Multi-Graph Convolutional Network for Relationship-Driven Stock Movement Prediction

Jiexia Ye, Juanjuan Zhao*, Kejiang Ye
Shenzhen Institutes of Advanced Technology
Chinese Academy of Sciences
Shenzhen, China
University of Chinese Academy of Sciences
Beijing, China
Email: {jx.ye,jj.zhao,kj.ye}@siat.ac.cn

Chengzhong Xu
State Key Lab of IOTSC, Department of Computer Science
University of Macau
Macau SAR, China
Email: {czxu}@um.edu.mo

***Abstract*—Stock price movement prediction is commonly accepted as a very challenging task due to the volatile nature of financial markets. Previous works typically predict the stock price mainly based on its own information, neglecting the cross effect among involved stocks. However, it is well known that an individual stock price is correlated with prices of other stocks in complex ways. To take the cross effect into consideration, we propose a deep learning framework, called Multi-GCGRU, which comprises graph convolutional network (GCN) and gated recurrent units (GRU) to predict stock movement. Specifically, we first encode multiple relationships among stocks into graphs based on financial domain knowledge and utilize GCN to extract the cross effect based on these pre-defined graphs. To further get rid of prior knowledge, we explore an adaptive relationship learned by data automatically. The cross-correlation features produced by GCN is concatenated with historical records and fed into GRU to model the temporal dependency of stock prices. Experiments on two stock indexes in China market show that our model outperforms other baselines. Note that our model is rather feasible to incorporate more effective stock relationships containing expert knowledge as well as learn relationship on the basis of data dynamically.**

***Index Terms*—GCN, Graph Convolutional Network, GRU, Relationship, Stock Movement, Stock Price.**

I. Introduction

Predicting the future status of stocks has always been of great interest by many investors for that a little improvement of prediction accuracy might yield a huge enormous gain. Both traditional finance and modern behavior finance believe that fluctuations of stock prices are information-driven. Information affects the beliefs and behaviors of investors, thus changing stock movement. Therefore, understanding how stock markets impound information into stock prices is paramount in stock prediction [10].

Recently, researches make substantial effort on modeling correlations between various information and stock prices by machine learning [1], [2], [3], [4] or deep learning [5], [6], [7], [8]. However, the core assumption of algorithms in most works is that stocks are independent of each other. They mainly focus on extracting the autocorrelation of an individual stock from its own historical information but neglect the cross effect of stocks over time, which is an important aspect of stock prices dynamics [9].

The cross interaction among stocks is attributed to various connections among corporations, such as the shared industry information [10], supply chain, payments network [11], business partnership and shareholder ownership [12]. These complex and multifaceted connections result in that the price of an individual stock is highly correlated with other stocks besides its own information. For instance, considerable empirical findings have confirmed a pronounced lead-lag pattern in stock prices, i.e., some stock prices lead or lag other stock prices [9], [10]. Therefore, it is natural to take the corporation relationships into consideration for better stock prediction.

However, there are three major challenges in utilizing these relationships: (1) design appropriate representation for corporation relationships, (2) find a model without independent instance assumption to extract the cross-correlation among stocks, (3) predict the target stock movement by jointly considering its historical observation and the cross-correlation with related stocks.

To address the first challenge, we follow previous works [11], [12] to embed the corporation relationships into graphs. In each graph, a node represents a listed company and the edge represents interaction between two listed companies. Besides inheriting the shareholding graph in [12], we novelly define an industry graph based on lead-lag theory [9] and a topicality graph based on common topical news impact [13]. In addition, we realize that these artificial relational graphs depend on solid financial knowledge and require more financial data. To overcome such limitations, we establish a dynamic graph on the basis of data. We analyze the effectiveness of these graphs and make necessary comparison.

After establishing relational graphs among stocks, we propose a deep learning framework called Multi-GCGRU by jointly combining Graph Convolutional Network (GCN) and Gated Recurrent Units (GRU) to tackle the second and the third challenges. GCN is a rapidly developed deep learning approach to handle the complexity of graph data. It has demonstrated its effectiveness in capturing interdependency between instances in a graph and has achieved state-of-the-

*Corresponding author: Juanjuan Zhao

art performance in many applications, such as molecular fingerprints [14], recommendation system [15] and traffic forecasting [16]. In this paper, we perform graph convolutions on the pre-defined graph structures to model various interactions among stocks. Since stock prediction is a time series task and GRU has been proved to be effective for processing sequential data [17], we utilize GRU to learn the temporal dependency from historical market data along with the cross-effect features produced by GCN. We test our Multi-GCGRU model on two real stock market indexes. The experimental results show that our model has better performance than baseline methods.

Specifically, our contributions can be summarized as follow.

(1) We take the cross effect among a collection of stocks into consideration for better stock prediction. We construct multiple graphs based on various corporation relationships to enrich the representations of cross effect. Further, to get rid of prior knowledge on financial domain, we explore a data-driven adaptive graph.

(2) We employ the Graph Convolutional Network (GCN) to process these graphs along with historical information to learn complicated interactions among related stocks, and produce new features for each individual stock in the collection which contains cross-impact information from other stocks.

(3) We concatenate the cross-effect features with historical market information at the same time slice for each stock. These combined features are fed into Gated Recurrent Units (GRU) to learn temporal pattern in stock prices.

(4) Note that our model can be easily extended to incorporate more effective relationships among stocks. Even without any pre-defined relationship, our model is able to learn a dynamic graph automatically from market price data.

II. Related Works

A. Stock price Prediction

Stock price prediction is a very challenge task due to the diverse and complicate factors, including corporate financial performance [18], industry information [10], public news [19], [13], social sentiment [20], [21]. Recently, various traditional machine learning and deep learning approaches have been proposed to extract valuable clues from different types of information sources for better stock prediction.

