Graph-Based Stock Recommendation by Time-Aware Relational Attention Network

JIANLIANG GAO, XIAOTING YING, CONG XU, JIANXIN WANG, and SHICHAO ZHANG, School of Computer Science and Engineering, Central South University, China
ZHAO LI, Alibaba Group, China

The stock market investors aim at maximizing their investment returns. Stock recommendation task is to recommend stocks with higher return ratios for the investors. Most stock prediction methods study the historical sequence patterns to predict stock trend or price in the near future. In fact, the future price of a stock is correlated not only with its historical price, but also with other stocks. In this article, we take into account the relationships between stocks (corporations) by stock relation graph. Furthermore, we propose a Time-aware Relational Attention Network (TRAN) for graph-based stock recommendation according to return ratio ranking. In TRAN, the time-aware relational attention mechanism is designed to capture time-varying correlation strengths between stocks by the interaction of historical sequences and stock description documents. With the dynamic strengths, the nodes of the stock relation graph aggregate the features of neighbor stock nodes by graph convolution operation. For a given group of stocks, the proposed TRAN model can output the ranking results of stocks according to their return ratios. The experimental results on several real-world datasets demonstrate the effectiveness of our TRAN for stock recommendation.

4

CCS Concepts: • **Information systems → Recommender systems**; **Data mining**; • **Computing methodologies → Neural networks**; • **Applied computing → Economics;**

Additional Key Words and Phrases: Stock relation graph, knowledge discovery, stock recommendation, relational attention network, time-aware

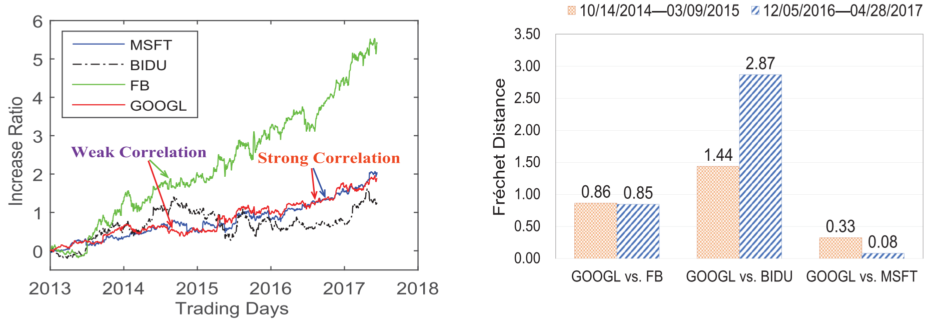
ACM Reference format:
Jianliang Gao, Xiaoting Ying, Cong Xu, Jianxin Wang, Shichao Zhang, and Zhao Li. 2021. Graph-Based Stock Recommendation by Time-Aware Relational Attention Network. *ACM Trans. Knowl. Discov. Data.* 16, 1, Article 4 (June 2021), 21 pages.
https://doi.org/10.1145/3451397

1 INTRODUCTION

Stock market plays an important role in the economic operations of modern society. As one of the largest financial markets, stock market has reached the total value of 99 trillion dollars in recent

The work is supported by the National Natural Science Foundation of China under Grant No.: 61873288, 61836016, CAAI-Huawei MindSpore Open Fund, Zhejiang Lab under Grant No.2019KE0AB01, and the Fundamental Research Funds for the Central Universities of Central South University (1053320212445).
Authors' addresses: J. Gao, X. Ying, C. Xu, J. Wang, and S. Zhang, School of Computer Science and Engineering, Central South University, Changsha 410083, China; emails: {gaojianliang, yingxiaoting, jxwang, zhangsc}@csu.edu.cn, csuxucong@163.com; Z. Li (corresponding author), Alibaba Group, Hangzhou 311500, China; email: lizhao.lz@alibabainc.com.

1556-4681/2021/06-ART4 $15.00
https://doi.org/10.1145/3451397



(a) Increase Ratio Trend of Stock Price

(b) Time-varying Correlation with Fréchet Distance

Fig. 1. An example of stock relations of four stocks (corporations) in NASDAQ stock market. (a) Shows the different correlations of stock price curves in the same sector-industry (Technology: Computer Software), where increase ratio is obtained by comparing to the *close* price on Jan 2, 2013. (b) Uses Fréchet distance to measure the time-varying correlations between different price curves over two price series in (a), where the smaller the Fréchet distance is, the stronger is the correlation strength between them.

years.[1] In order for seeking maximized profit over their investments, stock investors continuously attempted to predict the future trends of market. Recently, some evidences have indicated the predictability of stock markets [1, 2]. They treat stock historical sequence as typical time-series data [3], and then transform stock prediction into time-series analysis and prediction [4–6]. However, analysing stock market movements and price behaviours is extremely challenging because of the highly volatile and non-stationary nature of the markets.

Inspired by recommendation system [7–9], a good stock recommendation model should be able to recommend the stocks with higher return ratios to the investors from a group of stocks. Additionally, analysis and forecasting of the stock market is being studied using various data. Some researchers utilize time-series information as features to forecast stock market movement, such as stock price sequence [10]. Others dig into textual features as external knowledge combined with price features to predict, such as financial news [2, 11] and social media [12]. Besides the basic series information and textual information, connection among corporations is also utilized for stock prediction recently [13]. The **Efficient Market Hypothesis** (**EMH**) indicates that all relevant information will be reflected in the stock price, and the information from different sources can complement each other and affect stock prices [14, 15]. Therefore, it is natural to believe that the change of the stock price of a target corporation would be affected by corporations that are related in the real-world [1, 13]. In order to take this issue into consideration, we utilize the stock relations from the real market as an injection of explicit knowledge when making stock prediction. However, there are two major challenges for utilizing the relation among stocks (corporations):

—How to quantify the strength of the relation among stocks (corporations). Corporations are connected with each other broadly via various relationships such as the classification of sector-industry. Even if a group of stocks falls into the same category, there is a difference in the strength of their correlation with each other. As shown in Figure 1(a), the stock prices of four corporations belong to the same sector-industry (Technology: Computer Software) in the **National Association of Securities Dealers Automated Quotations** (**NASDAQ**) stock market, but the strengths of the correlation among them are very different. For example, MSFT (Microsoft Inc.) and GOOGL (Alphabet Inc.) exhibit similar trend on the stock

[1] https://data.worldbank.org/indicator/CM.MKT.TRAD.CD.

price with strong correlation over time, while the correlation strengths of FB (Facebook Inc.), and GOOGL (Alphabet Inc.) are quite different. These correlation strengths are also reflected in Figure 1(b) with Fréchet distance.

—How to quantify the time-varying strength of the relation among stocks (corporations). Although a correlation exists among corporations, the strength of the correlation changes over the time. Figure 1(b) shows the correlation change of different corporations measured by Fréchet distance [16] over two time periods (from 10/14/2014 to 03/09/2015 and from 12/05/2016 to 04/28/2017), where the smaller the Fréchet distance is, the stronger is the correlation strength between them. In this example, the strengths between GOOGL (Alphabet Inc.) and FB (Facebook Inc.), BIDU (Baidu Inc.), MSFT (Microsoft Inc.) remain essentially stable (from 0.86 to 0.85), decreased (from 1.44 to 2.87) and increased (from 0.33 to 0.08), respectively. Therefore, it is necessary to design a model to characterize the more detailed relation and capture the time-aware correlation strength between stocks.

In order to tackle these two challenges, we propose a **Time-aware Relational Attention Network (TRAN)** for graph-based stock recommendation. Firstly, we construct the stock relation graph-based on the classification of sector-industry to represent the real relation in stock market, and extract the stock historical feature and document feature with stock attribute from historical sequence and stock description document, respectively. Then, we interact dynamically the document feature and historical feature to account for the temporal property and detailed relational feature of the stock. Based on that, a TRAN is designed to capture time-aware stock relation strength, which can describe the more accurate relation strengths to improve the recommendation performance. Furthermore, the relation strength is treated as attention edge weight on stock relation graph and graph convolution operation is utilized to aggregate historical features of related stock nodes for representing each stock node more comprehensively. Finally, the output of graph convolution and the stock historical features are utilized to predict ranking of all stocks. In summary, the main contributions of this article are as follows:

—We propose a graph-based stock recommendation model based on both the historical sequence patterns and the stock relation graph, which utilizes the relations between the stocks (corporations) for ranking the return ratios in a group of stocks.

—We design a method termed TRAN for graph-based stock recommendation, which can capture the time-aware relation strength between stocks and improve the accuracy for stock recommendation.

