An Interpretable Framework for Stock Trend Forecasting

Lewen Wang\textsuperscript{1}, Zuoxian Ye\textsuperscript{2}

\textsuperscript{1} School of Information Science and Technology, Peking University, Beijing, 100871, China

\textsuperscript{2} School of Information Science and Technology, Peking University, Beijing, 100871, China

*Corresponding author’s e-mail: lwwang@pku.edu.cn

Abstract. Stock trend forecasting plays a critical role when investing in the stock market. Comparing with traditional technical analysis and fundamental analysis, deep learning models own better forecasting performance. However, the poor interpretability of deep learning models brings lots of limitations to its practical application since the lack of interpretability increases the investment risk. In this paper, we propose a graph-based framework, which owns good interpretability while maintaining forecasting performance. Specifically, the framework explains the stock returns by dividing the returns into multiple parts: 1) the part related to individual stock; 2) the part related to company business; 3) the part related to the corresponding industry; 4) the part associated with the whole market. Extensive experiments on real-world Chinese stock market data have demonstrated the effectiveness of our proposed framework for stock trend forecasting. Afterward, we try to illustrate the interpretability of our framework.

1. Introduction

Stock trend forecasting has been one of the critical research issues for many investors and researchers. Unlike the technical \cite{1} and fundamental \cite{2} analysis methods used in traditional methods, quantitative methods often use models to replace the human decisions, which use computer technology to mine patterns in historical market data to obtain useful information to make forecasting. To improve the predictive performance of the model in quantitative methods, many studies often utilize complex non-linear models to train the model based on large-scale historical data directly. However, the lack of interpretability brings high investment risk, making it a challenge to apply the model to the real trading in the stock market.

In this paper, we propose an interpretable stock return forecasting framework based on graph neural networks. Our proposed framework can be interpretable while maintain predictive performance. To maintaining predictive performance, the framework takes advantage of several graph layers to make forecasting of individual stock with the help of information from other related stocks. To acquire interpretability, we further divide the individual stock return into multiple parts, and each part corresponds to one graph convolutional layer. Accurately, the return of the particular stock is summarized as following four parts in this paper: 1) Return brought by the factors of the individual stock. For example, the company's expansion of production scale or increase of revenue will make stock prices tend to rise. 2) Return brought by the impact of the company's main business. 3) Return brought by the effects of the industry. 4) Return brought by the effects of the overall situation of the market.
This paper uses a large number of experiments and various evaluation metrics to prove the effectiveness of the proposed framework. At the same time, we carried out an interpretability analysis on the stock return forecasting. We explained the stock return according to the above four parts by comparing the predicted value with the actual market situation.

2. Related Work
In recent years, many methods have been proposed to improve the interpretability of deep learning models. In this paper, we roughly summarize the techniques into three categories. The first category refers to the model-independent interpretation method. The most classic one is Local Interpretable Model-Agnostic Explanations (LIME) [3]. The second category relates to the optimization-based approach, such as optimizing the structure of the trained model [4] and optimize the learning process to directly obtain a model that owns interpretability. The last category refers to visualization-based methods such as the deconvolution method [5], Lucid [6], and Class Activation Mapping [7].

In addition to the methods mentioned above, some interpretability methods take full account of the properties of the problem, and design model according to the properties to make models own interpretability. Take the time series forecasting problem as an example. Prophet [8] proposed by Facebook choose to divide the future trend of time series into three dependent items: trend item, seasonality item, and holiday item to make forecasting and explain the future trend. Similar to Prophet, in this paper, we propose an interpretable framework to make stock trend forecasting and explain the forecasting by dividing them into several dependent parts.

3. Graph-based Interpretable Stock Trend Forecasting Framework
To make the neural network interpretable to some extent while maintaining the performance, in this paper, we propose a Graph-Based Interpretable Stock Trend Forecasting Framework (GIFF). Intuitively, the complex influence between stocks may play a major role in the future trend of stocks. Some work [9,10] focus on predicting stock future trend based upon stock-wise information together with the influence between stocks and have achieved an absolute improvement. Inspired by this, we choose to utilize graph-based architecture to build connections between stocks to make forecasting. Furthermore, to obtain interpretability, we divide the stock returns into multiple parts, and each part corresponds to a graph layer in GIFF. In this part, we will introduce the framework of GIFF and then describe the learning method when training GIFF.