The input features of most works are mainly based on the historical market data (e.g., stock prices, trading volume). For instance, [7] utilized only historical prices to capture the multi-frequency trading patterns by a novel State Frequency Memory (SFM) recurrent network. [5] addressed the stochasticity of stock price variable to improve the generalization of prediction model by proposing an adversarial training solution. However, only historical prices data cannot entirely explain the volatility of stock price. Other types of information are complementary to enrich the input features and discover more concealed rules in stock price, such as public news [13], [19], texts from social medias [22], [23], web browsing data [24]. For example, [25] extracted events from news titles to model influence of events on stock price by a CNN-based framework. [23] presented a novel deep generative model to learn opinions from Twitter texts. [8] explored the mutual fund portfolio data to extract stock intrinsic properties for enhancing prediction. [26] integrated CEO's vocal features in a conference call into the model.

Despite tremendous efforts have been made to understand the principle of stock price movement, most of the works above mainly focused on combining a single stock's historical records with other textual information but overlooked the correlations among stocks. Only a few attempts [27] have been made to explore the cross effect among stocks which has been verified by [9]. In this paper, we pay attention to model the influences from other stocks on the target stock.

B. Graph Convolutional Networks

In the last couple of years, many attempts have been made to generalize neural networks on graph-structured data. Encouraged by the success of CNN, researchers have successfully re-defined the notion of convolution for graph data, called graph convolution. The corresponding neural network, i.e., graph convolution network (GCN) has gained much attention recently since it has demonstrated outstanding performance on a node classification task [28]. GCN takes the graph structure and node features as inputs and it can capture the complex interaction between nodes on graph by aggregating information from neighbors and doing non-linear transformation on features dimension. Such capacity enables GCN to achieve state-of-the-art performance in graph related applications [14], [15], [16]. In recent research, [12] applied GCN in stock prediction and they modeled the correlations among stocks based on a shareholding graph. However, such graph is rather limited due to the sparse cross-shareholdings among public corporations and it is insufficient to represent the complex correlations among stocks. We would compare our model with this method in experiments.

III. Problem Formulation

Stock price prediction can be divided into stock price return prediction [8] and stock movement prediction [27], [5], [22], where the former predicts the exact price of stock while the latter predicts the up or down of stock price. Due to the complexity and stochasticity of stock market, it is rather difficult to predict price return and stock movement is more achievable. Therefore, most works focus on stock movement prediction and so do we.

Usually, stock price movement aims to predict the movement of a target stock in a pre-selected stock collection on a target trading day with historical market information along with other information [23]. The mathematical formulation is as follow:

$$\hat{y}^{\mathbf{s}}_{\mathbf{d}} = f([x^{\mathbf{s}}_{\mathbf{d}-\mathbf{P}}, \cdots, x^{\mathbf{s}}_{\mathbf{d}-1}], \mathbf{E}; \Theta) \tag{1}$$

Where $\mathbf{s}$ is the target stock, $\mathbf{d}$ is the target day, $\mathbf{P}$ is the lag size, $x^{\mathbf{s}}_t \in \mathbb{R}^{\mathbf{F}}$ is the $\mathbf{F}$ historical features of the target stock at day t, $\mathbf{E}$ is external information, Θ is trainable parameter. $\hat{y}^{\mathbf{s}}_{\mathbf{d}} \in [0, 1]$ is the predicted probability at day $\mathbf{d}$.

However, such formulation treats each stock independently and overlooks its complex correlations with other related stocks. We encode various correlations among stocks as graphs and focus on exploring relationship-driven influence for stock prediction (as shown in Figure 1). Therefore, we re-formalize the problem as follow:

$$\hat{Y}_{\mathbf{d}} = f([X_{\mathbf{d-P}}, \cdots, X_{\mathbf{d-1}}], \mathbf{G}; \Theta) \tag{2}$$

Here $X_t \in \mathbb{R}^{\mathbf{N}\times\mathbf{F}}$ denotes a snapshot of stock collection $\mathbf{S}$ with $\mathbf{N}$ stocks at day t. $\mathbf{G}$ is the graph. $\hat{Y}_{\mathbf{d}} = [\hat{y}^1_{\mathbf{d}}, \cdots, \hat{y}^{\mathbf{N}}_{\mathbf{d}}] \in \mathbb{R}^{\mathbf{N}}$ is the predicted series labels of collection $\mathbf{S}$ at day $\mathbf{d}$. We use cross entropy function as the loss function:

$$\mathcal{L} = -\frac{1}{\mathbf{N}}\sum_{\mathbf{s}=1}^{\mathbf{N}}[y^{\mathbf{s}}_{\mathbf{d}} log(\hat{y}^{\mathbf{s}}_{\mathbf{d}}) + (1-y^{\mathbf{s}}_{\mathbf{d}})log(1-\hat{y}^{\mathbf{s}}_{\mathbf{d}})] \tag{3}$$

Where $Y_{\mathbf{d}} = [y^1_{\mathbf{d}}, \cdots, y^{\mathbf{N}}_{\mathbf{d}}]$ is denoted as the ground-truth series and $y^{\mathbf{s}}_{\mathbf{d}} \in \{0,1\}$ for that most works estimate the binary movement [5], [22] with 1 denoting as rise or positive, 0 denoting as fall or negative.

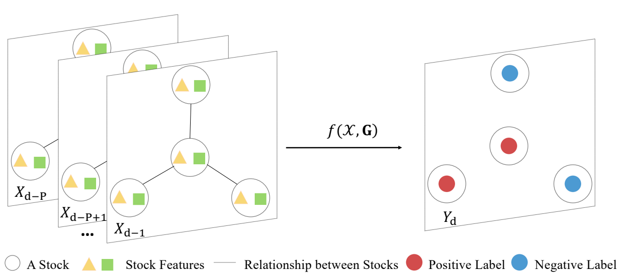

Fig. 1. Problem Formulation of Stock Movement Prediction Based on Graph

IV. MULTI-GCGRU

The Multi-GCGRU architecture in our paper aims to predict the stock price movement by considering both historical records of individual stock and cross effect from other related stocks. The first stage is to replicate relationships which can explain the cross effect among stocks by encoding them into graphs (e.g., Shareholding graph, Industry graph, Topicality graph). The second stage is to learn more complicated and dynamic cross-correlation behind stock collection by Multi-GCN. The third stage is to utilize GRU to learn the temporal dependency by historical records along with higher features containing cross effect produced by multi-GCN. Finally, a fully connected layer with sigmoid activation function is added to get the probability prediction. The details of the architecture are shown in Figure 2.

