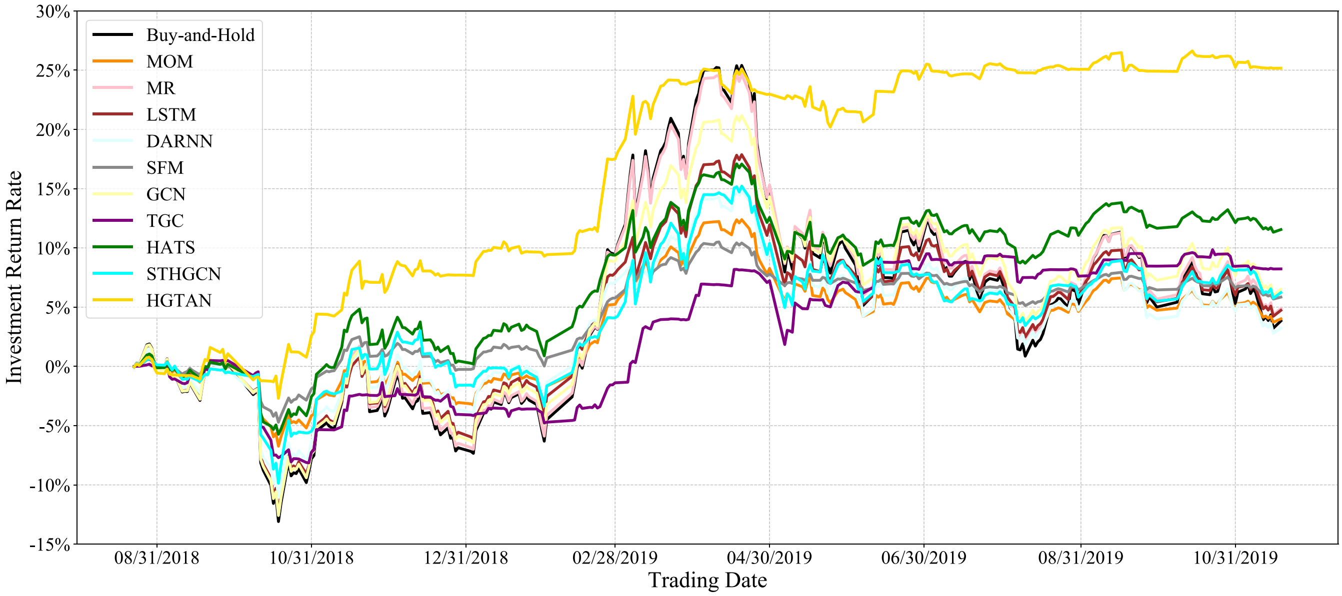


Fig. 4. The fluctuation curves of different methods regarding the cumulative investment return rate during the back-testing.

plots the fluctuation curves of different methods regarding IRR during the back-testing. From the results, we can see that:

- Most active trading strategies generated with different methods beat the passive buy-and-hold strategy, indicating that both technical indicator based and machine learning based models are capable of providing profitable trading signals to some extent. The only exception is DARNN that yields 3.23% return rate slightly inferior to 3.84% for the buy-and-hold strategy. This is perhaps surprising given the relatively good performance of DARNN for stock trend prediction as shown in Table 3. Therefore, it cannot be definitively concluded that the trend prediction ability and profitability of models are equivalent for stock investment.
- The proposed HGTAN earns significantly higher returns than the other methods with a cumulative rate of 25.17%. Notably, it achieves more than twice the return obtained by the second-best method, and six times that of the buy-and-hold strategy. On the other hand, HGTAN shows a stronger power of risk management, reaching the lowest maximum drawdown among all competitors. It is worthwhile to point out that all the other methods fail to yield a Sharpe ratio above one, implying that their returns are not high enough to compensate for the risk. By contrast, HGTAN results in a more desirable risk-adjusted return with a Sharpe ratio of 1.792. In other words, HGTAN is able to provide more return under the same risk. As a result, we can give a positive answer to the research question **RQ2**.
- By analyzing the investment return curves of different methods in Fig. 4, we notice that HGTAN produces approximately stable and continuous positive returns throughout the back-testing. Particularly, the advantage of HGTAN over the others mainly lies in its superior performance when the stock market is in a downtrend. Taking the period from 04/23/2018 to 05/10/2018 as an example, for which Fig. 5 shows the number of stocks with rising, falling, and steady trends per trading day. As can be seen, the market is in a bear stage at that time, and there are far more stocks going down than those going up or keeping steady. Meanwhile, Table 6 lists the accuracy of predicting stock trends of different methods per day in the same period. Clearly, HGTAN provides more accurate predictions in the bear stage, so that selling decisions can be made in time to avoid losses.

