

# LLM-Driven Knowledge Enhancement for Securities Index Prediction

Zaiyuan Di, Jianting Chen, Yunxiao Yang, Ling Ding and Yang Xiang\*

Tongji University, No. 4800, Cao'an highway, JiaDing District, Shanghai city, 201804, Shanghai, China

## Abstract

The securities market carries complex financial interactions, providing challenges to its prediction. To represent this complexity, researchers have utilized multi-source data, such as financial news and macro market indicators, for better performance. However, these efforts often ignore the internal knowledge among these data or suffer from the high cost of acquiring diverse knowledge. Thus, we propose a LLM-driven knowledge enhancement method for securities index prediction. Specifically, we collect the daily data of Shanghai Stock Exchange indexes and their related market indicators and model the internal knowledge among them as triplets. Then we leverage LLM as a knowledge base to acquire diverse knowledge efficiently. Finally, we integrate the knowledge and numeric multi-source data as a heterogeneous graph and apply a GNN model to predict the trend of securities indexes. Experiments demonstrate the effectiveness of our method in prediction and real-world backtest.

## Keywords

Stock Market Prediction, Large Language Models, Knowledge Enhancement

## 1. Introduction

Securities prediction has always been a challenging but engaging task. Many studies [1, 2, 3, 4, 5] have already proven the feasibility of predicting securities based on historical price data. These efforts have further made securities prediction a promising profit opportunity for investors and a classic time series prediction task with valuable research significance.

The securities market is a complex environment with various entities and events, often requiring time for their interactions to be fully reflected in price data. For example, the management change information in the announcement will affect its stock price, and the crude oil price will affect the stock price of airline companies. These facts inspire researchers to utilize external information for more accurate price prediction. Thus, multi-source data, such as sentiment information [6, 7, 8], semantic information from news and posts [9, 10, 11], and information on companies related to the predicted stock [12, 13], has been widely explored.

Substantial efforts have been made to leverage this multi-source data. Nevertheless, many treat these data in isolation, ignoring the relations' role in promoting prediction performance [14, 15], for example, representing all information as concatenated vectors [16, 17]. Compared with these methods, some work models the internal relations of data and incorporates them into predictions utilizing graph-based models, resulting in good performances. They usually represent the relations as knowledge graphs [12, 13] or correlation matrixes [18]. However, these methods still have the following challenges: 1) Since knowledge acquisition from tremendous raw data is expensive, it is not always applicable [12]. 2) Meanwhile, many rely on existing knowledge sources or correlation calculations, both suffer from the limited variety of relation and entity types [13, 18].

To overcome these challenges, we propose an economical and effective LLM-driven knowledge enhancement for securities index prediction. First, we collect the daily data of Shanghai Stock Exchange indexes (SSE indexes) and their related market indicators, such as the Northbound Capital and the Shenzhen Composite Index, as multi-source data to overcome the hysteresis of price data. Second, we establish the relationships between them by leveraging a large language model (LLM) as a knowledge

---

*The First international OpenKG Workshop: Large Knowledge-Enhanced Models, August 03, 2024, Jeju Island, South Korea*

\*Corresponding author.

✉ dizaiyuan@tongji.edu.cn (Z. Di); tjdxxyang@gmail.com (Y. Xiang)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

source, thus obtaining diverse knowledge inexpensively. Third, we integrate the collected numeric data and knowledge into a heterogeneous graph. Finally, we apply a heterogeneous graph-based neural network to learn the representation of SSE indexes and predict their price movement. The experiment results demonstrate the effectiveness of our method in prediction and real-world backtest. In conclusion, our contributions are as follows:

1) We propose a LLM-driven knowledge enhancement method for securities index prediction, leveraging LLM as an automated knowledge base to acquire various knowledge efficiently.

2) We implement a heterogeneous graph-based neural network to incorporate the numeric multi-source data and their knowledge, resulting in a good improvement in the downstream prediction task.

3) We validate the effectiveness of our method in securities index trend prediction, covering 175 securities indexes and 9 years of data. Additionally, we verify the method's real-world profit capabilities through backtest.

## 2. Related work

### 2.1. Deep learning in stock market prediction

Deep learning has been a promising approach to stock market prediction [19]. Recurrent models, such as RNN [20] and LSTM [21], are particularly prominent due to their ability to capture temporal information [19, 22]. In the previous work, the majority enhance their method by incorporating multiple numerical and textual data, e.g., technical indicators [23], social text [24], and financial news [25]. Nonetheless, these data often reflect a post-event status of the predicted object in isolation. Since events propagate between different entities in sequence, it causes a lead-lag effect in these data [12, 26].

Instead of concatenating the multi-source data as input features, some work also leverages the internal relationships of the data. For instance, Cheng et al. [12] integrate multi-modal information of multiple companies through a company knowledge graph. Matsunaga et al. [13] fuse price data of companies through their commercial relationships. In these works, complex relationships are reflected through knowledge, reflecting the role of knowledge in enhancing securities prediction.