—We empirically show that the proposed model outperforms existing stock recommendation methods on two real market datasets, which demonstrates the effectiveness of the proposed TRAN for stock recommendation.

2 RELATED WORK

Our work is directly related to the recent work on stock prediction with multi-source data, stock prediction with graph-based learning, and multiple stocks prediction and stock ranking.

2.1 Stock Prediction with Multi-Source Data

The EMH [14, 17] states that the price of a stock reflects all of the information available, and that every investor has a certain degree of access to the information. Although many researchers in the fields of finance, computer science, and other research communities have studied more than 50 years, the debate continues about what kinds of information is useful for stock market prediction [18]. In recent years, stock historical sequences and external information have been the most exploited for stock prediction methods. Therefore, stock prediction methods can be cataloged as historical sequences and external information approaches based on the data sources.

Most stock prediction methods study the historical sequence patterns to predict stock trend or price in the near future. They represent the indicator of a stock as a stochastic process and take the historical sequence of the indicator to fit the process, such as Autoregressive models [19], Kalman Filters [20], and other combined models [21]. For example, Lin selects the **Shanghai Stock Exchange** (**SSE**) Composite Index as research object, through the application of **generalised autoregressive conditional heteroscedasticity** (**GARCH**) type models to conduct stock empirical analysis [22]. However, traditional solutions lack the capability to model highly volatile market.

Recently, deep models have become a promising solution to substitute the mathematical models, especially **recurrent neural networks** (**RNNs**) [10, 23, 24]. Nelson et al. study the **Long Short-Term Memory** (**LSTM**) networks [25] on stock historical sequence to predict future trends of stock prices [23]. Zhang et al. propose a **State Frequency Memory** (**SFM**) recurrent network inspired by discrete fourier transform to discover multi-frequency trading patterns, which makes price prediction from past market sequence [10]. Wang et al. design a **Convolutional LSTM based Variational Sequence-to-Sequence model with Attention** (**CLVSA**) model based on stochastic recurrent networks to capture underlying features in raw financial sequence data [24].

In addition to historical price sequence, stock prediction also considers external information such as financial news, social media, and other various information. For example, Ding et al. capture the influence of financial news event to predict stock price movements [26]. Xu and Cohen incorporate both data from social media and historical price sequences, which mines user opinions to improve the accuracy of stock prediction [12]. Deng et al. combine event embeddings obtained by external knowledge information from the knowledge graph and price values together to forecast stock trend [2]. Li et al. take the Sino–US trade friction incident in 2018 as the research background to analyze the stability of China's stock market [27].

2.2 Stock Prediction with Graph-Based Learning

Graph-based learning aims at achieving better performance in the target task by incorporating the relationship among entities into the learning procedure [28]. Recently, instead of treating the stocks as independent of each other, some works have began to use the graph-based learning methods to obtain the information of correlations among stocks and make prediction. Chen et al. construct the corporation graph-based on financial investment fact, and utilize a popular graph-based learning method called **Graph Convolutional Network** (**GCN**) [29] to incorporate information of related corporations of a target company [13]. Feng et al. consider the relation between the stocks in the same sector or industry, which may affect related stocks to exhibit similar trends in their prices. Then, they adopt the idea of the PageRank to encode certain similarity information between two stocks in the same sector or industry [1]. Ying et al. take into account differences between stocks in the same sector or industry, and extract stock description documents to describe more detailed relations. Then, they design a graph-based model to capture the time-varying attention edge weight between the connected stocks [30]. Ye et al. employ GCN to model the influence of related stock based on three novel graphs which represent the shareholder relationship, industry relationship, and concept relationship among stocks based on investment decisions, and take their multiple relationships into consideration to improve the prediction accuracy [31].

However, the stocks (corporations) are related to some other stocks broadly via various relationships, it is still a challenging problem to express the relationship between stocks accurately, especially the stocks in the same sector or industry still present very different price trends, which indicates there are many other factors (e.g., business, capital) leading to more detailed relation in the same sector or industry. At the same time, the relation strength between stocks changes over time in the real markets, but the graph-based learning methods like GCN cannot capture time-aware relation strength over time.

2.3 Multiple Stocks Prediction and Stock Ranking

Fischer and Krauss [32] focus on 500 stocks in the S&P 500 index and adopt the LSTM networks to predict the multiple stocks. They select top-k stocks and bottom-k stocks from multiple stocks prediction to make long positions and short positions, respectively. This strategy with LSTM networks achieves better performance than other model such as **Random Forest** (**RF**) and vanilla Deep Neural Networks. However, they treat different stocks as independent sequences, which neglects the relative comparison between stocks. Therefore, stock ranking prediction has been proposed to rank the stocks with return ratios based on the comparison among multiple stocks recently. Feng et al. propose an objective function that combines both pointwise regression loss and pairwise ranking-aware loss to predict the stock ranking [1]. Chen et al. also combine pointwise regression loss and pairwise ranking-aware loss into a unified objective function to enhance stock trend prediction, but they add the regularization item to avoid obtaining an optimal constant ranking score [33]. Chiewhawan and Vateekul leverage **mean square error** (**MSE**) loss and pairwise ranking-aware loss to make multiple stock returns prediction [34].

Based on these methods, we think the multiple stocks prediction and stock ranking can provide new ideas for stock recommendation in terms of the optimization objective. Stock ranking prediction can be converted into the stock recommendation model, which is able to recommend the stocks with higher return ratios to the investors.

Difference to Existing Methods. Unlike the pioneering methods described above, our model utilizes both sequence information, textual information, and the classification of stock sector-industries to characterize the stock relation for better recommendation. Our proposed TRAN is able to describe the more detailed relation among the stocks, and capture time-aware relation strength between stocks to represent the volatility of the stock market better. Compared with the accuracy of the stock ranking prediction, TRAN can recommend the stock with higher return ratio from the perspective of the overall model objective more accurately and effectively.

3 PRELIMINARIES

For all S stocks, we get a set of historical sequence data and collect description documents of corporations that issue stocks. A set of stock historical sequence data at day t is represented by $\mathcal{X}_t = \{X_t^1, X_t^2, \ldots, X_t^S\} \in \mathbb{R}^{S\times T\times K}$, where T is the length of time-series and K is the dimension of features such as *open*, *high*, *low*, *close*. Description documents of corporations that issue stocks are represented by $D = \{d_1, d_2, \ldots, d_S\}$. For each stock s, it has own description document d_s and historical sequence data $X_t^s = \{x_{t-T+1}^s, x_{t-T+2}^s, \ldots, x_t^s\}$ at day t. In particular, every stock has its attribute of sector-industry classification. For example, MSFT (Microsoft Inc.) and GOOGL (Alphabet Inc.) belong to the same sector-industry classification (Technology: Computer Software). Subsequently, we define a stock relation graph-based on their sector-industry classifications as the explicit knowledge for stock recommendation, and formulate the stock recommendation problem.

Definition 3.1 (Stock Relation Graph). An undirected graph $G = \{V, E\}$, where V is a set of $|V| = S$ nodes representing stocks; E is a set of edges, indicating the relation between stocks. To determine the relation between stock i and j, their sector-industry classifications are converted into multi-hot binary vector C_i and C_j. If $C_i \cdot C_j \geq 1$, there is an edge connecting stock i and j, where "$\cdot$" is the inner product operation. Each node s generates a feature vector at each time step t from the historical sequence X_t^s, and each edge generates the time-aware relation strength at each time step t.

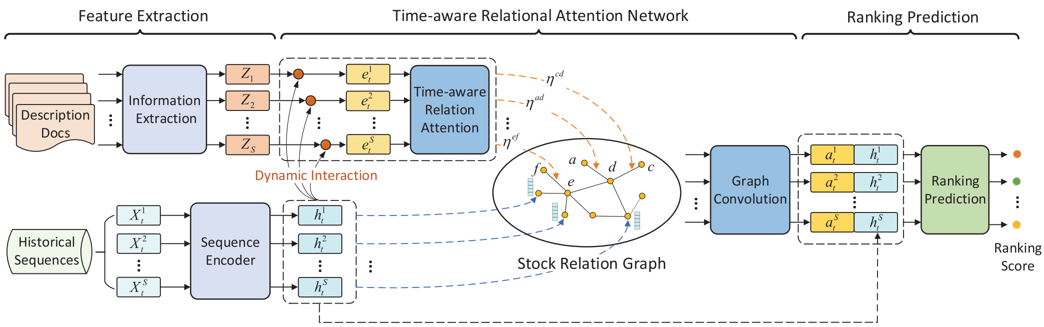


Fig. 2. The overall framework of stock recommendation model, which mainly consists of three components: Feature Extraction, TRAN, and Ranking Prediction.

