Information Sciences 607 (2022) 783–798

Contents lists available at ScienceDirect

Information Sciences

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

HGNN: Hierarchical graph neural network for predicting the classification of price-limit-hitting stocks

Cong Xu [a], Huiling Huang [a], Xiaoting Ying [a], Jianliang Gao [a,*], Zhao Li [b], Peng Zhang [c], Jie Xiao [d], Jiarun Zhang [e], Jiangjian Luo [f]

[a] Central South University, Changsha 410083, China
[b] Alibaba Group, Hangzhou 310052, China
[c] Guangzhou University, Guangzhou 510006, China
[d] Hangzhou Yugu Technology Co., Ltd., Hangzhou 311113, China
[e] University of California San Diego, CA 92093, USA
[f] Zhejiang Chika Internet Technology Co., Ltd., Zhejiang, China

ARTICLE INFO

Article history:
Received 3 January 2022
Received in revised form 25 April 2022
Accepted 4 June 2022
Available online 11 June 2022

Keywords:
Hierarchical graph neural network
Stock prediction
Price limit

ABSTRACT

In some stock markets, stock prices are not allowed to rise above a daily limit to restrain the surge of price (called price limit). When the price limit occurs, investors tend to chase the continuing upward momentum for profit-making. However, For the stocks that hit daily price limit, we observe whether they close at daily price limit will lead to the opposite price trends of the next trading day. Therefore, this work aims to predict whether a stock that hits its daily price limit will also close at the same price level (*i.e.*, Type I or Type II). The occurrence of price limit is driven by different levels of market state. For example, it can result from macro-economic changes of the whole market, or it can be traced to some industry-specific factors. A challenging task is to learn a better stock representation with less uncertainty by comprehensively considering the hierarchical property of market state. Accordingly, we design a novel hierarchical architecture, called Hierarchical Graph Neural Network (HGNN), to investigate the market state at hierarchical view for stock type prediction. In HGNN, we construct the stock relation graph and merge stock information hierarchically extracted from multiple views of market state, including node view, relation view and graph view, which takes both historical sequence pattern and stock relation into consideration. Our key innovation is the introduction of hierarchical structure makes the predictive model able to more comprehensively infer the hierarchical property of market state. Further, it also provides the deeper insight for the actual investment practice. To validate the effectiveness of our method, we conduct back-testing on the two-year historical data of more than 2500 main-board stocks in two China stock markets, SSE and SZSE. To support further study of the stock type prediction task, we have published two long-range stock datasets (Datasets are available at https://drive.google.com/file/d/1TXiAyqt3rHveuzdGT6YtswU1e-tBSFUe/view?usp=sharing). Extensive experiments show that our method outperforms the state-of-the-art solutions including ALSTM, GCN and GAT with the improvements of at least 3.54% on average in accuracy. In addition, the average return ratio of SSE and SZSE has improved by 18.57% and 8.75%, respectively.

* Corresponding author.
E-mail address: gaojianliang@csu.edu.cn (J. Gao).

https://doi.org/10.1016/j.ins.2022.06.010
0020-0255/

1. Introduction

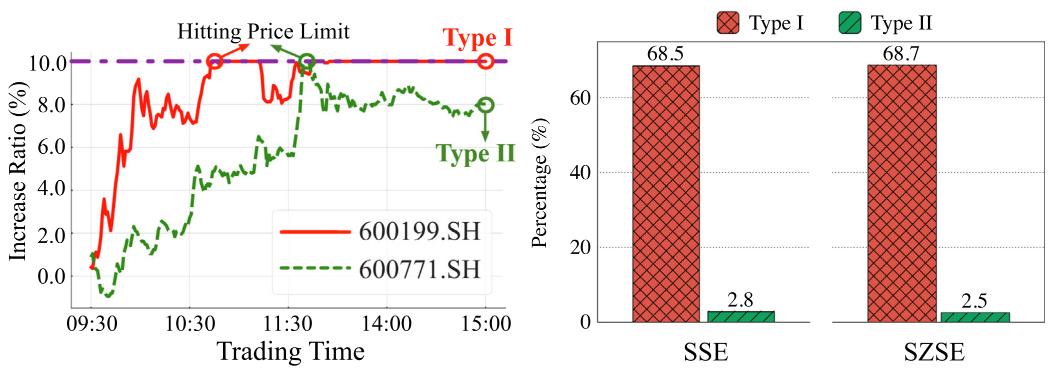
The stock market plays a vital role in the economic development of modern society, which is an indicator of financial health of an economy [1,2]. The application of data mining technology in stock market has become an attractive topic [3–5]. Previous attempts mainly focus on predicting the exact price [6,7] or price movement (up or down) [8,9], and they utilize price sequence and external knowledge [10] (e.g., financial news, social media, and knowledge graph) as features to predict. However, there is still considerable disagreement on these existing methods since they are inevitably limited in their ability to achieve better prediction performance due to the highly stochastic property of stock market [11,12]. Therefore, some studies have suggested that it is worthwhile to explore the market rules in stock market rather than directly predict stock price [13,14].

The rule of price limit is widely adopted to restrain the surge and crash of stock price [14]. **Price limit** refers to that stock price is not allowed to exceed a daily limit in a single trading session [15]. For example, the Chinese stock markets impose a daily price limit of 10% on regular stocks based on the closing price of previous trading day. If the price of a stock hits its daily price limit, it is called **price-limit-hitting** stock. As shown in Fig. 1(a), the two stocks are both price-limit-hitting stocks and they can be divided into two types according to their closing prices. One type (called **Type I** in this paper) is the price-limit-hitting stocks whose closing prices stay at the daily price limits finally, such as 600199.SH. Correspondingly, the price-limit-hitting stocks belong to the other type (called **Type II**) if their closing prices are lower than the daily price limits, such as 600771.SH.

It is worth predicting whether a price-limit-hitting stock will close with daily price limit (Type I) or not (Type II) on a trading day. According to the delayed price discovery hypothesis [16], stock price will keep rising in the subsequent period (usually the next trading day) after hitting the daily price limit theoretically. We have verified this hypothesis on Chinese stock market data from 01/01/2018 to 12/31/2019. We compare the daily price limits of current trading day to the opening prices of the next trading day. The results are shown in Fig. 1(b). We can see that in two markets, the majority of Type I stocks will rise at the opening of the next day, while only less than 3% of Type II stocks will rise at the opening of the next day. Therefore, it is particularly important to predict the types of price-limit-hitting stocks, which will lead to extremely opposite upcoming trend.

Previously, most studies are focused on how to obtain and utilize more data sources related stock market. Some researchers apply time series technology to stock historical sequence for stock prediction [4,17], others try to dig into textual information such as financial news, social media, etc [18,19]. Besides historical sequence and textual information, the rich relations between stocks have been employed for stock prediction recently [20,21]. Nevertheless, these methods often pursue the richness of data, but inevitably ignore the hierarchical information contained in the stock data itself. In addition, stock prediction generally refers to predicting future stock trend (up or down) or stock price [22,23]. However, these studies are not directly optimized towards the target of investment, i.e., gaining more profits [7,24].

In this work, our primary consideration is that how investors can be benefit from the price operation exist in the market rules. To implement this idea, our proposed objective is optimized towards the target of investment, i.e., gaining profits from the rule of price limit. And the occurrence of price limit is driven by the different levels of market state. Different levels can be understood as different factors that cause the movement of the stock price, such as macro-economic, industry-specific and stock-specific factors. [17]. The combined role of multiple levels of market state leads to the uncertainty of price limit. Therefore, the key to this work is to learn a better stock representation with less uncertainty by comprehensively considering the hierarchical property of market state. Accordingly, we propose to enhance the stock type prediction by tracing the devel-



(a) Price-Limit-Hitting Stocks (b) Percentage of stocks with rising price

Fig. 1. An example of motivation for price-limit-hitting stock classification. (a) shows two price-limit-hitting stocks and their final types on Mar. 18, 2019, where the increase ratio is obtained by comparing to their respective closing prices of previous trading day. (b) shows the percentage of stocks with rising price for Type I and Type II respectively. Rising price is obtained by comparing the opening price of the next trading day to the daily price limit of previous trading day. SSE and SZSE mean Shanghai Stock Exchange and Shenzhen Stock Exchange markets.

opment of the market state at hierarchical view (e.g., from the perspective of the stock itself, the industry-specific factors and the macro-economic changes of the whole market in this paper). We creatively introduce the hierarchical architecture to stock prediction model, which provides deeper insight into the real investment practice. And we design a new architecture, Hierarchical Graph Neural Network (HGNN), which can predict the stock type by integrating multiple levels of market state at hierarchical view. One view called *node view* is to strengthen the importance of target stock itself based on historical sequence. The second one called *relation view* is used to introduce the trend information of related stocks to infer target stock. Especially, the occurrence of price limit is related to the current state of stock market [15]. The third one called *graph view* can reflect the impact of the current market-wide trend on target stock. To prove the practicality of proposed method, we conduct extensive experiments on two real-world stock market datasets. Experimental results demonstrate that our proposal outperforms the baseline models. Further analysis proves the effectiveness of hierarchical architecture designed in the proposed framework. Three main contributions are proposed as follows:

- We propose a novel objective that aims to predict the type of price-limit-hitting stock. A major obstacle is the lack of data for stock prediction. We publish the datasets for price-limit-hitting stock prediction from two Chinese stock exchanges.
- We creatively propose to model multiple levels of market state at hierarchical view, which provides deeper insight into the real investment practice. Then design a Hierarchical Graph Neural Network to explicitly integrate the hierarchical information and infer the type of target stock.
- Extensive experiments demonstrate that proposed HGNN significantly outperforms the baselines. Further analysis suggests that the introduction of hierarchical architecture makes our model able to obtain a better representation of target stock.