5.3. Impact of group-wise relationships

In our study, we characterize the group-wise relationships of industry-belonging and fund-holding among stocks via hypergraph modeling. To verify the effectiveness of this scheme, we conduct the performance comparison between GCN and HGCN with different kinds of relationship information on the China's A-share dataset. Notably, GCN generates the stock embeddings based on the pairwise relationships, for which any pair of stocks are connected if they are both belonging to the same industry or held by the same fund; instead, HGCN works on the group-wise relationships that directly brings together all stocks associated with an industry or fund. Table 7 summarizes the comparison results, in which the suffixes '**-I**', '**-F**', and '**-M**' indicate the methods using the industry-belonging relationship information, the fund-holding relationship information, as well as the mixture of the two, respectively. From the table, we have the following findings:

- Overall, GCN is substantially worse than HGCN, whether exploiting the industry-belonging or fund-holding relationship information. The observation confirms our belief that simply compressing the group-wise relationships into pairwise ones inevitably causes the loss of information.
- It is difficult to tell which is better between HGCN-I and HGCN-F, but HGCN-M frequently obtains higher performance over them. The results show that the industry-belonging and fund-holding relationship information are independent but somewhat complementary to each other, and jointly leveraging them is beneficial to improving stock trend prediction. Therefore, a positive answer to the research question **RQ3** can be formed.

Note that several recent research endeavors [38,39] have provided the preliminary theoretic frameworks for analyzing the expressive power of GCNs. For example, Xu et al. [38] demonstrated that GNNs can be as powerful as the Weisfeiler-Lehman graph isomorphism test if the neighbor aggregation and graph-level readout functions are injective. Exploring the theoretic foundations and explanations of the GCN and HGCN models is indeed an important topic but beyond the scope of our work. We recommend the survey article [40] for readers interested in this topic.

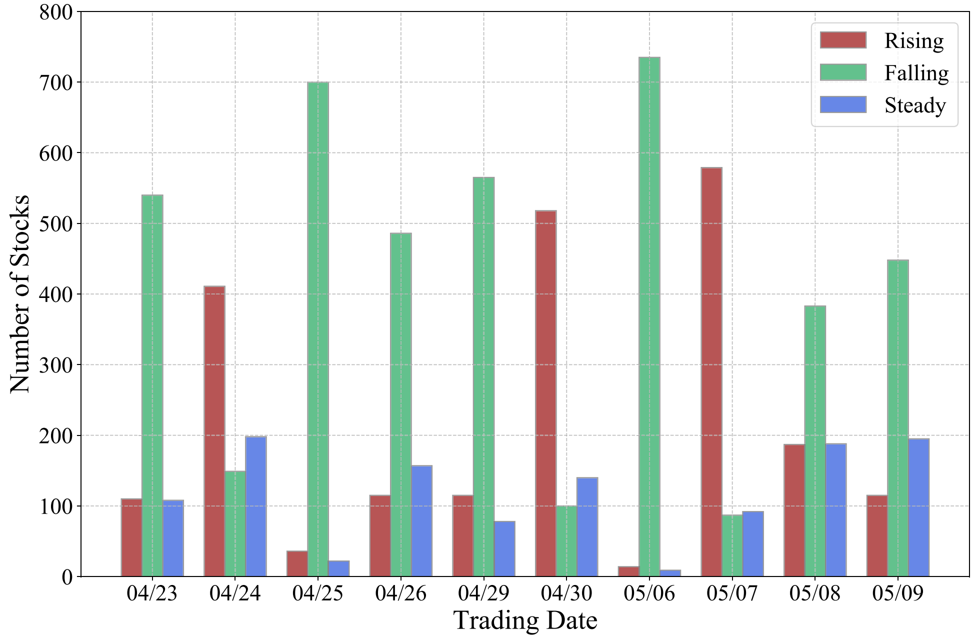


Fig. 5. Number of stocks with different trends per day when the market is in a bear period from 04/23/2018 to 05/10/2018.

