Multi-Channel Temporal Graph Convolutional Network for Stock Return Prediction

Jifeng Sun
Tsinghua-Berkeley Shenzhen Institute
ShenZhen, China

Jianwu Lin*
Tsinghua Shenzhen International
Graduate School ShenZhen,China
Lin.jianwu@sz.tsinghua.edu.cn

Yi Zhou,
Tsinghua Shenzhen International
Graduate School ShenZhen,China

*Corresponding author

***Abstract*—Stock return prediction can help investors make better investment decisions and trends of country's economics. However, most of methods for stock return prediction are based on time-series models, treating the stocks as independent from each other. Inter-relations among stocks' time series are out of consideration. In this work, a Multi-Channel Temporal Graph Convolutional Neural Network (MCT-GCN) is proposed to optimize stock movement prediction. Experiments show that its performance is greater than benchmark algorithms, LSTM in the S&P 500.**

Keywords—multi-channel, GCN, Stock, Related Time Series Modeling

I. INTRODUCTION

Nowadays, stock investment plays an increasingly important role in financial industry. As an important indicator of a country's macroeconomics, returns in stock market directly make great effect on the stability of financial markets and the healthy development of the economy. Stock return prediction can help investors not only make better investment decisions by predicting the future trends of a stock, but also understand trends of country's economics. However, it is a very challenging task.

Stock return prediction is based on the regularity of the development of the stock market and its history and current situation, and uses various methods of stock market information and accurate statistical survey data to predict the prospects of the stock market. For decades, scholars have explored various forecasting methods, and stock return prediction has also been a research hotspot. There are some traditional linear models such as autoregressive integrated moving average (ARIMA) models[1],or exponential smoothing[2] can be used in the stock market predicting . Deep learning methods, such as long short-term memory (LSTM)[4], are also widely accepted by the academic community and fits well in China A-share stock markets[3]. However, these models mainly focus on modeling individual time series, which don't take the interrelationship between stocks into consideration. But the relations among stocks' time series could be a very valuable signal for the stock return prediction.

For example, Apple and HP both make computers and are in competition with each other. The performance of their companies influences its own stock prices and each other's price mutually. Therefore, It could be the case that certain relation data can be transformed into a graph structure, and naturally integrate the consistency pattern and characteristic attributes of graph-structured data. In order to use graph-based learning to build an appropriate algorithm which fits the data, graph-based learning has been applied to various machine learning tasks to express entity relationships in non-Euclidean Spaces. With the popularity of graph neural networks, people have more abilities to analyze graph structured data when building deep learning models.

The main contributions of this work can be summarized as follows.

1) A multi-channel T-GCN method is proposed to model relation of time series among stocks effectively to leverage interrelationship between stocks into consideration. The numerical experiments shows that it is better than the existing baseline method LSTM based on individual time series.

2) The multi-channel model is constructed by combining the relationship which generated from trading price/volume data and real relationships among the corporations. The result proves that it has better performance than a single-channel model.

3) Compared with LSTM and GRU, due to fewer GRU parameters, the convergence speed is faster[18], because our model is relatively complex, we use GRU for time series modeling

4) The mutual information and Pearson correlation coefficient are used to generate two new kinds of relationship between each pair of stocks, which proves to be an effective measure in our experiments.

5) By comparing experiments on small data sets, we explored the reasons why T-GCN performs better than LSTM.

The remainder of this paper is organized as follows. In Section II, the related works in the stock return and time series prediction are introduced. In Section III, a short preliminaries is provided to help understand our work deeply. Detailed descriptions of our proposed framework are provided in Section IV. Section V explains the data used in our experiments. Section VI discusses our experimental results in and Section VII concludes our work.

II. RELATED WORK

As the graph neural network is relatively a new solution for stock return prediction, so the most of works in this area are conducted in the recent years.