By way of introduction, the rest of this paper is organized as follows. Section 2 discusses the related works and the differences between proposed model and other related methods. Section 3 provides several empirical evidence for the link between price limit and hierarchical property of stock market from practical market analysis. Section 4 defines the price-limit-hitting stock classification task and symbolic norms. Section 5 introduces proposed HGNN in detail. Section 6 describes the datasets and experimental setup. Section 7 presents the effectiveness analysis of proposed HGNN from different aspects, followed by conclusion in Section 8.

2. Related works

In recent years, stock prediction has been an attractive topic in the financial field. Many researches are devoted to predicting stock movement or price value based on historical sequence or stock relation.

2.1. Stock prediction with historical sequence

Most historical sequence based methods learn historical patterns to predict stock movement [4,22,23,17]. The traditional solutions for stock prediction are based on time-series analysis models, such as technical analysis [25], Kalman Filters [26], and Autoregressive models [27]. Moreover, machine learning has become the generally accepted solution with the capability in a highly volatile market. Zhang et al. propose a State Frequency Memory (SFM) recurrent network to make long and short-term predictions over time by capturing the multi-frequency trading patterns [6]. Qin et al. propose an Attentive-LSTM model with an attention mechanism to predict stock price movement [28]. Wang et al. propose a Convolutional LSTM based Variational Sequence-to-Sequence model with Attention (CLVSA) to extract the underlying features of the trend from raw financial sequence [29]. Liu et al. design a Multi-scale Two-way Deep Neural Network (MTDNN) to capture the multi-scale information on stock trading [17]. Ding et al. propose a Gaussian Transformer based approach to tackle the stock prediction [4].

2.2. Stock prediction with stock relation

Analyzing relation graph has become a focal point of study interests in both academic and industrial domains [21,30]. The relations between stocks are also important for stock prediction indicated by recent researches [20,7,18]. Recent works have begun to model the stock relations by constructing the relation graph, where each stock is represented as a node and each edge is built in terms of some predefined stock relations. For instance, Chen et al. utilize Graph Convolutional Network (GCN) [31] to aggregate related corporation information of the target company by constructing the graph of investment fact [32]. Feng et al. consider two types of relation to construct the graphs, including the stocks in the same sector or industry, and the upstream and downstream stocks in the supply chain. Then, they adopt the idea of the PageRank to embed the relation between stocks[7]. Furthermore, Li et al. propose to adopt the correlation matrix among stocks, which is calculated based on the correlation of historical market data. Then, they propose a Long Short Term Memory Relational Graph Convolution Networks model (LSTM-RGCN) to model the connection among stocks with their correlation matrix [18].

2.3. Stock prediction with different objectives

In general, stock prediction refers to predicting stock price movement (up or down) or price value in the near future. Stock prediction can be designed for different optimization objectives according to specific tasks. Recently, some researchers have focused on stock ranking prediction and overnight stock movement prediction. Feng et al. [7] believe that stock movement or price prediction is not suitable for guiding stock selection. They formulate stock prediction as a ranking task, for which the target is to directly predict a stock list ranked by stock return. Furthermore, Li et al. [18] argue that it is not very reliable to use financial news to predict the next day's price trend, because the time for the market to absorb the influence of news varies from a few minutes to a few hours, but usually less than a day. Subsequently, they only use the financial news published after the stock market is closed to predict the overnight stock movement.

Difference to Related Methods.

Compared with the related methods described above, our work mainly has the following significant differences: Instead of directly predicting price movement direction or price value, we propose a novel objective that aims to predict the type of price-limit-hitting stock based on the continuing influence of price limit rule. Considering enhancing generalization ability by absorbing the wide-ranging information of stock market, we propose Hierarchical Graph Neural Network (HGNN) to integrate the stock information extracted from multiple views, including *node view*, *relation view* and *graph view*. Extensive experiments demonstrate the superiority of proposed framework.

3. Hierarchical property of stock market

In the real market, the occurrence of price limit is usually driven by the different level of market state (e.g., from the perspective of the stock itself, the industry-specific factors and the macro-economic changes of the whole market in this paper). In this section, we will provide several empirical evidences for the link between price limit and hierarchical property of stock market from practical market analysis.

From the perspective of stock itself. According to the efficient market hypothesis (EMH) [33], stock prediction is impossible given that the market is efficient enough. However, the stock market is, in reality, not always as effective as expected and can be possible to predict by analyzing previous prices sequence. [6,7,4]. One well observed example is mean regression [34], which means that the stock price will fall when it is higher than the historical average. Consequently, proposed *node view* focuses on historical sequence to strengthen the importance of target stock itself for type prediction.

From the perspective of stock industry. It is natural to believe the price of target stock would be affected by the related stocks [32]. There is an attractive phenomenon that price-limit-hitting stocks will cluster to appear in a certain relation. We investigate the behavior of price-limit-hitting stocks in industry relation, and calculate the number of price-limit-hitting stocks belong to per industry on each trading day. The statistics are shown in Fig. 2, and it can be observed that the majority of price-limit-hitting stocks appear in the same industry on a trading day, which implies there is a strong linkage between the spread of stock industry relations and the price limit. Motivated by this, we leverage proposed *relation view* to model the effect of industry relation among stocks for enhancing stock type prediction.

From the perspective of stock market. In general, the positive sentiment on stock market leads more stocks to hit their daily price limits [15]. Fig. 3 shows the change in the number of price-limit-hitting stocks as the trend of the Shanghai Composite Index changes. It is well observed that the good momentum of stock market indicates there are more price-limit-hitting stocks before Sep. 9, 2019, while a falling market is accompanied by a small number of price-limit-hitting stocks after Sep. 9, 2019. The intuition is that the introduction of the whole market sentiment contributes to the analysis of price limit. We propose *graph view* to model the whole market sentiment by adaptively selecting and aggregating the information from all stocks.

4. Problem definition

In this section, we begin with notations definitions. For day t, a stock set containing S_t stocks is defined as $\mathscr{S} = \{s^1, s^2, \cdots, s^{M_t}\} \cup \{s^{M_t+1}, s^{M_t+2}, \cdots, s^{S_t}\}$. Especially, M_t price-limit-hitting stocks are $\mathscr{M} = \{s^1, s^2, \cdots, s^{M_t}\} \in \mathscr{S}$, and others are non-price-limit-hitting stocks. We get a set of stock historical sequence data for past T time-steps before day $t, \mathscr{X}_t = \{X_t^1, X_t^2, \cdots, X_t^{S_t}\} \in \mathbb{R}^{S_t \times T \times K}$, where K is the dimension of features (e.g., opening price, low price, high price, closing price). For M_t price-limit-hitting stocks, we calculate the technical indicators related price limit [35] (called limit-related indicators, including Moving Average, K Length, Rate of Change, Turnover Ratio, Amplitude, Bias Ratio) $D_t = \{d_t^1, d_t^2, \cdots, d_t^{M_t}\}$ to capture the suddenness of the price limit, through the minutely historical trade data before the price limit occurs. Table 1 demonstrates a set of detailed indicators with their calculation formulas, where on day $t, Limit_t, Open_t$, and $Close_t$ denote the price limit, opening price, and closing price, respectively; $Low_t, Amount_t, Volume_t$, and $Shares_t$ denote the lowest price, trading amount, trading volume, and outstanding shares before the occurrence of hitting daily price limit, respectively. In particular, each stock has its inherent industry categories $Industry_k$. For example, 600199.SH (Golden Seed Winery Co.,Ltd.) belongs to the liquor-making industry. Subsequently, we define the limited-related indicator and stock relation graph, and finally formulate the problem of price-limit-hitting stock classification.

Fig. 2. The connection between price limit and stock industry, which shows the number of price-limit-hitting stocks belong to per industry on each trading day from Sep. 2, 2019 to Sep. 6, 2019. For example, C39 refers to the industry code of electronic equipment.