Definition 3.2 (Return Ratio). As the return ratio indicates the expected revenue of the stock s, we define the return ratio as the ground-truth ranking score y^s_{t+1} :

$$y^s_{t+1} = \left(r^s_{t+1} - r^s_t\right)/r^s_t, \tag{1}$$

where r^s_t is the closing price at day t.

Definition 3.3 (Stock Recommendation). Given the stock relation graph G, stock historical sequences $\mathcal{X}_t = \{X^1_t, X^2_t, \ldots, X^S_t\}$, and stock description documents $D = \{d_1, d_2, \ldots, d_S\}$, our problem is to learn a function f which is able to get the $t+1$ day ranking list $\hat{\mathbf{y}}_{t+1} = \{\hat{y}^1_{t+1}, \hat{y}^2_{t+1}, \ldots, \hat{y}^S_{t+1}\}$ sorted by their ranking scores, and then top-1 stock is recommended to the investor. The mapping relation is represented as follows:

$$[\mathcal{X}_t, D, G] \xrightarrow{f} \hat{\mathbf{y}}_{t+1}. \tag{2}$$

4 METHODOLOGY

In this section, we introduce the detailed design of our model and Figure 2 illustrates its overall framework, including *Feature Extraction*, *TRAN*, and *Ranking Prediction*.

4.1 Overview

Our proposed stock recommendation model is based on stock ranking prediction, which can recommend the stocks with higher return ratios from a group of stocks. It relies on the sector-industry classifications of stocks to establish the stock relation graph that representing the relation among various stocks, and applies the graph-based learning method to capture the time-aware relation strengths between any two connected stocks.

The overall framework of the proposed stock recommendation is shown in Figure 2. It includes the following three components:

—*Feature Extraction*: The component takes the historical sequences and corporation description documents as input and extracts sequence features and document features with stock attributes for use in the subsequent network.

—*TRAN*: The component takes the sequence features and document features as input to the account for the temporal property and detailed relational feature of the stock. The time-aware relational attention mechanism is designed to obtain the time-aware relation strengths for further graph convolution operation, and the component outputs the representations with relational features for all stock nodes.

—*Ranking Prediction*: In order to strengthen the importance of stock itself, the component takes the sequence features and the representations with relational features as input to compute and sort the ranking scores of all stocks. Moreover, the ranking-based loss is backpropagated to the model when learning stock recommendation task.

4.2 Feature Extraction

Feature extraction module aims at obtaining features from the historical sequences and the description documents of stocks (corporations).

Sequence Encoder. The LSTM networks have been widely applied to process sequential data [23], which can boost the ability of RNNs to store longer term temporal information. Therefore, we also adopt the LSTM networks to encode the historical features of stocks over time. Given the historical sequence of stock s, $X_t^s = \{x_{t-T+1}^s, x_{t-T+2}^s, \ldots, x_t^s\}$, we input it into the LSTM networks. It is formulated as follows:

$$\begin{aligned} z_t^s &= tanh\left(W_z x_t^s + Q_z h_{t-1}^s + b_z\right), \\ i_t^s &= \sigma\left(W_i x_t^s + Q_i h_{t-1}^s + b_i\right), \\ g_t^s &= \sigma\left(W_g x_t^s + Q_g h_{t-1}^s + b_g\right), \\ c_t^s &= g_t^s \odot c_{t-1}^s + i_t \odot z_t^s, \\ o_t^s &= \sigma\left(W_o x_t^s + Q_o h_{t-1}^s + b_o\right). \\ h_t^s &= o_t^s \odot tanh(c_t^s), \end{aligned} \tag{3}$$

where $x_t^s \in \mathbb{R}^K$ is the original historical features; $tanh$ and σ are activation functions; $W_z, W_i, W_g, W_o \in \mathbb{R}^{U \times K}$, and $Q_z, Q_i, Q_g, Q_o \in \mathbb{R}^{U \times U}$ are learnable parameters and U is the number of hidden units; and $b_z, b_i, b_g, b_o \in \mathbb{R}^U$ are bias vectors. After recurrently processing the historical sequence, the encoder outputs the last hidden representation $h_t^s \in \mathbb{R}^U$ as the sequence feature.

Topic-based Information Extraction. We collect stocks (or the corresponding corporations) description documents to describe more detailed information, because the item description documents have been proved to valuable in the recommendation models [7, 35]. For stock recommendation task, stock description documents contain many factors (e.g., business, capital) that are helpful in describing more detailed relations. Inspired by the **Latent Dirichlet Allocation** (**LDA**) model [36, 37], we extract the topic distribution for each corporation description document as a document feature with stock attribute. For the stock s, the corresponding topics are obtained as follows:

(1) A word is defined to be an item from a vocabulary of size V represented by $\{w_1, w_2, \ldots, w_V\} \in \mathbb{R}^V$;
(2) A topic z_l, $l \in \{1, 2, \ldots, L\}$ is associated with a multinomial $\varphi_l = \{p_{l,1}, p_{l,2}, \ldots, p_{l,V}\}$ over the V words vocabulary, where $p_{l,v}$ refers to the probability that word w_v is generated from topic z_l;
(3) The document of stock s is a sequence of N_s words denoted by $d_s = \{w_1, w_2, \ldots, w_{N_s}\}$. Likewise, the document is associated with a multinomial $\theta_s = \{p_{s,1}, p_{s,2}, \ldots, p_{s,L}\}$ over L topics, where $p_{s,l}$ refers to the probability that topic z_l is generated from document d_s.

Therefore, in our stock recommendation model, the description documents of all stocks D are encoded as the features of topic distributions $Z = \{Z_1, Z_2, \ldots, Z_S\} \in \mathbb{R}^{L \times S}$.

4.3 Time-Aware Relational Attention Network

This module contains two core units: **Time-aware Relational Attention** (**TRA**) unit and **Graph Convolution (GC)** unit. TRA unit captures time-aware stock relation strength, and GC unit updates the stock representation based on stock relation graph.

TRA unit. In the real-world, the change of the stock price of a target corporation would be affected by corporations that are related, and the relation strength changes over time. Therefore, it is natural to believe that the descriptions of the time-aware relational strengths among multiple stocks are more accurate, and the performance of a stock recommendation task may be better. In order to describe more detailed relation and capture time-aware stock relation strength, we propose TRA unit. Firstly, we interact dynamically the document encoding Z_s and historical sequence encoding h_t^s to account for the temporal property and detailed relational feature of the stock. For stock s at the day t, we define the *Dynamic Interaction* function as

$$e_t^s = \psi\left(W_{f1}^T\left[h_t^s; W_{f2}Z_s\right] + b_f\right), \tag{4}$$

where ψ is an activation function; W_{f1}^T denotes the transpose of W_{f1}; $W_{f1} \in \mathbb{R}^{2U\times U}$ and $W_{f2} \in \mathbb{R}^{U\times L}$ are the parameter matrixs of a fully connected layer; and $b_f \in \mathbb{R}^U$ is the learnable bias. We obtain all S stocks latent embedding $E_t = \{e_t^1, e_t^2, \ldots, e_t^S\} \in \mathbb{R}^{S\times U}$.

Since, the strength of the relation between the stocks changes over time, we design a TRA unit to get time-aware relation strength from the dynamic interaction at each time step. Then, the relation strength is treated as edge weight η in the stock relation graph. Formally, the time-aware relation strength between stock s and stock j at each day t is computed by

$$\eta_t^{sj} = \frac{\exp(\alpha_t^s(j))}{\sum_{k\in\mathcal{N}_s}\exp(\alpha_t^s(k))}, \tag{5}$$

$$\alpha_t^s(j) = u_a^T\phi(W_a e_t^j + b_a), \tag{6}$$

where $\alpha_t^s(j)$ indicates the importance of stock node j's features to stock node s; $j \in \mathcal{N}_s$ and $\mathcal{N}_s$ is neighborhood of stock node s; ϕ is an activation function; $u_a \in \mathbb{R}^{M'}$, $W_a \in \mathbb{R}^{M'\times U}$, and b_a are parameters to be learned.