T. N. Kipf and Chen et al. [6] used investment information to build stock relationships and trade in A stock market. They used a GCN model and compare its performance with embedding models'. But it is still in its infancy. Therefore, this paper referred to methods in other fields for inspiration. For example, traffic flow prediction

2020 IEEE 18th International Conference on Industrial Informatics (INDIN) | 978-1-7281-4964-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/INDIN45582.2020.9442196

has always been very popular in the field of related time series modeling. Y Li et al[12] used two-way random walking to model the spatial dependence, used the encoder-decoder architecture to model the time dependence, and proposed the DCRNN architecture. It tested on the traffic data sets METR-LA and PEMS-BAY and achieved good results.

S. Deng et al. [8] noticed the traditional models' ignorance of the background knowledge of stocks and offered a knowledge-driven Model to capture inconsistent evolution of stream data. They used company relations and daily major financial news to build a knowledge graph, and used the Translating Embedding algorithm to build node embedding, and used a relatively novel TCN model for time series modeling.

Y. Chen and Z. Wei[6] pioneered the use of the graph neural networks to associate the related companies' information in stock market. In the work, they first conducted a graph which presents all involved corporations of a target company and got a distributed representation for each corporation via node embedding methods. The results of their experiments proved the superiority of GCN model in stock return prediction.

Q. Li et al[9] analyzed key reasons why GCNs worked and proposed a co-training approach and a self-training approach to train GCNs. The extensive experiments on benchmark significantly showed how to improve GCNs in learning very few labels, which provided ideas for our experiment.

L. Zhao and Y. Song[12] proposed a model temporal-graph convolutional network (T-GCN) for urban road network traffic prediction, which combines GCN and gated recursive unit (GRU) [13]. GCN is used to learn complex topologies to capture spatial dependencies, and gated recursive units are used to learn dynamic changes in traffic data to capture time dependencies.

So GCN's unique characters get more attention for prediction based on relations among time series.

III. PRELIMINARIES

In this paper, the goal of our method is to predict future stock returns from a stock relation network and historical stock price information. Several preliminary concepts are explained in advance before the description of our methodology.

A. *Data structure Definitions*

Stock relationship network G

The graph G = (V, E) is defined to describe a topological structure of the stock relationship network. Each stock is a vertex. V is the vertex set, V = $\{v_1, v_2, \ldots, v_n\}$, n is the number of vertices, and E is the edge set.

Adjacency matrix A

The adjacency matrix A represents the relationship of stocks, A $\in R_{n\times n}$. In this paper, the adjacency matrix expresses the relationship network. It is 1 if there is a connection between two stocks, otherwise it is 0. This will be explained in detail in the following section IV.

Feature matrix $X_{n\times m}$

The stock price information on the trading day are considered as features of the stock. The characteristics of each stock on each trading day are open price, highest price, lowest price, close price and trade volume, which is a 5-dimension data. The time step l represents l day. P represents the number of features, and $X_t \in R_{n\times I}$ is used to represent the price of each stock at the moment. The problem of correlated temporal financial time series prediction can be thought of as learning the mapping function f in the stock relationship network topology G and the feature matrix X, and then predicting the stock price information at the next T moments, as shown in the equation.

$$Xt+1 = f\,(G; (Xt-n, \cdots, Xt-1, Xt)) \qquad (1)$$

n is the length of the historical time series, and T is the length that needs to be predicted. In this paper, we define the length of T as 1, that is, the prediction target is the stock return information for the next trading day.

B. *Graph Convolutional Networks*

Generally, most of graph convolutional networks (GCNs) can be classified as either spectral or spatial.

The key idea of GCN based on spatial methods is to use the information dissemination mechanism on the graph, that is, the node status is continuously updated, and each moment of the update uses the state information of neighboring nodes at the previous moment. There are some models like LGNN[14], GGS-NN[15], GPNN[16], GGT-NN[17].

Graph neural network based on spectral method is another important class of graph neural network. Different from the spatial method, the key point of the spectral method is to use the graph Laplace matrix as a tool but not to explicitly use the information propagation mechanism on the graph.

