Use Information Flow Graph Attention Network to Optimize Stock Movement Prediction

Jifeng Sun, Huan Tong, Tsinghua-Berkeley Shenzhen Institute
tongh19@mails.tsinghua.edu.cn

Abstract

Stock prediction can help investors make better investment decisions by predicting the future trends of a stock. However, most methods for stock prediction are based on time-series models, treating the stocks as independent from each other. Relations between stocks like two stocks have a same investor or the companies are in cooperation relationship, are out of consideration. What’s more, it is observed that the relations between stocks not only exist, but also have the characteristics of two-way inequality. In this work, we design a new graph attention network, named Information Flow Temporal Graph Network （IFTGAT）, for stock prediction and compare the predicting accuracy with several baseline model like LSTM, Temporal Graph Convolutional Neural Networks（T-GCN）, Temporal Graph Attention Networks（T-GAT）. Experiments show that IFGAT defeats GCN's prediction results and can bring greater improvements to the stock prediction effect.

1 Introduction

Stock prediction has been a hotspot in the financial and academic circles. Since the birth of the stock market, specialists in many countries have tried various methods to predict the time series of stock prices. Traditional predicting models such as Kalman Filters, Autoregressive Models and their extensions mainly focus on modeling historical data of a stock and exploring its underlying laws in order to predict the future trend of stock prices, which does not take the interrelationship between stocks into consideration. Relations in stocks could be a very valuable signal for stock prediction. For example, the stock prices of Huawei and Samsung, both from mobile communication field, have similar long-term development trends and may function to each other. Therefore, we try to transform Euclidean data into graph-structured data to build a stock relationship graph, and naturally integrate the consistency pattern and characteristic attributes of graph-structured data.

In addition, we also found that there is a two-way inequality in the correlation between the two stocks, which means there is causality between the correlated stock price time series data. For instance, the impact of the supplier's stock price on the customer may be greater than the customer's impact on the supplier's stock price. The amount of information exchange between stocks explains not only the size magnitude but also the direction of the cause-effect relation. By using the time rate of information flowing from one sequence to another, we can use the method of information flow to measure the causation between stocks. Then we choose the graph attention neural network（GAT） to apply the relations into the stock prediction.

GAT is a new neural network structure that operates on graph-structured data. It uses a hidden attention layer to weight the sum of features of neighboring nodes. The weight of the features of the neighboring nodes is totally dependent on the features of the nodes and is independent of the graph structure. It makes up for the shortcomings that the graph convolutional neural network (GCN) needs to rely on graph structure on the features of neighboring nodes, and enhances the generalization ability of the trained model on other graph structures.

In this work, our input is a graph combing the information of historical time series data of each stock. Then we use the information flow graph neural network to output a graph with predicted closed prices of each stock. During the process, first we use a Long-Short Term Memory (LSTM) network to capture the sequential dependencies and learn a stock-wise sequential embedding. Then the sequential embeddings will be revised by accounting for stock relations. Finally, we feed the concatenation of sequential embeddings and relational embeddings to a fully connected layer to obtain the ranking score of stocks. To justify our proposed method, we employ it on two real-world markets, New York Stock Exchange (NYSE) and NASDAQ Stock Market (NASDAQ).

In summary, our innovation lies in designing a new graph neural network, named information flow graph neural network (IFGAT) based on information flow to extract causal relations between stocks. The comparison of the results of IFGAT and GCN confirms the superiority of our new model in predicting financial time series data.

2 Related work

As graph neural network is relatively a new solution for stock prediction, not much analysis has been done on this topic. The most relevant past work in this area was conducted in the recent 5 years.

S. Deng et al. noticed the traditional models' ignorance of the background knowledge of stocks and offered a knowledge-driven Model to capture inconsistent evolution of stream data. However, their work was still based on LSTM model. Y. Chen and Z. Wei pioneered the use of 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 prove the superiority of GCN model in stock prediction. Q. Li et al analyzed the key reason 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.