A. Graph Construction

The cross-autocorrelation of security returns over time has long been recognized in stock markets [9]. To capture the complex cross effect among a clique of stocks, we extract three types of relationships based on prior knowledge and construct three graphs respectively, including (1) Shareholding Graph $\mathbf{G_S} = (\mathbf{V}, \mathbf{E_S}, \mathbf{A_S})$ to encode shareholding influence [12], (2) Industry Graph $\mathbf{G_I} = (\mathbf{V}, \mathbf{E_I}, \mathbf{A_I})$ to encode lead-lag effect within industry [9], (3) Topicality Graph $\mathbf{G_T} = (\mathbf{V}, \mathbf{E_T}, \mathbf{A_T})$ to encode topical news impact [19], where $|\mathbf{V}| = \mathbf{N}$ refers $\mathbf{N}$ public corporations in the graph, $\mathbf{A} = (\mathbf{a}_{ij})_{\mathbf{N}\times\mathbf{N}}$ is the adjacency matrix representing the particular stock network. Element a_{ij} in $\mathbf{A}$ stands for the connection strength between company i and company j. We examine the effectiveness of these relationships in our experiments.

*1) **Shareholding Graph**:* [12] defined a weighted shareholding graph based on the financial fact that the performance of a listed company is likely to influence the stock price of its shareholder which is also a listed company and vice versa [29]. The mutual influence strength depends on the shareholding ratio. Therefore, an edge is attached to two listed corporations having shareholding relationship. The edge weight a_{ij} is the shareholding ratio in range of $[0, 1]$. However, we find that the cross-shareholdings among public corporations are rather rare empirically, which leads to a very sparse adjacent matrix and thus weakens its effective representation for the cross-effect among stocks.

*2) **Industry Graph**:* The cross effect among stocks has displayed a pronounced lead-lag structure [9], which refers that returns on some stocks systematically lead or lag those of others [30]. The main cause of lead-lag effect in equity market can be explained by industry information diffusion hypothesis [10], which argues that new information is usually incorporated into the stock prices of industry leaders before it spreads to other firms in the same industry. Therefore, lead-lag effect is related to the firm size and the stock returns of larger firms generally lead those of smaller ones [9] in the same industry.

In this paper, we construct an industry graph to model the lead-lag relationship. Since [10] has verified that the cross-industry lead-lag effect is rather weak, we mainly focus on intra-industry lead-lag effect. If two companies are in different industries, there is no edge between them. Otherwise, the influence from company i to company j is denoted as $a_{ij} = \frac{\mathbf{M_i}}{\mathbf{M_j}}$, where $\mathbf{M}$ denotes the firm size. Note that such influence is asymmetric. Returns of small firms are correlated with past returns of big firms, but not vice versa [10]. Since several studies [9] have shown that stock with larger capital size tend to be leading stock, we use registered capital to measure firm size in this paper.

*3) **Topicality Graph**:* The rapid development of Internet has accelerated the speed of news producing and broadcasting, enhancing the influence of news in investment behaviors. Extensive studies have been conducted on the correlations between news and stock prices [19], [13]. It can be observed that a stock responses to a group of similar news with the same topicality. For example, the news related with 2019-nCoV are likely to impact a pharmaceutical stock. On the other hand, the news belonging to one particular topicality impact many related stocks, leading to the similar volatility of stock prices. For instance, news related with 2019-nCoV have influenced pharmaceutical stocks (e.g. ABIO), entertainment stocks (e.g. Disney) and caterers stocks (e.g.YUM).

In this paper, we novelly define a topicality graph to model the correlation among stocks due to topical news

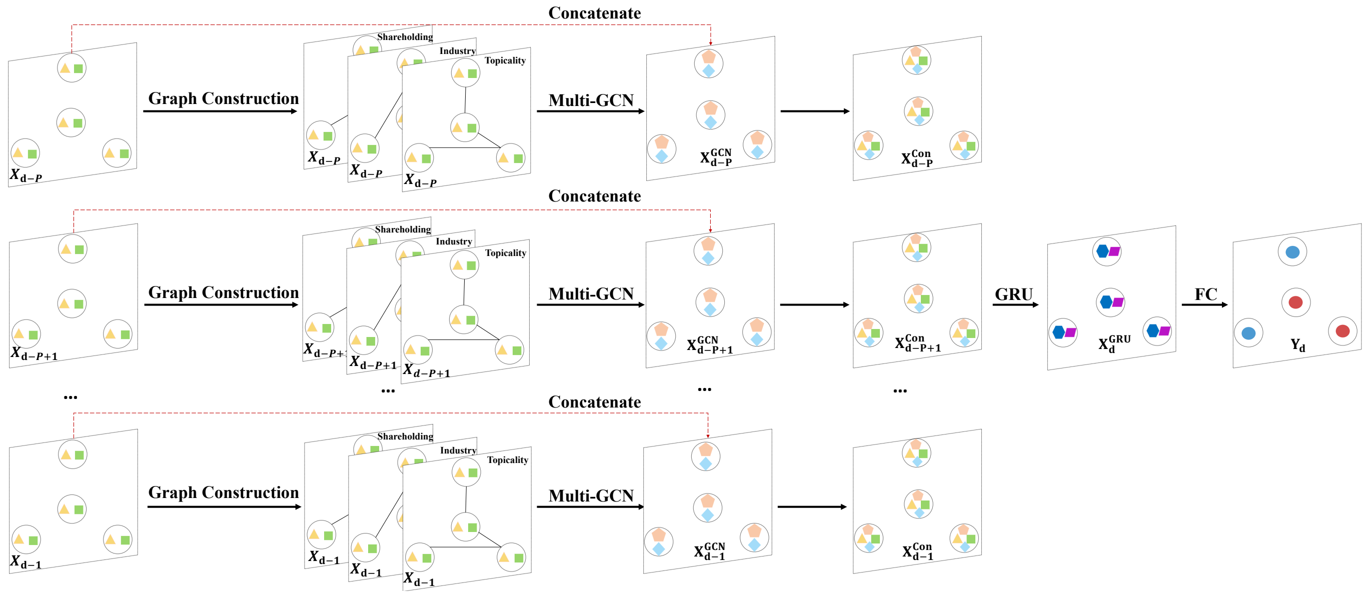

Fig. 2. The architecture of Multi-GCGRU; GCN: Graph Convolution Network; GRU: Gated Recurrent Units; FC: Fully connected layer.