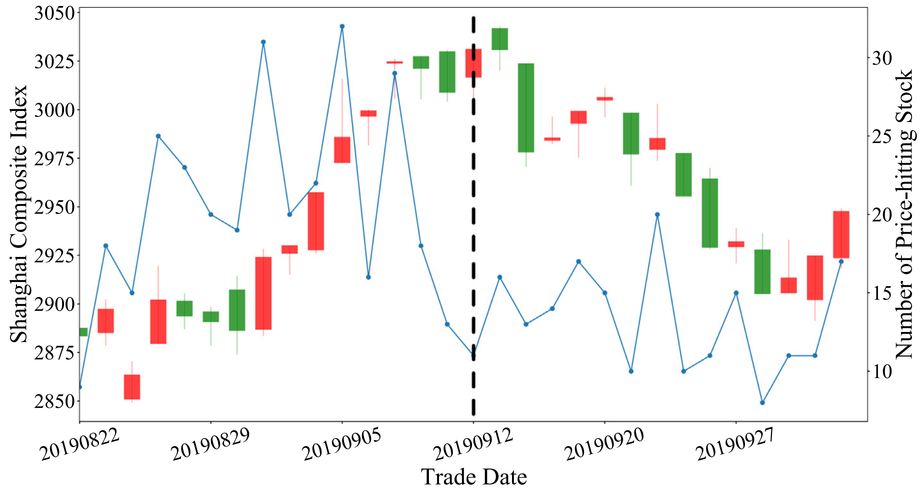


Fig. 3. The connection between price limit and the trend of the whole market, which shows the K-line of the Shanghai Composite Index and the number of price-limit-hitting stocks on each trading day from Aug. 22, 2019 to Oct. 10, 2019.

Table 1
Limit-related indicators of price-limit-hitting stocks.

Limit-related Indicator	Calculation Formula
Moving Average	$MA_{short} = Amount_t / Volume_t$
KLength	$KLength = Limit_t - Open_t$
Rate of Change	$ROC = Limit_t / Close_{t-1} - 1$
Turnover Ratio	$TOR = Volume_t / Shares_t$
Amplitude	$AMP = \frac{Limit_t - Low_t}{Close_{t-1}}$
Bias Ratio	$Bias = \frac{Limit_t - MA_{short}}{MA_{short}}$

Definition 1. Stock Relation Graph. An undirected graph $G_t = \{V_t, E_t, \mathcal{T}\}$, where V_t is a set of S_t nodes representing stocks on day t; E_t is a set of edges, indicating the relations between stock nodes; $\mathcal{T}$ is the type of stock node, including price-limit-hitting stock and non-price-limit-hitting stock. We construct edge between stock i and j by converting their industry categories into multi-hot binary vectors C_i and C_j. If $C_i \cdot C_j \geqslant 1$, there is an edge between stock i and j.

Problem 1. Price-Limit-Hitting Stock Classification. The price-limit-hitting stock classification is a binary classification task. Given the stock relation graph G_t, stock historical sequences $\mathcal{X}_t$ and limit-related indicators D_t, the task is to learn a function f which is able to get the type label of price-limit-hitting stock $\hat{\mathbf{y}}_\mathbf{t} = \{\hat{y}_t^1, \hat{y}_t^2, \cdots, \hat{y}_t^{M_t}\} \in \mathbb{R}^{M_t}$, where $\hat{y}_t^{m_t}$ is the predicted label of Type I (1) or Type II (0). The mapping relation is represented as follows:

$$[\mathscr{X}_t, D_t, G_t] \xrightarrow{f} \hat{\mathbf{y}}_{\mathbf{t}}. \tag{1}$$

5. Methodology

In this section, we introduce the detailed design of the Hierarchical Graph Neural Network (HGNN), and Fig. 4 illustrates the overall framework, which includes *Feature Extraction Layer*, *Hierarchical GNN Layer* and *Hierarchical Fusion Classification Layer*.

5.1. Feature extraction layer

This layer aims to extract the historical feature and limit feature from historical sequence and technical indicator related to price limit, respectively.

Historical Feature Extraction. The Long Short Term Memory (LSTM) is a popular approach to boosting the ability of recurrent neural networks [36]. The LSTM networks have been shown effective in capturing and characterizing the long-term dependencies in sequence data [29]. We utilize the LSTM networks to extract the historical feature over time. For the stock s_t with historical sequence $X_t^{s_t} = \{x_1^{s_t}, x_2^{s_t}, \cdots, x_T^{s_t}\}$, we feed it into the LSTM:

$$\begin{aligned} & i_t^{s_t}, f_t^{s_t}, o_t^{s_t} = f_{\theta^i}, f_{\theta^f}, f_{\theta^o}(h_{t-1}^{s_t}; x_t^{s_t}) \\ & u = tanh(W_u h_{t-1}^{s_t} + Q_u x_t^{s_t} + b_u) \\ & c_t^{s_t} = f_t^{s_t} \odot c_{t-1}^{s_t} + i_t^{s_t} \odot u \\ & h_t^{s_t} = o_t^{s_t} \odot tanh(c_t^{s_t}), \end{aligned} \tag{2}$$

where $x_t^{s_t} \in \mathbb{R}^K$ is the original historical sequence; f_θ is a feed forward network with sigmoid activation function and parameters θ; i, f, o indicate input, forget and output gates, respectively; $W_u \in \mathbb{R}^{U \times K}, Q_u \in \mathbb{R}^{U \times U}$ are learnable parameters; U is the number of hidden units; and $b_u \in \mathbb{R}^U$ is a bias vector. After recurrently projecting the input historical sequence, the extractor outputs the last hidden feature $h_t^{s_t} \in \mathbb{R}^U$ for stock s_t as the historical feature.

Limit Feature Extraction. Many stock market investors generally accept and use certain standard technical indicators as signals of future market trends [37,38]. Limit-related indicators for the price-limit-hitting stocks are combined value vectors by calculating the minute-level trade data before the occurrence of the price limit. We apply a Multiple-Layer Perceptron (MLP) to extract limit feature from limit-related indicators [39]. Given the limit-related indicators $d_t^{m_t}$ of stock m_t, it is formulated as follows:

$$l_t^{m_t} = MLP(d_t^{m_t}), \tag{3}$$

where $MLP(\cdot)$ is a multiple-layer perceptron consisting of hidden layers with *LeakyReLU* as the activation function. The limit feature $l_t^{m_t} \in \mathbb{R}^U$ calculated from high frequency data can be used to better capture the suddenness of price limit.

5.2. Hierarchical GNN layer

This layer aims to learn the hierarchical information of stock data including the **node-view** (NV), **relation-view** (RV) and **graph-view** (GV) representations by interacting the historical feature and limit feature.

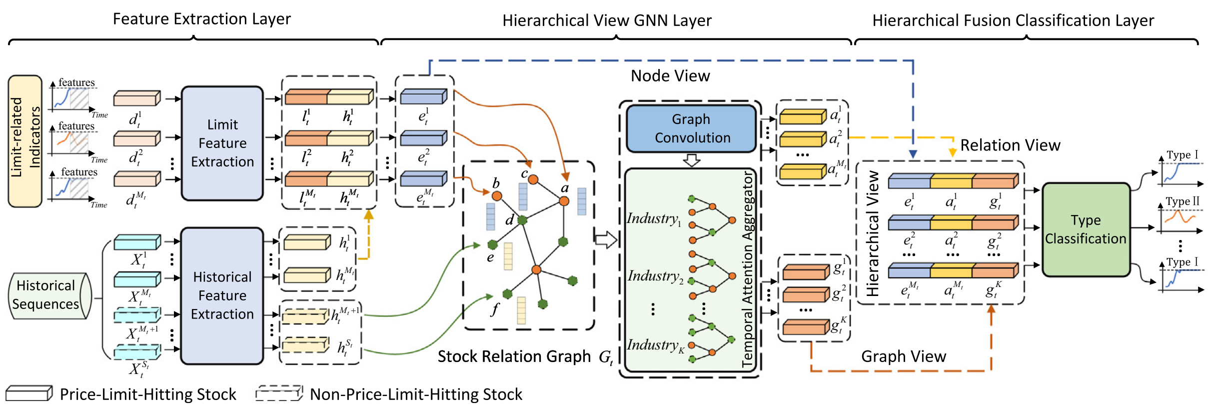


Fig. 4. The overall framework of Hierarchical Graph Neural Network (HGNN) for the price-limit-hitting stock type classification.

Graph Convolution. We first initialize the hidden representation h_t of all S_t stock nodes by the LSTM networks. The stock relation graph has two types of stock nodes, including price-limit-hitting node m_t and non-price-limit-hitting node z_t. For normal stock node z_t, we utilize $h_t^{z_t}$ as node representation $e_t^{z_t} = h_t^{z_t}$. For price-limit-hitting stock node m_t, its representation is not only related to the historical feature, but also related to the limit feature. Therefore, we interact the historical feature and limit feature to obtain the **node-view** representation for stock m_t:

$$e_t^{m_t} = \psi(W_f^T[h_t^{m_t} \oplus l_t^{m_t}] + b_f), \tag{4}$$

where ψ is an activation function; $\oplus$ represents a concatenate operation; $W_f \in \mathbb{R}^{2U \times U}$ is the parameter matrix of a fully connected layer; and $b_f \in \mathbb{R}^U$ is the learnable bias. By initializing the features of all S_t stock nodes, we obtain node-view representations $E_t^{S_t} = \{e_t^1, e_t^2, \cdots, e_t^{S_t}\} \in \mathbb{R}^{S_t \times U}$ for all S_t stocks.