P. Veličković et al. introduced an attention-based architecture to perform node classification of graph-structured data via masked self-attentional layers to solve the problem of prior methods based on graph convolutions. The attention architecture conducted a self-attention strategy to compute the hidden representations of each node in the graph which can be applied to graph nodes with different degrees by specifying arbitrary weights to the neighbors. The attention strategy is the key point for GAT to get rid of the limitations of graph structure. X. Liang unraveled the cause-effect relation between time series. He used information flow to measure the causation between dynamical events and showed that causality analysis can be rigorously formulated and quantitatively realized, and that the resulting formula turns out to be remarkably concise and very easy to compute. The conclusion has been applied successfully to the investigation of the cause-and-effect relation between the two climate modes, El Ni˜no and the Indian Ocean Dipole (IOD), which have been linked to hazards in far-flung regions of the globe.

While these papers have made large strides in using machine learning to solve problems. They did not provide a perfect strategy which takes the two-way unequal relationship between stocks into account. Our work captures the ideas of information flow and GAT model. We will propose a new graph neural network (IFGAT) and apply it to improve the accuracy of stock prediction.

3 Dataset and Features

In order to conduct extensive evaluations, we fetched data on our own to construct a large number of stock datasets.

The basic data source is the stocks traded on the NASDAQ exchange market from January 2, 2013 to August 12, 2017, with a total of 3,274 stocks. We chose the NASDAQ market because it is more unstable and can better prove the prediction level of our model. Based on the original data, we have retained stocks that meet the following two conditions: 1) Since January 2, 2013, its trading day has exceeded 98%. 2) During the collection period, the transaction price has never been lower than $ 5 per share. The purpose of this is to 1) eliminate the interference factors of abnormal patterns that may be caused by intermittent sequences. 2) Make sure the stock you choose is not a penny stock which is too risky for the average investor. After filtering, a total of 1026 stocks from the NASDAQ market were generated.

After determining the types and amounts of stocks, we need to further collect the three kinds of data required for the experiment: 1) historical price data, 2) sector-industry relations, and 3) Wiki relations between their companies.

3.1 Historical price data

We selected the daily trading data of each stock and predict the closing price on the t + 1 trading day based on the closing price of the last t trading days. For the historical price of each stock, we have made a unified treatment to reflect their price level more fairly. First collect the daily closing price of each stock from January 2, 2013 to August 12, 2017. Then divide the price of each stock by the maximum value of the entire data set to get the normalized closing price. In addition to the rise and fall of a single stock, we also considered the impact of the stock price ranking on the predicting result, and ranked each stock by the stock return. The real ranking of stock i is set to its 1-day yield $r_i^{t+1} = (P_i^{t+1} - P_i^t)/P_i^t$ where P_i^t is the closing price on day t. In addition to the normalized closing prices, we also calculate four consecutive characteristics: 5, 10, 20, and 30-day moving averages, which represent weekly and monthly trends, respectively. Finally, we divided the sequential data into three time periods for training (2013-2015), verification (2016), and evaluation (2017) in chronological order. As we can see from Table 3.1, there are 756, 252, 237 days for training, verification, and evaluation respectively.

Stocks#	Training	Validation	Testing
	days#	days#	Days#
	01/02/2013	01/04/2016	01/03/2017
	12/31/2015	12/30/2016	12/08/2017
1026	756	252	237

Table 3.1: Statistics of the sequential data from NASDAQ

3.2 Sector-industry Relations

Each stock belongs to an industry, so we need to sort out the relationship between each stock and industry under the same industry node. We collect a hierarchy of NASDAQ stocks from an official company list maintained by NASDAQ, for example (Amazon; Computer Software: Programming, Data Processing). It is calculated that there are 112 types of industries in the Nasdaq market. Due to sparse industry relationship data, only about 5% of stock pairs have at least one type of industry relationship.

3.3 Wiki Company-based Relations

In addition to industry relations, we also need to explore the impact of company relations such as investment relations and supply relations on stocks. We chose Wikidata for corporate relations, which is one of the largest and most active open domain knowledge bases with more than 42 million projects and 367 million sentences. We divided company relationships into first and second order. As shown in Figure 3.1, if there are statements that use A and B as subjects and objects, respectively, then companies A and B have a first-order relationship. If the statements of companies A and B have the same object, they have a second-order relationship. For example, Oracle acquired Aconex. Then Oracle and Aconex belong to the first-order relationship. After an exhaustive exploration of recent Wikidata dumps, we obtained 5 first-order relationships and 53 pairs of second-order relationships, respectively. We then summarize the number of relationship types and the ratio of stock pairs with at least one Wiki-based relationship in Table 3.2. As can be seen, there are a total of 97 kinds of corporate relationships between the stock pairs of the NASDAQ exchange.