GC unit. In order to describe each stock node more comprehensively, the features of the node itself and the structural information are needed. So, we apply the graph-based learning method to aggregate the features of neighbor stock nodes with time-aware relation strength. General GCN model [29] cannot capture the time-aware relation strength, the adjacency matrix is fixed in GCN model. Therefore, we employ the graph convolution unit to update the node representation in the stock relation graph with time-aware edge weight. For stock s, we will aggregate the features of its all neighbor stock nodes in the graph to get the relational feature representation $a_t^s \in \mathbb{R}^U$ at each day t. The graph convolution operation is formulated as follows:

$$a_t^s = \sum_{j\in\mathcal{N}_s}\frac{\eta_t^{sj}}{degree(j)}h_t^j, \tag{7}$$

where $degree(j)$ is the degree of stock node j.

4.4 Ranking Prediction

This module calculates ranking score for each stock and outputs the ranking list of all stocks. The output a_t^s of TRAN is the updated representation of stock node. Instead of directly making prediction from a_t^s, we concatenate a_t^s with historical embedding h_t^s as the final representation m_t^s, which is conducive to strengthening the importance of stock s itself:

$$m_t^s = \left[a_t^s; h_t^s\right]^T, \tag{8}$$

where $m_t^s \in \mathbb{R}^{2U}$. Then, we deploy a full connection layer as the predictive function to calculate the ranking score $\hat{y}_{t+1}^s$:

$$\hat{y}_{t+1}^s = LeakyReLU\left(W_p^T m_t^s + b_p\right), \tag{9}$$

where $W_p \in \mathbb{R}^{2U}$ and b_p are the learnable parameters. Finally, we can get the ranking list for all S stocks $\mathbf{\hat{y}_{t+1}} = \{\hat{y}_{t+1}^1, \hat{y}_{t+1}^2, \ldots, \hat{y}_{t+1}^S\} \in \mathbb{R}^S$.

4.5 Objective Function

Generally, previous stock prediction methods treat different stocks as independent sequences. In order to meet the needs of stock recommendation, we should consider the relative comparison between stocks to select the stock with maximum return ratio. Therefore, we use a combination of pointwise regression loss and pairwise ranking-aware loss to optimize our model. Firstly, we select the square error loss between the ground-truth and predicted return ratio to account for the accuracy of return ratio prediction for all stocks. Then, we apply the pairwise ranking-aware loss to calculate the relative ranking error for every stock pair. Finally, the combined loss for both functions is backpropagated to the model when learning recommendation task, which is written as follows:

$$L(\mathbf{\hat{y}_{t+1}}, \mathbf{y_{t+1}}) = \|\mathbf{\hat{y}_{t+1}} - \mathbf{y_{t+1}}\|^2 + \alpha \sum_i \sum_j max\left\{0, -\left(\hat{y}_{t+1}^i - \hat{y}_{t+1}^j\right)\left(y_{t+1}^i - y_{t+1}^j\right)\right\}, \tag{10}$$

where $\mathbf{\hat{y}_{t+1}} = \{\hat{y}_{t+1}^1, \hat{y}_{t+1}^2, \ldots, \hat{y}_{t+1}^S\} \in \mathbb{R}^S$ and $\mathbf{y_{t+1}} = \{y_{t+1}^1, y_{t+1}^2, \ldots, y_{t+1}^S\} \in \mathbb{R}^S$ denote the predicted ranking list and the ground-truth ranking list for all stocks at day $t+1$, respectively; $\hat{y}_{t+1}^i$ and y_{t+1}^i denote the predicted and the ground-truth ranking score of stock i at day $t+1$, respectively; and α is a hyperparameter to balance the two loss terms.

5 EXPERIMENTAL SETUP

In this section, we introduce the details of the datasets, the baselines for our experiments and the corresponding experimental setup.

5.1 Datasets

We use stock historical sequence data, stock description document, and stock relation graph in our experiments, and the datasets are published on GitHub.[2]Table 1 shows the detailed statistics of these datasets.

Stock Historical Sequence. We use the stocks from NASDAQ and **New York Stock Exchange** (**NYSE**) stock markets that have historical sequences between 01/02/2013 and 12/08/2017, including 1,026 and 1,737 stocks, respectively [1]. In detail, the entire dataset of stock historical sequence is divided into the training set (from 01/02/2013 to 12/31/2015), the validation set (from 01/04/2016 to 12/30/2016) and the testing set (from 01/03/2017 to 12/08/2017). The lengths are 756, 252, and 237, respectively.

Stock Description Document. We collect the description documents of the stocks and their corresponding companies from Yahoo Finance.[3] Totally, 5,130 and 5,955 description documents

[2]https://github.com/xiaoting135/TRAN.
[3]https://finance.yahoo.com.

Table 1. Statistics of the Datasets

Datasets		NASDAQ	NYSE
Stock Historical Sequence	#Training Days	756	756
	#Validation Days	252	252
	#Testing Days	237	237
Stock Description Document	#Descriptions	5130	5955
	#Words	319297	427932
Stock Relation Graph	#Nodes	1026	1737
	#Edges	52586	98065

are collected in NASDAQ and NYSE, respectively. These description documents are then applied to train the topic-based extraction model.

Stock Relation Graph. For stock historical sequence data, there are 112 and 130 kinds of sector-industry classifications in the NASDAQ and NYSE markets [1], which are used to construct stock relation graph. Table 1 shows the summary statistics for stock relation graph.

5.2 Baselines

We compare our proposed TRAN with the following baselines, which forms the comprehensive set of regression-based and ranking-based stock recommendation methods as follows:

- **LSTM**: It is the vanilla LSTM, which obtains the sequential embedding from historical data and then a full connection layer is used to predict the return ratio. Additionally, LSTM is optimized by the regression loss [38].
- **SFM**: Inspired by discrete fourier transform, SFM regression network can learn trading patterns via discovering multi-frequency patterns to make the prediction [10].
- **Rank_LSTM**: Rank_LSTM treats stock prediction as a return ratio ranking task based on the LSTM network, which only utilizes the historical sequence data [1].
- **GCN**: GCN is a popular graph-based learning method recently [29]. We replace TRAN of our proposed model with a common GCN layer.
- **GAT**: **Graph Attention Network** (**GAT**) is the attention-based graph learning model, which leverages masked self-attentional layers to graph convolutions [39]. We replace TRAN of our proposed model with a GAT layer.
- **Relational Stock Ranking_Explicit (RSR_E)**: RSR_E is a stock ranking prediction model based on stock relations, which adds relational embedding layer to Rank_LSTM. The relational embedding layer is used explicit modeling to capture relation from sector-industry classifications [1].
- **Relational Stock Ranking_Implicit (RSR_I)**: RSR_I is a variant of RSR_E through replacing explicit modeling with implicit modeling in the relational embedding layer [1].
- **TRAN (with Doc2vec)**: Doc2vec [40] is a widely used method in text mining. To explore the relational influence of information extracted from description documents, we replace the topic-based model with the Doc2vec in the text encoder of our model.

5.3 Experimental Setup

Market Simulation. Similar to [41], we adopt a market simulation strategy to evaluate the performance through calculating the cumulative return ratio on testing period (from 01/03/2017 to 12/08/2017). At first, we assume that the investor spends the same amount of money on every trading day, and the stock market is liquid enough to satisfy investor's trading requests. Additionally, the cumulative return ratio does not take into account the transaction cost and buy/sell spread. Based on the assumptions, the investor applies recommendation model to get a ranking list with predicted return ratio of each stock at trading day t. The investor buys the top-1 stock at the closing price on day t and sells it at the closing price on trading day $t + 1$.

Evaluation Metrics. In order to evaluate the accuracy and stability, we adopt three metrics as follows:

—**MSE**: MSE evaluates the volatility between the ground-truth and predicted ranking scores over all stocks on every trading day. MSE is formally defined below:

$$MSE = \frac{1}{Q}\sum_{q=1}^{Q}\left\|\hat{\mathbf{y}}_{\mathbf{q}} - \mathbf{y}_{\mathbf{q}}\right\|^2, \tag{11}$$

where Q is the number of testing days, $\hat{\mathbf{y}}_{\mathbf{q}} = \{\hat{y}_q^1, \hat{y}_q^2, \ldots, \hat{y}_q^S\}$ and $\mathbf{y}_{\mathbf{q}} = \{y_q^1, y_q^2, \ldots, y_q^S\}$ are the predicted ranking list and the ground-truth ranking list for all stocks on testing day q, respectively; $\hat{y}_q^i$ and y_q^i denote the predicted and the ground-truth ranking score of stock i on testing day q, respectively.