### 2.2. Knowledge enhancement

Knowledge enhancement has become increasingly crucial in stock market prediction [16, 27, 28, 29]. Generally, the challenge of knowledge enhancement lies in the acquisition and incorporation of knowledge. For acquisition, many efforts have been made to obtain knowledge from unstructured raw data [30, 31] and existing knowledge sources [32, 33, 34, 35] such as open-source knowledge graphs. For incorporation, it is a common practice to concatenate unstructured knowledge (e.g., semantic information) and historical price data into vectors [36, 37], making the recurrent model widely used. Besides, graph-based models are also intuitive ways to incorporate structural knowledge, such as triplets, into predictions. Among these methods, homogeneous graph-based [38] and heterogeneous [39, 40] graph-based neural networks are both widely used.

However, the primary methods are confronted with acquisition costs and knowledge diversity. Recently, large language models have demonstrated their potential for knowledge acquisition, engaging many researchers in information extraction [41, 42, 43]. In this paper, we keep further focusing on reducing the cost of knowledge acquisition by using a large language model and integrating semantic and structural information in knowledge and numerical data with a graph-based model.

## 3. Method

### 3.1. Problem description

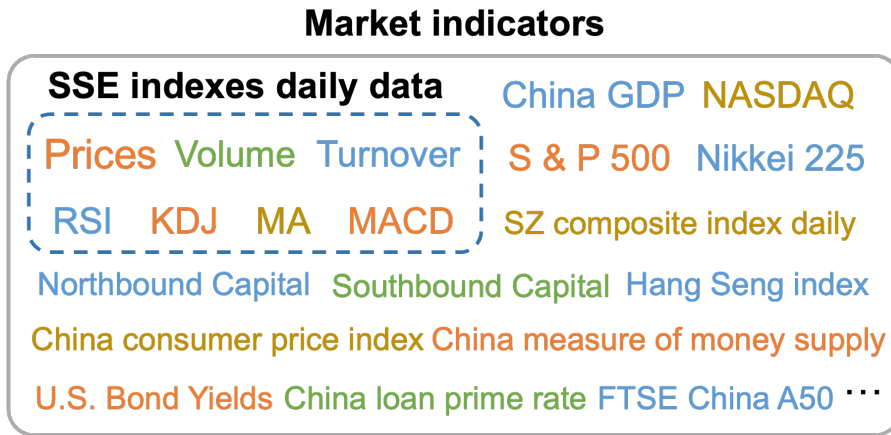
Our objective is to predict the daily trend of SSE indexes, framing it as a binary classification task. Assuming today is day  $t$ , we use the average closing price over the past  $\alpha$  days as the baseline. If the

closing price on day  $\beta$  in the future exceeds the baseline, it is considered an rising trend and labeled as a positive sample. The formula for setting the labels is as follows:

$$l^t = \begin{cases} 1, & c_{t+\beta} > \frac{\lambda}{\alpha} \sum_{i=0}^{\alpha-1} c_{t-i} \\ 0, & otherwise \end{cases} \quad (1)$$

where  $c_t$  represent the closing price on day  $t$ , and  $\lambda \geq 1$  is a constant that determines the threshold for classifying the trend. Using the average as a baseline can avoid the impact of market fluctuations.  $\beta$  represents the prediction horizon.

The data used for trend prediction comes from multiple sources. In addition to SSE indexes trading data  $D_{SSE}$ , it also includes other data sources  $D_1, D_2, \dots, D_r$ . These data sources have varying dimensions and granularity. In subsequent sections, we will introduce how to organize these data sources into input data for the prediction task.



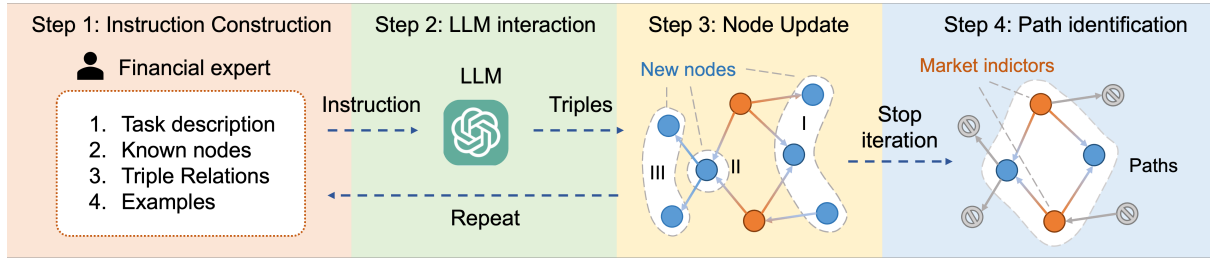
**Figure 1:** The data source of the SSE index prediction task

### 3.2. LLM-driven knowledge enhancement

Generally, the data used for SSE index prediction comes from two sources: SSE index trading data and other market indicators, as shown in the Figure 1. Trading data has a direct relationship with the SSE indexes, but the available historical data often exhibits lagging characteristics. On the other hand, market indicators refer to other economic market indicators related to the SSE indexes. These indicators can reflect more environmental factors and provide guidance for predicting the trends of SSE indexes. Traditional methods typically fuse these two types of data by simply appending market indicators to the daily trading data, thereby expanding the input feature dimensions. However, they overlooks the semantic information and connections behind the market indicators, which are crucial for improving prediction performance.