In order to simulate the propagation of relations between stocks and learn the hidden state for each stock more comprehensively, we design a graph convolution operation to adaptively absorb the relational effect from different stocks. For stock node s_t, we will aggregate the relational information of its all neighbor nodes with a learnable way to get the **relation-view** representation $a_t^{s_t} \in \mathbb{R}^U$ on each day t. The graph convolution operation is formulated as follows:

$$a_t^{s_t} = \sum_{j_t \in \mathcal{N}_{s_t} \cup \{s_t\}} \frac{1}{\sqrt{degree(j_t) \cdot degree(s_t)}} (\pi \cdot e_t^{j_t}), \tag{5}$$

where $degree(s_t)$ denotes the degree of stock node $s_t; j_t \in \mathcal{N}_{s_t} \cup \{s_t\}$ and $\mathcal{N}_{s_t}$ is the neighborhood of stock node s_t in the stock relation graph. The neighboring node features are first transformed by a weight matrix π. In particular, one of all M_t price-limit-hitting stocks m_t's relation-view feature is $a_t^{m_t} \in \{a_t^1, a_t^2, \cdots, a_t^{M_t}\}$.

Temporal Attention Aggregator. According to the empirical experiment in Section 3, industry-level and market-level (collectively termed **graph view** in paper) information can provide more clues for the occurrence of price limit. Given a stock with their relation embeddings $a_t^{s_t} \in \mathbb{R}^U$ (the output of Graph Convolution layer) and the industry it belongs to $industry_k$, proposed Temporal Attention Aggregator aims to learn the hidden embeddings that encode the graph-view information. Instead of directly presenting the formulation of Temporal Attention Aggregator, we detail how we design the component to shed some lights on its rationale.

a) Uniform Embedding Aggregator. Our first inspiration comes from the graph pooling research. Learning the high-level representation of graphs is very important for graph-based analysis tasks. Moreover, the pooling operation has shown its effectiveness in image and natural language processing tasks. So it is natural that the pooling operation is introduced into the graph data processing. A general example is the *AvgPooling* method that aggregates the importance score of a single vertex to graph-level representation. Since stock graph-level representation stands for certain information covered a set of related stocks, we consider relating their embeddings through the similar aggregation process as in graph pooling analysis:

$$g_t = \sum_{s_t \in \mathcal{S}_t} \tfrac{1}{d_k} a_t^{s_t}, \tag{6}$$

where $\mathcal{S}_t$ represents all stocks normally traded in day t, d_k is the number of stocks meeting the condition $s_t \in \mathcal{S}_t$ (*i.e.*, the number of stocks normally traded in day t). After such the aggregation in the stock relation graph, the graph-view representation g_t encodes the overall impact coming from all stocks that in day t. It describes the state of the whole market by considering the movements of all stocks.

b) Weighted Embedding Aggregator. Considering that different stocks have different impacts or importance on modeling the state of the whole market, a static transform is applied when aggregating the relation embeddings:

$$g_t = \sum_{s_t \in \mathcal{S}_t} \frac{weight_{s_t}}{d_k} a_t^{s_t}, \tag{7}$$

where $weight_{s_t}$ stands for the impact strength of stock s on modeling the state of the whole market, and we term it as the importance-strength module. For example, suppose we have two stocks with different market values, the intuition is that the movement of stocks with high market value has a more significant on the market state. $weight_{s_t}$ is similar to the stock index in reality, and its composition has been static for a long period of time. We can give static importance strength for different stock in the stock relation graph, which allows the embedding aggregation process to account for the importance of each stock node.

c) Time-aware Attention Aggregator. A limitation of the above weighted aggregation process is that the importance-strength module outputs a fixed weight regardless of the market evolution. However, stock market is highly stochastic and the statuses of all stocks are continuously evolving. Assuming the stock to have a static weight limits the ability to model market state. To address this limitation, we propose to design a learnable importance-strength module to adaptively learn the impact of each stock on the market state and define the Time-aware Attention aggregation process as follows:

$$g_t = \sum_{s_t \in \mathcal{S}_t} \frac{w(s_t)}{d_k} a_t^{s_t}, \tag{8}$$

where $w(s_t)$ is modified into a learnable importance-strength module that aims to learn the varying importance strength of single stock to generate graph view. It allows the importance-strength module to estimate the impact of single stock based on market evolution, which is very desirable.

Next, we describe two designs of the time-sensitive importance-strength module, which differ in whether to model the impact of single stock from the perspective of the whole market or the industry (See Section 3 for more evidence).

• **Market-Oriented Temporal Attention Aggregator.** In the real stock market, the movement of a single stock is related to the movement trend of the whole market. The most straightforward approach is introducing the stock index to guide the prediction, yet the stock index is easy to be distorted under the differentiated market, that is, the stock index does not keep pace with the market. Therefore, we propose to design the temporal market-oriented graph-view representation to model the time-aware varying market state by reshaping all stocks' relation-view features in stock relation graph. This design can comprehensively infer the state of the whole market. In order to introduce temporal property and consider the importance of different stocks, we apply the **attention aggregator** to learn the temporal market-oriented representation from all the stocks with attention scores. Formally, the importance-strength module of the Market-Oriented Temporal Attention Aggregator is computed by:

$$\begin{aligned} w(s_t) &= \frac{\exp(\eta_t^{s_t})}{\sum_{j_t} \exp(\eta_t^{j_t})}, \\ \eta_t^{s_t} &= u_a^T \phi(W_a a_t^{s_t} + b_a), \end{aligned} \tag{9}$$

where $\eta_t^{s_t}$ indicates the importance of stock node s_t in the stock relation graph; ϕ is an activation function; $u_a \in \mathbb{R}^{M\prime}, W_a \in \mathbb{R}^{M\prime \times U}$, and b_a are parameters to be learned.

• **Industry-Oriented Temporal Attention Aggregator.** The Market-Oriented Temporal Attention Aggregator has introduced market evolution and temporal property into the importance-strength module. However, we consider that it is not fine-grained enough to directly model the linkage between the whole market and certain target stock. In the Industry-Oriented Temporal Attention Aggregator, we propose a more fine-grained importance-strength module based on the industry facts of stocks. The intuition is that the movement of single stock is more directly related the trend of each industry. Industry-Oriented Temporal Attention Aggregator can better indicate what market state the target stock belongs to. Formally, we define the corresponding aggregation process as:

$$\begin{aligned} g_t^k &= \sum_{\{s_t | s_t \in Insudtry_k\}} \frac{w(s_t)}{d_k} a_t^{j_t}, \\ w(s_t) &= \frac{\exp(\eta_t^{s_t})}{\sum_{j_t} \exp(\eta_t^{j_t})}, \\ \eta_t^{s_t} &= u_a^T \phi(W_a a_t^{s_t} + b_a), \end{aligned} \tag{10}$$

where $Insudtry_k$ stands for a certain stock industry k, d_k is the number of all stocks belonging to $Insudtry_k, j_t$ belongs to the neighborhood of stock node s_t.

The Uniform Embedding Aggregator and Weighted Embedding Aggregator cannot capture the temporal market evolution properties as designed in our Time-aware Attention Aggregator. In addition, to further refine the market state, we progressively design the market-oriented and industry-oriented state modeling modules.

5.3. Hierarchical fusion classification layer

This layer aims to integrate the information from different views including node view, relation view and graph view into stock representation and predict the type of price-limit-hitting stock.

Hierarchical Fusion. The node-view feature $e_t^{m_t}$ is conducive to strengthening the importance of price-limit-hitting stock itself. In addition, the relation-view feature a^{m_t} and graph-view feature g_t respectively reflect the expression of relational information under different graph structures, where relation-view feature contains the relational information among the neighborhood, and graph-view feature indicates the trend of the whole market. Considering to incorporate the information of different views, we concatenate $e_t^{m_t}$, a^{m_t}, and g_t as the final representation $H_t^{m_t} \in \mathbb{R}^{3U}$ of stock m_t:

$$H_t^{m_t} = [e_t^{m_t} \oplus a_t^{m_t} \oplus g_t]^T, \tag{11}$$

Type Classification.

We deploy a fully connected layer to predict the type label $\hat{y}_t^{m_t}$ of price-limit-hitting stock m_t:

$$\hat{y}_t^{m_t} = softmax(W_p^T H_t^{m_t} + b_p), \tag{12}$$

where $W_p \in \mathbb{R}^{3U}$ and b_p are the learnable parameters.

Loss Function. The task is formulated as a binary classification problem. In general, one property of cross entropy loss is that even examples that are easily classified lead to a loss with non-trivial magnitude [40]. Therefore, we utilize the focal loss

[40,41] as the loss function, which acts as a more effective alternative to cross entropy loss for dealing with hard classified examples.

$$loss = \begin{cases} -\alpha(1-\hat{y}^{m_t})^{\gamma} log\hat{y}^{m_t}, & y = 1 \\ -(1-\alpha)\hat{y}^{m_t\,\gamma} log(1-\hat{y}^{m_t}), & y = 0 \end{cases}, \tag{13}$$

where α is the balanced parameter to balance the importance of positive and negative samples; $\gamma \geqslant 0$ is tunable focusing parameter, and it aims to down-weight easy examples and thus focus training on hard examples.

