IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 July 2022, accepted 9 August 2022, date of publication 16 August 2022, date of current version 23 August 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3199008*

RESEARCH ARTICLE

# ML-GAT:A Multilevel Graph Attention Model for Stock Prediction

**KUN HUANG[1], XIAOMING LI[2], FANGYUAN LIU[3], XIAOPING YANG[2], AND WEI YU[2]**

[1]School of Science, Zhejiang University of Science and Technology, Hangzhou 310023, China
[2]College of International Business, Zhejiang Yuexiu University, Shaoxing 312000, China
[3]College of Business and Management, UCSI University, Kuala Lumpur 56000, Malaysia

Corresponding author: Wei Yu (weiyu@zyufl.edu.cn)

This work was supported in part by the Key Research and Development Program of Zhejiang Province under Grant 2022C01083, in part by the National Science Foundation of China under Grant 62102262, in part by the Development Project of Xinjiang Production and Construction Corps 12th under Grant SR202103, in part by the Practice Conditions and Practice Base Construction of Ministry of Education under Grant SR202102624032, and in part by the Graduate Program of Zhejiang Yuexiu University of Foreign Languages under Grant ZY2021004.

**ABSTRACT** Stock market volatility research has long been the focus of industry and academia, and stock trend forecasting is challenging. Existing research focuses more on how to aggregate historical price features into graph networks, ignoring the effects of other information such as news and current events on forecasts. Most existing graph-based learning methods create stock graphs by manually constructing stock relationships, ignoring the complexity of stock relationships. Based on these, we propose a novel multilevel graph attention network (ML-GAT) for predicting the stock market trends. To be more specific, we initialize the node representations captured by each feature extraction module, then update the stock nodes in the isomorphic graph converted from Wikidata using ML-GAT, selectively aggregate the information of various relation types, aggregate the information to the node representation, and finally feed the result to a particular forecast layer for forecasting, completing the trend forecast of 423 stocks in the S&P 500 index and 286 stocks in the CSI 300 index. By comparing 5 popular approaches, the experimental research verifies ML-GAT'S state-of-the-art performance in prediction tasks. In comparison with the best performance benchmark model, the F1-score and accuracy increased by 11.82% and 12.6%, respectively, and the average daily return and Sharpe rate rose by 5.06 percent and 94.81 percent, respectively. More significantly, our model's stock linkages are interpretable and consistent with real-world interactions.

**INDEX TERMS** Stock trend prediction, graph neural network, stock relation, financial news.

## I. INTRODUCTION

Stock market forecasting, which belongs to the intersection of finance and computer technology, is one of the most attractive problems and has received extensive attention from investors and scholars. A substantial number of academics utilize regression methods to forecast future stock prices [1], but only a small number of scholars employ classification methods to predict future stock market trends [2]. However, a burgeoning number of investors are increasingly interested in the future direction of the stock market. At present, many approaches have been applied to anticipate stock trends, ranging from traditional statistical methods for deep learning, such as convolutional neural networks (CNNs) [3], [4], recurrent neural networks (RNNs) [5], and long short-term memory (LSTM) [6]. While these complicated models have been used to model time series with excellent forecasting performances, most past research has focused solely on financial time series.

Therefore, to produce more accurate predictions, more comprehensive information must be introduced and merged. Under ideal market conditions, the price of a stock represents all available information in the market, including news, events, and past market conditions, according to the efficient market hypothesis [7]. However, most earlier studies [2], [8] concentrated on single or partial information, neglecting

The associate editor coordinating the review of this manuscript and approving it for publication was Yiqi Liu.

additional market information. Stocks do not exist in isolation, and stock price fluctuations will be influenced by events in connected companies. Leading stocks, for example, can boost the market of the entire sector, so related stocks have strong linkages. However, the majority of current research [5], [9] predicts a single stock in isolation, ignoring stock correlation. As a result, consistently aggregating information such as news events, past market prices, and stock linkages is crucial to formulating accurate predictions.

In fact, a small number of researchers have integrated information from various sources to predict stock trends. By merging data from multiple sources with an attention-enhancing LSTM model, Zhang *et al.* [10] significantly improved stock prediction performance. The intrinsic properties of the financial market and the correlation properties that exist between stocks have inspired researchers to introduce graph structures to better solve the problems of stock trend prediction, aiming to use graph structures to process information from different sources in the market, but these works are still in their infancy. Chen *et al.* [11] improved prediction accuracy by constructing a graph containing all relevant company relationships but only integrating company-related information; Li *et al.* [12] modelled stock relationships by utilizing graph convolutional networks to forecast overnight stock movements but only integrating news data; Wang *et al.* [13] proposed an adaptive hierarchical temporal relational graph network (HTAR) to characterize and predict stock evolution but ignored the impact of news events; Matsunaga *et al.* [14] and Chen *et al.* [15] used a graph neural network to model and analyse stock-related factors but ignored the impact of investor sentiment on prediction. Although complex graphical models have been introduced to process information in the financial field, current methods simply integrate information from different sources into graphs and cannot efficiently utilize stock market information. Therefore, how to efficiently mine and integrate multisource data in the stock market is still an important challenge.

In this paper, we investigate how graph-based learning methods can be used to efficiently aggregate multisource data in the stock prediction task. First, stock market data features are extracted based on the three dimensions of text data, numerical data and relational data. Based on textual data such as news, we apply the BERT model to capture the textual features of financial markets. Although there have been studies applying BERT to predict stock trends based on news, the fusion of BERT to graph neural network-based methods has not yet been explored. While previous studies based on graph neural networks, such as that by Feng *et al.* [1], have ignored the limitations of GNN for processing text, we optimize the extraction features of unstructured news through advanced natural language processing technology, thereby improving the predictive ability of the prediction model. However, this still does not capture the relationships between stocks, and it is difficult to accurately predict the trends of related stocks. From this, we propose the multilevel graph attention network (ML-GAT), a new module based on the graph attention network. It can assign different weights to different relations and selectively aggregate the node representations captured from the feature extraction module into the stock graph network using the designed multilevel attention mechanism. This result is then fed to the prediction layer for training. Extensive experiments on public datasets demonstrate the effectiveness of our proposed method.

In summary, the main contributions of this research are as follows:

- We proposed a new stock relationship modelling model, ML-GAT, which can assign different weights to different relationship types and selectively aggregate node representations captured from the feature extraction module into a stock relationship graph, leveraging the stock network graph topological information and market characteristics to facilitate stock trend forecasting.
- We applied BERT to the financial field, learned the feature representation of news and embedded them into the stock graph for prediction, and explored the method of integrating BERT with the graph neural network. Experiments show that the node representation learned by BERT is beneficial to stock prediction.
- We conducted extensive experiments in the standard S&P 500 index and the CSI 300 index and showed that our proposed method outperforms all benchmark models.