3.1. Architecture of GIFF
As illustrated in Figure 1, GIFF contains three parts, named a sequential embedding module, a residual graph module, and a forecasting module, which are elaborated as follows.

Since the historical trend of the stock is one of the most influential factors in predicting its future trend, a sequential embedding module is proposed to capture the pattern along time. As one of the most classic time-series architecture, many studies have shown that LSTM is effective in the problem of stock trend prediction [11,12]. Therefore, we choose to utilize LSTM as the sequential embedding module to produce sequential embedding \( e_t^v \) for each stock \( v_i \) at time \( t \) as Equation (1), where \( x_t^v \) refers to the stock-wise features for stock \( v_i \) at time \( t \).

\[
e_t^v = \text{LSTM}(x_t^v, x_{t-N}^v, ..., x_{t-N+1}^v)
\]  

Based on the assumption that stock trends are independent of each other, many existing quantitative models merely based on stock-wise information to predict the future trend of individual stocks. In the actual scenario, there exists a certain correlation between stocks, which can be greatly helpful when making forecasting [13,14]. Therefore, we utilize the residual graph module to use pre-defined relations between stocks to model the stock market and further make forecasting. Notably, the residual graph module contains three layers, they are a business-based layer, an industry-based layer, and a fully-connect layer in sequence, which is introduced as follows.
1) The business-based layer builds connections between stocks that operate the same business. Intuitively, the information from other stocks which operating the same business may benefit forecasting for the return brought by the corresponding business.

2) The industry-based layer builds connections between stocks that belong to the same industry. Similar to the business-based layer, the information from other stock belonging to the same industry may benefit forecasting for the return brought by the corresponding industry.

3) The fully-connect layer builds connections between each stock pairs. The motivation for this approach is to make individual stocks utilize the overall information in the market to obtain the return brought by the whole market.

The three layers introduced above are arranged to form the residual graph module. Similar to the Temporal Graph Convolution Layer in [13], considering that the influence between two stocks may have varying impacts on their future trend, we apply a non-uniform coefficient $g(e_i^t, e_j^t)$ when propagating the embedding from stock $v_j$ to $v_i$:

$$
\tilde{e}_i^t = \Sigma_{(j|a_{ji}>0)} g(e_i^t, e_j^t)e_j^t
$$

$$
\alpha_{ji} = \sigma(W^T e_i^T e_j^T T + b)
$$

$$
\sigma(e_i^t, e_j^t) = \frac{e^{a_{ji}}}{\Sigma e^{a_{ji}}}
$$

where $a_{ji}$ refers to the number of connections between stock $v_j$ and $v_i$. In the business-based layer, $a_{ji}$ refers to the number of businesses operated by both stock $v_i$ and stock $v_j$. In the industry-based layer, if stock $v_i$ and stock $v_j$ belong to the same industry, the value of $a_{ji}$ is 1, otherwise is 0. In the fully-connect layer, the value of $a_{ji}$ is 1 for each $v_j$ and $v_i$. $W$ and $b$ are model parameters to be learned and $\sigma$ refers to activation function, which is specified as Sigmoid in this paper.

Notably, the stock information flows in the graph module with the form of stock embedding. Finally, the forecasting module is proposed to combine the embedding linearly to produce forecasting for each stock. As a regression task, the predicted value $p(v_i^t)$ for the stock $v_i$ will be compared with the rising percent $r(v_i^t)$ to calculate loss. The model parameters will be updated by optimizing the target as following:

$$
\min \Sigma_{i=1}^n (p(v_i^t) - r(v_i^t))^2
$$
where \( n \) refers to the number of stocks at time \( t \).