Finally, we introduce the overall training process of Hierarchical Graph Neural Network (HGNN) in Algorithm 1, which explains the input, the acquisition of hierarchical information, model optimization and the output, etc. In the pseudo-code, step 2 shows the process of constructing the stock relation graph. Step 6 to 10 are the key points of the algorithm, which explain how to obtain hierarchical information from multiple market states. Step 12 and 13 explain the optimization process of the proposed model.

Algorithm 1 Hierarchical Graph Neural Network (HGNN)

Input: Daily-level historical data $\mathscr{X}$. Limit-related indicators $\mathscr{D}$. Industry categories $k \in \{1, \ldots, K\}$.
Output: Model parameters Θ,
1: Randomly initialize the model parameters Θ, the parameters of Temporal Attention Aggregator Φ and learning rate λ.
2: Construct the undirected graph $G = \{V, E, \mathscr{T}\}$according to industry categories K, $V \in \{1, \ldots, S\}$, $\mathscr{T}$ is the type of stock node.
3: **while** not reach stopping criteria **do**
4: **for all** $\mathscr{D}^{batch} \leftarrow \mathscr{D}^{train}$ **do**
5: **for all** s ← S **do**
6: Encode high-level historical embedding h_s by Eq. 2
7: Obtain Limit-related embedding l_s by Eq. 3
8: Generate the *node-view* representation e by Eq. 4
9: Generate the relation-view representation α by graph convolution operation in Eq. 5
10: Generate the graph-view representation g by Temporal Attention Aggregator in Eq. 10
11: **end for**
12: Calculate the loss $\mathscr{L}_{HGNN}$ for batch $\mathscr{D}^{batch}$ by Eq. 13;
13: Do backpropagation and update parameters Θ:
$\Theta^{(new)} = \Theta^{(old)} - \lambda \frac{\partial \mathscr{L}_{HGNN}}{\partial \Theta^{(old)}}$
14: **end for**
15: **end while**
16: **return** Θ

---

6. Experimental setup

In this section, we describe the statistics of datasets, the baselines in our experiment, and the corresponding experimental setup.

6.1. Datasets

We collect the stock historical sequence data, including the daily historical trade data and the minutely historical trade data before the price limit occurs, then construct the stock relation graph in our experiments. The details of these datasets are described in Table 2.

Table 2
Statistics of the datasets.

Datasets		SSE	SZSE
Stock Historical Sequence	#Train samples (01/01/2018 to 08/07/2019)	8064	10044
	#Test samples (08/08/2019 to 31/12/2019)	2016	2512
Stock Relation Graph	#Nodes	1392	1312
	#Edges	15559	16880

Stock Historical Sequence. For stock historical sequence, we first collect the daily historical trade data in the Shanghai Stock Exchange (SSE) and the Shenzhen Stock Exchange (SZSE) from 01/01/2018 to 12/31/2019, which have 10080 and 12556 valid price-limit-hitting stock samples, respectively. We divide the historical sequence into two time periods in chronological order for development set from 01/01/2018 to 08/07/2019 and test set from 08/08/2019 to 12/31/2019. Moreover, we extract the limit-related indicators from minutely historical trade data to capture and represent the suddenness of the price limit.

Stock Relation Graph. We get the 110 kinds of industry categories in the SSE and SZSE markets from China Securities Regulatory Commission[1], which are used to construct the stock relation graph with 1392 and 1312 stock nodes, respectively.

6.2. Baselines

In this paper, predicting the type of price-limit-hitting stock is first proposed, we convert some widely known models as baselines. The details of implementation are described in the following:

- **Naive Bayes [18]:** This model takes limit-related indicators as input and applies Naive Bayes classifier to predict the type of price-limit-hitting stock. The limit-related indicators are normalized by zero-mean normalization.
- **Logistic Regression (LR) [32]:** This model also takes the normalized limit-related indicators as the input, but applies Logistic Regression classifier to predict the type.
- **Support Vector Machines (SVM) [42]:** SVM is an effective nonlinear classifier, which takes normalized limit-related indicators for prediction.
- **eXtreme Gradient Boosting (XGBoost) [43]:** XGBoost is a scalable machine learning system for tree boosting, which is fed normalized limit-related indicators.
- **Long Short Term Memory (LSTM) [44]:** LSTM is one of the most powerful deep learning models for time series forecasting, which only utilizes the historical sequence as input.
- **Attentive LSTM (ALSTM) [11]:** Feng et al. proposed the Attentive LSTM (ALSTM) model for stock predictions, which utilizes a temporal attentive layer to aggregate sequence information.
- **Graph Convolutional Network (GCN) [31]:** GCN is the state-of-the-art graph-based learning method to incorporate relation. We apply this method by adding a common GCN layer after the LSTM network. The stock historical sequence and relation graph are fed into the GCN layer.
- **Graph Attention Network (GAT) [45]:** GAT is the attention-based graph neural network, which leverages self-attentional layers to address the shortcomings of prior methods based on graph convolutions. We add a GAT layer after the LSTM network and feed the historical sequence and relation graph into the GAT layer.
- **HGNN_M:** Our proposed HGNN with Market-Oriented Temporal Attention Aggregator.
- **HGNN_I:** Our proposed HGNN with Industry-Oriented Temporal Attention Aggregator.

6.3. Experimental setup

Market Simulation.

Inspired by previous market strategy [7], we design a more practical market simulation strategy to evaluate the performance through conducting back-testing on the test datasets of two markets. We assume that stock market is liquid enough to satisfy investors' trading requests, and that the investor buys the price-limit-hitting stock m at daily price limit $Limit_t^m$ of the day t and sells it at the opening price $Open_{t+1}^m$ of the next day $t+1$. Additionally, we do not take into account the transaction costs since the overnight return is higher than transaction costs [46]. For buying a set of M price-limit-hitting stocks, we formulate the *Average Return Ratio* (ARR) as follows:

$$ARR = \sum_{m=1}^{M} \frac{Open_{t+1}^m - Limit_t^m}{Limit_t^m} / M, \tag{14}$$

Evaluation Metrics. To evaluate the performance of the type classification of price-limit-hitting stock, we select two metrics as evaluation metrics: Accuracy (Acc) and F1-score (F1) [17]. It should be noted that the higher the ACC and F1 values, the better the performance in our experiment. The calculation formula is as follows:

$$Acc = \frac{TP+TN}{TP+TN+FN+FP}, \tag{15}$$

$$F1 = \frac{2TP}{2TP+FN+FP}, \tag{16}$$

where TP, FP, TN and FN stand for true positive, false positive, true negative and false negative, respectively. The investors are most concerned about the return ratio on investment. Therefore, *Average Return Ratio* (ARR) also needs to be considered in the experiment.

[1] http://www.csrc.gov.cn/csrc_en/index.shtml

Parameter Settings. Our proposed model is implemented with Pytorch and optimized by Adam with learning rate of 0.001. In MLP module, we set the number of hidden layers as 2. For the window size of historical sequential input and the number of hidden units in LSTM, we select them via grid-search within the ranges of [4, 6, 8, 12, 16] and [8, 16, 32, 64]. And we also tune α and γ in loss function within the ranges of [0, 0.2, 0.4, 0.6, 0.8] and [0.1, 1, 2, 3]. We report the mean testing performance over ten different runs, including the average values and standard deviations of the evaluation metrics.

7. Results and analysis

We conduct a series of experiments to evaluate the performance of our proposed HGNN. We implement not only classical models but also the advanced deep learning model for comparison, and analyze the effectiveness of proposed HGNN from different aspects, including the overall performance, the ablation study of hierarchical fusion, study on back-testing strategies and case study of integrated gradients on stock relation.

7.1. Overall performance

The experimental results of different models are presented in Table 3. In all cases, our proposed HGNN_M and HGNN_I achieve better results in both SSE and SZSE markets compared to other models. For example, the accuracy values are 63.74% and 63.59% for HGNN_I in SSE and SZSE markets, respectively; the values of F1 score are 63.97% and 63.07% for HGNN_I in SSE and SZSE markets, respectively. Additionally, the stability of proposed model is also verified by the standard deviations of the evaluation metrics, especially HGNN_I. The performance of HGNN_I has been maintained at a relatively stable level in both SSE and SZSE markets. The results prove the advantage and effectiveness of proposed model in the classification task of price-limit-hitting stock.

The best classification performance of the historical sequence based model (LSTM, ALSTM) and limit-related indicator based models (Naive Bayes, LR, SVM, and XGBoost) can be improved to a similar level(59+%). And the best classification performance of graph-based method (GCN, GAT) can achieve better results, which indicates considering and combining information from different data sources effectively can lead to better results in the price-limit-hitting stock classification. We believe that the relations among the stock provide more useful information, while other models without stock relation graph cannot infer the stock price movement only by relying on stock trend information. Moreover, compared to graph-based method (GCN, GAT), proposed model has increased by an obvious improvement in both markets. It strongly proves the role of integrating hierarchical information based on introducing stock relations, so as to obtain more the clues of price limit.