The rest of this paper is organized as follows: in Section 2, we review some previous achievements in the field of stock forecasting and the application of graph neural networks to stock forecasting tasks; in Section 3, we present the overall framework of our research and introduce the ML-GAT model in detail; in Section 4, we explain the source of our dataset and the specific information of the experimental setup, and the experimental results of various models are presented and analysed; finally, we propose conclusions and recommendations for future work based on the limitations of this study's current work.

## II. RELATED WORK

### A. STOCK MARKET PREDICTION

Academia has proposed different approaches for research on stock market forecasting. General methods focus on technical analysis, taking historical stock prices as input and using machine learning algorithms such as SVM [17] and RF [18] to model the stock. With the improvement in computational power, and the stock market being characterized by highly nonlinear and nonstationary fluctuations, different deep learning techniques have been used for stock market forecasting. Hoseinzade and Haratizadeh [3] proposed a CNN-based framework that successfully combined various types of information for prediction, and experiments showed that deep learning methods outperformed shallow methods. Wu *et al.* [4] proposed a novel convolutional neural network framework SSACNN based on CNN, which significantly improved the accuracy of stock trading prediction task. In his subsequent research [6], he proposed a new method for convolution of stock sequence arrays based on CNN and LSTM. Zhao *et al.* [5] proposed an RNN-based prediction model and introduced an attention mechanism to focus on key information. Wu *et al.* [6] proposed a multi-input multi-output associative deep recurrent neural network model based on LSTM, and the experimental results show that it is superior to other prediction models. With the increase in public information in the stock market, researchers have begun to focus on fundamental analysis, using potential factors affecting a company or industry as a predictive attribute. Ding *et al.* [19] used a novel neural tensor network to extract features from news events to predict stock price movements. Wang *et al.* [20] proposed a novel framework for improving the performance of investment opinion mining and individual stock forecasting.

### B. REPRESENTATION OF FINANCIAL TEXTS

Some studies have shown that financial news will affect stock price trends [25], [39], [44]. With the development of natural language processing technology, the technology for extracting features from financial text data has made great progress. In the field of the stock market, many researchers attempt to better predict stock market trends by obtaining financial text features. Most of the early stock prediction studies [21] introduced the bag of words method to vectorize textual information. In recent years, thanks to the development of deep learning, the performance of word embedding techniques has been improved. Liu *et al.* [22] implemented word2vec to extract deep semantic features of social media, which improved the accuracy of stock trend prediction. Chen *et al.* [26] improved the accuracy of using sentiment in Chinese news to predict stock trends by generating key sentences for news items via TextRank and word2vec. The SDAE model proposed by Vincent *et al.* [27] uses an autoencoder to map text to low-dimensional vectors. Tang and Chen *et al.* [23] extracted more features from news headlines through GloVe, and the proposed method outperformed the baseline.

In this paper, the BERT model is introduced to represent the financial text features as vectors. The main framework of the BERT algorithm is the transformer, which is a self-supervised learning model based on a rich corpus. By learning the main representation of the sentence, the bidirectional relationship in the sentence can be fully captured. Although there have been studies citing pretrained models such as BERT to predict stock trends based on text data, the combination of BERT and ensemble learning had not been explored until this work, which achieved good results.

### C. GRAPH NEURAL NETWORK

The models used in traditional stock forecasting tasks cannot fully reflect the complex relationships between stocks. In view of the non-Euclidean characteristics of financial market information, researchers introduced a graph neural network method to learn the distribution representation between companies. Chen *et al.* [11] introduced GNN into the stock market, constructed a graph of related companies based on real market investment events, and captured inter-company relationships through a graph convolutional neural network to make more accurate predictions. Feng *et al.* [28] collected the relationships between industries and companies to embed into a GNN model, captured interstock relationships in a time-sensitive manner, and customized a ranking system for stocks through a temporal graph convolutional network. Cheng *et al.* [29] proposed a multimodal graph neural network model that learns to fuse multimodal input information features into heterogeneous graphs for financial market time series forecasting. Kim *et al.* [30] used a stacked graph neural network to capture time-series features and global information, dynamically perceive stock relationships, and recommend high-return stocks. This paper also verifies the applicability and practicability of the graph model in the stock market. Yin *et al.* [43] use a multi-Hawkes Process to initial a correlation graph between stocks and propose a novel machine learning model.

Different types of relationships among stocks reflect different qualities of information. However, most of the existing work has not studied which relationship data are more beneficial for the task of predicting stock trends. Therefore, we selectively aggregate information from data of different relation types through the graph attention neural network.

## III. METHODOLOGY

In this section, we first introduce the overall framework of our proposed method and then elaborate on how we model the three dimensions of textual, numerical, and relational data in financial markets. Unlike other studies that rely heavily on cost-intensive relational modelling routes, our framework is based on publicly available corporate relational data. The overall framework is shown in Figure 1. After the original data is processed, different feature vectors are obtained through different feature extraction modules, and then aggregated to form a stock relationship network diagram containing price characteristics and text features. Finally, the multi-layered attention layer of ML-GAT processes the stock relationship network diagram for trend prediction. After describing the various parts of the framework, we will detail our proposed novel relational modelling module structure.

### A. FEATURE EXTRACTION MODULE

In modelling the stock market graph, we treat the stock of each company as a node, and the characteristics of each node represent the current state of each company under the movement of the stock price. Node characteristics are also updated as market conditions change over time. Different types of data will have different effects on stock volatility. This paper uses historical prices and financial news as indicators of stock price changes and Wikidata's company relationship data as the stock correlation.