Table 6
Trend prediction accuracy of different methods per day when the market is in a bear period from 04/23/2018 to 05/10/2018.

Date	04/23	04/24	04/25	04/26	04/29	04/30	05/06	05/07	05/08	05/09
MOM [35]	43.27%	36.94%	48.02%	38.39%	44.46%	37.86%	48.29%	40.90%	37.60%	36.41%
MR [28]	16.23%	**52.77%**	9.76%	17.68%	19.13%	64.78%	7.12%	<u>73.48%</u>	25.33%	17.81%
LSTM [21]	24.80%	46.44%	14.38%	22.43%	22.03%	<u>66.49%</u>	8.31%	69.92%	30.87%	20.32%
DARNN [7]	35.36%	43.27%	20.19%	30.74%	24.14%	61.87%	14.12%	57.26%	30.61%	25.86%
SFM [8]	20.05%	30.87%	14.38%	<u>56.86%</u>	62.27%	22.56%	<u>52.59%</u>	22.03%	**48.68%**	<u>53.30%</u>
GCN [23]	22.43%	<u>48.02%</u>	15.44%	21.50%	23.22%	63.72%	11.35%	71.11%	28.36%	18.73%
TGC [13]	**69.39%**	21.24%	<u>77.97%</u>	43.40%	<u>65.57%</u>	65.44%	15.17%	15.30%	<u>48.15%</u>	22.69%
HATS [14]	37.86%	38.26%	31.40%	38.13%	43.67%	50.53%	48.55%	50.26%	44.59%	33.25%
STHGCN [18]	21.24%	27.44%	15.04%	23.88%	14.78%	**66.62%**	12.67%	**75.20%**	26.65%	19.26%
HGTAN	<u>57.78%</u>	22.03%	**88.26%**	**60.16%**	**72.43%**	16.62%	**76.52%**	31.27%	44.99%	**56.07%**

Table 7
Classification performance comparison between GCN and HGCN with different kinds of relationship information.

Method	5 trading days				10 trading days				20 trading days			
	Accuracy	Precision	Recall	F_1	Accuracy	Precision	Recall	F_1	Accuracy	Precision	Recall	F_1
GCN-I	37.45%	35.63%	<u>34.79%</u>	35.19%	37.21%	35.23%	33.96%	34.55%	<u>37.88%</u>	36.74%	<u>35.66%</u>	36.19%
GCN-F	37.63%	33.85%	34.47%	34.15%	37.23%	34.01%	34.29%	34.14%	36.48%	32.51%	33.31%	32.90%
GCN-M	37.24%	37.23%	33.54%	35.22%	37.44%	<u>39.07%</u>	34.49%	36.62%	37.30%	**39.28%**	34.16%	36.54%
HGCN-I	37.55%	**37.83%**	34.62%	**36.15%**	38.29%	37.98%	35.01%	36.43%	37.70%	37.76%	35.64%	**36.67%**
HGCN-F	<u>38.03%</u>	36.76%	**35.23%**	<u>35.97%</u>	<u>38.32%</u>	37.65%	**36.23%**	<u>36.92%</u>	37.19%	36.58%	**36.52%**	<u>36.55%</u>
HGCN-M	**38.15%**	<u>37.60%</u>	33.72%	35.55%	**38.78%**	**39.14%**	<u>35.35%</u>	**37.15%**	**38.13%**	<u>38.14%</u>	34.92%	36.44%

5.4. Contribution of triple attention mechanism

In the proposed HGTAN, we introduce a triple attention mechanism consisting of the intra-hyperedge, inter-hyperedge, and inter-hypergraph attention modules to hierarchically measure the importance of nodes, hyperedges, and hypergraphs during the process of information propagation. To investigate the contribution of the triple attention mechanism, we devise three variants of HGTAN, named '$\mathbf{EW_{intraE}}$', '$\mathbf{EW_{interE}}$', and '$\mathbf{EW_{interG}}$', in which the intra-hyperedge, inter-hyperedge, and inter-hypergraph attention modules are replaced by assigning equal weights to different nodes, hyperedges, and hypergraphs, respectively. Table 8 compares the performance between HGTAN and the variant methods. It can be seen that when any of the intra-hyperedge, inter-hyperedge, and inter-hypergraph attention modules is missing, the variant methods become clearly inferior to HGTAN in most cases. This underlines the necessity of simultaneously weighing the importance of nodes, hyperedges, and hypergraphs for guiding the information propagation in stock hypergraphs, and also indicates that the triple attention mechanism plays a critical role in HGTAN. Therefore, we can give a positive answer to the research question **RQ4**.