### 3.2. Learning method

In the learning process, we propose to firstly layer-by-layer pre-training and then end-to-end fine-tuning the model. The two-stage learning method is designed to improve the interpretability of GIFF. We choose to firstly pre-train the sequential embedding module and forecasting module. Secondly, we perform the next round of pre-training on the business-based layer to fit residual of the sequential embedding module, then fixing the parameters in the industry-based layer and the fully-connect layer in sequence. The purpose of the layer-by-layer pre-training is to ensure that the stock returns learned in each layer are those that cannot be explained by the previous layer. Specifically, the sequential embedding module will reveal part of the return based on the stock-wise information. For the return that LSTM cannot explain, we try to utilize information from other stocks that operate the same business to explain it. The following layers in the residual graph module are the same. After the process of layer-by-layer pre-training is completed, we need to end-to-end fine-tune the framework to complete the training process.

Besides, it should be noticed that when training GIFF, each batch should contain information of all the stocks on one trading day so that the information could propagate between stocks. The situation is the same during the valid and test period.

The pre-defined relation used in this paper can be replaced by any relation between stocks, such as upstream-downstream relation, as long as they can independently explain a part of the stock return.

### 4. Experimental Result and Analysis

In this paper, we try to utilize a group of experiments to demonstrate the effectiveness of GIFF in stock trend forecasting. Furthermore, we will analyze the industry-based layer in the residual graph module to show the interpretability of GIFF.

#### 4.1. Data description

We conduct experiments with stock data from the most famous Chinese Securities Index 300 (CSI300) and Chinese Securities Index 500 (CSI500). CSI300 aims to reflect the situation of large market capitalization stocks, and CSI500 reflects the overall performance of mid-to-large and small-to-mid capital A-shares. In the experiment, we set one trading day as the forecasting cycle. The stock data includes the highest price, the lowest price, the open price, the close price, the volume-weighted price, and the trading volume. The pre-defined relation data, including the business and industry information of stocks, corresponding to the business-base layer and industry-based layer in the residual graph module. The data from the CSI300 stock pool utilized ranges from 01/01/2007 to 12/31/2019. The training data ranges from 01/01/2007 to 12/31/2015, and valid data ranges from 01/01/2016 to 12/31/2016, and test data ranges from 01/01/2017 to 12/31/2019.

#### 4.2. Compared methods and evaluate metrics

To demonstrate the effectiveness of GIFF in stock trend forecasting, we conduct a series of comparative experiments. The first group of comparative experiments utilizes MLP and LSTM to regress future rise percent of individual stock based on stock-wise information. The second group experiments add consideration of the influence between stocks. In this paper, Graph Attention Networks (GAT) [15] are utilized.

We employ a set of metrics to show the effectiveness of the GIFF. Notably, both the return and the forecasting on the same trading day will be normalized to calculate Mean Square Error (MSE) to quantify the prediction error. Information Coefficients (IC) can measure the correlation between forecasted return and real return. Information Coefficient Information Ratio (ICIR) is used to quantify the IC balanced by volatility during the test period. Finally, a straightforward strategy Long-short (LS) is utilized to calculate Annualized Return (AR) and Sharpe Ratio (SR) to measure excess return and excess return balanced by risk. LS strategy refers to invest in the market by buying long 10% stocks
with the highest forecasted return and short selling 10% stocks with the lowest forecasted return. Notably, the transaction fee is not considered.

4.3. Experimental result
The performance comparison of different methods in CSI300 and CSI500 is demonstrated in Table1 and Table2, respectively.

### Table 1: Experimental results of the CSI300 stock pool.

|        | IC  | ICIR | MSE   | AR  | SR  |
|--------|-----|------|-------|-----|-----|
| MLP    | 0.092 | 0.765 | 1.832 | 0.751 | 0.626 |
| LSTM   | 0.095 | 0.776 | 1.825 | 0.772 | 0.631 |
| GAT    | 0.096 | 0.893 | 1.822 | 0.796 | 0.688 |
| GIFF   | 0.098 | 0.925 | 1.811 | 0.799 | 0.691 |

### Table 2: Experimental results of the CSI500 stock pool.

|        | IC  | ICIR | MSE   | AR  | SR  |
|--------|-----|------|-------|-----|-----|
| MLP    | 0.113 | 1.085 | 1.725 | 0.901 | 0.852 |
| LSTM   | 0.121 | 1.092 | 1.721 | 0.917 | 0.847 |
| GAT    | 0.123 | 1.117 | 1.711 | 0.936 | 0.883 |
| GIFF   | 0.123 | 1.118 | 1.711 | 0.938 | 0.883 |

As shown in Table1 and Table2, GIFF outperforms other models almost in both of the stock pools. Besides, the performance of GIFF and GAT are generally better than the first group of experiments (MLP and LSTM). This phenomenon shows that the utilization of the relation between stocks does help to forecast future trends for stocks.