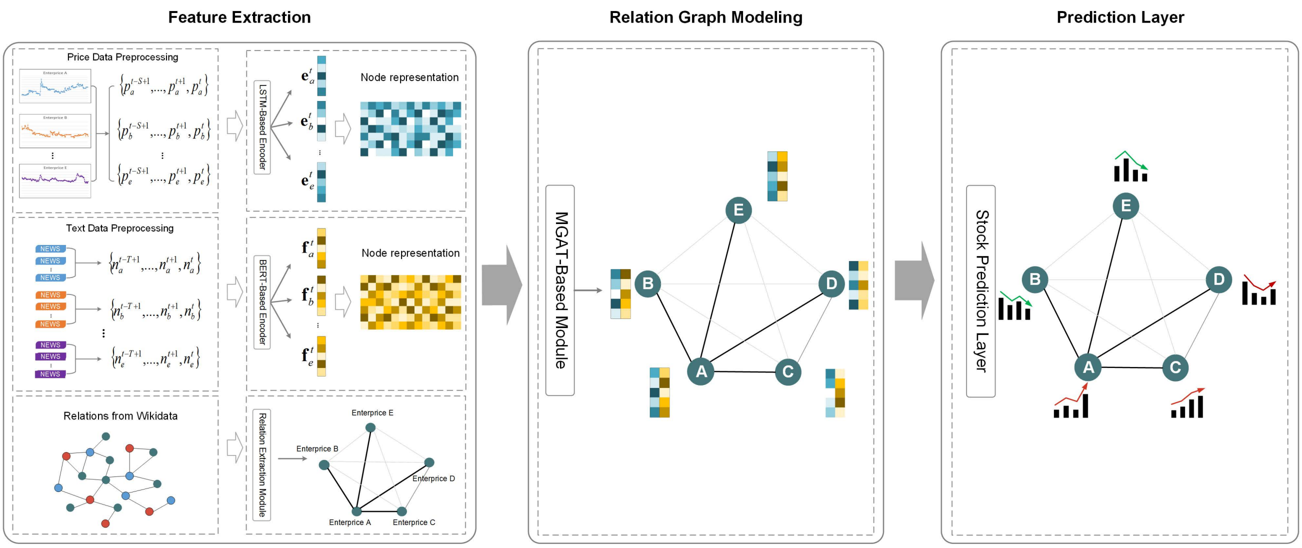


**FIGURE 1. General framework of stock prediction using relational data. It includes a feature extraction layer, graph modelling layer and prediction layer. In the stock relational graph, the lowercase a, b, c, d, and e denote the different stocks, and the symbols A, B, C, D, and E correspond to the company with stock a,b,c,d, and e, respectively.**

### 1) LSTM-BASED ENCODER MODULE

In financial markets, historical stock price data have short-term and long-term dependencies. The time span differs, as does the stock price performance. Therefore, we introduce an LSTM-based encoding model, which can adequately capture long-term dependencies. In addition, [28], [30], [40] also take LSTM as the model feature extraction module.

Many different raw features of historical stock prices are input into the feature extraction module, such as opening price and closing price. In this paper, the daily historical closing price is preprocessed first, and the calculated price change rate is used as the input of LSTM. The price change rate is calculated in Section 4.2. We denote the historical price change rate of stock i at time step t as $R_i^t = \left\{r_i^{t-S+1}, \ldots, r_i^{t+1}, r_i^t\right\} \in \mathbb{R}^{S\times D}$ (S represents the sequence length, and D represents the feature dimension at each time step). We take the time-series data $R_i^t$ as input into the LSTM network, and the last hidden state of the output $\left(\boldsymbol{h}_i^t\right)$ is used as the sequential embedding $\left(\boldsymbol{e}_i^t\right)$ $\left(\boldsymbol{h}i^t = \boldsymbol{e}_i^t\right)$ of the following network. That is, we have:

$$i_i^t, f_i^t, o_i^t = f_{\theta j}, f_{\theta f}, f_{\theta e}\left(W\left[h_i^{-1}; R_{\mathrm{i}}^t\right] + b_{(i,f,o)}\right) \tag{1}$$

$$c_i^t = i_i^t \odot \mathrm{u}_{\mathrm{i}}^{\mathrm{t}} + f_i^t \odot \mathrm{c}_{\mathrm{i}}^{\mathrm{t-1}} \tag{2}$$

$$u_i^t = \tan\left(\mathrm{W}^{(u)}R_i^t + \mathrm{U}^{(u)}h_i^{t-1} + b^{(u)}\right) \tag{3}$$

$$h_i^t = o_i^t \odot \tanh\left(\mathrm{c}_{\mathrm{i}}^{\mathrm{t}}\right) \tag{4}$$

In simple terms, we can obtain:

$$\mathbf{E}^{\mathrm{t}} = \mathrm{LSTM}\left(\mathcal{R}^t\right) \tag{5}$$

where $\mathcal{R}^t = \left[\mathrm{R}_1^t, \ldots, \mathrm{R}_i^t, \ldots, \mathrm{R}_M^t\right] \in \mathbb{R}^{M\times S\times D}$, $\mathbf{E}^t = \left[\mathbf{e}_1^t, \ldots, \mathbf{e}_i^t, \ldots, \mathbf{e}_M^t\right] \in \mathbb{R}^{M\times U}$ represents the sequential embedding of M stocks, and U represents the embedding size (the number of hidden units in the LSTM).

### 2) BERT-BASED ENCODER MODULE

The input embedding of BERT consists of the token embedding layer, the segment embedding layer, and the position embedding layer. The token embedding layer is a vector that transforms each word into a fixed dimension. In particular, each input is preceded by a special token "[CLS]". The segment embeddings mainly distinguish between two different kinds of sentences, each of which is marked with the token "[SEP]" in the input. The position embedding is obtained by training.

To illustrate the BERT extraction of stock news text data features in the financial field in more detail, we will form T pieces of the news collected from stock i in time period t as $\mathbf{N}_i^t = \left\{n_i^{t-T+1}, \ldots, n_i^{t+1}, n_i^t\right\} \in \mathbb{R}^{T\times H}$ (T represents the number of text paragraphs, and H represents the feature dimension of each text) and feed it into the pretrained encoding model.The BERT-based encoder module preprocesses the input sentence by truncation or completion, adding the [CLS] symbol at the beginning of the sentence of each input text and adding the [SEP] symbol at the end of the sentence. Related news item $n_i^t$ for stock i is converted to $[CLS] + n_i^t + [SEP]$, where the output vector corresponding to [CLS] can represent the semantic feature information of the sentence.where the output vector corresponding to [CLS] can represent the semantic feature information of the sentence. After many deep-level feature learning operations of the BERT model, the [CLS] vector in the previous layer of the output classification result can best represent the semantic information of the text. For the news text feature extraction task, only the upper layer of the output classification result needs to be extracted from