5.5. Visualization and case study

To gain an intuitive understanding about the representation learning ability of HGTAN, we took the embeddings of 300 randomly chosen China stocks generated by HGTAN, and projected them into a two-dimensional space using the t-SNE algorithm [41]. Fig. 6 displays the projection results. Taking the industries of banking, airlines, and liquor as examples, we can see that some stocks

Table 8
Classification performance comparison between HGTAN and its variants.

Method	5 trading days				10 trading days				20 trading days			
	Accuracy	Precision	Recall	F_1	Accuracy	Precision	Recall	F_1	Accuracy	Precision	Recall	F_1
EW_{intraE}	37.54%	38.97%	34.79%	36.76%	38.92%	39.35%	35.36%	37.65%	39.36%	40.27%	37.62%	38.90%
EW_{interE}	38.67%	38.50%	33.81%	36.00%	39.63%	39.06%	34.68%	36.74%	38.74%	41.16%	37.52%	39.26%
EW_{interG}	38.97%	**39.20%**	36.43%	37.76%	39.35%	41.41%	37.23%	39.21%	39.85%	39.31%	**39.29%**	39.30%
HGTAN	**39.51%**	38.90%	**36.96%**	**37.89%**	**39.83%**	**41.72%**	**37.32%**	**39.37%**	**40.02%**	**41.77%**	39.03%	**40.32%**

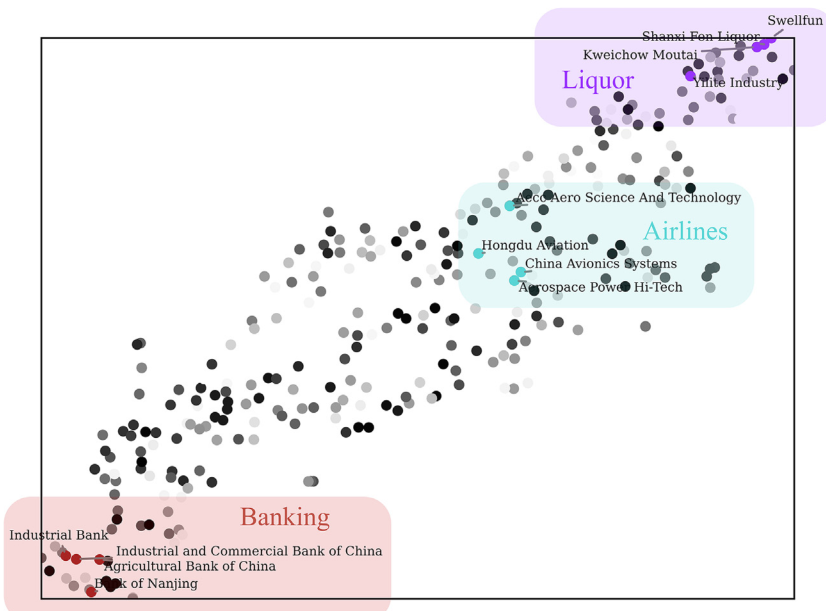
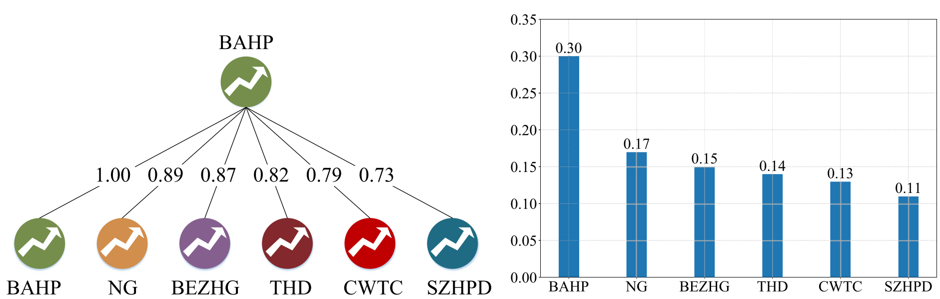


Fig. 6. Visualization of the learned embeddings of 300 randomly chosen stocks. Some stocks belonging to the industries of banking, airlines, and liquor are marked by colored points.