We propose a knowledge enhancement method based on Large Language Models (LLM), leveraging LLMs to automatically embed knowledge into market indicator data. We argue that the knowledge of market indicators is reflected in the relationships between them. For example, there is a connection between "Southbound Capital (SC)" and the "Hang Seng Index (HSI)". SC reflects the capital flowing from mainland China into Hong Kong, while HSI is an index of the Hong Kong stock market influenced by SC. To characterize the relationships between market indicators, we adopt a approach that establishes relational paths formed by multiple triplet links. In the example above, we take SC and HSI as nodes and form a triplet link based on their relationship:  $[(SC, index\ status, Capital\ from\ Mainland), (Capital\ from\ Mainland, participate\ in, Hong\ Kong\ market), (HSI, index\ status, Hong\ Kong\ market)]$ .

To extract relationships among numerous market indicators, we leverage a LLM to automatically uncover the connections between market indicators. The specific process is illustrated in the following steps:



**Figure 2:** The process of automatically uncovering relational paths between market indicator nodes using an LLM

**Step 1: Instruction Construction.** We construct instructions with known nodes for input into the LLM. These instructions involve the task description, ontology constraints, output format, examples, and other relevant information. The instructions must enable the LLM to fully understand our task of extracting relationships between market indicators.

**Step 2: LLM Interaction.** We input the constructed instructions into the LLM and obtain the feedback results. We then validate these results and extract triplets from the successful outcomes.

**Step 3: Node Update.** After saving the newly generated triplets, we update the known node pool with the newly generated nodes. We repeat Step 1 until the maximum number of iterations is reached.

**Step 4: Path Identification.** Once the maximum number of iterations is reached, we identify paths from any two market indicator nodes as start and end points using the triplet set. Irrelevant triplets that are not on the identified paths are removed.

The iterative process generates intermediate nodes for market indicators, fully utilizing the knowledge base and reasoning capabilities of the LLM to uncover complex multi-hop and cross paths. After this process, multiple market indicators, including SSE indexes, form a connected heterogeneous graph  $G = (V, E)$ .

Furthermore, we align data with knowledge to form the input for the SSE index prediction model. Assuming the prediction model aims to forecast the trend on day  $t$ , we organize the input data into a graph structure  $G^t = (V, E, A^t)$ . The node set  $V$  is divided into two categories based on their sources: market indicator nodes  $V_m$ , which include SSE indexes among others, and intermediate nodes  $V_p$ , which generated by the LLM and reflect the relationships between market indicators.

The market indicator nodes have corresponding data values as predictive support, serving as node feature attributes  $\mathcal{A}(n_i, t)$ , where  $n_i \in V_m$ . For instance, the SSE indexes node uses trading data from the period  $[t - \gamma, t]$  as its attribute values  $\mathcal{A}(n_i, t) = D_{SSE}^{[t-\gamma, t]}$ . These attribute values reflect the specific state of the nodes at the target time. Differences in these states directly influence the prediction results.

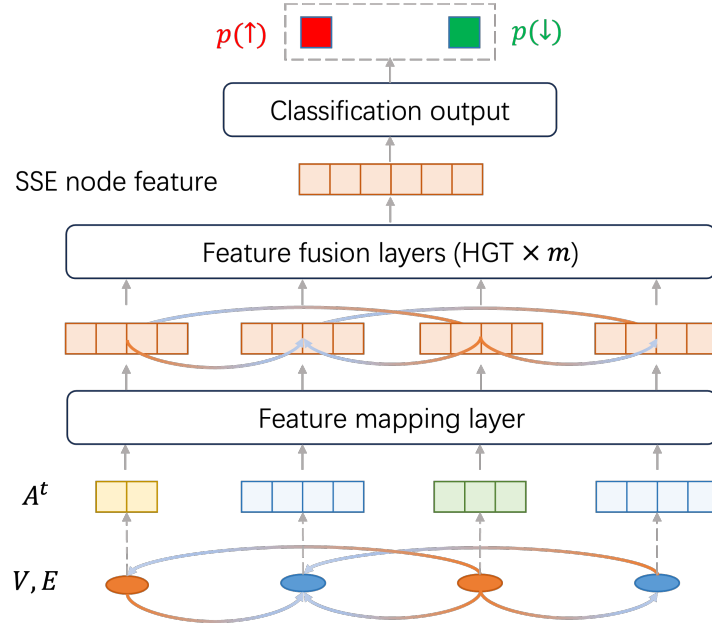
The intermediate nodes  $V_p$  do not have specific numerical values but possess clear semantics. Therefore, we number all intermediate nodes and use one-hot encoding as their attributes, i.e.,  $\mathcal{A}(n_i, t) = \text{onehot}(n_i, |V_p|)$ . These intermediate node encodings reflect the path semantics, guiding the information aggregation between market indicators. The attributes of all nodes form the attribute set  $A^T$ . This graph, enriched with knowledge for the prediction  $l^t$ , is then input into the model.