impact. The stock datasets we collect from a public API (https://tushare.pro/) have already contained topical information for each stock, where a stock has more than one topicality and a topicality is shared by more than one stock. Based on these datasets, we measure the connection strength by the number of topicalities which are shared by two listed corporations. Intuitively, the more topicalities shared by corporations, the more similar their prices volatility might be. Specifically, if company i owns $\mathbf{M_i}$ topicalities, company j owns $\mathbf{M_j}$ topicalities and they share $\mathbf{T_{ij}}$ topicalities, the connection strength from i to j is denoted as $a_{ji} = \frac{\mathbf{T_{ij}}}{\mathbf{M_j}}$. Similarly, the connection strength from j to i is denoted as $a_{ij} = \frac{\mathbf{T_{ij}}}{\mathbf{M_i}}$. Note that if there is no topicality shared by two companies, their connection strength is zero.

B. Graph Convolutional Network

In this section, we introduce the architecture of graph convolutional network proposed in our paper. We first introduce the traditional graph convolutional layer proposed by [32]. Then we novelly propose its variant, i.e., multi-graph convolutional layer, to incorporate multiple pre-defined stock graph structures into our model. As an alternative, we also design a dynamic graph convolutional layer to learn graph topology from data automatically in case that we don't have sufficient domain knowledge and data to pre-define a stock graph. Finally, the mathematical formalization of our graph convolutional network is defined.

*1) **Graph Convolutional Layer:*** When processing graph data, researchers hope to extract high level representations containing graph structure information. Inspired by convolution on images, [31] defined convolution on graph in spectral domain. The normalized graph Laplacian matrix $\mathbf{L} = \mathbf{I_N} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ is decomposed as $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ to get a graph Fourier basis $\mathbf{U}$ which contains the graph topology information, where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$, $\mathbf{U}$ is the eigenvectors matrix and $\mathbf{\Lambda}$ is the eigenvalues matrix. The graph signal $x \in \mathbb{R}^{\mathbf{N}}$ is first transformed to the spectral domain by the graph Fourier basis $\mathbf{U}$, then filtered by a parameterized kernel Θ, finally transformed back by the inverse graph Fourier basis $\mathbf{U}^T$ to get the convolution result: $y = \Theta *_{\mathcal{G}} x = \mathbf{U}\Theta\mathbf{U}^T x$, where $*_{\mathcal{G}}$ is the graph convolution operator. However, all the nodes in the graph are considered during convolution by the kernel Θ with $\mathbf{N}$ parameters. To extract spatial localization in graph, [32] restricted the kernel Θ as a polynomial of $\mathbf{\Lambda}$ to get the $\mathbf{K}$-hop graph convolution:

$$y = \Theta(\mathbf{\Lambda}) *_{\mathcal{G}} x = \mathbf{U}(\sum_{k=0}^{\mathbf{K}-1}\theta_k\mathbf{\Lambda}^k)\mathbf{U}^T x = \sum_{k=0}^{\mathbf{K}-1}\theta_k\mathbf{L}^k x \tag{4}$$

Where $\theta \in \mathbb{R}^{\mathbf{K}}$ is a vector of polynomial coefficients. $\mathbf{K}$ is the kernel size of graph convolution, which determines the maximum radius of the convolution from central node.

In this paper, we generalize this definition to a signal $X \in \mathbb{R}^{\mathbf{N}\times\mathbf{F}}$ with $\mathbf{F}$ input features and $\mathbf{C}$ filters as $Z = (\sum_{k=0}^{\mathbf{K}-1}\theta_k\mathbf{L}^k)XW$, where $Z \in \mathbb{R}^{\mathbf{N}\times\mathbf{C}}$ is the convolved signal matrix and $W \in \mathbb{R}^{\mathbf{F}\times\mathbf{C}}$ is a trainable parameter. The corresponding graph convolutional layer is denoted as follow:

$$H^{(l+1)} = \boldsymbol{\rho}((\sum_{k=0}^{\mathbf{K}-1}\theta_k\mathbf{L}^k)H^{(l)}W^{(l)}) \tag{5}$$

Where $\mathbf{L}$ represents a graph based on one particular corporation relationship. $H^{(l)} \in \mathbb{R}^{\mathbf{N}\times\mathbf{F}}$ is the input at l-th layer and $H^{(l+1)} \in \mathbb{R}^{\mathbf{N}\times\mathbf{C}}$ is its output. $W^{(l)} \in \mathbb{R}^{\mathbf{F}\times\mathbf{C}}$ is the corresponding trainable parameter. $\boldsymbol{\rho}(\cdot)$ denotes the activation function, e.g., tanh, sigmoid, ReLU.

There are two main steps in graph convolution. The first step is to aggregate the information from surrounding stocks by multiplying Laplacian matrix and features matrix. Then a fully connected layer is implemented on the aggregated features to create high level representations for each stock. Note that with small $\mathbf{K}$, the feature aggregation will focus on close neighbors being reached with small number of hops. Increasing the value of $\mathbf{K}$ enables model to capture larger spatial dependency.

*2) **Multi-Graph Convolutional Layer**:* To model the cross effect from the constructed graphs, we propose the multi-graph convolution as follow:

$$H^{(l+1)} = \boldsymbol{\rho}((\sum_{k=0}^{\mathbf{K}-1} \theta_k (W_S \mathbf{L_S}^k + W_I \mathbf{L_I}^k + W_T \mathbf{L_T}^k)) H^{(l)} W^{(l)}) \tag{6}$$

Where $\{\mathbf{L_S}, \mathbf{L_I}, \mathbf{L_T}\}$ are the Laplacian matrices corresponding to adjacency matrices $\{\mathbf{A_S}, \mathbf{A_I}, \mathbf{A_T}\}$. $\{W_{\mathbf{S}}, W_{\mathbf{I}}, W_{\mathbf{T}}\} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ are the trainable coefficients respectively.