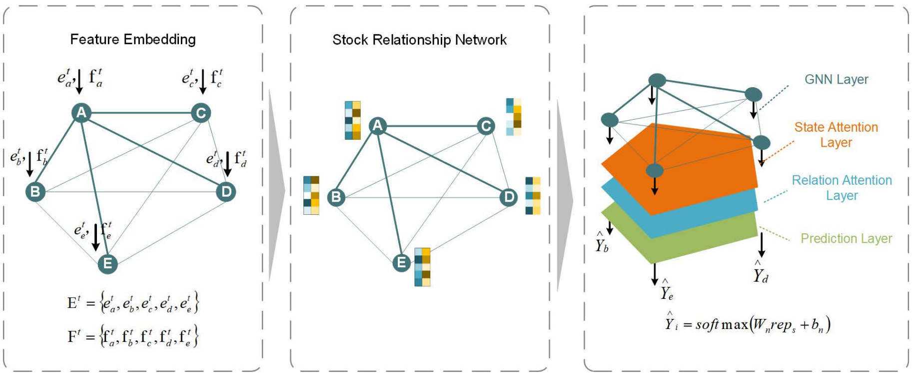


**FIGURE 2.** Overview of the ML-GAT model.

the text features. That is, only the feature vector of the last layer of the transformer corresponding to the symbol [CLS] in front of the sentence is extracted. That is, we have:

$$\mathbf{F}^t = \text{BERT}\left(\mathcal{M}^t\right) \tag{6}$$

where $\mathcal{M}^t = \left[\mathbf{N}_1^t, \ldots \mathbf{N}_i^t, \ldots, \mathbf{N}_M^t\right] \in \mathbb{R}^{M \times T \times H}$, $\mathbf{F}^t = \left[\mathbf{f}_1^t, \ldots, \mathbf{f}_i^t, \ldots, \mathbf{f}_M^t\right] \in \mathbb{R}^{M \times V}$ represents the text feature embedding vector of M stocks, and V represents the embedding size.

### 3) RELATION EXTRACTION MODULE

In the relation extraction module, we initially extract the enterprise relations in the public Wikidata data. Wikidata contains relationships between various entities (such as countries, companies, and individuals) and is a heterogeneous graph with different types of nodes and edges. In this study, we are only interested in the node types of stock-related companies. However, there are often several types of edges between these companies, and the relationships between them are very sparse. Aiming at this problem, we are inspired by [34] and use meta-paths to deal with heterogeneous graphs, converting complex heterogeneous graphs into homogeneous graphs with only stock (company) nodes.

We represent Wikidata as a heterogeneous network $G = (V, E, T)$, where each node v and each link e are distributed and their mapping functions $\phi(v) : V \rightarrow T_V$ and $\varphi(e) : E \rightarrow T_E$ are associated. $T_V$ and $T_E$ represent collections of node objects and relation types. The task of the relationship extraction module is to learn the d-dimensional latent representation $\mathbf{X} \in \mathbb{R}^{|V| \times d} (d \ll |V|)$ and capture the structural relationships between the target nodes. We then show how to use meta-path bootstrapping with a maximum of only 2 hops to guide the random walk process in heterogeneous networks, given a meta-path scheme: $\mathcal{P} : V_1 \overset{R_1}{\rightarrow} V_2 \overset{R_2}{\rightarrow} V_3$. The probability of transition at step i is defined as follows (7), as shown at the bottom of the next page. where $v_t^i \in V_t$ and $N_{t+1}\left(v_t^i\right)$ represent the $V_{t+1}$ type of the field of node $v_t^i$.

## B. RELATION GRAPH MODELLING

The function of the relational graph modelling module is to update the stock (company) node in the isomorphic graph transformed by the relational extraction module in the previous section. The main function of the graph neural network is to exchange information between adjacent nodes, then aggregate the information of adjacent nodes, and to finally add it to each node's representation. Node features are key to the success of the graph prediction task, so we need to effectively aggregate different types of relationship information collected from different nodes. To this end, this paper proposes a novel GNN-based multilevel graph attention network for relational graph modelling (ML-GAT). Figure 2 depicts the detailed structure of ML-GAT. By adding multiple layers of attention mechanisms at different levels and assigning different weight values to information screening, our model can collect information on different types of relationships from different nodes and filter out information that is invalid for trend prediction. Our method is important for accurately predicting stock trends because there are many different types of relationships between stocks, some of which are irrelevant to stock forecasting.

Through the stock data feature extraction module, the price embedding vector $\boldsymbol{e}_i^t \in \mathbb{R}^h$ and news text embedding vector $\mathbf{f}_i^t \in \mathbb{R}^h$ of stock i at time t can be obtained.For simplicity, the superscript t is uniformly omitted in the following text, assuming that all vector representations are computed at the same time t. Furthermore, since the model is implemented based on a graph neural network, we need to know the set of neighbour nodes of the target node i in each relation type. We denote the relation type m of node i as $\boldsymbol{r}_m \in N_i^{r_m}$ and the embedding vector of relationship type $\boldsymbol{r}_m$ as $\mathbf{e}_{r_m} \in \mathbb{R}^d$. Our goal is to selectively collect information about different relationships from neighbouring nodes.

In the first-level state attention layer, ML-GAT selects relevant information of the same type of relationship from a set of adjacent nodes. The attention mechanism is mainly used to calculate different weights according to the selected relationship type $\boldsymbol{r}_m$. Before calculating the state attention

**TABLE 1. Statistics of historical price data.**

| Index | Stocks | Training Days | Validation Days | Testing Days |
|---|---|---|---|---|
| S&P 500 | 423 | 08/02/2013-23/05/2017 1080days | 24/05/2017-27/03/2018 213days | 27/03/2018-29/08/2019 316days |
| CSI 300 | 286 | 08/02/2013-23/05/2017 1080days | 24/05/2017-27/03/2018 213days | 27/03/2018-29/08/2019 316days |

coefficient, it is necessary to concatenate the relation type embedding vector $\mathbf{e}_{r_m}$ and the node representations of nodes i and j into a vector, where $j \in \mathrm{N}_i^{r_m}$. We denote the connection vector as $x_{ij}^{r_m} \in \mathbb{R}^{2h+d}$. The formula for calculating the attention coefficient of the relationship type $\boldsymbol{r}_m$ between target node i and target node j is:

$$\alpha_{ij}^{r_m} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{a}_r^{\mathrm{T}}\left(\mathrm{W_s x_{ij}^{r_m}} + \mathrm{b_s}\right)\right)\right)}{\sum_{k \in N_i^{r_m}} \exp\left(\mathrm{LeakyReLU}\left(\vec{a}_r^{\mathrm{T}}\left(\mathrm{W_s x_{ik}^{r_m}} + \mathrm{b_s}\right)\right)\right)} \tag{8}$$

where $W_s \in \mathbb{R}^{2h+d}$ and $\mathrm{b_s} \in \mathbb{R}$ are the learnable parameters used to compute the state attention coefficient.