### 3.3. GNN-based securities index prediction

Based on the graph data  $G = (V, E, A)$ <sup>1</sup>, we then construct a GNN model  $\phi_{gnn}(G)$  to predict the trend of the index. The feedforward computation process of this model, as illustrated in Figure 3, consists of three main components: feature mapping, feature fusion, and classification output.

**Feature mapping** aims to map different types of nodes into a unified vector space. In the input heterogeneous graph, the node set  $V$  contains  $k$  types of nodes, and each type of node corresponds to a fixed set of feature attributions. We prepare a feature mapping function for each type of node. Given a node  $n_i$  of type  $j$  and its associated feature values  $\mathbf{a}_i$ , the computation of feature mapping is as follows:

<sup>1</sup>omitting the superscript  $t$



**Figure 3:** The model structure of securities index prediction

$$\mathbf{x}_i = f_j(\mathbf{a}_i) = W_j \mathbf{a}_i + \mathbf{b}_j, \quad (2)$$

where  $W_j$  and  $\mathbf{b}_j$  represent the weights and bias, respectively. The vector  $\mathbf{x}_i$  denotes the feature vector of node  $n_i$ .

**Feature fusion** is the process of using graph neural networks to aggregate information from nodes and edges. The feature mapping step outputs vector representations for all nodes in the graph, denoted as  $X = \{\mathbf{x}_s, \mathbf{x}_1, \dots, \mathbf{x}_{n-1}\}$ . We employ a Heterogeneous Graph Transformer (HGT) model to learn the representations of nodes and the topological structure. The fusion vector representation of node  $n_i$  is denoted as  $h_i^l$ , where  $l$  indicates the layer number of HGT, and the initial fusion vector is  $\mathbf{h}_i^0 = \mathbf{x}_i$ . The entire network consists of  $m$  layers, and the feedforward process for each layer is calculated as:

$$\mathbf{h}_i^{l+1} = \phi_{hgt}(\mathbf{h}_i^{l+1}, \{(\mathbf{h}_j^l, e_{ij}) | n_j \in \mathcal{N}(n_i)\}), \quad (3)$$

where  $\mathcal{N}(n_i)$  represents the neighbor nodes of  $n_i$ , and  $e_{ij}$  represents the edge between nodes  $n_i$  and  $n_j$ . The HGT model, based on the Transformer architecture, can aggregate features from different types of nodes and edges. The vector representation of SSE index nodes includes not only the node's original features but also additional data features and the semantic knowledge underlying the data.

**Classification output** predicts the trend of the SSE index based on the high-level representation. Given the fusion feature representation  $h_s^m$  of the SSE index node, we use a fully connected neural network and the softmax function to calculate the probability of the index rising or falling. This computation is defined as:

$$p = \text{Softmax}(W \mathbf{h}_s^m + \mathbf{b}). \quad (4)$$

The model output  $p$  is a 2-dimensional vector, with each element corresponding to the probabilities of the index rising and falling, respectively. During model training, we utilize the cross-entropy function as the loss function and update all learning parameters through gradient descent.

## 4. Experiment

In this section, we prepare model prediction and market backtesting experiments to validate our method.

Table 1

Results of Prediction Experiment <sup>2</sup>				
Model	Acc	P	R	F1
LSTM	0.6801	0.6505	0.6512	0.6366
Bi-LSTM	0.6890	0.6557	0.6611	0.6465
GAT	0.6836	0.6587	0.6451	0.6296
HGT	0.6868	0.6676	0.6576	0.6432
KHGT	<b>0.7183*</b>	<b>0.6870*</b>	<b>0.6805*</b>	<b>0.6793*</b>

#### 4.1. Experiment setting

**Datasets.** We collect the price and trading data of 175 SSE indexes from 2013 to 2021, along with 6 technical indicators and 12 market indicators. With the collected data, 364,314 prediction samples are formed. We split the datasets and conduct the experiments by year. In the prediction experiment, we adopt a time-series cross-validation, as referenced in [44]. The samples of each year is split into five parts: Jan to Apr, May to Jun, Jul to Aug, Sep to Oct, and Nov to Dec. For the  $i$ -th fold validation, we take the first  $i$  parts as a training set and the  $i + 1$ -th part as a validation set. In the backtesting experiment, we use the samples from Jan to Oct as a training set and those from Nov to Dec as a test set.