Intuitively, different relationships are likely to contribute differently for stock prediction. However, it is hard to assign them weights artificially. Therefore, we leave it to the algorithm and hope to learn the weights from data automatically. Note that our multi-graph convolution is not limited to the relationships above. It can be easily extended to incorporate more effective relationships for better stock prediction.

*3) **Dynamic Graph Convolutional Layer**:* All the predefined relationships require prior knowledge in financial domain and need more financial data which is unavailable sometimes. To get rid of the expert knowledge, we design a dynamic Laplacian matrix learned by data and the corresponding layer is defined as follow:

$$H^{(l+1)} = \boldsymbol{\rho}(\hat{\mathbf{L}} H^{(l)} W^{(l)}) \tag{7}$$

Here, $\hat{\mathbf{L}} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ is trainable and can be initialized just as other trainable parameters. We compare the performance of data-driven relationship with that of hand-crafted relationships in the experiments.

*4) **Graph Convolutional Network**:* In this paper, following Kipf' work [28], we design our graph convolutional network with two graph convolutional layers. The formula of our GCN is defined as follow:

$$X_t^{\mathbf{GCN}} = \boldsymbol{tanh}(f(\mathbf{L}) \boldsymbol{tanh}(f(\mathbf{L}) X_t W^1) W^2) \tag{8}$$

Here, $f(\mathbf{L})$ can represent one pre-defined relationship (see Equation 5), or the combination of various relationships (see Equation 6), or dynamic relationship (see Equation 7). We utilize $\boldsymbol{tanh}(\cdot) \in [-1, 1]$ as activation function in this paper. $X_t \in \mathbb{R}^{\mathbf{N}\times\mathbf{F}}$ is the input features matrix at day t, $X_t^{\mathbf{GCN}} \in \mathbb{R}^{\mathbf{N}\times\mathbf{C}}$ is the output of GCN at day t, which is fed into GRU later. Equal graph convolution operation with the same kernel is implemented on each day in parallel.

C. Gated Reccurent Unit

RNN has shown its powerful capacity to process time-series tasks to capture long-term dependency and recent stock prediction studies. [8], [26], [7] have demonstrated its effectiveness. Among various variants of RNN (e.g., vanilla RNN, Long Short Term Memory Network (LSTM), GRU), GRU is more complex than RNN which enables it to handle larger long-term dependency and is simpler than LSTM with fewer parameters which enables it to have shorter training time. But [17] has demonstrated that GRU is as effective as LSTM empirically in many applications. Thus we choose GRU for stock price prediction, which is a typical time-series task.

In this paper, we consider not only the historical market data (e.g., trading prices and trading volume) but also the cross-correlation among stocks for stock movement prediction. We concatenate the high level cross-effect features produced by GCN with historical market data to form new features for stocks. These new features are put into GRU to discover trading patterns for future stock trend. Our GRU hidden layer is formulated mathematically as follow:

$$\begin{aligned} r_t &= \boldsymbol{\sigma}([H_{t-1}, X_t, X_t^{\mathbf{GCN}}] \cdot W_r + b_r) \\ u_t &= \boldsymbol{\sigma}([H_{t-1}, X_t, X_t^{\mathbf{GCN}}] \cdot W_u + b_u) \\ \hat{H}_t &= \boldsymbol{tanh}([r_t \odot H_{t-1}, X_t, X_t^{\mathbf{GCN}}] \cdot W_h + b_h) \\ H_t &= u_t \odot H_{t-1} + (1 - u_t) \odot \hat{H}_t \end{aligned} \tag{9}$$

Where $X_t \in \mathbb{R}^{\mathbf{N}\times\mathbf{F}}$ is the historical records of stock collection at time t and $t \in [\mathbf{d} - \mathbf{P} + 1, \cdots, \mathbf{d}]$. $X_t^{\mathbf{GCN}} \in \mathbb{R}^{\mathbf{N}\times\mathbf{C}}$ is the output of GCN which contains cross-effect information at time t. $H_{t-1} \in \mathbb{R}^{\mathbf{N}\times\mathbf{H}}$ is the hidden state at time $t-1$. r_t is the reset gate, u_t is the update gate. $\boldsymbol{\sigma} \in [0, 1]$ is the sigmoid activation function. Operator $\cdot$ is the matrix multiplication, $\odot$ is the element-wise product. The output layer of GRU is $X_t^{\mathbf{GRU}} = H_t \cdot W_g$, where $X_t^{\mathbf{GRU}} \in \mathbb{R}^{\mathbf{N}\times\mathbf{G}}$, $W_g \in \mathbb{R}^{\mathbf{H}\times\mathbf{G}}$.

D. Predictor

The final output of GRU is $X_{\mathbf{d}}^{\mathbf{GRU}} \in \mathbb{R}^{\mathbf{N}\times\mathbf{G}}$ where $\mathbf{d}$ is the target day. A fully connected layer with sigmoid function is stacked on GRU to transform the high level features to get the final probability prediction of stocks in collection. The formulation of Predictor is as follow:

$$\hat{Y}_{\mathbf{d}} = \boldsymbol{\sigma}(X_{\mathbf{d}}^{\mathbf{GRU}} W) \tag{10}$$

Where $\hat{Y}_{\mathbf{d}} \in \mathbb{R}^{\mathbf{N}\times\mathbf{1}}$ is the probability prediction at day $\mathbf{d}$ and $W \in \mathbb{R}^{\mathbf{G}\times\mathbf{1}}$ is trainable parameter.