Based on the calculated state attention coefficients, we calculate the output feature of the company relationship as having a weighted average of all nodes with the formula:

$$\mathrm{Z_i^{r_m}} = \sigma\left(\sum_{j \in N_i^{r_m}} \alpha_{ij}^{r_m} W_s \mathrm{e_j}\right) \tag{9}$$

Through the calculation of the above formula, the vector representation of each relationship between stocks is obtained, which can be regarded as the summary information of the relationship. The vector $\mathrm{Z_i^{r_m}}$ includes summary information from the relationship type $\boldsymbol{r}_m$ of stock i. For example, the subsidiary relationship vector representation can summarize all subsidiaries of the target stock company. As with human investment decision-making behaviour, our model should prioritize that important information for trading decisions from all the aggregated information. Therefore, in the second-level relation attention layer of ML-GAT, the attention mechanism is used to assign importance weights to many pieces of information.

We concatenate the summary information vector $\mathrm{Z_i^{r_m}}$ of the relationship, the current node representation $\mathrm{e_i}$, $\mathrm{f_i}$ of the stock company and the embedding vector of the relationship type $\mathbf{e}_{r_m}$ and denote the concatenated vector as $v_i^{r_m} \in \mathbb{R}^{2\,h+d}$ as the feature input for the next attention layer. Similar to the previous formula, in the second-level attention layer, the attention coefficient is calculated by the formula:

$$\alpha_i^{r_m} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{a}_r^{\mathrm{T}}\left(\mathrm{W_r v_i^{r_m}} + \mathrm{b_r}\right)\right)\right)}{\sum_{k \in \Phi} \exp\left(\mathrm{LeakyReLU}\left(\vec{a}_r^{\mathrm{T}}\left(\mathrm{W_r v_i^{r_k}} + \mathrm{b_r}\right)\right)\right)} \tag{10}$$

where $W_\mathrm{r} \in \mathbb{R}^{2\,h+d}$ and $b_r \in \mathbb{R}$ are the learnable parameters. The weighted vectors of each type are added to form an aggregated relation representation, which is calculated as a weighted average of the source-hidden features with a sigmoid function, calculated as:

$$e_i^{\mathrm{r}} = \sigma\left(\sum_{k \in \Phi} \alpha_i^{(r_k)} W_r \mathrm{Z_i^{r_k}}\right) \tag{11}$$

Finally, the node and its node representation are added to obtain the representation of the target node i:

$$\mathrm{rep}_s = \mathrm{e_i^r} + \mathrm{e_i} + \mathrm{f_i} \tag{12}$$

### C. PREDICTION LAYER

For the target stock price prediction task, we use a shallow neural network implementation. We represent the prediction task as a classification problem: that is, we divide the future price trend into three categories up, neutral, down, and the task settings are described in detail later. The prediction network is a simple linear transformation layer defined as:

$$\hat{Y}_i = \mathrm{softmax}\left(W_n rep_s + b_n\right) \tag{13}$$

where $W_\mathrm{n} \in \mathbb{R}^{\mathrm{d}\times 1}$ and $b_n \in \mathbb{R}^l$ are the weight matrix and variance, respectively. where l is the number of target prediction classes, l=3 in this paper.

Finally, we train the model on the full stock relationship data using the likelihood cross-entropy loss in the output layer:

$$Loss_{\mathrm{node}} = -\sum_{i \in Z_u} \sum_{c=1}^{l} Y_{ic} \ln \hat{Y}_{ic} \tag{14}$$

where $Y_{ic}$ is a ground truth label of the $c_{th}$ movement class of stock i and $Z_u$ denotes the set of all stocks in the dataset.

$$\rho\left(v^{i+1} \mid v_t^i, \mathcal{P}\right) = \begin{cases} \frac{1}{\left|\mathrm{N_{t+1}}(v_t^i)\right|} & \left(v^{i+1}, v_t^i\right) \in \mathrm{E}, \phi\left(v^{i+1}\right) = t+1 \\ 0 & \left(v^{i+1}, v_t^i\right) \in \mathrm{E}, \phi\left(v^{i+1}\right) \neq t+1 \\ 0 & \left(v^{i+1}, v_t^i\right) \notin \mathrm{E} \end{cases} \tag{7}$$

## IV. EXPERIMENTS

In this section, we conduct extensive experiments to investigate the effectiveness and limitations of the proposed method and compare our method with several popular methods in finance from recent years. We first introduce the experimental data sources and experimental settings and then report the results of each experiment.

### A. DATA GATHERING AND PREPARATION

**Price data** Our research focuses on developed stock markets in the world, selecting 423 stocks of U.S. enterprises in the S&P 500 index and 286 stocks of Chinese companies in the CSI 300 index, respectively. We obtain historical price data for stocks in the S&P-500 Composite index from the Yahoo Finance website (https://finance.yahoo.com/) and for stocks in the CSI 300 index from the China Stock Market & Accounting Research Database. Our relational data come from Wikidata, which does not include all stocks in the index, so some stocks that have no relation to other companies (stocks) in Wikidata are excluded. For example, for the S&P 500 index, after removing irrelevant stocks, we included stocks of the remaining 423 companies as target stocks. The historical price dataset for this paper is described in detail in Table 1. Many research works directly feed the original features of historical prices, such as opening and closing prices, to a specific feature extraction module, and we use the historical price change rate as the input of LSTM. The rate of change in the price of a stock at time t can be calculated by: $R_i^t = \frac{P_i^t - P_i^{t-1}}{P_i^{t-1}}$ where $P_i^t$ and $P_i^{t-1}$ are the closing prices of stock i at time t and t-1, respectively. **Financial news data** For financial news, we extract news related to the target stock from websites such as Yahoo Finance within a specific time interval, and our final text dataset consists of nearly 150,000 texts about the target stock. **Corporate relation data** As a rich source of entity relationships, Wikidata contains a large number of corporate entities and corporate relationships, which may impact on stock. Our third type of data is relational data from Wikidata (https://www.wikidata.org/wiki/Wikidata: Main_Page). Wikidata is a free collaborative knowledge base that currently contains 48 million items (e.g., Google Inc., etc.) and hundreds of millions of sentences (e.g., Apple, founded by Steve Jobs). Inspired by the work of [28], we extract 9 kinds of first-order relations and 62 kinds of second-order relations from [35] through a meta-path with a maximum of only 2 hops.

### B. EXPERIMENT SETTINGS

As mentioned above, we divide the stock price training data into three categories according to the price change rate: up, neutral, down. See Section 4.1 for the calculation of the stock price change rate.We set the labels of price changes for the training data as follows:

$$f = \begin{cases} up & R_i^t \geq r_{up} \\ \text{neutral} & r_{\text{down}} \leq R_i^t \leq r_{up} \\ \text{down} & R_i^t \leq r_{\text{down}} \end{cases} \tag{15}$$

Here, we set $r_{up} = 0.6$, $r_{down} = -0.6$.