Sector-Industry Relation		Wiki Company-based Relations	
Relation Types#	Relation Ratio (Pairwise)	Relation Types#	Relation Ratio (Pairwise)
112	5.00%	97	0.21%

Table 3.2: Statistics of sector-industry relation and Wiki relation data in the NASDAQ dataset

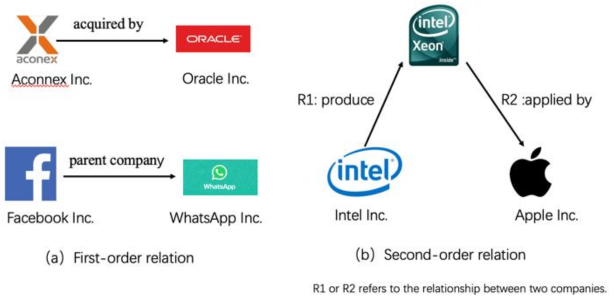


Fig. 3.1 Examples of first- and second-order company relations

4 Methods

In this paper, the goal of our method is to predict future stock prices from stock relation network and historical stock price information.

4.1 Problem Definition

Definition 1: Stock relation network. We use the graph G = (V, E) to describe the topological structure of the stock relationship network. Each stock is a vertex. V is the vertex set, V = {v1, v2 ,…, vN}, N is the number of vertices, and E is the edge set.. The adjacency matrix A represents the relationship of stocks, $A \in R^{N \times N}$.

In this paper, we use the adjacency matrix to express the relationship network. We selected a total of 42 company associations on wiki data, and constructed a total of 42 matrix graphs. It is 1 if there is a connection between two stocks, otherwise it is 0. This has been explained in detail in the previous section. Because these matrices are very sparse, we have compressed them through a fully connected layer. Therefore, the initial input of the model is N * N * types, and after compression, the total is N * N * 1 dimensions.

Definition 2: Feature matrix $X^{N \times P}$. We consider the stock price information on the trading day as the characteristics of the stock. The characteristics of each stock on each trading day are [open, high, low, close, volume], 5-dimensional data. The time step we choose is 4, which represents 4 days. P represents the number of features, and $Xt \in R^{N \times i}$ 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.

$$[X_{t+1},\cdots,X_{t+T}] = f\,(G;\ (X_{t-n},\ \cdots,\ X_{t-1},\ X_t)) \qquad (1)$$

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 price information for the next trading day.

4.2 Overview

In this paper, to capture the stock relations and temporal dependences from the dataset at the same time, we propose a new kind of network model names information flow temporal graph attention network((IFTGAT)). we take the network architecture of T-GCN and GAT as the prototype, add the Information Flow attention mechanism we designed to the model, and design our own network structure. Then we verified the effectiveness of our model through comparative experiments with other models to test its performance.

4.3 Methodology

4.3.1 Relation Attention Modeling

As mentioned earlier in the article, we assume that there is a two-way unequal transmission of information between stocks and it has been verified in experiments. We hope that the model can take this relationship into account during training, so according to the idea of information Flow, we designed a Graph attention mechanism, we call this attention mechanism Information Flow Attention or IF Attention. Information flow describes the causal relationship between two time series. It was published in xxx. It was originally widely used in climate prediction. The causality is measured by the time rate of information flowing from one sequence to another. More detailed process about Information Flow can refer to the paper. In our article, we only use its conclusions. The formula in the paper is as follows:

$$T_{2\to1} = \frac{C_{11}C_{12}C_{2,d1} - C_{12}^2 C_{1,d1}}{C_{11}^2 C_{22} - C_{11}C_{12}^2} \qquad \textbf{\textit{(2)}}$$

$$\dot{X}_{j,n} = \frac{X_{j,n+k} - X_{j,n}}{k\Delta t} \qquad \textbf{\textit{(3)}}$$

Where c stands for xxx and x stands for xxx. After inputting the normalized time series data, the output result is a floating point number that roughly falls between -1 and 1.

As we all know, whether it is CNN or GCN, it can be approximated as a kind of parameter feature transformation in local space and weighted average of surrounding points. Regarding the original Graph Attention Layer, you can refer to GRAPH ATTENTION NETWORKS ICLR 2018. In the original text, there are four steps to graph attention:

1.Doing feature transformation of nodes using learnable parameters W to get $W^{(l)}\boldsymbol{h}_{\boldsymbol{i}}^{(l)}, W^{(l)}\boldsymbol{h}_{\boldsymbol{j}}^{(l)}$.