—**Mean Reciprocal Rank (MRR)**: MRR is a widely used metric for ranking performance evaluation, which evaluates the predicted rank of the top-1 return ratio stock in the ground-truth. MRR is defined as

$$MRR = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{rank_q}, \tag{12}$$

where $rank_q$ is the real $rank$ of the predicted top-1 stock in the ground-truth on the qth testing day.

—**Investment Return Ratio (IRR)**: IRR is our main metric, which is the sum of return ratio of every testing day based on the market simulation strategy. IRR is defined by

$$IRR = \sum_{q=1}^{Q} R_q, \tag{13}$$

where R_q is the real return ratio of the predicted top-1 stock in the ground-truth on the qth testing day. Note the better performance is smaller value of MSE (≥ 0) and larger value of MRR ([0,1]) and IRR.

Parameter Settings. As the scales of prices vary among different stocks, we normalize the historical sequence feature of each stock to the range [-1,1]. Our proposed model is implemented with TensorFlow and optimized by Adam with a learning rate of 0.001. Depend on perplexity results for LDA, the topic numbers of NASDAQ and NYSE are 50 and 60, respectively. For the length of sequential input T and the number of hidden units U in LSTM, we select them via grid-search within the ranges of [2, 4, 8, 16] and [16, 32, 64, 128]. And , we tune α in loss function within [0.1, 1, 5, 10]. We report the mean testing performance when our proposed model and other methods perform best on the validation set over 10 different runs.

6 EXPERIMENTAL RESULT

We analyze the experimental results from different aspects to verify the effectiveness of our proposed method, including the overall performance, the information extraction model effect, the sector-wise performance, the different market simulation strategies, time complexity analysis, and sensitivity analysis.

Table 2. Experimental Results of Different Models

Method	NASDAQ			NYSE		
	MSE	MRR	IRR	MSE	MRR	IRR
LSTM	3.81e-4±2.20e-6	3.64e-2±1.04e-2	0.13±0.62	2.31e-4±1.43e-6	2.75e-2±1.09e-2	-0.90±0.73
SFM	5.20e-4±5.77e-5	2.33e-2±1.07e-2	-0.25±0.52	3.81e-4±9.30e-5	4.82e-2±4.95e-3	0.49±0.47
Rank_LSTM	3.79e-4±1.11e-6	**4.17e-2±7.50e-3**	0.68±0.60	2.28e-4±1.16e-6	3.79e-2±8.82e-3	0.56±0.68
GCN	3.80e-4±2.66e-6	3.66e-2±6.69e-3	0.39±0.49	2.27e-4±1.12e-6	4.07e-2±7.54e-3	0.82±0.71
GAT	3.79e-4±1.46e-6	3.51e-2±8.01e-3	0.61±0.55	2.28e-4±1.24e-6	4.05e-2±7.66e-3	1.04±0.37
RSR_E	3.82e-4±2.96e-6	3.16e-2±3.45e-3	0.20±0.22	2.29e-4±2.77e-6	4.28e-2±6.18e-3	1.00±0.58
RSR_I	3.80e-4±7.90e-7	3.17e-2±5.09e-3	0.23±0.27	2.26e-4±5.30e-7	4.51e-2±2.41e-3	1.06±0.27
TRAN (with Doc2vec)	3.80e-4±2.47e-6	3.67e-2±6.93e-3	0.73±0.35	2.26e-4±1.31e-6	4.55e-2±7.73e-3	1.17±0.40
TRAN	**3.79e-4±3.90e-7**	3.81e-2±4.37e-3	**0.92±0.25**	**2.26e-4±2.30e-7**	**4.91e-2±4.82e-3**	**1.38±0.35**

Bold values are the best results of the evaluation metrics in the table.

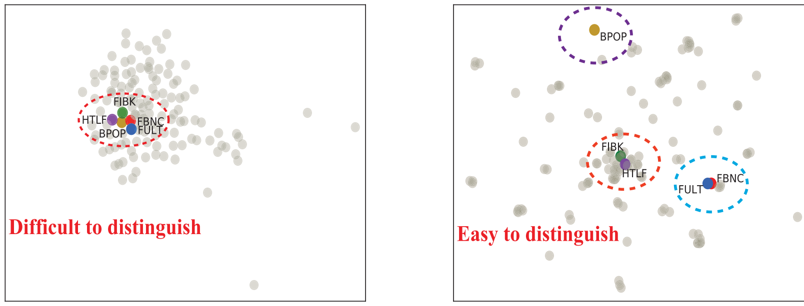
6.1 Overall Performance

As shown in Table 2, our proposed TRAN achieves better results in both NASDAQ and NYSE markets, comparing to other state-of-the-art models, including regression-based models (SFM and LSTM) and ranking-based models (Rank_LSTM, GCN, GAT, RSR_E, and RSR_I). It can be seen that Rank_LSTM outperforms both SFM and LSTM on the two markets with great improvement. This result shows that the ranking-based models can achieve better results than the regression-based models in stock recommendation task, which has been proven in previous studies [1]. At the same time, this result also indicates the strength of *Ranking Prediction* component based on the ranking-based objective function.

As the most important metric in stock recommendation, the highest IRR of our model presents great improvement than all baselines in the markets. For example, the IRR values are 0.92 and 1.38 for our model in NASDAQ and NYSE markets, respectively. This result verifies the advantage and practicability of the proposed stock recommendation model. Meanwhile, the performance of IRR varies greatly under different runs of a method. It is reasonable since the daily return ratios of all stocks are different in our datasets, which shows that as long as there is a slight change in the stock ranking, the IRR value will change greatly. Additionally, the stability of our model is also verified by the MSE metric. In both NASDAQ and NYSE markets, the MSE values of TRAN are the smallest compared with other baselines. For the metric of MRR, our model also achieves the best result in NYSE market.

Impact of Stock Relation Graph. In the proposed TRAN, stock relation graph is used to formulate the relations among multiple stocks. Rank_LSTM method predicts stock without considering the relations. Therefore, the impact of stock relation graph can be observed by comparing to the Rank_LSTM method. The results of our model prove the effectiveness of considering stock relations in stock recommendation, especially in the NYSE market. However, our method fails to beat Rank_LSTM regarding MRR in the NASDAQ market. There are two reasons about worse MRR. The one reason could be attributed to the instability of stock relations. The relation between stocks with long-term correlation can be encoded more efficiently. The NYSE market is considered to have more well-established companies and more quality stocks than the NASDAQ market [42]. According to the above reason, the stock relations are more effective in the NYSE market. Another reason is that the stock ranking predict model needs to minimize the pointwise regression loss and pairwise ranking-aware loss, which would lead to a tradeoff between accurately predicting absolute value of return ratios and their relative order [1].

Impact of Time-aware Relational Attention Network. For the first time, we design time-aware relational attention mechanism for stock recommendation. From the overall perspective, the comparison between Rank_LSTM method and our model can also demonstrate the strength



(a) Doc2vec (b) Topic Model

Fig. 3. Two-dimensional projection of stock node embedding learned for stock description document in the same sector-industry (Finance: Major Banks) of the NASDAQ market. (a) and (b) are stock node embeddings by Doc2vec model and topic-based model, respectively, where five stocks are represented in five different colors and marked with their stock codes next to them. The five stocks cluster together in the Doc2vec model, which shows Doc2vec is difficult to distinguish more detailed relations among stocks. But a topic-based model can distinguish the potential relations from multiple stocks by separating these stocks into different groups.

of TRAN component. In order to explain the advantages of TRAN in detail, we compare with other graph-based learning methods. GCN method takes stock relations into consideration, but ignoring the different relation strengths among multiple stocks and the temporal properties of stock relations. Although, GAT only captures the relation attention strength without the temporal properties, and RSR_I and RSR_E simply utilize the relation weight obtained from the historical sequence, they still perform better than GCN. These results indicate that the descriptions of the time-aware relational strengths among multiple stocks are more accurate, and the performance of stock recommendation task is better. Meanwhile, our model outperforms GCN, GAT, RSR_I, and RSR_E in terms of all metrics on two datasets. This result proves the effectiveness of our proposed TRAN, that is, TRAN can obtain the property of detailed time-aware relation from the attributes of stocks and the dynamic trend of historical sequence, and capture time-aware relation strength to aggregate sequence feature.