(a) Correlation coefficients regarding the price movement direction.

(b) Attention weights.

Fig. 7. The correlation coefficients between BAHP and each neighbor, and the attention weights of the neighbors in the hyperedge of park development industry.

in the same industry are embedded close to each other, which are marked by colored points in the figure. The finding suggests that the representations learned from HGTAN indeed reflect the industry relatedness between different stocks to some extent.

Given a target stock, HGTAN can learn the attention weights to indicate the importance of local neighbors of the stock within the same hyperedge. Here, we present a case study for the stock Beijing Airport High-Tech Park (BAHP), in which its five neighbors[8] in the hyperedge of park development industry are considered. We show the correlation coefficient between the price movement directions of BAHP and each neighbor in Fig. 7(a), as well as the attention weights assigned to the neighbors in Fig. 7(b). By comparison, we can observe that there exists the consistency between the neighbor lists sorted by correlation coefficient and attention weight, i.e., the neighbors having more similar price trends to BAHP are assigned higher attention weights. Moreover, BAHP gets the highest attention weight, which is reasonable in the sense that a node itself often plays the most important role in learning its representation. Such results demonstrate that HGTAN discriminates between the neighbors belonging to the same industry, and effectively identifies those meaningful ones.

[8] Nanjing Gaoke (NG), Beijing Electronic Zone High-Tech Group (BEZHG), Tianjin High-Tech Development (THD), China World Trade Center (CWTC), and Shanghai Zhangjiang High-Tech Park Development (SZHPD)

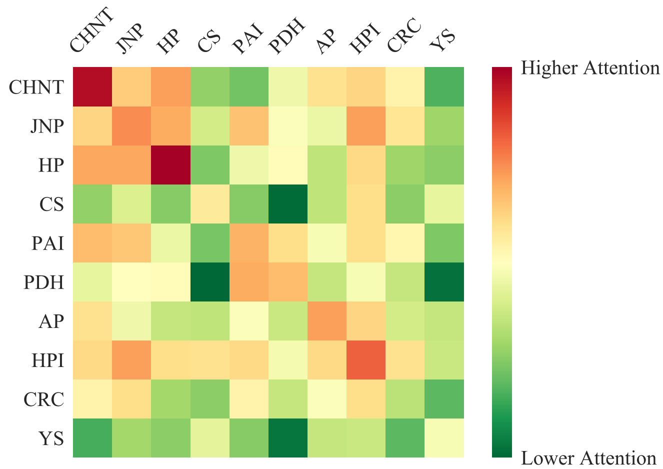


Fig. 8. Heatmap depicting the attention weights between each pair of stocks in a fund.

As for the fund-holding relationships, we take the Xingquan Global Vision Equity Securities Investment Fund as an example, and obtain its top 10 constituent stocks[9] in the fourth quarter of 2018. Fig. 8 visualizes the attention weights between each pair of stocks in the fund. As can be seen, the first three stocks, including CHNT, JNP, and HPG, pay much more attention to each other. A possible reason is that they all belong to the pharmaceutical industry. The phenomenon indicates that HGTAN is also able to capture the underlying correlations between the stocks held by the same fund.

6. Conclusion

Inspired by the observation that different stocks are naturally connected as a collective group, we present a collaborative temporal-relational modeling framework for end-to-end stock trend prediction, in which a hypergraph tri-attention network (HGTAN) is introduced to model the stock group-wise relationships of industry-belonging and fund-holding after capturing the temporal dynamics of stocks. HGTAN is equipped with the hierarchical intra-hyperedge, inter-hyperedge, and inter-hypergraph attention modules, which simultaneously measures the importance of nodes, hyperedges, and hypergraphs for guiding the information propagation in stock hypergraphs. In the experiments, we demonstrate that our approach substantially outperforms state-of-the-art methods on real-world dataset, and show its superior profitability through an investment simulation. We also perform detailed ablation analysis on its key components, as well as visualization and case studies to provide more insights into our approach.

Building upon the current study, our future work will be carried out along three directions. Firstly, we intend to collect more diverse stock-related data like online financial news and social media contents, and integrate these additional cues in improving stock trend prediction. Secondly, we will explore alternative trading strategies to generate more reliable trading signals based on the predictions about future stock trends. Finally, we will test on more real-world stock markets to further validate the adaptability of our approach.