Let the adjacency matrix of weighted undirected graph G be A, and the element A (i, j) in the i-th row and j-th column of the matrix is the weight of the edge (i, j). The degree matrix D is defined as

$$D(i,i) = \sum_{j=1}^{n} A(i,j) \qquad (2)$$

The symmetric normalized Laplace matrix of graph G can be defined as

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \qquad (3)$$

Kipf T N and Welling M . [11] proposed the graph convolutional network (GCN). The adjacency matrix A is adjusted to $\widetilde{A}$, here, $\widetilde{A} = A + I_N$, is the adjacency matrix of the undirected graph G with added self-connections. I_N is the identity matrix. Then the convolution layer is defined as:

$$H^{l+1} = \sigma\left(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} H^l W^l\right) \qquad (4)$$

Where $\boldsymbol{W^l}$ is a parameter matrix with a dimension of $\boldsymbol{d^{l+1}} \times \boldsymbol{d^l}$, $\widetilde{D}$ is the degree matrix of $\widetilde{A}$.

C. Gated Recurrent Unit (GRU)

The GRU is a gating mechanism in the recurrent neural networks, it was introduced by 2014 by Kyunghyun Cho et al. [13]. The GRU is similar to long short-term memory. The research shows that the performance of GRU on some tasks such as speech signal modeling is similar to LSTM.
What' more, because of its having fewer parameters than LSTM, GRU runs faster. The Architecture of GRU is as shown in Figure 3.1, h_{t-1} indicates the hidden state at time t-1; x_t is the stock information at time t; u_t and r_t in the figure represent the update gate and the reset gate, respectively. The update gate is used to control the degree to which the previous state information is converted to the current state. The reset gate controls how much previous information is written into the current state h_t. The GRU uses the hidden state at time t-1 and the current stock feature X t as inputs to obtain the stock price state at the next time. The model can effectively preserve the historical price status and capture temporal dependencies.

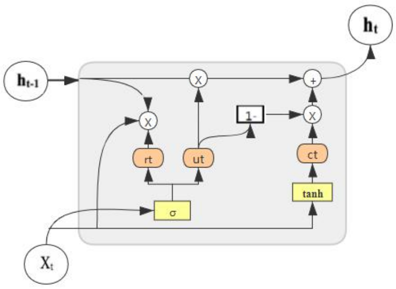

Fig. 3.1 the Structure of the Gated recurrent units.

D. Mutual Information MI

The mutual information (MI) [1] of two random variables is a measure of the mutual dependence between the two variables.
Mutual information has been used as a criterion for feature selection and feature transformations in machine learning. And it is used in determining the similarity of two different clustering of dataset. In this paper, MI describes the similarity between different stock return time series and generate a stock relational graph according to the values of the MI matrix.
The definition of MI is:

$$I(X;Y) = D_{KL}(P_{(X,Y)} \| P_X \otimes P_Y) \qquad (5)$$

In the formula, $\boldsymbol{D_{KL}}$ is the Kullback–Leibler divergence. $\boldsymbol{P_{(X,Y)}}$ is the joint distribution of X and Y and their marginal distributions are $\boldsymbol{P_X}$ and $\boldsymbol{P_Y}$.

E. Pearson correlation coefficient

In statistics, the Pearson correlation coefficient, also known as the Pearson product-moment correlation coefficient (PPMCC or PCC), is used to measure the linear correlation between two variables X and Y, with a value between -1 and 1. In this paper, we also use PCC to describe the linear correlation between different stock sequences and generate a stock graph according to the PCC values .
The formula of the PCC is:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \qquad (6)$$

$cov(X,Y)$ is the covariance, σ_X is the standard deviation of X; σ_Y is the standard deviation of Y.

IV. METHODOLOGY

A. Overview

This section first introduces our entire framework. The model is called Multi-Channel Temporal Graph Convolutional Network (MCT-GCN), which is on the basis of Time-graph convolutional neural network (T-GCN) [12]. The entire framework is shown in Figure 4.1. The model consists of two parts: Relational Modeling Module and Temporal Modeling Module. The relational modeling module uses a multi-channel design. The architecture consists of three GCN channels.