V. Experiment

A. Datasets

To demonstrate the effectiveness of our model, we collect our datasets from a public API ((https://tushare.pro/), which are the best-know CSI (China Securities Index) 300 index and CSI 500 index in Chinese stock market. CSI 300 is composed of three hundred large-cap listed corporations with good liquidity. CSI 500 consists of constituent stocks chosen from top 500 mid-cap and small-cap listed companies. Their versions are defined every half a year and we fix them on 2015 January, following Lis work [27]. Each stock in our datasets has three kinds of attributes: (1) Input features: opening price, high price, low price, trading amount. Note that all the input features are Min-max normalized. (2) Relationship features: shareholder and shareholding ratio, industry category, registered capital, topicality. They are utilized to construct relational graphs. (3) Label feature: closing price. Given the closing price of a stock at day t as P_t, if $P_t > P_{t-1}$, we attach price movement at this day as positive with $y_t = 1$, otherwise as negative $y_t = 0$.

We retrieve the historical data from 1st June 2015 to 5th December 2019 with the length being 1121 trading days. All prices are adjusted for dividends and splits. We delete the delisted stocks during the collection period. Finally, it remains 287 stocks in CSI 300 and 489 stocks in CSI 500. To solve the problem that some stocks lack trading data for temporary suspension in some trading days, we align the historical trading days of all the stocks and fill up the missing data with trading data in most recent day.

We split the dataset into three parts: the 70% for training, then 10% for validation and the last 20% for testing. Details of the division of these two indexes are shown in Table V-A.

TABLE I
THE SPLIT OF DATASET

Indexes	Training set	Validation set	Testing set	Total
CSI 500	383,719	54,817	109,633	548,169
CSI 300	225,209	32,173	64,345	321,727

B. Evaluation Metrics

The stock movement prediction is a binary classification problem. Several metrics [21] are selected to justify the effectiveness of all the approaches, i.e., Accuracy (ACC), Precision, Recall, F1-score and Matthews Correlation Coefficient (MCC) [5]. ACC measures the ratio of correct predictions over all examples. Precision focuses on the correct prediction ratio of example predicted as positive class. Recall is used to measure the fraction of positive examples that are correctly classified. F1-score is the harmonic mean of Precision and Recall. MCC can avoid bias due to data skew. All metrics are calculated on all the constituent stocks in each CSI index. The formulas of Accuracy and MCC are as follow.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{12}$$

In the confusion matrix, TP is true positive, TN is true negative, FP is false positive, FN is false negative.

C. Baselines

The baselines in this paper can be divided into two groups. The first group only takes the historical records of the target stock as input for its movement prediction, which contains (1) Logistic Regression, (2) ARIMA (Autoregressive Integrated Moving Average) [33], (3) SVM (Support Vector Machine) [1], (3) RF (Random Forest) [34], (4) ANN (Artificial neural network) [6], (5) LSTM (Long Short Term Memory) [35].

The second group considers both historical information and stock relationships, which contains (1) GCN-S: The GCN model with shareholding graph in [12], (2) GCGRU-S: Our model with Shareholding Graph, (3) GCGRU-I : Our model with Industry Graph, (4) GCGRU-T: Our model with Topicality Graph, (4) Multi-GCGRU: Our model with three graphs above, (5) GCGRU-D: Our model with Dynamic Graph.

D. Parameter Settings

We set the length of lag $\mathbf{P} = 5$ for all the baselines for that there are 5 trading days in a week. The optimized length of $\mathbf{P}$ is explored in our Multi-GCGRU. We train Multi-GCGRU utilizing Adam optimizer [5] with an initial learning rate of 0.01 and setting the mini-batch size as 32. Following Kipf' work [28], we set $\mathbf{K} = 1$ and build our GCN module with two layers and the number of corresponding hidden units are tuned within the ranges of [8, 16, 32, 64] on the validation set. We also tune the number of layers and the corresponding hidden units in GRU module with ranges of [2, 3, 4, 5] and [8, 16, 32, 64]. The best performance is observed on the GCN units being [16, 32] and GRU units being [16, 32, 32]. Our model is implemented with Tensorflow 2.1.

E. Experiment Analysis

In this paper, we aim to answer the following research questions:

(1) Does taking the cross effect among stocks into consideration enhance the stock movement prediction? Does our proposed model provide a better solution to incorporate the cross effect?

(2) Which kind of corporation relationship is more effective for stock prediction and why? Can we get rid of artificial relationships based on prior knowledge and learn the relationship on the basis of data?

(3) How does our proposed Multi-GCGRU framework perform with different length of historical information?

To answer the research questions above, we conduct various experiments and deliver the final experiment results on the test datasets. Table II answers the first and second questions and Table III answers the third question.

1) ***Effectiveness of Cross Effect among Stocks****:* In this paper, we design two groups of approaches. The first group only considers the auto-correlation of an individual stock by taking its historical records as input. The second group considers both the auto-correlation of the target stock and cross-correlation among stocks by taking historical market data long with corporation relationships as input. By comparing the performance of these two groups, we can test the effectiveness of cross effect among stocks for prediction.

As we can see in Table II, statistic methods perform worst for their linear and stationarity assumptions against the non-linear and dynamic properties in stock data. LSTM outperforms ANN by almost 3% in accuracy which justifies that temporal dependency exists in stock prices. The performance of RF is nearly as better as LSTM probably due to the randomness in RF which can probably model the stochasticity in stock market. Besides, the GCN-S [12] based on shareholding relationship performs slightly better than LSTM, which indicates that the cross effect is at least as important as the temporal dependency. When considering both cross effect represented by shareholding relationship and temporal pattern in our GCGRU-S, the performance increases nearly 1% in accuracy. All the models integrated with relational features achieve better performance than those without relationships,