[9] Changchun High and New Tech (CHNT), Jiangsu Nhwa Pharmaceutical (JNP), Haisco Pharmaceutical (HP), Huaneng Power International (HPI), Citic Securities (CS), Avicopter PLC (AP), Poly Developments and Holdings (PDH), China Railway Construction (CRC), Ping An Insurance (PAI), and Yonghui Superstores (YS)

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.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62077033 and Grant 61701281, by the Shandong Provincial Natural Science Foundation under Grant ZR2020KF015, and by the Taishan Scholar Program of Shandong Province under Grant tsqn202211199.

References

[1] E.F. Fama, The behavior of stock-market prices, J. Bus. 38 (1) (1965) 34–105.
[2] P.-Y. Yang, Z.-R. Lai, X. Wu, L. Fang, Trend representation based log-density regularization system for portfolio optimization, Pattern Recognit. 76 (2018) 14–24.
[3] T. Yin, C. Liu, F. Ding, Z. Feng, B. Yuan, N. Zhang, Graph-based stock correlation and prediction for high-frequency trading systems, Pattern Recognit. 122 (2022) 108209.
[4] D. Cheng, F. Yang, S. Xiang, J. Liu, Financial time series forecasting with multi–modality graph neural network, Pattern Recognit. 121 (2022) 108218.
[5] A.A. Adebiyi, A.O. Adewumi, C.K. Ayo, Comparison of ARIMA and artificial neural networks models for stock price prediction, J. Appl. Math. 2014 (1) (2014) 1–7.
[6] M. Ballings, D. Van den Poel, N. Hespeels, R. Gryp, Evaluating multiple classifiers for stock price direction prediction, Expert Syst. Appl. 42 (20) (2015) 7046–7056.
[7] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, G.W. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 2627–2633.
[8] L. Zhang, C. Aggarwal, G.-J. Qi, Stock price prediction via discovering multi-frequency trading patterns, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 2141–2149.
[9] Q. Ding, S. Wu, H. Sun, J. Guo, J. Guo, Hierarchical multi-scale gaussian transformer for stock movement prediction, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 4640–4646.
[10] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of the 5th International Conference on Learning Representations, 2017.
[11] L. Bai, L. Cui, Y. Jiao, L. Rossi, E.R. Hancock, Learning backtrackless aligned-spatial graph convolutional networks for graph classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2) (2022) 783–798.
[12] L. Cui, L. Bai, X. Bai, Y. Wang, E.R. Hancock, Learning aligned vertex convolutional networks for graph classification, IEEE Trans. Neural Netw. Learn. Syst. (2021), doi:10.1109/TNNLS.2021.3129649.
[13] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, T.-S. Chua, Temporal relational ranking for stock prediction, ACM Trans. Inf. Syst. 37 (2) (2019) 1–30.
[14] R. Kim, C.H. So, M. Jeong, S. Lee, J. Kim, J. Kang, Hats: a hierarchical graph attention network for stock movement prediction. arXiv preprint arXiv:1908.07999.
[15] A. Bretto, Hypergraph theory, An introduction. Math. Eng.. Cham: Springer (2013).
[16] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 3558–3565.
[17] S. Bai, F. Zhang, P.H.S. Torr, Hypergraph convolution and hypergraph attention, Pattern Recognit. 110 (6) (2021) 107637.
[18] R. Sawhney, S. Agarwal, A. Wadhwa, R.R. Shah, Spatiotemporal hypergraph convolution network for stock movement forecasting, in: Proceedings of the 20th IEEE International Conference on Data Mining, 2020, pp. 482–491.
[19] X. Yan, Z. Guosheng, Application of kalman filter in the prediction of stock price, in: Proceedings of the 5th International Symposium on Knowledge Acquisition and Modeling, 2015, pp. 197–198.
[20] Z. Li, D. Yang, L. Zhao, J. Bian, T. Qin, T.-Y. Liu, Individualized indicator for all: Stock-wise technical indicator optimization with stock embedding, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 894–902.