2. Stitch the embedding of two nodes. Note that || represents stitching here; then dot product the stitched embedding and a learnable weight vector $\vec{a}^{(l)^T}$.

3. use $LeckyReLU$ to enhance the ability of non-linear expression to obtain

4. use $\boldsymbol{Softmax}$ function to normalize the above results and finally get the probability of attention $a_{ij}^{(l)}$.

The four-step formulas are as follows:

$$\boldsymbol{z}_i^{(l)} = W^{(l)}\boldsymbol{h}_{\boldsymbol{i}}^{(l)} \qquad \textbf{(4)}$$

$$e_{ij}^{(l)} = LeckyReLU\left(\vec{a}^{(l)^T}(\boldsymbol{z}_i^{(l)}||\boldsymbol{z}_{\boldsymbol{j}}^{(l)})\right) \qquad \textbf{(5)}$$

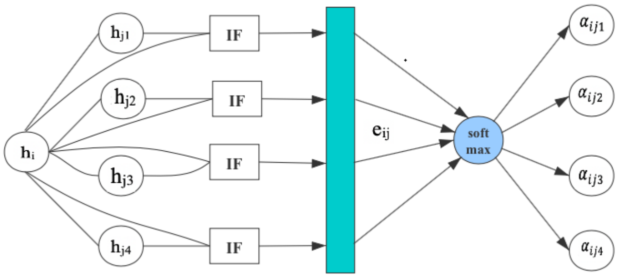
$$a_{ij}^{(l)} = \frac{exp\left(e_{ij}^{(l)}\right)}{\sum_{k\in N_i}\exp\,(e_{ik}^{(l)})} \qquad \textbf{(6)}$$

$$h_i^{(l+1)} = \sigma(\textstyle\sum_{j\in N_i}\alpha_{ij}^{(l)}z_j^{(l)}) \qquad \textbf{(7)}$$

In the original paper, graph attention is a kind of attention based on learning totally. From the formula, we can see that it does not rely on any prior knowledge. This may achieve amazing results on some tasks, however in financial time serial modeling, due to the high noise and sharp fluctuations of financial data, people often construct various prior knowledge indicators to process financial data when modeling. Therefore, it is meaningful to introduce a priori information into the attention mechanism.

Compared with the original graph attention mechanism, the formula is very similar. The difference is that the new mechanism uses the information flow process instead of the stitching operation.

The former is usually called additive attention, and our method can be understood as a special multiplicative attention []



$$\boldsymbol{z}_i^{(l)} = W^{(l)}\boldsymbol{h}_{\boldsymbol{i}}^{(l)} \qquad \textbf{(8)}$$

$$IF_{ij}^{(l)} = informationFlow(\boldsymbol{z}_{\boldsymbol{i}}^{(l)}, \boldsymbol{z}_j^{(l)}) \qquad \textbf{(9)}$$

$$e_{ij}^{(l)} = LeckyReLU\left(\vec{a}^{(l)^T}IF_{ij}^{(l)}\right) \qquad \textbf{(10)}$$

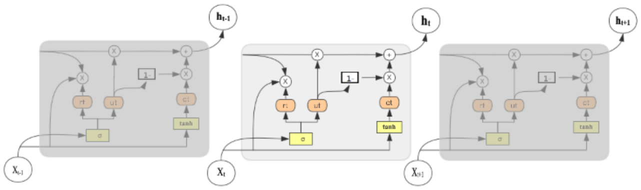
$$a_{ij}^{(l)} = \frac{exp\left(e_{ij}^{(l)}\right)}{\sum_{k\in N_i} \exp\left(e_{ik}^{(l)}\right)} \qquad (11)$$

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in N_i} \alpha_{ij}^{(l)} z_j^{(l)}\right) \qquad (12)$$

4.3.2 Temporal Dependence Modeling

Modeling time dependencies is another key issue in related-stock forecasting work. The most widely used The neural network model used to process the sequence data are LSTM [49] and GRU[50] now. They both use a gating mechanism to memorize long-term information, which are efficiency methods for various tasks. because GRU's simple model structure can speed up the training progress, on the other hand, it is also used in [], so we also used it in the model structure in the end.