TABLE II
THE EXPERIMENTAL RESULTS

Input Feature	Models	CSI300 Accuracy	CSI300 Precision	CSI300 Recall	CSI300 F1	CSI300 MCC	CSI500 Accuracy	CSI500 Precision	CSI500 Recall	CSI500 F1	CSI500 MCC
Historical Records	LR	0.5145	0.9746	0.5133	0.6724	0.0228	0.5149	0.9723	0.5148	0.6732	0.0117
	SVM	0.5197	0.9498	0.5165	0.6691	0.0412	0.5253	0.9662	0.5202	0.6763	0.0636
	RF	0.5375	0.9298	0.5271	0.6728	0.0957	0.5433	0.9900	0.5294	0.6899	0.1587
	ANN	0.5191	0.9724	0.5158	0.6740	0.0463	0.5202	0.9900	0.5170	0.6792	0.0576
	LSTM	**0.5435**	0.9756	0.5291	0.6861	0.1443	**0.5461**	0.9662	0.5318	0.6860	0.1384
Historical Records & Corporation Relationships	GCN-S	0.5472	0.9609	0.5317	0.6845	0.1421	0.5463	0.9675	0.5423	0.6950	0.0717
	GCGRU-S	0.5505	0.9321	0.5346	0.6795	0.1338	0.5521	0.9635	0.5458	0.6969	0.0938
	GCGRU-I	0.5598	0.9561	0.5392	0.6895	0.1739	0.5678	0.9814	0.5540	0.7082	0.1655
	GCGRU-T	0.5628	0.9512	0.5412	0.6899	0.1782	0.5751	0.9837	0.5581	0.7122	0.1916
	GCGRU-D	0.5602	0.9442	0.5402	0.6871	0.1667	0.5697	0.9844	0.5549	0.7097	0.1756
	Multi-GCGRU	**0.5754**	0.9603	0.5484	0.6981	0.2171	**0.5885**	0.9894	0.5658	0.7199	0.2377

proving the effectiveness of the cross effect represented by corporation relationships.

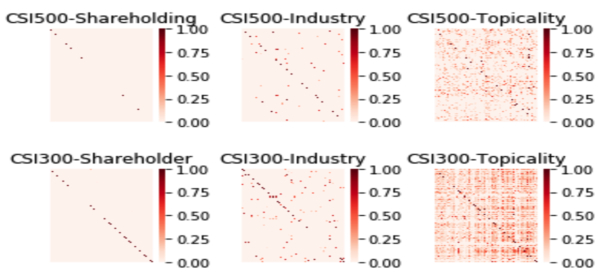

Fig. 3. The Visualization of Relationship Matrices

*2) **Relationships Comparison**:* We have pre-defined three corporation relationships based on financial knowledge, i.e., shareholding relationship, industry relationship, topicality relationship. As shown in Table II, model fed with shareholding relationship has the worst ACC performance, while performance of model with topicality relationship is the best on both CSI300 and CSI500. The results indicate that the impact from common news on stock price is stronger than that of shareholder. This can be explained by that many investors are not familiar with the shareholding structure of the corporation but they are more sensitive with public news, especially those good news and bad news. The model integrated with industry relationship also performs better than that with the shareholding relationship by at least 1% increase of accuracy. Compared with model based on topicality relationship, it has nearly the same performance in CSI 300 and a better performance in CSI500. To further explore the correlation between relationships and their performances, we visualize the corresponding matrices (as shown in Figure 3) and has an interesting finding that the shareholding matrix is the sparsest and the topicality is the densest while the industry matrix is in the middle. Perhaps the dense matrix contains more information helpful for prediction than the spare matrix. The result that the Multi-GCGRU with three relationships performs better than model with any single one can enhance this inference. The results above show that some relationship is more effective than other relationship, indicating that we can futher improve the performance of the proposed GCGRU through adding more effective relationship and we leave it to interested readers.

However, all the pre-defined matrices depend on domain knowledge and extra financial data. To overcome such limitations, we explore a dynamic matrix learned by data automatically. The results show that the performance of model with data-driven matrix is between that with industry relation and topicality relation. Although the data-driven relationship doesn't have the best performance, it can also be a choice when financial knowledge and data are insufficient.

TABLE III
MULTI-GCGRU WITH DIFFERENT LAG SIZES

Length	CSI300 ACC	CSI300 MCC	CSI500 ACC	CSI500 MCC
3-days	0.5623	0.1513	0.5752	0.1742
5-days	0.5754	0.2171	0.5885	0.2377
7-days	**0.5790**	0.2196	**0.5901**	0.2821
9-days	0.5769	0.1869	0.5783	0.1965
11-days	0.5705	0.1378	0.5691	0.1221

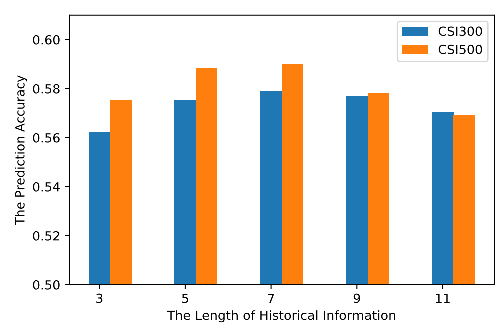

Fig. 4. Accuracy of Multi-GCGRU with Different Lag Sizes

*3) **The Length of Historical Information**:* We conduct experiments on different length of days, specifically [3, 5, 7, 9, 11] previous days. As shown in Table III and Figure 4, our model achieves the best performance at 7 days , with the accuracy 57.90% on CSI300 dataset and 59.01% on CSI500. The worst performance occurs at 3-days with accuracy 56.23% on CSI300 and at 11-days with accuracy 56.91% on CSI500. Therefore, the length of historical information has an impact on prediction performance. And according to the experiments, the best window size is 7 days.

VI. CONCLUSION

The price movement of an individual stock is inevitably influenced by other related stocks. This paper justifies that taking cross effect among stocks can effectively improve the prediction accuracy. Our contribution is that we model the cross effect by encoding various corporation relationships into graphs, based on which we propose a Multi-GCGRU framework to capture both auto-correlation and cross-correlation properties in stock prices. We first employ GCN to extract cross effect among stocks based on three pre-defined graphs. Then we utilize GRU to process the cross-effect features produced by GCN along with historical market information to model temporal pattern in stock prices. Further, we explore a data-driven graph to overcome dependency on prior financial knowledge. Our model can be easily extended to incorporate more effective relationships among stocks.