[21] D.M.Q. Nelson, A.C.M. Pereira, R.A. de Oliveira, Stock market's price movement prediction with LSTM neural networks, in: Proceedings of the 30th International Joint Conference on Neural Networks, 2017, pp. 1419–1426.
[22] R. Akita, A. Yoshihara, T. Matsubara, K. Uehara, Deep learning for stock prediction using numerical and textual information, in: Proceedings of the 15th International Conference on Computer and Information Science, 2016, pp. 1–6.
[23] Y. Chen, Z. Wei, X. Huang, 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, 2018, pp. 1655–1658.
[24] L. Bai, L. Cui, Z. Zhang, L. Xu, Y. Wang, E.R. Hancock, Entropic dynamic time warping kernels for co-evolving financial time series analysis, IEEE Trans. Neural Netw. Learn. Syst. 34 (4) (2023) 1808–1822.
[25] L. Bai, Y. Jiao, L. Cui, L. Rossi, Y. Wang, P.S. Yu, E.R. Hancock, Learning graph convolutional networks based on quantum vertex information propagation, IEEE Trans. Knowl. Data Eng. 35 (2) (2023) 1747–1760.
[26] S. Agarwal, K. Branson, S. Belongie, Higher order learning with graphs, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 17–24.
[27] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, P. Talukdar, HyperGCN: a new method for training graph convolutional networks on hypergraphs, Adv. Neural Inf. Process. Syst. (2019) 1509–1520.
[28] F. Feng, H. Chen, X. He, J. Ding, M. Sun, T.-S. Chua, Enhancing stock movement prediction with adversarial training, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2018, pp. 5843–5849.
[29] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, T-Gcn: a temporal graph convolutional network for traffic prediction, IEEE Trans. Intell. Transp. Syst. 21 (9) (2020) 3848–3858.
[30] G. Kling, L. Gao, Calendar effects in chinese stock market, Annal. Econ. Finance 6 (1) (2005) 75–88.
[31] Y. Hao, H.-H. Chu, K.-Y. Ho, K.-C. Ko, The 52-week high and momentum in the taiwan stock market: anchoring or recency biases? Int. Rev. Econ. Finance 43 (2016) 121–138.
[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: Proceedings of the 6th International Conference on Learning Representations, 2018.
[33] S.P. Borgatti, M.G. Everett, A graph-theoretic perspective on centrality, Soc. Netw. 28 (4) (2006) 466–484.
[34] D. Bowen, M.C. Hutchinson, N. O'Sullivan, High-frequency equity pairs trading: transaction costs, speed of execution, and patterns in returns, J. Trad. 5 (3) (2010) 31–38.
[35] T.J. Moskowitz, Y.H. Ooi, L.H. Pedersen, Time series momentum, J. Financ. Econ. 104 (2) (2012) 228–250.
[36] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[37] H. Yu, G.V. Nartea, C. Gan, L.J. Yao, Predictive ability and profitability of simple technical trading rules: recent evidence from southeast asian stock markets, Int. Rev. Econ. Finance 25 (2013) 356–371.
[38] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks? in: Proceedings of the 7th International Conference on Learning Representations, 2019.
[39] H. Maron, H. Ben-Hamu, H. Serviansky, Y. Lipman, Provably Powerful Graph Networks, Curran Associates Inc., 2019.
[40] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: a review of methods and applications, AI Open 1 (2020) 57–81.
[41] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008) 2579–2605.

Chaoran Cui received his Ph.D. degree in computer science from Shandong University in 2015. He is now a professor with School of Computer Science and Technology, Shandong University of Finance and Economics. His research interests include information retrieval, recommender systems, multimedia, and machine learning.

Xiaojie Li received her B.E. degree in digital media technology from Shandong University of Finance and Economics in 2019. She is currently pursuing the master degree in computer application technology at Shandong University of Finance and Economics. Her research interests include data mining, graph neural networks, and machine learning.

Chunyun Zhang received her Ph.D. degree from Beijing University of Posts and Telecommunication in 2015. She is now an associate professor in Machine Learning and Data Mining Center, Shandong University of Finance and Economics. Her current research interests include machine learning and natural language processing.

Weili Guan received her bachelor's degree from Huaqiao University, China, in 2009, and the Graduate Diploma and master's degrees from the National University of Singapore, Singapore, in 2011 and 2014, respectively. She is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Monash University Clayton Campus, Australia. Her research interests include multimedia computing and information retrieval.

Meng Wang received his Ph.D. degree in the Department of Electronic Engineering and Information Science from the University of Science and Technology of China in 2008. He is now a professor at the Hefei University of Technology. His current research interests include multimedia content analysis, computer vision, and pattern recognition.