**Evaluation Metrics.** In the prediction experiment, we apply the accuracy (Acc), precision (P), recall (R), and Macro-F1 score (F1) as metrics to evaluate the prediction performance of different models. In the backtesting experiment, we define the daily return ( $R$ ) as follows:

$$R^t = \frac{1}{|I^t|} \sum_{i \in I^t} \frac{P_i^t - P_i^{t-1}}{P_i^{t-1}}, \quad (5)$$

where  $P_i^t$  denotes the price of index  $i$  at time  $t$ , and  $I^t$  denotes the number of indexes in the portfolio held by the model at time  $t$ . We then use the average daily return (DR) and the Sharpe ratio (SR) to measure the profitability of different models. In the Sharpe ratio, we use the 1-year China Government Bond Yield as the reference for the risk-free rate. To align with the daily return  $R$ , the annual Bond Yield is divided by 365.

**Implementation Details.** For the label setting, we take  $\alpha = 5$ ,  $\beta = 2$ , and  $\lambda = 1.01$  as specified in Eq. 1. The heterogeneous graph generated by the LLM consists of 30 nodes and 56 edges, with the time interval for node features set to  $\gamma = 5$ . During training, we use the Adam optimizer with a batch size of 512 and a learning rate of 0.001, and we implement early stopping to prevent overfitting. In our GNN model, the embedded vector dimension for feature mapping is set to 60, the hidden vector dimension of the HGT is 90, the number of attention heads is 4, and the number of layers is  $m = 6$ . For the baseline settings, recurrent models have 4 layers with a hidden vector dimension of 360; graph-based models use an embedded vector dimension of 100, a hidden vector dimension of 120, 3 attention heads, and 4 layers.

#### 4.2. Prediction experiment

We compare our model, Knowledge-enhanced HGT (KHGT), with several baselines, including recurrent models such as LSTM and Bi-LSTM, and graph-based models such as GAT and HGT. Their average performance over 9 years is presented in Table 1. The results indicate that KHGT outperforms all baselines. Furthermore, yearly comparison results are illustrated in Figure 4, showing that KHGT demonstrates strong generalization, achieving the highest Macro-F1 score and accuracy in most years from 2013 to 2021.

By comparing graph-based models and recurrent models, we observe that graph-based models generally do not outperform recurrent models. Specifically, GAT lags behind other baselines in terms of recall and F1 score, and HGT is slightly inferior to Bi-LSTM in terms of accuracy, recall, and F1 score. This indicates

<sup>2\*</sup> denotes that the improvement is significantly in a paired t-test ( $p < 0.05$ ), same as below.