6.2 Information Extraction Model Effect

In order to evaluate the effect of information extraction model in our stock recommendation method, we replace topic-based model with Doc2vec model to extract information from stock description documents in NASDAQ and NYSE markets. As shown in Table 2, the result of our TRAN is better than TRAN with Doc2vec model, which indicates the relational information extracted from topic-based model outperforms Doc2vec model. To demonstrate it, we use the t-Distributed **Stochastic Neighbor Embedding** (t-**SNE**) to project stock node embeddings learned from stock description documents to two-dimensional space. Figure 3 shows the stock node embeddings that belong to the same sector-industry (*Finance: Major Banks*) in the NASDAQ market from two models. Take five stocks that colored as examples, they cluster together from Doc2vec model in Figure 3(a), but they are separated into three groups by topic model in Figure 3(b). This experiment reveals two interesting phenomena as follows:

—Topic-based model can distinguish these stocks into several groups to provide more detailed stock relations in the same sector-industry, which plays a positive role in dynamic interaction of TRAN. For Doc2vec model, it can identify companies in the same sector or industry

Table 3. Performance of TRAN on Ranking Stocks in Different Sectors with Respect to IRR, Where Ranking is the Ranking of IRR Value in All Sectors and All Effective Sectors in Both Markets Are 12

Market	Sector	#Stocks	IRR	Ranking
NASDAQ	Finance	222	0.15	4
	Technology	158	0.60	1
	Consumer Services	116	0.35	2
	Health Care	95	-0.11	11
	Capital Goods	90	0.16	3
NYSE	Consumer Services	260	1.20	1
	Finance	198	0.26	2
	Capital Goods	167	0.05	6
	Energy	108	0.08	5
	Public Utilities	99	0.11	4

actually, but the stock relation graph has roughly described this relation. Thus, Doc2vec model is difficult to distinguish more detailed relations among stocks in the same sector-industry.

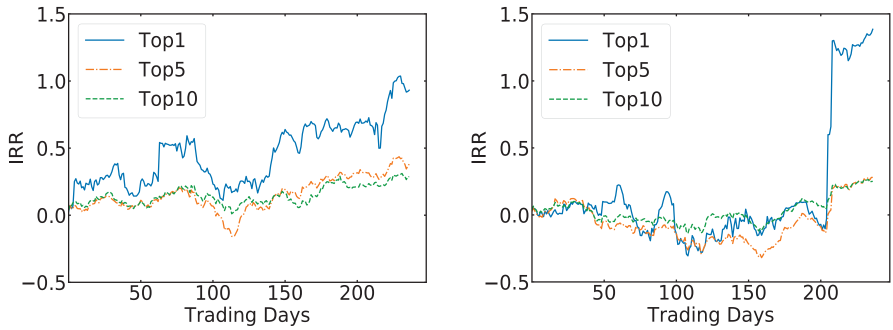
—Some words related to the corporation's products and business appear frequently in the stock description documents. Topic-based model represents topics of the document by word probabilities, and then it is utilized for description document encoding through capturing the topic distributions. Therefore, topic-based model is more suitable for mining stock relations than Doc2vec in our model.

6.3 Study on Sector-Wise Performance

In order to study whether the performance is sensitive to sectors, we apply our model to evaluate its performance over the stocks in each sector, i.e., separately counting the IRR value for each sector in the two markets. Note that the number of effective sectors in two market datasets is 12. We only list the performance on effective sectors with the top-5 stocks and present their IRR rankings in all sectors, which is shown in Table 3. We can see that the IRR result and ranking perform well on sectors with the top-5 stocks. For example, the top-5 effective sectors in NYSE market are Consumer Services, Finance, Capital Goods, Energy and Public Utilities, and their IRR rankings are 1, 2, 6, 5, and 4, respectively. This result further indicates the effectiveness of considering sector-industry relation on the stock market. Additionally, we can find that the performance in NYSE market is better than NASDAQ market. In NASDAQ market, the sector of Health Care ranks 11 in the IRR ranking. This phenomenon also proves once again that the stock relation is unstable in NASDAQ market, which is coherent with the results in Table 2 and the analysis in Impact of Stock Relation Graph of Section 6.1. Going further, it also means our model can consider the stocks in each single sector separately.

6.4 Study on Different Market Simulation Strategies

In reality, investors always choose multiple stocks to avoid risks. Therefore, we study the performance of our proposed TRAN under three different markets simulation strategies, named *Top1*, *Top5*, and *Top10*, buying stocks with top-1, top-5, and top-10 highest expected return ratio, respectively. For instance, with the market simulation strategy of *Top10*, we equally split the budget to trade the top-10 ranked stocks and calculate the IRR value by summing the mean return ratio



(a) NASDAQ

(b) NYSE

Fig. 4. Comparison on market simulation strategies (Top1, Top 5, and Top10) about IRR based on prediction of TRAN.

Table 4. Performance of TRAN as Compared with Ideal Investment Strategies in Two Markets

Method	NASDAQ			NYSE		
	Top1	Top5	Top10	Top1	Top5	Top10
Market	**3.40**	**2.36**	**1.99**	**2.42**	**1.90**	**1.47**
Selected	1.63	0.81	0.27	2.24	1.78	1.39
TRAN	0.92	0.24	0.21	1.38	0.50	0.37

Bold values are the best results of the evaluation metrics in the table.

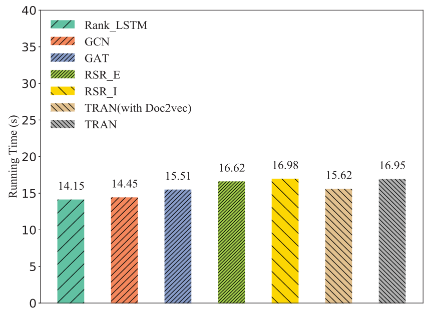
of the 10 selected stocks on each testing day. The performance comparison of these strategies with the predictions of TRAN is shown in Figure 4. From the figure, we can observe the following phenomena:

—In general, the highest return ratios should be obtained by selecting the most top stocks. Therefore, the *Top1* strategy achieves the highest return ratio of the range of almost one year. However, the volatility of the *Top1* strategy is greater than that of the *Top5* and *Top10* strategies. It is reasonable to suppose that selecting only top-1 stock from more than 1,000 is a highly risk strategy.

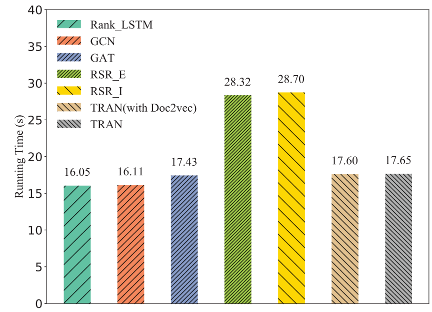
—The IRR performance on most testing days follows the order of *Top1*>*Top5*>*Top10*. The reason may be that the ranking algorithm can accurately rank the relative order of future return ratios [1]. Once the order of stock ranking is accurate, we can buy and sell stocks with higher expected return ratios (such as the top-1 stock) to achieve the higher cumulative return ratios according to the market simulation strategy.

Additionally, in order to better evaluate the achieved IRR performance, we compare two more ideal investment strategies in two markets; (1) **Market**: We select the stocks with highest return ratios (*Top1*, *Top5,* and *Top10*) during the testing period from the whole market of the real-world. (2) **Selected**: We select the stocks with highest return ratio (*Top1*, *Top5,* and *Top10*) during the testing period from two datasets of our methods. Table 4 shows the IRR result of the compared investment strategies. From the table, we have the following observations:

—When trading the same number of stocks (*Top1*, *Top5,* and *Top10*), our proposed model performs significantly worse than the ideal investment strategies. However, this result is acceptable because it is not easy to accurately select the stocks with highest return ratio in



(a) NASDAQ



(b) NYSE

Fig. 5. The running time comparison of different ranking-based models on two markets.

the range of almost one year. Meanwhile, it can also reflect that there is a lot of space for improvement in the stock recommendation task.

—In NYSE market, the *Top1* of our method achieves the IRR value comparable to the investment strategy, Market and Selected under *Top10*. In NASDAQ market, we find the *Top1* setting of our method is better than the *Top5* of the investment strategy Selected. These results show the competitivity of our method, which can further prove the effectiveness of TRAN in the stock recommendation task.

6.5 Time Complexity Analysis

The major time complexity of our stock recommendation model depends on the time complexity of LSTM and TRAN. The time complexity of LSTM is $O\left(l \cdot n^2\right)$ where l is the length of sequence and n is the number of hidden units in LSTM. The time complexity of TRAN in our model is divided into two parts. One part is the TRA unit and its time complexity is $O\left(|V|^2 \cdot d\right)$, and the other is GC unit and its time complexity is $O(|V|FF' + |E|F')$, where $|V|$ and $|E|$ are the number of nodes (stocks) and edges in the stock relation graph, respectively; d is the dimensionalities of input features in TRA unit and $|V| \gg d$; F and F' are the dimensionalities of input and output features in GC unit, respectively. Additionally, the operation of the TRA unit can be parallelized across all edges, and the computation of output features can be parallelized across all nodes. It does not require eigendecompositions or similar costly matrix operations. Therefore, the time complexity of TRAN would be $O(|V|FF' + |E|F')$, which is on par with GCN and GAT layers.