B. Relational Modeling Module

The function of relationship modeling is to better aggregate the stock nodes， so that a stock can be better combined with the information of other associated stocks. Information from neighboring nodes is aggregated and then added to each node representation. This paper focuses on three types of graph relationships, the corporate relational data from Wiki data, the stock relational graph generated according to the values of the MI matrix of the stock returns and the stock relational graph generated according to the values of the PCC matrix of the stock returns.
The corporate relationship graph obtains a total of 72 types of relational data from Wiki data, and then excludes some stocks that have no relationship with other stocks from the price data. So it gets 423 stock sequences. Therefore, the shape of the corporate relationship data is [423,423,72]. However, because the matrix of each kind of relationship among companies is too sparse, two following approaches are applied to compress all relational graph into a graph.
1) Add the matrices Å directly in the third dimension to get the adjacency matrix A and input it into Graph Convolution layer together with stock feature matrix X. The layer can be expressed as:

$$A = \sum_{i=1}^{72} \text{Å}_i \qquad (7)$$

$$f(X,A) = Relu(\hat{A}XW) \qquad (8)$$

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \qquad (9)$$

W represents its weight matrix in the layer, and Relu() represents its activation function.
2) Input the matrices Å to a full-connected neural network to get the adjacency matrix A and input it into Graph Convolution layer together with stocks feature matrix X. The layer can be expressed as:

$$A = W_1 \text{Å} + \text{b} \qquad (10)$$

$$f(X,A) = Relu(\hat{A}XW_2) \qquad (11)$$

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \qquad (12)$$

W_1 represents its weight matrix of the full-connected layer, and b is its bias of the full-connected layer. W_2 representsits weight matrix in the layer.

Intuitively, method 2 uses the fully connected neural network to compress the matrix Å, but the actual experimental effect is very poor compared to method 1.

Another two types relationships, MI relational graph and PCC relational graph, are not sparse. In order to remove fake correlation effect, the largest first 1% elements of the matrix are set to 1, and the rest of elements are set to 0.

Because the relationships of three categories are quite different and not suitable to compress into one graph, , Multi-Channel GCN layer is proposed in this paper to solve this problem, which carries out graph convolution operations on three kinds of graph relations respectively and then add them one by one.

$$F(X,Å) = \sigma\left(\sum_{i=1}^{3} f(X,Å_i)\right) \tag{13}$$

Å represents these three kinds of graphs mentioned above. After the above operations, the features are extracted from relationship graph and X.

C. Temporal Modeling Module

Stock return trends in the stock market can be considered as a typical type of spatial-temporal graph. The previous analysis uses a single stock (company) as a node. According to the market's momentum effect Fama and French in 1993[20], we know that the future return of the stock is affected by the current performance and the historical performance, and it may change over time. Therefore, a feature extraction module is built to obtain this temporal dependency in the stock market.

In deep learning, commonly used module that can be used to extract time dependencies recurrent neural networks, such as LSTM, GRU and etc. This paper selects GRU because of its superiority, especially in training efficiency according to [18]. The steps of the temporal modeling can be expressed as:

$$F = F(X,Å) \tag{14}$$

$$u_t = \sigma(Wu[F,h_{t-1}] + b_u) \tag{15}$$

$$r_t = \sigma(Wu[F,h_{t-1}] + b_r) \tag{16}$$

$$c_t = \tanh\left(Wc[F,(r_t * h_{t-1})] + b_c\right) \tag{17}$$

$$h_t = u_t * h_{t-1} + (1 - u_t) * c_t \tag{18}$$

The description of the temporal modeling module is what was mentioned above.

To sum up, MCT-GCN model is designed to handle the complex correlation and time dependence in the related stock prediction. On the one hand, the multi-channel design enhances the effect of model prediction. On the other hand, the gated recursive unit is used to capture the dynamic change of stock return to obtain the time dependence of stock data and finally realize the related stock return forecasting task.