Trading strategies have been extensively studied [41], [42].

To measure the profitability of the model proposed in this paper, we simulate stock trading using some common trading strategies. We constructed a portfolio based on the predicted values obtained from the predictive model. Since it is divided into three classes at training time, the prediction vector of the model is three-dimensional, and the value of each dimension represents the predicted probability of each class. Every day, the stock is bought, sold, or held, and if the probability of upside is highest, the stock is bought with all the currently available capital at the closing price of the day. If the probability of decline is highest, then the is sold stock at the closing price of the day. Otherwise, no transaction is made that day.

We implement our model in TensorFlow, using the Adam optimizer [37] to tune parameters for 100 epochs on a single NVIDIA Tesla K80 GPU, with a learning rate of 5e-4, a weight decay of 5e-5, and a batch size of 32. To mitigate overfitting, we apply dropout [36] at the end of each layer and set its value to 0.5. We use leaky ReLU as the activation function. Depending on the actual experimental situation, the results vary widely for each iteration, so all of our experiments were repeated 10 times independently.

### C. EVALUATION METRICS

To compare the performance of the proposed method and the benchmark model, we chose common metrics to evaluate model classification and profitability. Stock trend prediction is a typical classification prediction task, so we first selected two evaluation indicators widely used in classification tasks: accuracy and F1-score. The calculation formulas are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{16}$$

$$F1 = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{17}$$

where TP=True Positive, TN=True Negative, FP=False Positive, FN=False Negative; Recall $= \frac{\text{TP}}{\text{TP+FP}}$ and Precision $= \frac{\text{TP}}{\text{TP+FN}}$.

We can obtain a macro F1 value by averaging the calculated F1 scores for each category.

To evaluate the profitability of the proposed methods, we use the following two metrics to compare the profitability of each method.

The formula for calculating the return of a portfolio is:

$$\text{Return}_i^t = \frac{1}{|F^{t-1}|} \sum_{i \in F^{t-1}} \frac{p_i^t - p_i^{t-1}}{p_i^{t-1}} \tag{18}$$

where $F^{t-1}$ denotes a group of stocks included in the portfolio at time t-1, $p_i^t$ denotes the price of stock i at time t, $|\cdot|$ and denotes the number of items in the set portfolio.

The Sharpe rate is an indicator that considers both return and risk and can be used to measure the performance of

**TABLE 2.** Parameter settings of various methods.

| Methods | Parameters and Description | Methods | Parameters and Description |
|---|---|---|---|
| MLP | Hidden layers: 128,64;<br>Optimizer: Adam<br>Learning rate: 0.0001<br>Epochs: 100 | TGC | Hidden: LSTM layers: 64,64<br>MLP layer: 1<br>Optimizer: Adam<br>Learning rate: 0.001<br>Epochs: 100 |
| CNN | Convolutional layer1: 16filters 2*2<br>Convolutional layer2: 32filters 2*2<br>Convolutional layer3: 64filters 2*2<br>Max pooling layer: 2*2<br>Fully connected layer: 500<br>Optimizer: rmsprop<br>Learning rate: 0.01<br>Epochs: 100 | GCN | Convolutional layer1: 64filters 2*2<br>Convolutional layer2: 64filters 2*2<br>Optimizer: Adam<br>Learning rate: 0.01<br>Epochs: 100 |
| LSTM | LSTM layers: 60,50,50,50<br>Dropout layer: 0.2<br>Optimizer: rmsprop<br>Learning rate: 0.01<br>Epochs: 100 | ML-GAT | The length of time series: 50<br>Hidden: MLP layer: 2<br>LSTM layer: 128<br>Optimizer: Adam<br>Learning rate: 5e-4<br>Dropout layer: 0.5<br>Activation function: Leaky_ReLU<br>BERT: BATCH_SIZE=32<br>MAX_token_LENGTH=128<br>Epochs: 100 |

investment risk compared to return. It is calculated as follows:

$$\text{Sharpe}_a = \frac{E\left[R_a - R_f\right]}{\sigma_p} \tag{19}$$

where $R_a$ denotes the portfolio rate of return, $R_f$ denotes the risk-free rate, and $\sigma_p$ denotes the standard deviation of the portfolio rate of return. We used the 13-week Treasury bill as the risk-free rate.

### D. BASELINE METHODS

In this section, we describe several benchmark models chosen for this paper.

(1)**MLP** Multilayer perceptron is one of the most widely used neural networks applied to stock forecasting. In this paper, we use a simple multilayer perceptron model with four layers, including two hidden layers and one layer prediction layer.

(2)**CNN [3]** Convolutional neural networks are fast in terms of modelling time series and are therefore widely used. We used a CNN network with 3 convolutional layers.

(3)**LSTM [31]** As a widely used deep learning model for time series forecasting tasks, the effectiveness of LSTM has been verified in many previous studies. In our experiments, we encode the final predictions on historical price data through a classical LSTM model with 2 layers.

(4)**GCN [15]** Using a GCN model with 2 convolutional layers and 1 prediction layer, the stock graph is reconstructed with historical information representing the relationship of the target company using historical price data as input to the node.

(5)**TGC [28]** The temporal graph convolution module proposed by Feng *et al.* is used to model stock relations. This module uses an LSTM to encode the current state of the stock and feeds the latest state to the GCN to explore the relationships between companies. The TGC model summarizes all the related information of the adjacent nodes of the target stock, but the ML-GAT summarizes the information of different relationship types and assigns different weights to the information of different relationship types.

**TABLE 3.** Results of using the best 10 relations.

| Relation Type | F1-score |
|---|---|
| Industry-Legal form | 0.4576 |
| Parent organization-Owner of | 0.4561 |
| Industry-Product or material produced | 0.4552 |
| Owned by-Subsidiary | 0.4547 |
| Founded by-Founded by | 0.4543 |
| Follows | 0.4535 |
| Parent organization | 04521 |
| Complies with-Complies with | 0.4502 |
| Subsidiary-Owner of | 0.4491 |
| Owner of-Parent organization | 0.4484 |

**TABLE 4.** Results of using the worst 10 relations.

| Relation Type | F1-score |
|---|---|
| Board member | 0.3112 |
| Lnstance of-Legal form | 0.3084 |
| Location of formation-Country | 0.3075 |
| Stock Exchange | 0.3053 |
| Country-Location of formation | 0.2952 |
| Country of origin-Country | 0.2948 |
| Country-Board member | 0.2886 |
| Country-Country of origin | 0.2851 |
| Instance of-Instance of | 0.2748 |
| Stock Exchange-Stock Exchange | 0.2665 |

Table 2 presents the parameters adopted by the ML-GAT-based method and other benchmark models.