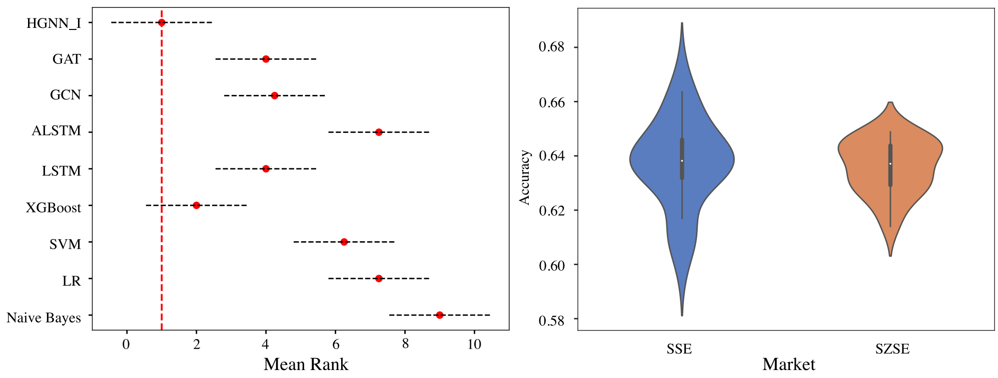
Moreover, compared with HGNN_M, HGNN_I can achieve better results. In HGNN_M, the Market-Oriented Temporal Attention Aggregator can introduce market evolution into the importance-strength function. However, it is not fine-grained enough to directly model the relation between the whole market and a certain stock. For HGNN_I, the Industry-Oriented Temporal Attention Aggregator can capture the more fine-grained importance-strength information, and can better indicate what market state the individual stock belongs to.

7.2. Statistics test

To prove whether the obtained experimental results are significant, we perform a statistics test with the Friedman test and distribution analysis of experimental results. The Friedman test with the corresponding post hoc tests is employed for the comparison of more classifiers over multiple datasets [47]. The Friedman test was used to evaluate the statistical significance of any difference in the mean ranks of each approach. We calculate the Friedman test statistic $p-value$=.000 <.05, which indicates there are statistically significant differences in the performance of each method. Therefore, we conduct a post hoc test (Bonferroni-Dunn test) to determine which methods have statistical differences in performance. Fig. 5(a) shows a critical difference diagram of Bonferroni-Dunn test. From the figure, we can see that proposed HGNN is ahead of the other

Table 3
Experimental results of the compared methods.

Model	SSE		SZSE	
	Acc(%)	F1(%)	Acc(%)	F1(%)
Naive Bayes	46.21 ± 1.31	44.37 ± 1.60	47.80 ± 1.19	45.76 ± 0.99
LR	58.59 ± 1.03	59.43 ± 0.95	57.42 ± 0.34	58.27 ± 0.32
SVM	58.04 ± 2.11	58.90 ± 2.05	58.16 ± 1.06	59.01 ± 1.05
XGBoost	60.59 ± 0.80	61.11 ± 0.69	59.10 ± 1.73	59.87 ± 1.67
LSTM	59.74 ± 2.43	58.70 ± 1.62	58.52 ± 1.97	57.43 ± 1.34
ALSTM	57.45 ± 2.00	57.42 ± 1.38	56.67 ± 1.38	56.03 ± 0.73
GCN	60.54 ± 1.91	59.54 ± 1.43	60.57 ± 1.75	58.94 ± 1.74
GAT	61.56 ± 1.99	60.55 ± 1.56	59.79 ± 1.84	59.15 ± 1.76
HGNN_M	63.33 ± 1.15	63.36 ± 1.11	63.29 ± 1.48	62.96 ± 1.13
HGNN_I	**63.74** ± **1.64**	**63.97** ± **1.05**	**63.59** ± **0.96**	**63.07** ± **1.13**



(a) Critical difference diagram of Bonferroni-Dunn test (b) Distributions on HGNN over Multiple Tests

Fig. 5. Statistics test of HGNN. (a) is critical difference diagram of Bonferroni-Dunn test, which shows the mean ranks of nine methods over the two stock datasets. (b) shows the distributions of classification accuracy with HGNN over multiple experiments. The most predictions fall to the side of superior performance based on multiple experiments.

baseline methods in mean rank. It indicates the significant different rankings (performance) between our HGNN and the other baseline methods. Moreover, we conduct the distribution analysis of experimental results. Fig. 5(b) shows the distributions of classification accuracy with HGNN over multiple experiments. As can be seen, the distribution of classification accuracy with HGNN is left skewed ($Mean < Median < Mode$) in both markets, which indicates that most predictions fall to the side of superior performance based on multiple experiments [48]. It also justifies that the experimental results of proposed HGNN are significant.

7.3. Ablation study

We conduct two different ablation experiments to analyze the effectiveness of different factors in our model, including the effect of hierarchical information fusion and the aggregation module of Temporal Attention Aggregator.

The Effect of Hierarchical Information Fusion. To further understand the effectiveness of the hierarchical information in stock data, an ablation experiment of integrating hierarchical information is performed in HGNN_M and HGNN_I (Hereinafter referred to as HGNN). The variations are tested under information integration of different views including node view, relation view and graph view. We remove the information of each view respectively to reveal the capability of each single view in HGNN. For example, *HGNN without RV* denotes the information of relation view is removed from the integration of hierarchical information. The performance of ablation experiment with hierarchical fusion is shown in Table 4. From the result, proposed *HGNN* with the information of a complete view achieves the best performance in both markets. The lack of hierarchical fusion causes varying degrees of loss in performance in both markets. It also proves that the hierarchical structure of HGNN can effectively improve the prediction performance, and HGNN obtains more useful information from multiple views to infer the movement of price-limit-hitting stocks while other models do not.

The Effect of Attention Aggregator in Temporal Attention Aggregator.

The movement of one stock is related to the market state [18]. Therefore, it is natural that the trend of market can help reflect the strength of price limit. In order to verify whether our proposed model can capture the time-aware market state for the type prediction task, we design a *Temporal Attention Aggregator* to encode the market state with temporal property. Owing to HGNN_I outperforms better than HGNN_M, and to reduce experimental result redundancy, we only show the relevant ablation study experiments of HGNN_I on Temporal Attention Aggregator. By replacing the attention aggregator with

Table 4
The effect of hierarchical fusion on HGNN. We remove node view (NV), relation view (RV) and graph view (GV), respectively. For example, *HGNN_M(w/o NV)* denotes that the node view is removed from hierarchical fusion.

Model	SSE		SZSE	
	Acc(%)	F1(%)	Acc(%)	F1(%)
HGNN_M	**63.33 ± 1.15**	**63.36 ± 1.11**	**63.29 ± 1.48**	**62.96 ± 1.13**
HGNN_M (w/o NV)	62.21 ± 1.72	60.56 ± 1.21	62.31 ± 2.23	60.69 ± 1.34
HGNN_M (w/o RV)	61.92 ± 1.62	62.00 ± 1.41	62.94 ± 1.12	62.87 ± 1.01
HGNN_M (w/o GV)	62.33 ± 1.63	62.29 ± 1.44	62.48 ± 2.08	62.43 ± 1.84
HGNN_I	**63.74 ± 1.64**	**63.97 ± 1.05**	**63.59 ± 0.96**	**63.07 ± 1.13**
HGNN_I (w/o NV)	62.09 ± 2.43	62.12 ± 1.30	61.33 ± 1.41	59.31 ± 0.71
HGNN_I (w/o RV)	60.72 ± 2.09	61.70 ± 1.73	62.45 ± 1.20	62.37 ± 0.77
HGNN_I (w/o GV)	62.33 ± 1.63	62.29 ± 1.44	62.48 ± 2.08	62.43 ± 1.84

other aggregator method, we perform the ablation study to verify the effectiveness of modeling the market state obtained by the attention aggregator in Temporal Attention Aggregator module. The results of the ablation study as shown in Table 5.

It is essential to verify the effectiveness of the internal attentive mechanism in *Attention Aggregator* module. Therefore, we replace *Attention Aggregator* with *Mean Aggregator*, *Max Aggregator* and *Min Aggregator* respectively in our proposed *Temporal Attention Aggregator*, and remain the same of the training process and datasets. Our original HGNN_I implemented achieves the best performance in both markets than other aggregator methods. Besides, HGNN_I with *Mean Aggregator*, *Max Aggregator* and *Min Aggregator* cannot adaptively learn the importance of a certain stock to generate accurate market state from stock relation graph and lead to the prediction performance loss. These results confirm that each stock has different degree of reflection on the whole market state, and integrating the temporal feature of each stock node adaptively can better represent the time-aware state of the market.