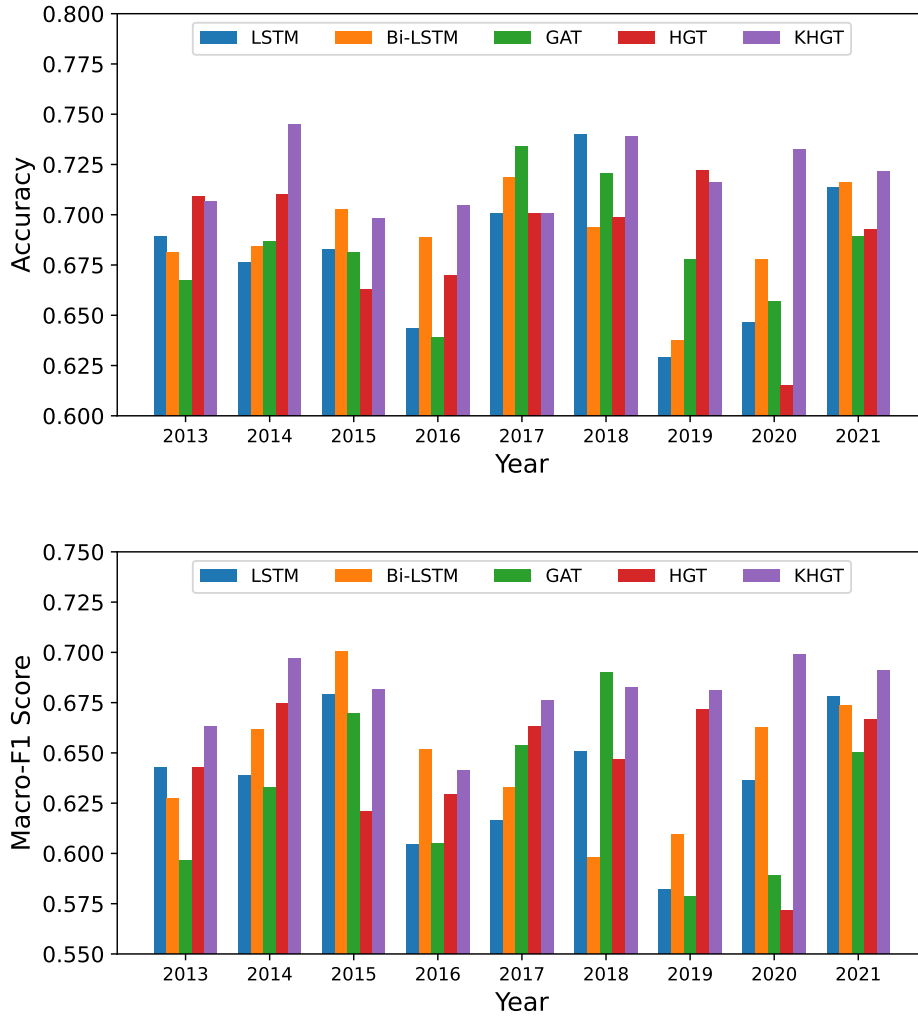


Figure 4: Performance comparison of models from 2013 to 2021

Table 2

Results of backtesting experiment			
Model	DR	SR	F1
Market	0.0009	0.0683	—
LSTM	0.0019	0.1824	0.6366
Bi-LSTM	0.0022	0.1909	0.6465
GAT	0.0012	0.1089	0.6296
HGT	0.0018	0.1747	0.6432
<b>KHGT</b>	<b>0.0029</b>	<b>0.2660</b>	<b>0.6793</b>

that the structural information in GAT and HGT does not provide a significant predictive enhancement. The superior performance of KHGT is attributed to the contribution of knowledge enhancement.

### 4.3. Backtesting experiment

To assess the real-world profitability of KHGT, we backtest it over a period of 9 years. We adopt a straightforward and effective trading strategy, as referenced in [12]: buy when the prediction is "rise" and sell when the prediction is "fall". The trading volume is set proportional to the prediction probability and inversely proportional to the index price. For simplicity, we assume that the index is tradable and ignore

**Table 3**

Results of intervention experiment				
Deletion Ratio	Acc		F1	
	Mean	Std	Mean	Std
1.00	0.6959	0.0056	0.6623	0.0072
0.75	0.7094	0.0049	0.6673	0.0084
0.50	0.7031	0.0123	0.6705	0.0085
0.25	0.7077	0.0048	0.6746	0.0035
0	<b>0.7274*</b>	<b>0.0045</b>	<b>0.6905*</b>	<b>0.0035</b>

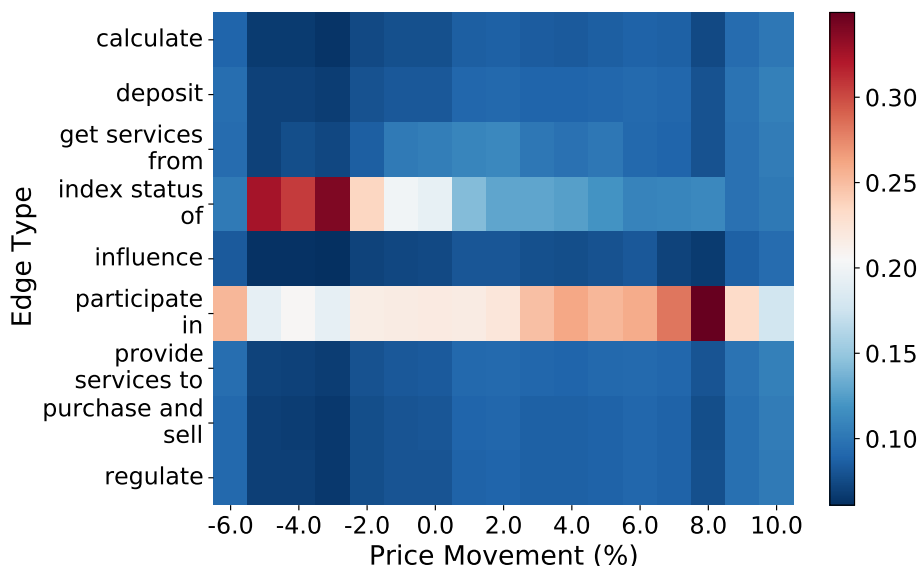
transaction fees.

Table 2 presents the average performance of each model, where "Market" refers to always holding all indexes. The results demonstrate that KHGT outperforms all baselines across both metrics. For the baselines, graph-based models generally lag behind recurrent models, consistent with the findings from the prediction experiment. Notably, the profitability of KHGT remains optimal across both metrics.

#### 4.4. Effectiveness of knowledge

**Intervention experiment.** To further verify the effectiveness of knowledge enhancement, we conducted an intervention experiment. We randomly deleted a portion of the knowledge paths between market indicators. A deletion proportion of 1 indicates no knowledge enhancement. For each proportion setting, we randomly selected paths and repeated the experiment three times, reporting the average results.

Table 3 presents the model performance under different proportions. The performance decreases as the number of knowledge paths increases. When eliminating all knowledge, its performance will degrade to the same level as HGT. This result implies that the LLM's knowledge has a strong enhancement effort. Without domain experts, our method can fully exploit the relationships between market indicators, thereby enhancing the predictive ability of the model in an economical and effective manner.



**Figure 5:** Visualization of attention weights

**Visualization of Attention.** We train the model with the samples from 2021 and calculate the average attention weight of each edge type. The attention weights in KHGT are visualized in Figure 5. The x-axis represents the price movement of the prediction target, and the y-axis represents the edge type.