To compare the computational efficiency of the ranking-based stock recommendation model, we set same batch size and parameter settings for all methods and run them on a same GPU device, and record the running time (measured in seconds wall-clock time) per epoch of these methods on NASDAQ and NYSE stock markets, respectively. As shown in Figure 5, we have the following observations:

—The running time of Rank_LSTM is 14.15s and 16.05s in NASDAQ and NYSE markets, which is not much faster than the running time of other models that based on Rank_LSTM and adding graph-learning (e.g., GCN and GAT). This result shows that the running time of Rank_LSTM accounts for a large proportion for other models with graph-learning. Additionally, our proposed model spends similar time to graph-learning baseline models (e.g., GCN and GAT) on two markets, which proves the TRAN component in our model does have similar efficiency to GCN and GAT layers.

—Due to the larger scale of the stock relation graph constructed on NYSE market, the running time on NYSE market is longer than that on NASDAQ market for all models. For example, the

Table 5. The Average IRR Values of TRAN for the Prediction of Different Sequential Length Input T in Two Markets

Market	T=2	T=4	T=8	T=16
NASDAQ	0.12	0.52	**0.92**	0.54
NYSE	0.48	1.19	**1.38**	0.66

Bold values are the best results of the evaluation metrics in the table.

Table 6. The Average IRR Values and Standard Deviations of TRAN for 1-step, 3-step and 5-step Prediction in Two Markets

Market	1-step	3-step	5-step
NASDAQ	**0.92±0.25**	0.64±0.99	0.58±0.97
NYSE	**1.38±0.35**	0.89±0.92	0.86±1.46

Bold values are the best results of the evaluation metrics in the table.

running times of our proposed TRAN (with Doc2vec) and TRAN have increased by 12.68% and 4%, respectively. For the RSR_E and RSR_I methods, which also take the temporal properties of stock relations into account, their running time has increased by 70.40% and 69.02%, respectively. This result shows that our proposed TRAN is more scalable than RSR_E and RSR_I models that also consider the temporal properties of stock relations.

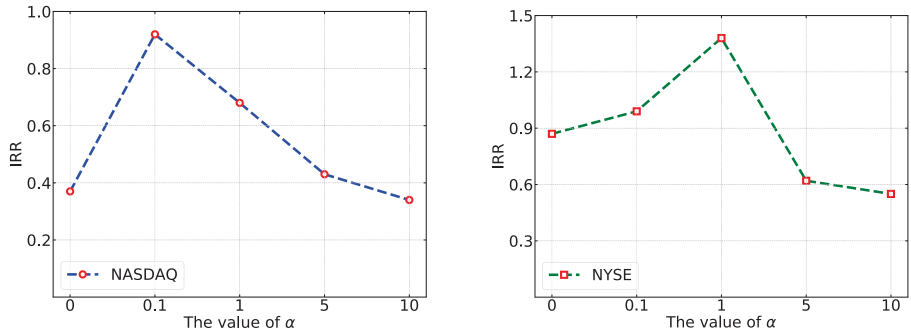
6.6 Sensitivity Analysis

We change several parameters in our model to analyze the sensitive of our model, including the different sequential length input T, the prediction of step n, and the α in objective function.

Impact of Different Sequential Length Input T**.** To explore how well our model performs with the different sequential length input T, we set T is 2, 4, 8, and 16, respectively. Table 5 summarizes the performance of comparison on different sequential length input T with respect to the average IRR values.

We can see that a change in the sequential length input T can affect the IRR performance of our model. The IRR values of our proposed TRAN become better as the sequential length input T ($T \leq 8$) increases for each market. Specifically, our model achieves best IRR results 0.92 and 1.38 in NASDAQ and NYSE markets when T is 8, respectively. However, the IRR performance becomes worse when T increases again, that is, the IRR values are 0.54 and 0.66 in NASDAQ and NYSE markets when T increased to 16. The reason could be attributed to different amounts of information contained in different sequential lengths. At first, the shorter sequential length may contain insufficient information. With the increase of T value, there will be more information about helpful price information to be embedded in the sequence feature. However, when T reaches a large value, the longer sequential length contains the redundant information, some useless information will be embedded in the sequence feature as well. Thus, the IRR value increases firstly and then decreases.

Impact of n**-step Prediction.** We set n-step prediction is predicting the $t + n$ day ranking list $\mathbf{\hat{y}_{t+n}}$ and recommending the top-1 stock to the investors at day t, where $\mathbf{\hat{y}_{t+n}} = \{\hat{y}^1_{t+n}, \hat{y}^2_{t+n}, \ldots, \hat{y}^S_{t+n}\}$ and $n \geq 1$. We choose the average IRR values and standard deviations over the 10 different runs as our evaluation metrics to compare the performance in the two markets. We compare the one-step, 3-step, and 5-step prediction results. Among them, the 1-step prediction is a



(a) NASDAQ (b) NYSE

Fig. 6. The IRR values by TRAN based on different α values (0, 0.1, 1, 5, and 10) in the two markets. It verifies the effect of α in balancing the pointwise regression loss and pairwise ranking-aware loss.

short-term prediction, which predicts the stock ranking list of the next day. The 3-step prediction implies the half-week stock ranking list. With the stock market only opening on weekdays, the 5-step prediction indicates the ranking list in the next week. In other words, the 5-step prediction is usually a challenging task since it typically covers one weekend. The comparison of the IRR results is reported in Table 6.

From Table 6, we can see that the results in NYSE market outperform that of the NASDAQ market. Moreover, the IRR performance becomes worse as the prediction step increases for each market, especially the standard deviations. These results indicate our proposed TRAN can perform well for short-term prediction, but with the increase of prediction step, the volatility of two markets return ratios are increasing.

Impact of the α in Objective Function. To verify the effect of different α in balancing the pointwise regression loss and pairwise ranking-aware loss, we run the experiments for five values of α (0, 0.1, 1, 5, and 10) in NASDAQ and NYSE markets, respectively. Figure 6 shows the IRR values by TRAN based on different α values in NASDAQ and NYSE markets. From the figure, we have following observation. The IRR value of TRAN with $\alpha = 0$ is 0.37 in NASDAQ market test set. With the increase of the α value to 0.1, the IRR value increases to 0.92. While when α values reach to 1, 5, and 10, the corresponding IRR values drop to 0.68, 0.43, and 0.34, respectively. In NYSE market test set, the trend in IRR values is similar. The IRR value is 0.87 with $\alpha = 0$, and the IRR value achieves the best 1.38 with the increase of the α value to 1. But α reaches to 5 and 10, the corresponding IRR values become worse. This is because of the following: (1) At first, with the increase of α value, the pairwise ranking-aware loss has a positive effect on the stock ranking of return ratio, and thus the IRR values increase. (2) However, when α reaches a large value, the pointwise regression loss is ignored, resulting in the decline of the accuracy of the return ratio, and further affecting the calculating of relative ranking error based on the return ratio. Thus, the IRR values decrease.

7 CONCLUSIONS

The relation between stocks is one of the most important factors for stock recommendation. It is difficult to utilize the relation because of their potential time-varying properties. In this article, we have proposed TRAN for graph-based stock recommendation based on return ratio ranking. Our model is able to capture time-aware relation strength from the interaction of historical sequence and description document. By constructing stock relation graph, graph convolution operation is applied to aggregate the features of the related stock nodes with dynamic attention. Our stock recommendation model can rank the stocks according to their expected return ratios. Experimental results on different aspects show that our stock recommendation model can achieve better performance. More specifically, TRAN achieves the best investment return ratio by 0.92 on the NASDAQ dataset and 1.38 on the NYSE dataset, respectively.