7.4. Study on different back-testing strategies

In reality, investors always chase the price-limit-hitting stocks to seek overnight return. We investigate the performance of average return ratio (ARR) about the baselines and proposed method. According to the market simulation, Fig. 6 illustrates the results of back-testing regarding the average return ratio (ARR) in both markets, and average return ratio means the return ratio of trading once (buy a price-limit-hitting stock at daily price limit and sell it at the opening price of the next trading day). As can be seen, the gaps in ARR are obvious between baselines, which indicates that accurately selecting price-limit-hitting stocks that perform well is a highly difficult operation. In all cases, HGNN_M and HGNN_I can obtain better average return ratios than other models, and HGNN_I achieves 0.83% and 0.87% for average return ratio in two markets, respectively. Meanwhile, it also emphasizes the significant potential to gains profits from price limit and the necessity of the price-limit-hitting stock classification.

Additionally, in order to better evaluate the achieved performance of average return ratios, we choose the best model HGNN_I for comparison and set two contrasting investment strategies in both markets: (1) **Optimal**: we select and trade the stocks belonging to Type I of the ground-truth during the testing period from two datasets of our method. (2) **Whole**: we trade all the price-limit-hitting stocks during the testing period from two datasets, whether they are Type I or Type II. **Optimal** is the most desirable investment strategy, while **Whole** strategy can be seen as investors chasing the price-limit-hitting stocks enthusiastically due to bullish signal brought by price limit. Table 6 shows the results of two investment strategies regarding the average return ratios in SSE and SZSE markets. **Whole** achieves negative return ratios in two mar-

Table 5
The Effect of attention aggregator in Temporal Attention Aggregator.

Model	SSE		SZSE	
	Acc(%)	F1(%)	Acc(%)	F1(%)
HGNN_I	**63.74 ± 1.64**	**63.97 ± 1.05**	**63.59 ± 0.96**	**63.07 ± 1.13**
HGNN_I (with Mean Aggregator)	61.94 ± 1.70	62.84 ± 1.30	61.83 ± 1.63	61.92 ± 1.15
HGNN_I (with Max Aggregator)	61.44 ± 2.15	62.30 ± 1.65	61.56 ± 1.57	61.72 ± 1.08
HGNN_I (with Min Aggregator)	61.35 ± 1.58	62.36 ± 1.29	61.23 ± 2.01	61.41 ± 1.48

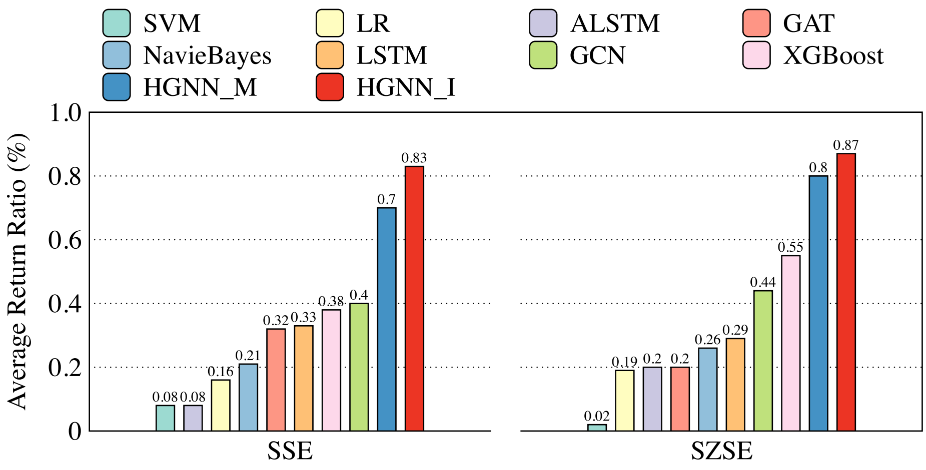
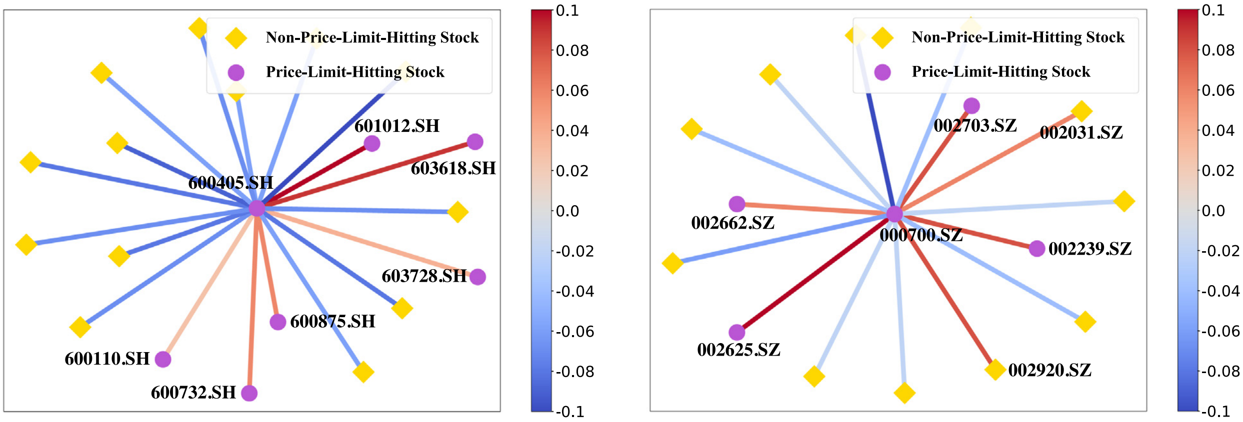


Fig. 6. Comparison on market simulation strategies of different models about Average Return Ratio (ARR).

Table 6
Average Return Ratio (ARR) performance of HGNN_I as compared to different investment strategies in two markets.

Markets	Optimal	Whole	HGNN_I
SSE	**3.85%**	−0.03%	0.83%
SZSE	**3.89%**	−0.29%	0.87%



(a) SSE **(b) SZSE**

Fig. 7. Case study of integrated gradients on the edge of stock relation graph in electrical equipment industry of SSE market and auto parts industry of SZSE market, where the circular nodes represent the price-limit-hitting stocks, the square nodes represent the non-price-limit-hitting stocks, and the central node is predicted correctly by proposed HGNN_I as Type I. The color of the edges with heatmaps indicates the value of integration gradients, and the greater the absolute value of the integrated gradient can be understood as the greater the role the edge plays in the prediction.

kets, which can confirm that buying the price-limit-hitting stocks is a highly risky operation. Moreover, it further verifies that the price-limit-hitting stock classification task is significant and necessary, and proposed model is effective for this task. Naturally, there is a significant gap between proposed model and the ideal investment strategy **Optimal**. This result is acceptable since selecting the stock performing well is a highly risky operation based **Whole**, but it also reflects that there is still huge space for improvement in the price-limit-hitting stock type classification task.

7.5. Case study of integrated gradients on the edge

In order to evaluate the effect of relations between stocks in proposed HGNN, we choose the better model HGNN_I and then generate explanation edge heatmaps for predicted importance using the *integrated gradients* described in [49,50]. The *integrated gradients* is an explainable AI technique recently, and it aims to explain the relations between a model's predictions in terms of its features. In our experiment, we set the integrated gradients to measure the contribution of each stock relation in the industry relation graph to predicted importance. We exemplify the price-limit-hitting stocks classified correctly by proposed HGNN as Type I as central nodes on January 4, 2021. Fig. 7 shows the results of integrated gradient on the relation in electrical equipment industry of SSE market and auto parts industry of SZSE market. We observe that the neighbor nodes belonging to the price-limit-hitting stocks generally play a more positive and vital role in the prediction of the central node. It shows the stronger connection between price-limit-hitting stock pairs than regular stock pairs, and also illustrates the clustering phenomenon of price-limit-hitting stocks in Fig. 2 from the perspective of representation learning. Moreover, it further proves the effectiveness of industry relation among stocks to the price-limit-hitting stock classification task.

Moreover, we apply the industry relation to build the stock relation graph, which is preferred by most studies and has been proved to have significant performance [7,20,32]. And it is hard to assume one certain type of known stock relation is superior to others when constructing the stock relation graph. For industry relations, it may not be fine-grained enough for an accurate predictive model in some cases. For example, Fig. 7(a) shows a general case in prediction, that is, the neighbor price-limit-hitting stock nodes generally play a more positive and important role in the prediction of the central price-limit-hitting node. However, a few non-price-limit-hitting or even falling stocks will also have a strong impact on the central price-limit-hitting node, which deviates from original intention of proposed method (like 002920.SZ in Fig. 7(b)). It indicates that there is a vulnerability of the prediction under exceptional circumstances if only industry relations are used. Consequently, in order to avoid misprediction as much as possible, we not only pay attention to the information of related stocks, but also integrate hierarchical information of stock market into the predictive model.