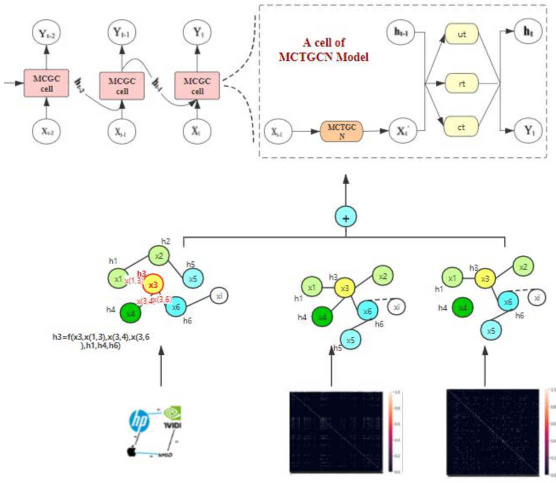

Fig. 4.1 The whole process of graph-related financial time series prediction

V. DATASET AND FEATURES

The data sets in this paper is accessible through: https://github.com/dmis-lab/hats. The basic data source is the stocks in S&P 500 from February 8, 2013 to June 17, 2019 and423 stocks having relationships with other stocks are selected. Three kinds of additional data are required for the experiment.

- Historical return data
- Wiki relations between their companies
- Two kinds of relationship data: MI matrix and PCC matrix which are generated from the history price data

A. Historical return data

Firstly, the daily closing price of each stock from February 8, 2013 to June 17, 2019 is collected to calculate the daily returns. Then divide the daily return of each stock by the maximum value of the entire data set to get the normalized return. The 80% data is used for training, and 20% for out of sample testing..

B. Wiki Company-based relations

In order to explored the impact of company relations such as investment relations and supply relations on stocks. Wiki data for corporate relations is used, which is one of the largest and most active open domain knowledge bases with more than 42 million projects and 367 million sentences. As mentioned before, 72 kinds of relationships is used from this data set. As shown in Figure 5.1, if there are statements that company A and B are in a competitive relationship, then companies A and B have a relationship r1. If there is a supply relationship between the two companies, they have a relationship r2.

C. Generated relational data

Besides the company correlation data, the two kinds of relational data, MI matrix and PCC matrix, are generated from historical return data.

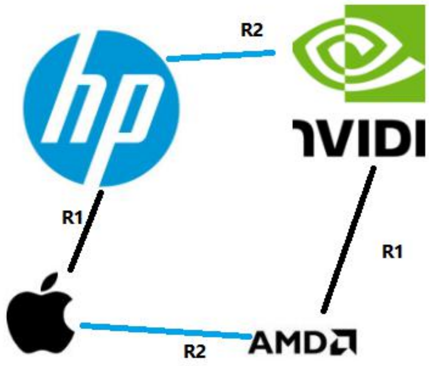

Fig. 5.1 Examples of two kinds of company relations

VI. NUMERICAL EXPERIMENT

This experiment uses two data sets. The first one is the S&P 500 data mentioned above. In addition, in order to explore the reason why MCT-GCN is more effective than LSTM, an experiment is conducted on the stock data of Nvidia and AMD. This experiment, for the adjacency matrix representing the relationship between two stocks, tries relationship weights with different values.

For application, our model can be applied to regression and classification task. For the regression problem, we use the model to predict the actual future return of the stock. For the classification problem, We performed classification on the following three types of labels: [up, neutral, down] according to the return of each stock.

General settings

All the models in our experiments uses a 16-day lookback sequence length and the normalized daily return data as our input feature. For the T-GCN and MCT-GCN, the dimension of the input is [batch size, sequence length, node numbers, feature size]. For LSTM, the dimension of the input is [node numbers, sequence length, feature size].

The hyper parameters of the models mainly include learning rate, batch size, training epoch, and the number of hidden layers.

Among these hyper parameters, some are very important like the number of hidden units and directly related to the quality of the model or even whether it converges. This experiment just selects some parameters based on past experience and the method recommended in [13].

This experiment manually adjusts and sets the learning rate to 0.001, the batch size to 16, the training epoch to 100, number of hidden layers of the model to 64 and the time step is set to 16. Because financial data has the problem of short dependence, the first four weeks are used to predict the return on the seventeenth day.