### Figure 2: Comparison between cumulative forecasting and cumulative excess return for the electronics industry.

### Figure 3: Comparison between cumulative forecasting and cumulative excess return for the commercial industry.

4.4. Interpretability analysis
In this part, we will discuss the interpretability of GIFF. Take the industry-based layer in the residual graph module as an example. This part tries to figure out the correlation between the average industry return with the forecasting given by the industry-based layer.

Take the electronics industry as an example, we calculate the forecasting of the electronics industry return by averaging industry-based layer residual forecasting for stocks in the electronics industry. We calculate the electronics industry’s return as the average excess return of the stocks in the electronics
industry. The calculation of the commercial industry is the same as the electronics industry. The comparison between the forecasting and return of the the same industries is demonstrated in Figure2 and Figure3. The prediction and return of the industry in the examples have an absolute correlation when the return of the industry shows stability pattern (continue to rise or fall) for a period of time. In this way, GIFFF owns interpretability to account for the return brought by the corresponding industry. Limited by space, the interpretability analysis for the business-based layer and fully-connect layer will not show in this paper.

5. Conclusion
Though its outstanding forecasting performance, the lack of interpretability brings many limitations in utilizing quantitative models in the practical investment scenario. In this paper, we propose a graph-based framework, which owns interpretability while maintaining forecasting performance. The framework divides the return into four independent parts to make forecasting. Extensive experiments demonstrate the effectiveness of the framework we proposed. Furthermore, we analyze industry-based returns to show the interpretability of the framework.

Acknowledgments
The authors are supported by the National Key R&D Program of China 2018YFB1004100.

References
[1] Zhu, Y., & Zhou, G. (2009). Technical analysis: an asset allocation perspective on the use of moving averages. Journal of financial economics, 92(3), 519-544.
[2] Tan, N. (2017). Fundamental analysis and the cross-section of stock returns: a data-mining approach. Cfa Digest, 47(10).
[3] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?": explaining the predictions of any classifier.
[4] Frosst, N., & Hinton, G. (2017). Distilling a neural network into a soft decision tree.
[5] Zeiler, M. D., Taylor, G. W., & Fergus, R. (2012). Adaptive deconvolutional networks for mid and high level feature learning. 2011 International Conference on Computer Vision. IEEE.
[6] Olah, C., Satyanarayan, A., Johnson, I., Carter, S., & Mordvintsev, A. (2018). The building blocks of interpretability.
[7] Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., & Batra, D. (2016). Grad-cam: why did you say that? visual explanations from deep networks via gradient-based localization.
[8] Taylor, Sean, J., Letham, & Benjamin. (2018). Forecasting at scale. American Statistician.
[9] Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2014). News impact on stock price return via sentiment analysis. Knowledge Based Systems, 69(oct.), 14-23.
[10] Tan, N. (2017). Fundamental analysis and the cross-section of stock returns: a data-mining approach. Cfa Digest, 47(10).
[11] Minami, S. (2018). Predicting equity price with corporate action events using lstm-rnn. Journal of Mathematical Finance, 08(1), 58-63.
[12] Zhuge, Q., Xu, L., & Zhang, G. (2017). Lstm neural network with emotional analysis for prediction of stock price. Engineering Letters, 25(2), 167-175.
[13] Feng, F., He, X., Wang, X., Luo, C., Liu, Y., & Chua, T. S. (2019). Temporal relational ranking for stock prediction. Papers.
[14] Mehtab, S., & Sen, J. (2020). Stock price prediction using convolutional neural networks on a multivariate timeseries.
[15] Velčíkovíč, Petar, Cucurull, G., Casanova, A., Romero, A., Liò, Pietro, & Bengio, Y. (2017). Graph attention networks.