8. Conclusion

In this paper, we propose a novel classification objective that aims to predict the type of price-limit-hitting stocks, due to the different types of price-limit-hitting stocks that will lead to extremely different profits for stock investments. The combined role of multiple market states (e.g., from the perspective of industry sector, the whole market, etc.) leads to the uncertainty of price limit. The core task of this study is to learn a better stock representation with less uncertainty by comprehensively considering the hierarchical property of the market state. Our contribution is to propose a Hierarchical Graph Neural Network (HGNN) combined framework, which enables the integration of multiple market states extracted from historical sequence patterns and stock relations. Specifically, we creatively propose to introduce hierarchical architecture to stock prediction model, which provides deeper insight into the real investment practice. Moreover, we innovatively apply this idea to the investment practice of predicting the type of price-limit-hitting stock. In addition, the proposed hierarchical structure can be easily adapted to other stock prediction task. Experimental results show that HGNN can achieve high performance and investment return ratio for classifying the type of price-limit-hitting stock on two real stock markets.

In the future, the following directions can be investigated: 1) Stock-related news has proven helpful in capturing stock trends. Therefore, we argue that the financial news will contribute to predicting stock type. We plan to introduce financial text mining methods into HGNN to explore more available media data such as financial reports and social media texts for strengthening the hierarchy. 2) Considering the stock market is liquid and the diversity of stock relations, the stock relation graph using only predefined industry relations in HGNN is not inevitably fine-grained enough for an accurate type prediction. We will give priority to how to build a more adaptive stock relation graph by incorporating linkages between stocks into the learnable modules. 3) Furthermore, we will investigate the performance of HGNN under real market conditions such as efficiency in real-time data stream and design the more abundant portfolios of multiple investment strategies for back-testing.

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.

References

[1] Y. Shi, W. Li, L. Zhu, K. Guo, E. Cambria, Stock trading rule discovery with double deep q-network, Appl. Soft Comput. 107 (2021) 107320.
[2] C. Xie, D. Rajan, Q. Chai, An interpretable neural fuzzy hammerstein-wiener network for stock price prediction, Inf. Sci. 577 (2021) 324–335.
[3] Z. Qi, Z. Bu, X. Xiong, H. Sun, J. Cao, C. Zhang, A stock index prediction framework: Integrating technical and topological mesoscale indicators, in: 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), IEEE, 2019, pp. 23–30.
[4] Q. Ding, S. Wu, H. Sun, J. Guo, J. Guo, Hierarchical multi-scale gaussian transformer for stock movement prediction, in: Proceedings of International Joint Conference on Artificial Intelligence, 2020, pp. 4640–4646.
[5] G. Gao, Z. Bu, L. Liu, J. Cao, Z. Wu, A survival analysis method for stock market prediction, in: 2015 International Conference on Behavioral, Economic and Socio-cultural Computing (BESC), IEEE, 2015, pp. 116–122.
[6] L. Zhang, C. Aggarwal, G.-J. Qi, Stock price prediction via discovering multi-frequency trading patterns, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2017, pp. 2141–2149.
[7] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, T.-S. Chua, Temporal relational ranking for stock prediction, ACM Trans. Inform. Syst. 37 (2) (2019) 1–30.
[8] C. Chen, L. Zhao, J. Bian, C. Xing, T.-Y. Liu, Investment behaviors can tell what inside: exploring stock intrinsic properties for stock trend prediction, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2376–2384.
[9] W. Chen, M. Jiang, W.-G. Zhang, Z. Chen, A novel graph convolutional feature based convolutional neural network for stock trend prediction, Inf. Sci. 556 (2021) 67–94.
[10] Y. Shi, Y. Zheng, K. Guo, X. Ren, Stock movement prediction with sentiment analysis based on deep learning networks, Concurrency and Computation: Practice and Experience 33 (6) (2021) e6076.
[11] F. Feng, H. Chen, X. He, J. Ding, M. Sun, T.-S. Chua, Enhancing stock movement prediction with adversarial training, in: Proceedings of International Joint Conference on Artificial Intelligence, 2019, pp. 5843–5849.
[12] C. Cappelli, R. Cerqueti, P. D'Urso, F. Di Iorio, Multiple breaks detection in financial interval-valued time series, Expert Syst. Appl. 164 (2021) 113775.
[13] H. Li, D. Zheng, J. Chen, Effectiveness, cause and impact of price limit–evidence from china's cross-listed stocks, J. Int. Financial Markets, Instit. Money 29 (2014) 217–241.
[14] T. Chen, Z. Gao, J. He, W. Jiang, W. Xiong, Daily price limits and destructive market behavior, J. Econometr. 208 (1) (2019) 249–264.
[15] I.M. Sifat, A. Mohamad, Trading aggression when price limit hits are imminent: Nardl based intraday investigation of magnet effect, J. Behav. Exp. Finance 20 (2018) (2018) 1–8.
[16] K.A. Kim, S.G. Rhee, Price limit performance: evidence from the tokyo stock exchange, J. Finance 52 (2) (1997) 885–901.
[17] G. Liu, Y. Mao, Q. Sun, H. Huang, W. Gao, X. Li, J. Shen, R. Li, X. Wang, Multi-scale two-way deep neural network for stock trend prediction, in: Proceedings of International Joint Conference on Artificial Intelligence, 2020, pp. 4555–4561.
[18] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, Q. Su, Modeling the stock relation with graph network for overnight stock movement prediction, in: Proceedings of International Joint Conference on Artificial Intelligence, 2020, pp. 4541–4547.
[19] Q. Li, J. Tan, J. Wang, H. Chen, A multimodal event-driven lstm model for stock prediction using online news, IEEE Trans. Knowl. Data Eng. 33 (10) (2020) 3323–3337.
[20] R. Cheng, Q. Li, Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 55–62..
[21] H. Wang, S. Li, T. Wang, J. Zheng, Hierarchical adaptive temporal-relational modeling for stock trend prediction, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event/ Montreal, Canada, 19–27 August 2021, ijcai.org, 2021, pp. 3691–3698..
[22] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 753–763.

[23] H. Lin, D. Zhou, W. Liu, J. Bian, Learning multiple stock trading patterns with temporal routing adaptor and optimal transport, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1017–1026.
[24] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, R.R. Shah, Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach, Proc. AAA I (2021) 497–504.
[25] R.D. Edwards, J. Magee, W.C. Bassetti, Technical analysis of stock trends, CRC Press, 2018.
[26] R. Bisoi, P.K. Dash, A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented kalman filter, Appl. Soft Comput. 19 (2014) 41–56.
[27] V. Golosnoy, B. Gribisch, R. Liesenfeld, The conditional autoregressive wishart model for multivariate stock market volatility, J. Econometr. 167 (1) (2012) 211–223.
[28] 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 International Joint Conference on Artificial Intelligence, 2017, pp. 2627–2633.
[29] J. Wang, T. Sun, B. Liu, Y. Cao, H. Zhu, Clvsa: a convolutional lstm based variational sequence-to-sequence model with attention for predicting trends of financial markets, in: Proceedings of International Joint Conference on Artificial Intelligence, 2019, pp. 3705–3711.
[30] Y. Lu, C. Shi, L. Hu, Z. Liu, Relation structure-aware heterogeneous information network embedding, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 4456–4463.
[31] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of International Conference on Learning Representations, 2017, pp. 1–14.
[32] Y. Chen, Z. Wei, X. Huang, Incorporating corporation relationship via graph convolutional neural networks for stock price prediction, in: Proceedings of ACM International Conference on Information and Knowledge Management, 2018, pp. 1655–1658.
[33] E.F. Fama, The behavior of stock-market prices, J. Business 38 (1) (1965) 34–105.
[34] J.M. Poterba, L.H. Summers, Mean reversion in stock prices: Evidence and implications, J. Financ. Econ. 22 (1) (1988) 27–59.
[35] 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 ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 894–902.
[36] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[37] K.-J. Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (1–2) (2003) 307–319.
[38] C.-F. Tsai, Y.-C. Hsiao, Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches, Decis. Support Syst. 50 (1) (2010) 258–269.
[39] J. Huang, W.X. Zhao, H. Dou, J.-R. Wen, E.Y. Chang, Improving sequential recommendation with knowledge-enhanced memory networks, in: Proceedings of International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 505–514.
[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[41] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.
[42] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, A. Iosifidis, Using deep learning to detect price change indications in financial markets, Proceedings of European Signal Processing Conference (2017) 2511–2515.
[43] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2016, pp. 785–794.
[44] D.M. Nelson, A.C. Pereira, R.A. de Oliveira, Stock market's price movement prediction with lstm neural networks, in: Proceedings of International Joint Conference on Neural Networks, 2017, pp. 1419–1426.
[45] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: Proceedings of the International Conference on Learning Representations, 2018, pp. 1–12.
[46] M. Seasholes, G. Wu, Profiting from predictability: Smart traders, daily price limits, and investor attention, University of California, Berkeley, working paper (2004)..
[47] J. Demšar, Statistical comparisons of classifiers over multiple data sets, The, J. Mach. Learn. Res. 7 (2006) 1–30.
[48] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, Moments of a distribution: Mean, variance, skewness, and so forth, Numer. Recipes (1992) 604–609.
[49] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the International Conference on Machine Learning, 2017, pp. 3319–3328.
[50] R. Sayres, A. Taly, E. Rahimy, K. Blumer, D. Coz, N. Hammel, J. Krause, A. Narayanaswamy, Z. Rastegar, D. Wu, et al, Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy, Ophthalmology 126 (4) (2019) 552–564.