Loss Function

In this paper, Our model is applied to applied to regression and classification task.

This paper uses Root Mean Squared Error (RMSE) to evaluate the model performance of the regression task. $\boldsymbol{Y_t}$ and $\boldsymbol{\hat{Y}_t}$ denote the real return and the predicted return respectively. L_{reg} is an L2 regularization and λ is set as 0.001.

$$reg\ loss = \left\|Y_t - \hat{Y}_t\right\| + \lambda L_{reg} \tag{19}$$

For classification task, we use cross-entropy loss to train models.

$$Loss = -\frac{1}{n}\sum_i \sum_{j=1}^{l} Y_{ij}\log(\boldsymbol{\hat{Y}_{ij}}) \tag{20}$$

Y_{ij} is the true movement class of stock i. $\boldsymbol{\hat{Y}_{ij}}$ means the predicted class.

Methods

LSTM is one of the most powerful deep learning models for time series forecasting. Many previous works have proven the effectiveness of LSTM. LSTM network in this experiment is with 2 layers and a hidden size of 128. The RMSProp optimizer is used to train LSTM, which is known to be suitable for RNN-based models.

Measurements

All measurements for the experiments are defined as below.

Root Mean Squared Error (RMSE):

$$RMSE = \left\|Y_t - \hat{Y}_t\right\| \tag{21}$$

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_t - \hat{Y}_t\right| \tag{22}$$

Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}(Y_t-\hat{Y}_t)^2}{\sum_{i=1}(Y_t-\bar{Y})^2} \tag{23}$$

Explained Variance Score (Var):

$$var = \frac{Var\{Y-\hat{Y}\}}{Var\{Y\}} \tag{24}$$

Accuracy can be calculated as follows.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{25}$$

Specifically, for the regression work, RMSE and MAE are used to measure the prediction error of the regression task: the smaller the value is, the better the prediction effect is. Both R2 and Var are the correlation coefficients, which measure the ability of the prediction result to represent the actual data: the larger the value is, the better the prediction effect is.

Accuracy measures the performance of the classification work. The label of prediction can be defined as True Positive(TP), True Negative(TN), False Positive(FP), or False Negative (FN). The Accuracy can be calculated as formula 25.

Model Type	Data Type	S&P 500				
		RMSE	*MAE*	*R^2*	*VAR*	*Accuracy*
LSTM	Return	0.09357	0.06739	0.00156	0.00156	0.3733
T-GCN	Return+ Relation	0.05914	0.03341	0.60071	0.60074	0.383
MCT-GCN	Return+ Relation+ PCC	0.05890	0.03315	**0.60395**	**0.60400**	**0.3910**
MCT-GCN	Return+ Relation+ PCC+ mutual information	**0.05878**	**0.03306**	0.60564	0.60565	0.3921

Table 6.1: Performance comparison among MCT-GCN and other baseline methods on S&P 500 on the Test Dataset

Model Type	Relationship weight	Nvidia &AMD				
		RMSE	*MAE*	*R^2*	*VAR*	*Accuracy*
LSTM(NVIDIA+ AMD)	-	0.12865	0.08709	0.05000	0.00160	0.3762
LSTM(AMD)	-	0.15880	0.10419	0.42200	0.02327	0.3731
LSTM(NVIDIA)	-	0.07579	0.05436	0.18270	0.01178	0.3735
T-GCN	0	0.08035	0.05201	0.66504	0.59174	0.3801
	0.01	0.07844	0.04863	0.61028	0.61151	0.3846
	0.05	0.07850	0.04914	0.60969	0.60978	0.3890
	0.1	0.07886	0.05026	0.60607	0.60621	0.3820

Table 6.2: Performance comparison between T-GCN with different relation weights and LSTM on NVIDIA and AMD on the Test Dataset

In the table 6.1, the experiment result on the test dataset shows that the prediction performance of MCT-GCN is better than T-GCN and LSTM, which can improve the prediction performance of stocks. In addition, the MCT-GCN using company relation data, and two kinds of generated relational data mainly get best performance on all measure index.