REFERENCES

[1] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T. S. Chua. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems* 37, 2 (2019), 27:1–27:30. DOI: http://dx.doi.org/10.1145/3309547

[2] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan, and H. Chen. 2019. Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In *Proceedings of the 27th World Wide Web Conference*. 678–685. DOI: http://dx.doi.org/10.1145/3308560.3317701

[3] Josif Grabocka, Nicolas Schilling, and Lars Schmidt-Thieme. 2016. Latent time-series motifs. *ACM Transactions on Knowledge Discovery from Data* 11, 1 (2016), 1–20. DOI: http://dx.doi.org/10.1145/2940329

[4] Mu-Yen Chen. 2014. A high-order fuzzy time series forecasting model for internet stock trading. *Future Generation Computer Systems* 37 (2014), 461–467. http://dx.doi.org/10.1016/j.future.2013.09.025

[5] Jason Lines, Sarah Taylor, and Anthony Bagnall. 2018. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data* 12, 5 (2018), 52:1–52:35. http://dx.doi.org/10.1145/3182382

[6] Sina Sajadmanesh, Sogol Bazargani, Jiawei Zhang, and Hamid R Rabiee. 2019. Continuous-time relationship prediction in dynamic heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data* 13, 4 (2019), 1–31. DOI: http://dx.doi.org/10.1145/3333028

[7] Hung-Hsuan Chen. 2018. Behavior2Vec: Generating distributed representations of users' behaviors on products for recommender systems. *ACM Transactions on Knowledge Discovery from Data* 12, 4 (2018), 1–20. DOI: http://dx.doi.org/10.1145/3184454

[8] Yu Lei and Wenjie Li. 2019. Interactive recommendation with user-specific deep reinforcement learning. *ACM Transactions on Knowledge Discovery from Data* 13, 6 (2019), 1–15. DOI: http://dx.doi.org/10.1145/3359554

[9] Huiting Liu, Huimin Liu, Qiang Ji, Peng Zhao, and Xindong Wu. 2020. Collaborative deep recommendation with global and local item correlations. *Neurocomputing* 385 (2020), 278–291. DOI: http://dx.doi.org/10.1016/j.neucom.2019.12.087

[10] L. Zhang, C. Aggarwal, and G. J. Qi. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2141–2149. DOI: http://dx.doi.org/10.1145/3097983.3098117

[11] Qikai Liu, Xiang Cheng, Sen Su, and Shuguang Zhu. 2018. Hierarchical complementary attention network for predicting stock price movements with news. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1603–1606. DOI: http://dx.doi.org/10.1145/3269206.3269286

[12] Y. Xu and S. B. Cohen. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 1970–1979. DOI: http://dx.doi.org/10.18653/v1/P18-1183

[13] Y. Chen, Z. Wei, and X. Huang. 2018. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *Proceedings of the ACM International Conference on Information and Knowledge Management*. 1655–1658. DOI: http://dx.doi.org/10.1145/3269206.3269269

[14] Burton G. Malkiel and Eugene F. Fama. 1970. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance* 25, 2 (1970), 383–417. DOI: http://dx.doi.org/10.2307/2325486

[15] Guang Liu and Xiaojie Wang. 2018. A numerical-based attention method for stock market prediction with dual information. *IEEE Access* 7 (2018), 7357–7367. DOI: http://dx.doi.org/10.1109/ACCESS.2018.2886367

[16] P. K. Agarwal, R. B. Avraham, H. Kaplan, and M. Sharir. 2014. Computing the discrete Fréchet distance in subquadratic time. *SIAM Journal on Computing* 43, 2 (2014), 429–449. DOI: http://dx.doi.org/10.1137/130920526

[17] Eugene F. Fama. 1965. The behavior of stock-market prices. *The Journal of Business* 38, 1 (1965), 34–105. DOI: http://dx.doi.org/10.1086/294743

[18] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 2327–2333.

[19] H. Xie, J. Bian, M. Wang, and H. Qiao. 2014. Is technical analysis informative in UK stock market? Evidence from decomposition-based vector autoregressive (DVAR) model. *Journal of Systems Science and Complexity* 27, 1 (2014), 144–156. DOI: http://dx.doi.org/10.1007/s11424-014-3280-9

[20] R. Bisoi and P. K. Dash. 2014. A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter. *Applied Soft Computing* 19 (2014), 41–56. DOI: http://dx.doi.org/10.1016/j.asoc.2014.01.039

[21] Chenglin Xiao, Weili Xia, and Jijiao Jiang. 2020. Stock price forecast based on combined model of ARI-MA-LS-SVM. *Neural Computing and Applications* 32, 10 (2020), 5379–5388. DOI: http://dx.doi.org/10.1007/s00521-019-04698-5

[22] Zhe Lin. 2018. Modelling and forecasting the stock market volatility of SSE composite index using GARCH models. *Future Generation Computer Systems* 79 (2018), 960–972. DOI: http://dx.doi.org/10.1016/j.future.2017.08.033

[23] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira. 2017. Stock market's price movement prediction with LSTM neural networks. In *Proceedings of the International Joint Conference on Neural Networks*. 1419–1426. DOI: http://dx.doi.org/10.1109/IJCNN.2017.7966019

[24] J. Wang, T. Sun, B. Liu, Y. Cao, and H. Zhu. 2019. CLVSA: A convolutional LSTM based variational sequence-to-sequence model with attention for predicting trends of financial markets. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3705–3711. DOI: http://dx.doi.org/10.24963/ijcai.2019/514

[25] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780. DOI: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[26] X. Ding, Y. Zhang, T. Liu, and J. Duan. 2015. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 2327–2333.

[27] Yanshuang Li, Xintian Zhuang, Jian Wang, and Weiping Zhang. 2020. Analysis of the impact of Sino-US trade friction on China's stock market based on complex networks. *The North American Journal of Economics and Finance* 52 (2020), 101185. DOI: http://dx.doi.org/10.1016/j.najef.2020.101185

[28] Yuqiang Fang, Ruili Wang, Bin Dai, and Xindong Wu. 2015. Graph-based learning via auto-grouped sparse regularization and kernelized extension. *IEEE Transactions on Knowledge and Data Engineering* 27, 1 (2015), 142–154. DOI: http://dx.doi.org/10.1109/TKDE.2014.2312322

[29] T. N. Kipf and M. Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*. 1–14.

[30] Xiaoting Ying, Cong Xu, Jianliang Gao, Jianxin Wang, and Zhao Li. 2020. Time-aware graph relational attention network for stock recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 2281–2284.

[31] Jiexia Ye, Juanjuan Zhao, Kejiang Ye, and Chengzhong Xu. 2020. Multi-view graph convolutional networks for relationship-driven stock prediction. arXiv:2005.04955. Retrieved from https://arxiv.org/abs/2005.04955.

[32] Thomas Fischer and Christopher Krauss. 2018. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research* 270, 2 (2018), 654–669. DOI: http://dx.doi.org/10.1016/j.ejor.2017.11.054

[33] C. Chen, L. Zhao, J. Bian, C. Xing, and T. Y. Liu. 2019. Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2376–2384. DOI: http://dx.doi.org/10.1145/3292500.3330663

[34] Tanawat Chiewhawan and Peerapon Vateekul. 2020. Stock return prediction using dual-stage attention model with stock relation inference. In *Proceedings of the 12th Asian Conference on Intelligent Information and Database Systems*. 492–503. DOI: http://dx.doi.org/10.1007/978-3-030-41964-6_42

[35] Wayne Xin Zhao, Jinpeng Wang, Yulan He, Ji-Rong Wen, Edward Y Chang, and Xiaoming Li. 2016. Mining product adopter information from online reviews for improving product recommendation. *ACM Transactions on Knowledge Discovery from Data* 10, 3 (2016), 1–23. DOI: http://dx.doi.org/10.1145/2842629

[36] D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3, Jan (2003), 993–1022.

[37] Ting Hua, Chang-Tien Lu, Jaegul Choo, and Chandan K. Reddy. 2020. Probabilistic topic modeling for comparative analysis of document collections. *ACM Transactions on Knowledge Discovery from Data* 14, 2 (2020), 1–27. DOI: http://dx.doi.org/10.1145/3369873

[38] Wei Bao, Jun Yue, and Yulei Rao. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS One* 12, 7 (2017), e0180944. DOI: http://dx.doi.org/10.1371/journal.pone.0180944

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*.

[40] Q. Le and T. Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*. 1188–1196.

[41] M. Dixon, D. Klabjan, and J. H. Bang. 2017. Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance* 6, 3-4 (2017), 67–77. DOI: http://dx.doi.org/10.3233/AF-170176
[42] G. W. Schwert. 2002. Stock volatility in the new millennium: How wacky is nasdaq? *Journal of Monetary Economics* 49, 1 (2002), 3–26. DOI: http://dx.doi.org/10.3233/AF-170176

Received July 2020; revised November 2020; accepted February 2021