As shown in Figure 5, ht−1 indicates the hidden state at time t-1; xt is the stock information at time t; ut and rt 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 ht. 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 timing dependencies.

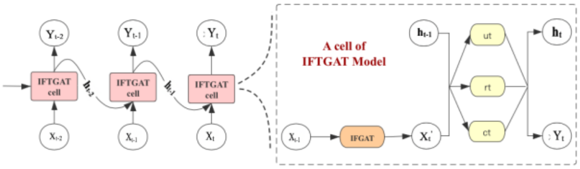


the Structure of the gated recursive unit model.

4.3.3 Information Flow Graph Attention Network

The following figure shows the entire process of the network (IFTGAT) we designed. The figure on the left represents the entire process of stock correlation-time series prediction we designed. The internal structure of the IFGAT cell is shown on the right.

ht-1 represents the output at time t-1, IFGAT has a graph convolution process for information flow graph attention, while ut, rt are the update and reset gates at time t, and ht represents the output at time t.



The whole process of graph-related financial time series prediction is shown in the figure. The right side represents the internal structure of the IFTGAT cell that makes up the recursive model. IFGAT represents the IF graph attention convolution process.

The specific calculation process is shown below. IFGAT (A, X t) represents the IF graph attention convolution process and is defined by Equation 2. W and b represent weights and biases during training

$$if = IFGAT(A, X_t) \qquad (13)$$

$$u_t = \sigma(Wu[if, h_{t-1}] + b_u) \qquad (14)$$

$$r_t = \sigma(Wu[if, h_{t-1}] + b_r) \qquad (15)$$

$$c_t = \tanh\left(Wc[if, (r_t * h_{t-1})] + b_c\right) \qquad (16)$$

$$h_t = u_t * h_{t-1} + (1 - u_t) * c_t \qquad (17)$$

To sum up, the IFTGAT model can handle the complex correlation and time dependence in the related stock prediction. On the one hand, through the information flow rule, the model specifies the attention weight to the neighbor of each node, focusing on those nodes with larger effects while ignoring some nodes with smaller effects, which enhances the effect of model prediction. On the other hand, the gated recursive unit is used to capture the dynamic change of stock price to obtain the time dependence of stock data and finally realize the related stock prices forecasting task.

4.3.4 Loss Function

The first term is used to minimize the mean square error between the real stocks' price and the prediction. We use $\boldsymbol{Y_t}$ and $\boldsymbol{\widehat{Y}_t}$ to denote the real price and the predicted price. The second term Lreg is an L2 regularization

$$\boldsymbol{loss} = \left|\left|\boldsymbol{Y_t} - \boldsymbol{\widehat{Y}_t}\right|\right| + \lambda L_{reg} \qquad (18)$$

5 Experiments

5.1 Hyperparameter

The hyperparameters of the IFTGAT model mainly include:

learning rate, batch size, training epoch, and the number of hidden layers. We usually determine relatively good hyperparameters based on empirical criteria or some methods such as grid search. Among these hyperparameters, some hyperparameters are very important like the number of hidden units and directly related to the quality of the model or even whether it converges. For the relationship of training time, we did not test the effects of multiple sets of parameters too much. We just selected some parameters based on past experience and the method recommended in the paper. In the future, we will test more parameters and adopt more methods that can improve the model.

In the experiment, we manually adjust and set the learning rate to 0.001, the batch size to 32, the training epoch to 200, number of hidden layers of the model to 64 and The time step is set to 4, because financial data has the problem of short dependence, so the first four days are used to predict the price on the fifth day.

5.2 Model evaluation

We use $\boldsymbol{Y_t}$ and $\widehat{\boldsymbol{Y}}_{\boldsymbol{t}}$ to denote the real price and the predicted price. In this paper, we use Root Mean Squared Error (RMSE) to evaluate the model performance.