In table 6.2, the experiment result shows that:

1) For T-GCN, the performance of the model is improved after adding relationship weighting mechanism, and the value of weight of the relationship has additonal influence on the performance of the models .

2) For LSTM, the model can not fit the trend of different stocks well at same time. From the performance of the three different inputs, when the model input is NVIDIA and AMD data, the R2 of the model is smaller than that of the model when NVIDIA and AMD are input separately, indicating that the LSTM can not fit different stocks at the same time.

3) The last and most importantly, the result by comparing the performance of T-GCN and LSTM shows that the main improvement of the model is brought by the parameter W in the graph convolution, which increases the model's learning ability for different stocks. When the relation weight is 0, A is a diagonal matrix and the graph convolution actually degenerates into the following form:

$$f(X,A) = Relu(XW) \tag{26}$$

$$A = I \tag{27}$$

VII. Conclusion and Future works

This paper proposes a new type of neural network based on the phenomena in the stock market, which can defeat the existing classic time series models LSTM and T-GCN in stock prediction tasks. It is a novel method for the prediction of stock returns of related companies. At the same time, the underlying reasons why T-GCN have good performance in stock prediction tasks is explored.

This paper only compares commonly used deep learning models, but classic machine learning models such as ARIMA, LA and other models have not been introduced.

In future research, we will explore more ways to optimize model parameters, such as the width and depth of the hidden layer of the neural network, the number of training rounds, and the learning rate. We will also try more deep learning model components, such as Dropout and Batch Normalization.

In addition, it is also worthwhile to test the performance of the model on the data of the A-share market.

References

[1] Box, G.E.P., Jenkins, G.M.: Time Series Analysis, Forecasting and Control. Holden-Day, San Francisco (1970)

[2] Brown, R.G.: Smoothing, Forecasting and Prediction. Courier Dover Publications, Mineola (2004)

[3] Durham, G.B.: SV mixture models with application to S&P 500 index returns. J. Financ. Econ. 85, 822–856 (2007)

[4] Chen, Kai, Yi Zhou, and Fangyan Dai. "A LSTM-based method for stock returns prediction: A case study of China stock market." 2015 IEEE international conference on big data (big data). IEEE, 2015.

[5] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.

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

[7] Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting[J]. arXiv preprint arXiv:1707.01926, 2017.

[8] Feng, F., He, X., Wang, X., Luo, C., Liu, Y., Chua, T.S., 2019. Temporal relational ranking for stock prediction. ACM Transactions on Information Systems (TOIS) 37, 27.

[9] Deng S, Zhang N, Zhang W, Chen J, Pan JZ, Chen H. Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network. InCompanion Proceedings of The 2019 World Wide Web Conference 2019 May 13 (pp. 678-685). ACM.

[10] Li Q, Han Z, Wu XM. Deeper insights into graph convolutional networks for semi-supervised learning. InThirty-Second AAAI Conference on Artificial Intelligence 2018 Apr 29.

[11] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. ArXiv:1609.02907, 2016

[12] Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting[J]. arXiv preprint arXiv:1707.01926, 2017.

[13] Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. IEEE Transactions on Intelligent Transportation Systems. 2019 Aug 22.

[14] Cho, Kyunghyun; van Merrienboer, Bart; Gulcehre, Caglar; Bahdanau, Dzmitry; Bougares, Fethi; Schwenk, Holger; Bengio, Yoshua (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". arXiv:1406.1078

[15] Bandinelli N, Bianchini M, Scarselli F. Learning long-term dependencies using layered graph neural networks. In: The 2010 International Conference on Neural Networks (IJCNN). Barcelona: IEEE, 2010, 1–8

[16] Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks. ArXiv:1511.05493, 2015

[17] Liao R, Brockschmidt M, Tarlow D, et al. Graph partition neural networks for semi-supervised classification. ArXiv: 1803.06272, 2018

[18] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. 2014 Dec 11.

[19] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[20] Fama, Eugene F., and Kenneth R. French. "Common risk factors in the returns on stocks and bonds." Journal of (1993)..