VII. Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments.This work is supported in part by the National Key R&D Program of China (No. 2019YFB2102100), and by National Natural Science Foundation of China (No. 61802387), and by Science and Technology Development Fund of Macao S.A.R (FDCT) under number 0015/2019/AKP, and by Shenzhen Discipline Construction Project for Urban Computing and Data Intelligence, and by China's Post-doctoral Science Fund (No.2019M663183), and by National Natural Science Foundation of Shenzhen (No.JCYJ20190812153212464).

References

[1] L. Luo, S. You, Y. Xu, and H. Peng, "Improving the integration of piece wise linear representation and weighted support vector machine for stock trading signal prediction," *Appl. Soft Comput.*, vol. 56, pp. 199–216, 2017.

[2] Y. Chang and M. Lee, "Incorporating markov decision process on genetic algorithms to formulate trading strategies for stock markets," *Appl. Soft Comput.*, vol. 52, pp. 1143–1153, 2017.

[3] F. Jin, W. Wang, P. Chakraborty, N. Self, F. Chen, and N. Ramakrishnan, "Tracking multiple social media for stock market event prediction," in *Industrial Conference on Data Mining*, 2017, pp. 16–30.

[4] J. Chang and W. Tu, "A stock-movement aware approach for discovering investors' personalized preferences in stock markets," in *ICTAI,Volos, Greece*, 2018, pp. 275–280.

[5] F. Feng, H. Chen, X. He, J. Ding, M. Sun, and T.-S. Chua, "Enhancing stock movement prediction with adversarial training," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 5843–5849.

[6] H. Liu and B. Song, "Stock trends forecasting by multi-layer stochastic ANN bagging," in *ICTAI, Boston, MA, USA*, 2017, pp. 322–329.

[7] L. Zhang, C. Aggarwal, and G.-J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *SIGKDD*, 2017, pp. 2141–2149.

[8] C. Chen, L. Zhao, J. Bian, C. Xing, and T.-Y. Liu, "Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction," in *SIGKDD*, 2019, pp. 2376–2384.

[9] A. W. Lo and A. C. MacKinlay, "When are contrarian profits due to stock market overreaction?" *The Review of Financial Studies*, 1990.

[10] Hou and Kewei, "Industry information diffusion and the lead-lag effect in stock returns," *The Review of Financial Studies*, vol. 20, no. 4, pp. 1113–1138, 2007.

[11] E. Letizia and F. Lillo, "Corporate payments networks and credit risk rating," *EPJ Data Sci.*, vol. 8, no. 1, pp. 21:1–21:29, 2019.

[12] Y. Chen, Z. Wei, and X. Huang, "Incorporating corporation relationship via graph convolutional neural networks for stock price prediction," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1655–1658.

[13] Z. Hu, W. Liu, J. Bian, X. Liu, and T. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in *WSDM, Marina Del Rey, CA, USA, February*, 2018, pp. 261–269.

[14] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[15] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *SIGKDD*, 2018, pp. 974–983.

[16] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, 11 2019.

[17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS Workshop on Deep Learning*, 2014.

[18] M. Kamp, M. Boley, and T. Gartner, "Beating human analysts in nowcasting corporate earnings by using publicly available stock price and correlation features," in *ICDM, Philadelphia, Pennsylvania, USA*, 2014, pp. 641–649.

[19] W. S. Chan, "Stock price reaction to news and no-news: drift and reversal after headlines," *Journal of Financial Economics*, vol. 70, no. 2, pp. 223 – 260, 2003.

[20] N. Rekabsaz, M. Lupu, A. Baklanov, A. Dur, L. Andersson, and A. Hanbury, "Volatility prediction using financial disclosures sentiments with word embedding-based IR models," in *ACL, Vancouver, Canada, July 30 - August 4*, 2017, pp. 1712–1721.

[21] M. G. Sousa, K. M. Sakiyama, L. de Souza Rodrigues, P. H. Moraes, E. R. Fernandes, and E. T. Matsubara, "BERT for stock market sentiment analysis," in *ICTAI, Portland, OR, USA*, 2019, pp. 1597–1601.

[22] Q. Li, L. Jiang, P. Li, and H. Chen, "Tensor-based learning for predicting stock movements," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[23] Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," in *ACL*, 2018, pp. 1970–1979.

[24] I. Bordino, N. Kourtellis, N. Laptev, and Y. Billawala, "Stock trade volume prediction with yahoo finance user browsing behavior," in *ICDE*, 2014, pp. 1168–1173.

[25] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[26] Y. Qin and Y. Yang, "What you say and how you say it matters: Predicting financial risk using verbal and vocal cues," in *ACL*, 2019, p. 390.

[27] C. Li, D. Song, and D. Tao, "Multi-task recurrent neural networks and higher-order markov random fields for stock price movement prediction: Multi-task rnn and higer-order mrfs for stock price classification," in *SIGKDD*, 2019, pp. 1141–1151.

[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR, April*, 2017, pp. 1–12.

[29] N. Attig, W.-M. Fong, Y. Gadhoum, and L. H. Lang, "Effects of large shareholding on information asymmetry and stock liquidity," *Journal of Banking & Finance*, vol. 30, no. 10, pp. 2875 – 2892, 2006.

[30] M. Hoffmann, M. Rosenbaum, and N. Yoshida, "Estimation of the lead-lag parameter from non-synchronous data," *Bernoulli*, vol. 19, no. 2, pp. 426–461, 2013.

[31] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *ICLR, Banff, AB, Canada, April 14-16*, 2014.

[32] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS, Barcelona, Spain*, 2016, pp. 3837–3845.

[33] Y. Chen and Y. Hao, "Integrating principle component analysis and weighted support vector machine for stock trading signals prediction," *Neurocomputing*, vol. 321, pp. 381–402, 2018.

[34] C. Krauss, X. A. Do, and N. Huck, "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500," *Eur. J. Oper. Res.*, vol. 259, no. 2, pp. 689–702, 2017.

[35] Z. Zhao, R. Rao, S. Tu, and J. Shi, "Time-weighted LSTM model with redefined labeling for stock trend prediction," in *ICTAI, Boston, MA, USA*, 2017, pp. 1210–1217.