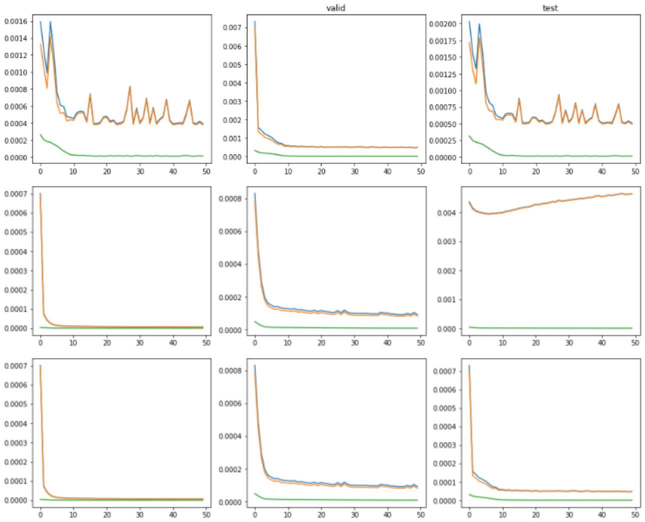
5.3 Training

For input layer, first 60% of all data is used for training, the middle 20% for validation, and the final 20% for testing. All model are trained using the Adam optimizer.

5.4 Environment

The models run on pytorch1.x with RTX2080Ti,cuda10.1

5.5 Experimental Results



What is shown in this figure is the prediction of the lstm model's performance loss during the training process. From left to right, it is the performance on the training set, validation set, and test set.
The blue line is the root-mean-square error mse, which can measure whether the model is effective, and the green one is the stock loss's rank loss, that is, the predicted loss between the relative ranking of stocks and the loss of real income ranking. In actual trading, it can effectively reflect whether the model is suitable for trading, because for real investment scenarios, the fund manager will choose the stocks in the strategy model that predict better returns. The orange line is the sum of the two . From top to bottom is the performance of the model on lstm, T-GCN and IFTGAT.

6 Result analysis

From the performance effects of the various graphs, we can see that although the graph neural network model's initial epoch loss during training is greater than the recurrent neural network model, after a period of iteration, the prediction ability is stronger than the classic time series deep learning model Like LSTM. The reason for the sudden change in expression on the training set of t-gcn is still under investigation
In addition, our model IFTGAT has achieved the best results in several graph neural network models, but at the same time we must admit that IFTGAT model training takes a long time. Due to lack of time, here we There is no comparison of the training costs of each model. We will give specific training costs and ideas for model optimization in subsequent studies.
From the results comparison chart, we can see that the loss of the several models we use has been decreasing on the data set, which can effectively fit the data. The interesting thing is that the performance of the several models on the validation and test sets is better In the performance on the training set, we guess that this may be related to the characteristics of the financial data itself, because this phenomenon has not been found in other data, and we will explore this issue in subsequent studies.

7 Conclusion

Proofs must be written in their own paragraph separated by at least 2pt and no more than 5pt from the preceding and succeeding paragraphs. Proof paragraphs should start with the keyword ``Proof." in 10pt italics font. After that the proof follows in regular 10pt font. At the end of the proof, an unfilled square symbol (qed) marks the end of the proof.

Proof. This paragraph is an example of how a proof should look like. □

8 Future Work

Due to time, this project has a lot of work that can be added in the future.

1. Enrich the backtesting mechanism. There is no backtesting for investment in this experiment, and the analysis of the results is not intuitive.

2. More models. In this project, we only compared commonly used deep learning models, but classic machine learning models such as armia, LA and other models have not been introduced. In addition, deep learning models such as TCN and LSTM should also be used. Compare with

3. Refine the model. In this project, the choice of parameters
4. More data. In this project, due to the selection of the data source, the latest one-year data was not introduced into the model. In addition, it is also worthwhile to test the performance of the model on the data of the A-share market

References

[Abelson *et al.*, 1985] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs.* MIT Press, Cambridge, Massachusetts, 1985.

[Baumgartner et al., 2001] Robert Baumgartner, Georg Gottlob, and Sergio Flesca. Visual information extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Databases*, pages 119–128, Rome, Italy, September 2001. Morgan Kaufmann.

[Brachman and Schmolze, 1985] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April-June 1985.

[Gottlob et al., 2002] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences*, 64(3):579–627, May 2002.

[Gottlob, 1992] Georg Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, June 1992.

[Levesque, 1984a] Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212, July 1984.

[Levesque, 1984b] Hector J. Levesque. A logic of implicit and explicit belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, pages 198–202, Austin, Texas, August 1984. American Association for Artificial Intelligence.

[Nebel, 2000] Bernhard Nebel. On the compilability and expressive power of propositional planning formalisms. *Journal of Artificial Intelligence Research*, 12:271–315, 2000.