We observe that "index status of" and "participate in" play relatively significant roles in falling and rising markets, respectively. "Index status of" connects indicators to financial objects, reflecting the impact of market indicators on the objects they measure. "participate in" connects these financial objects to markets, reflecting the convergence and interaction of these objects. The prominence of these two relations suggests that the model tends to focus on changes in specific indicators during falling markets and the interplay of multiple indicators during rising markets.

## 5. Conclusion

This work proposes an LLM-driven knowledge enhancement method for the task of securities index prediction. The innovation of this method lies in leveraging the rich knowledge within LLM, significantly reducing the cost and improving the efficiency of acquiring market index-related knowledge. By interacting with LLM through instructions, we establish triplet paths among market indices, thereby forming graph data that embodies implicit knowledge. Utilizing the graph-structured market data, we construct a GNN-based prediction model, which significantly outperforms traditional models.

**Limitations and future work.** The quality of the knowledge obtained from the LLM depends on itself and the design of the instructions. Knowledge quality is a crucial constraint affecting the model's predictive capability. Therefore, ensuring the quality of the knowledge is a critical issue that needs to be addressed in future work.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 72071145).

## References

- [1] T Chenowethl, Z Obradovic, and S Lee. "Technical trading rules as a prior knowledge to a neural networks prediction system for the S&P 500 index". In: *IEEE Technical Applications Conference and Workshops. Northcon/95. Conference Record*. Citeseer. 1995, p. 111.
- [2] Bernd Freisleben. "Stock market prediction with backpropagation networks". In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer. 1992, pp. 451–460.
- [3] Gia Shuh Jang et al. "An intelligent stock portfolio management system based on short-term trend prediction using dual-module neural networks". In: *Proc. of the 1991 International Conference on Artificial Neural Networks*. Vol. 1. 1991, pp. 447–52.
- [4] Darmadi Komo, Chein-I Chang, and Hanseok Ko. "Stock market index prediction using neural networks". In: *Applications of Artificial Neural Networks V*. Vol. 2243. SPIE. 1994, pp. 516–526.
- [5] Bing Yang, Zi-Jia Gong, and Wenqi Yang. "Stock market index prediction using deep neural network ensemble". In: *2017 36th chinese control conference (ccc)*. IEEE. 2017, pp. 3882–3887.
- [6] Saeede Anbaee Farimani et al. "Investigating the informativeness of technical indicators and news sentiment in financial market price prediction". In: *Knowledge-Based Systems 247* (2022), p. 108742.
- [7] Qing Li et al. "A tensor-based information framework for predicting the stock market". In: *ACM Transactions on Information Systems (TOIS)* 34.2 (2016), pp. 1–30.
- [8] Yaowei Wang et al. "EAN: Event attention network for stock price trend prediction based on sentimental embedding". In: *Proceedings of the 10th ACM conference on web science*. 2019, pp. 311–320.

- [9] Xin Liang et al. “F-HMTC: Detecting Financial Events for Investment Decisions Based on Neural Hierarchical Multi-Label Text Classification.” In: *IJCAI*. 2020, pp. 4490–4496.
- [10] Jun Wang et al. “Essential tensor learning for multimodal information-driven stock movement prediction”. In: *Knowledge-Based Systems* 262 (2023), p. 110262.
- [11] Hongfeng Xu, Donglin Cao, and Shaozi Li. “A self-regulated generative adversarial network for stock price movement prediction based on the historical price and tweets”. In: *Knowledge-Based Systems* 247 (2022), p. 108712.
- [12] Dawei Cheng et al. “Financial time series forecasting with multi-modality graph neural network”. In: *Pattern Recognition* 121 (2022), p. 108218.
- [13] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. “Exploring graph neural networks for stock market predictions with rolling window analysis”. In: *arXiv preprint arXiv:1909.10660* (2019).
- [14] Ryo Akita et al. “Deep learning for stock prediction using numerical and textual information”. In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE. 2016, pp. 1–6.
- [15] Zhige Li et al. “Individualized Indicator for All: Stock-wise Technical Indicator Optimization with Stock Embedding”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 894–902.
- [16] Shumin Deng et al. “Knowledge-driven stock trend prediction and explanation via temporal convolutional network”. In: *Companion proceedings of the 2019 world wide web conference*. 2019, pp. 678–685.
- [17] Shunrong Shen, Haomiao Jiang, and Tongda Zhang. “Stock market forecasting using machine learning algorithms”. In: *Department of Electrical Engineering, Stanford University, Stanford, CA* (2012), pp. 1–5.
- [18] Sheng Xiang et al. “Temporal and Heterogeneous Graph Neural Network for Financial Time Series Prediction”. In: *Proceedings of the 31st ACM International Conference on Information amp; Knowledge Management. CIKM '22*. ACM, Oct. 2022. DOI: 10.1145/3511808.3557089. URL: <http://dx.doi.org/10.1145/3511808.3557089>.
- [19] Weiwei Jiang. “Applications of deep learning in stock market prediction: recent progress”. In: *Expert Systems with Applications* 184 (2021), p. 115537.
- [20] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323 (1986), pp. 533–536. URL: <https://api.semanticscholar.org/CorpusID:205001834>.
- [21] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [22] Luca Di Persio and Oleksandr Honchar. “Recurrent Neural Networks Approach to the Financial Forecast of Google Assets”. In: 2017. URL: <https://api.semanticscholar.org/CorpusID:67797781>.
- [23] Simone Merello et al. “Ensemble Application of Transfer Learning and Sample Weighting for Stock Market Prediction”. In: *2019 International Joint Conference on Neural Networks (IJCNN)* (2019), pp. 1–8. URL: <https://api.semanticscholar.org/CorpusID:203606095>.
- [24] Gary Ang and Ee-Peng Lim. “Learning Dynamic Multimodal Implicit and Explicit Networks for Multiple Financial Tasks”. In: *2022 IEEE International Conference on Big Data (Big Data)*. 2022.
- [25] Ziniu Hu et al. “Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction”. In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 2018, pp. 261–269.
- [26] Matthew L O’Connor. “The cross-sectional relationship between trading costs and lead/lag effects in stock & option markets”. In: *Financial Review* 34.4 (1999), pp. 95–117.