**TABLE 5.** Classification accuracy scores for stock prediction tasks.

| F1-score | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MLP | | CNN | | LSTM | | GCN | | TGC | | ML-GAT | |
| Index | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 |
| 1 | 0.3462 | 0.4330 | 0.3986 | 0.3329 | 0.3873 | 0.4202 | 0.3989 | 0.4250 | 0.4337 | 0.4449 | 0.4983 | 0.5847 |
| 2 | 0.3331 | 0.3464 | 0.3977 | 0.3013 | 0.3821 | 0.4736 | 0.4307 | 0.4816 | 0.4528 | 0.4268 | 0.5224 | 0.6665 |
| 3 | 0.3728 | 0.4422 | 0.3861 | 0.4549 | 0.4155 | 0.5006 | 0.4142 | 0.4590 | 0.4476 | 0.5431 | 0.5035 | 0.5979 |
| 4 | 0.3661 | 0.4244 | 0.4058 | 0.3654 | 0.3958 | 0.4776 | 0.4192 | 0.3977 | 0.4352 | 0.4809 | 0.5183 | 0.6017 |
| 5 | 0.3749 | 0.3977 | 0.3895 | 0.4514 | 0.4129 | 0.5056 | 0.4335 | 0.4412 | 0.4449 | 0.4841 | 0.4980 | 0.5120 |
| 6 | 0.3776 | 0.3415 | 0.3984 | 0.3316 | 0.3862 | 0.3523 | 0.4011 | 0.3614 | 0.4311 | 0.3607 | 0.5221 | 0.5386 |
| 7 | 0.3302 | 0.2758 | 0.3892 | 0.3400 | 0.4104 | 0.4077 | 0.4308 | 0.4811 | 0.4574 | 0.3750 | 0.5078 | 0.5959 |
| 8 | 0.3494 | 0.2770 | 0.3733 | 0.4632 | 0.4138 | 0.4598 | 0.4317 | 0.5305 | 0.4318 | 0.4473 | 0.5386 | 0.5541 |
| 9 | 0.3653 | 0.4279 | 0.4012 | 0.3959 | 0.4120 | 0.3219 | 0.3953 | 0.4184 | 0.4289 | 0.5190 | 0.4941 | 0.5112 |
| 10 | 0.3525 | 0.4244 | 0.3991 | 0.3952 | 0.3882 | 0.3456 | 0.3903 | 0.3206 | 0.4484 | 0.4270 | 0.4985 | 0.5285 |
| Average | 0.3568 | 0.3790 | 0.3939 | 0.3832 | 0.4004 | 0.4265 | 0.4146 | 0.4317 | 0.4412 | 0.4509 | 0.5102 | 0.5691 |
| Accuracy | | | | | | | | | | | | |
| | MLP | | CNN | | LSTM | | GCN | | TGC | | ML-GAT | |
| Index | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 |
| 1 | 0.3267 | 0.3696 | 0.2912 | 0.3483 | 0.4467 | 0.4335 | 0.4345 | 0.3632 | 0.4779 | 0.4366 | 0.5172 | 0.5264 |
| 2 | 0.2711 | 0.3122 | 0.3423 | 0.4287 | 0.4365 | 0.4123 | 0.4515 | 0.4459 | 0.4674 | 0.5150 | 0.4972 | 0.3985 |
| 3 | 0.2759 | 0.2798 | 0.3601 | 0.3677 | 0.4212 | 0.3270 | 0.4519 | 0.4032 | 0.4443 | 0.4440 | 0.5094 | 0.5126 |
| 4 | 0.3017 | 0.4202 | 0.3497 | 0.3906 | 0.4498 | 0.3912 | 0.4564 | 0.4423 | 0.4498 | 0.4636 | 0.5036 | 0.5866 |
| 5 | 0.3181 | 0.4023 | 0.3136 | 0.4503 | 0.4317 | 0.3650 | 0.4481 | 0.4505 | 0.4569 | 0.3827 | 0.5184 | 0.6507 |
| 6 | 0.2875 | 0.3079 | 0.3543 | 0.4253 | 0.4616 | 0.3931 | 0.4384 | 0.4531 | 0.4473 | 0.4956 | 0.5069 | 0.6162 |
| 7 | 0.3095 | 0.4005 | 0.2885 | 0.4243 | 0.4412 | 0.3751 | 0.4356 | 0.2938 | 0.4694 | 0.5016 | 0.5174 | 0.4912 |
| 8 | 0.3073 | 0.3829 | 0.3429 | 0.4336 | 0.4345 | 0.4444 | 0.4332 | 0.4140 | 0.4665 | 0.3706 | 0.5193 | 0.6524 |
| 9 | 0.3039 | 0.3759 | 0.2827 | 0.2865 | 0.4574 | 0.3980 | 0.4238 | 0.3896 | 0.4769 | 0.4047 | 0.4983 | 0.6249 |
| 10 | 0.3199 | 0.3376 | 0.3528 | 0.3693 | 0.4474 | 0.4882 | 0.4480 | 0.4828 | 0.4693 | 0.4235 | 0.4977 | 0.6383 |
| Average | 0.3022 | 0.3589 | 0.3278 | 0.3924 | 0.4428 | 0.4028 | 0.4421 | 0.4138 | 0.4626 | 0.4438 | 0.5085 | 0.5698 |

### E. ANALYSIS OF THE EFFECT OF USING DIFFERENT RELATIONAL DATA

Before the stock market trend forecasting task, we explored the impact of different types of relational data on stock forecasting. In this paper, we use a basic GCN model that effectively handles graph-structured data, but which cannot distinguish between different types of relations, with the aim of fairly measuring the different degrees of influence of different types of relations and thus optimizing the existing types of relations. We use a GCN with two convolutional layers and one prediction layer, whose forward propagation is defined as follows:

$$Y^{GCN} = \text{softmax}\left(\widehat{A}\,\text{Re}\,LU\left(\widehat{A}\,\text{Re}\,LU\left(\widehat{A}X'W^{(0)}\right)W^{(1)}\right)W^{(2)}\right) \tag{20}$$

where $\widehat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ and $\tilde{A} = A + I$ is the adjacency matrix. Changing the relation type changes the adjacency matrix of the GCN input. Therefore, we input each relation type separately to the GCN as its adjacency matrix. Finally, the 10 best and 10 worst relation types with their F1 scores on the test set are listed in the table 3&4.