- [27] Dawei Cheng et al. “Knowledge graph-based event embedding framework for financial quantitative investments”. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020, pp. 2221–2230.
- [28] Jue Liu, Zhuocheng Lu, and W. P. Du. “Combining Enterprise Knowledge Graph and News Sentiment Analysis for Stock Price Prediction”. In: *Hawaii International Conference on System Sciences*. 2019. URL: <https://api.semanticscholar.org/CorpusID:102352388>.
- [29] Liping Wang et al. *Methods for Acquiring and Incorporating Knowledge into Stock Price Prediction: A Survey*. 2023. arXiv: 2308.04947 [q-fin.ST].
- [30] Wei Li et al. “Modeling the stock relation with graph network for overnight stock movement prediction”. In: *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*. 2021, pp. 4541–4547.
- [31] Bhaskarjit Sarmah et al. *Learning Embedded Representation of the Stock Correlation Matrix using Graph Machine Learning*. 2022. arXiv: 2207.07183 [q-fin.CP].
- [32] Fuli Feng et al. “Temporal Relational Ranking for Stock Prediction”. In: *ACM Transactions on Information Systems* 37.2 (Mar. 2019), pp. 1–30. ISSN: 1558-2868. DOI: 10.1145/3309547. URL: <http://dx.doi.org/10.1145/3309547>.
- [33] Xiaojie Li et al. “Hypergraph-Based Reinforcement Learning for Stock Portfolio Selection”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 4028–4032. DOI: 10.1109/ICASSP43922.2022.9747138.
- [34] Ramit Sawhney et al. “Deep attentive learning for stock movement prediction from social media text and company correlations”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 8415–8426.
- [35] Xiaoting Ying et al. “Time-aware Graph Relational Attention Network for Stock Recommendation”. In: *CIKM ’20*. 2020, pp. 2281–2284.
- [36] Heyan Huang et al. “News-driven stock prediction via noisy equity state representation”. In: *Neurocomputing* 470 (2022), pp. 66–75.
- [37] Wasiat Khan et al. “Stock market prediction using machine learning classifiers and social media, news”. In: *Journal of Ambient Intelligence and Humanized Computing* (2022), pp. 1–24.
- [38] Yingmei Chen, Zhongyu Wei, and Xuanjing 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). URL: <https://api.semanticscholar.org/CorpusID:53037746>.
- [39] Hao Qian et al. *MDGNN: Multi-Relational Dynamic Graph Neural Network for Comprehensive and Dynamic Stock Investment Prediction*. 2024. arXiv: 2402.06633 [q-fin.ST].
- [40] Wentao Xu et al. “REST: Relational Event-driven Stock Trend Forecasting”. In: *Proceedings of the Web Conference 2021. WWW ’21*. ACM, Apr. 2021. DOI: 10.1145/3442381.3450032. URL: <http://dx.doi.org/10.1145/3442381.3450032>.
- [41] Di Lu et al. *Event Extraction as Question Generation and Answering*. 2023. arXiv: 2307.05567 [cs.CL].
- [42] Ji Qi et al. *Preserving Knowledge Invariance: Rethinking Robustness Evaluation of Open Information Extraction*. 2023. arXiv: 2305.13981 [cs.CL].
- [43] Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. *Aligning Instruction Tasks Unlocks Large Language Models as Zero-Shot Relation Extractors*. 2023. arXiv: 2305.11159 [cs.CL].
- [44] Vitor Cerqueira, Luis Torgo, and Igor Mozetič. “Evaluating time series forecasting models: an empirical study on performance estimation methods”. In: *Machine Learning* 109.11 (Oct. 2020), pp. 1997–2028. ISSN: 1573-0565. DOI: 10.1007/s10994-020-05910-7. URL: <http://dx.doi.org/10.1007/s10994-020-05910-7>.