We found that the use of relational data does not always lead to good results from our experiments, and in the worst case, the introduction of relational data significantly reduces the performance of stock market forecasts. From the table, we can obtain that the best relationship performance is 19.11% higher than the worst relationship performance. Therefore, we use the 10 best relations obtained in our experiments to create relational graphs for stocks for forecasting.

### F. COMPARISON RESULTS OF DIFFERENT METHODS

#### 1) CLASSIFICATION ACCURACY

Table 5 summarizes the classification accuracy results of different index stocks based on different methods in stock trend prediction experiments. From the results table, we can see that among the 3 benchmark models that do not consider modelling of the relationships between stocks, LSTM performs better than the other models in both F1-score and accuracy. Therefore, when compared with the relational modelling module, we simply compare the results of our model with the results of the LSTM model. In terms of the F1-score, all models with relational modelling modules perform better than LSTM. However, not all models that consider modelling of the relationships between stocks are more accurate than LSTM, and GCN performs slightly worse than LSTM in terms of accuracy. In particular, we can determine that our ML-GAT model generally outperforms the other models in 10 repeated experiments.

#### 2) PROFITABILITY

We calculate the daily returns of the portfolio according to the trading strategies in the previous section, and the profitability results of different models are summarized in Table 6. On average, ML-GAT and TGC achieved relatively high average daily returns, with ML-GAT yielding the most competitive daily returns, far greater than TGC and other

**TABLE 6. Profitability results of the stock prediction task.**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Average Daily Return | | | | | | | | | | | |
| | MLP | | CNN | | LSTM | | GCN | | TGC | | ML-GAT | |
| Index | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 |
| 1 | -0.0279 | -0.0170 | 0.0804 | 0.0876 | 0.0756 | 0.0852 | 0.0744 | 0.0765 | 0.1532 | 0.1675 | 0.1152 | 0.1218 |
| 2 | 0.0161 | 0.0349 | 0.0851 | 0.0918 | 0.0557 | 0.0632 | -0.0557 | -0.0391 | 0.1457 | 0.1641 | 0.1114 | 0.1334 |
| 3 | -0.0605 | -0.0547 | -0.0627 | -0.0432 | -0.1037 | -0.0877 | 0.0907 | 0.1067 | 0.1588 | 0.1699 | 0.2130 | 0.2381 |
| 4 | -0.0538 | -0.0534 | 0.0784 | 0.0979 | 0.0580 | 0.0583 | 0.0531 | 0.0605 | -0.1295 | -0.1233 | 0.0268 | 0.0394 |
| 5 | -0.0025 | -0.0001 | 0.0408 | 0.0568 | 0.0883 | 0.1040 | 0.0852 | 0.0927 | 0.1420 | 0.1581 | 0.1155 | 0.1169 |
| 6 | 0.0630 | 0.0772 | -0.0212 | -0.0141 | 0.0731 | 0.0798 | 0.0450 | 0.0572 | 0.0923 | 0.1004 | 0.0382 | 0.0535 |
| 7 | -0.0501 | -0.0484 | 0.0493 | 0.0579 | -0.0914 | -0.0808 | 0.0861 | 0.0974 | 0.0774 | 0.0778 | 0.1051 | 0.1325 |
| 8 | 0.0862 | 0.0974 | -0.0540 | -0.0500 | 0.0445 | 0.0517 | -0.1064 | -0.1035 | -0.0587 | -0.0419 | 0.0242 | 0.0441 |
| 9 | -0.0111 | 0.0004 | 0.0479 | 0.0548 | -0.0296 | -0.0115 | 0.1165 | 0.1349 | 0.0427 | 0.0527 | 0.3991 | 0.4048 |
| 10 | 0.0344 | 0.0444 | 0.0722 | 0.0739 | 0.1171 | 0.1317 | 0.0973 | 0.1133 | 0.0948 | 0.1023 | 0.0440 | 0.0499 |
| Average | -0.0006 | 0.0081 | 0.0316 | 0.0413 | 0.0288 | 0.0394 | 0.0486 | 0.0597 | 0.0719 | 0.0828 | 0.1193 | 0.1334 |
| | Sharpe Ratio | | | | | | | | | | | |
| | MLP | | CNN | | LSTM | | GCN | | TGC | | ML-GAT | |
| Index | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 | S&P 500 | CSI 300 |
| 1 | 1.6805 | 1.8883 | -1.5475 | -1.3032 | 1.4447 | 1.5709 | 2.3555 | 2.9921 | 1.4076 | 0.4140 | 1.3997 | 1.5005 |
| 2 | -0.2416 | -0.0616 | 1.3258 | 0.4882 | -2.0754 | -1.5823 | 1.0749 | 1.1266 | 1.8283 | 1.9604 | 1.4691 | 1.6582 |
| 3 | 1.7833 | -1.6257 | 1.2976 | 1.8615 | 1.9577 | 2.3893 | -0.2295 | 1.1076 | 0.7839 | 1.2226 | 1.2263 | 1.5720 |
| 4 | 1.3725 | 1.4773 | -0.6959 | -0.2802 | 1.0672 | 1.1074 | -0.2419 | 0.2806 | -2.0261 | -1.5876 | 2.4552 | 2.7469 |
| 5 | -1.4729 | -0.3308 | 2.1807 | 2.6321 | 1.8222 | 1.8355 | 0.4541 | 0.7677 | 1.3664 | 1.7170 | 2.0245 | 1.4554 |
| 6 | -1.8405 | -1.5776 | 0.3249 | 0.3603 | 1.4248 | 1.5229 | -0.9291 | -0.8633 | 1.5023 | 1.8412 | 1.1347 | 1.2069 |
| 7 | 1.5891 | 2.4691 | -1.6977 | -1.2014 | 1.4555 | 0.4968 | -1.2779 | -1.1375 | 1.2599 | 0.6794 | 2.2177 | 2.6335 |
| 8 | -0.1580 | 0.3293 | 1.9360 | 1.3091 | 1.2072 | 0.2192 | 2.0270 | -1.8652 | 0.7183 | 0.8443 | 1.9186 | 2.0400 |
| 9 | 0.0119 | 0.2824 | 1.8290 | 1.4280 | -1.4811 | -1.0903 | 0.7792 | 1.8938 | 0.6135 | 0.8898 | 3.0144 | 1.7539 |
| 10 | 1.5192 | 1.9732 | -0.6809 | -0.3923 | 1.0610 | 1.0778 | 1.1681 | 1.5553 | 1.0720 | 1.5318 | 2.0282 | 2.4270 |
| Average | 0.4243 | 0.4824 | 0.4272 | 0.4902 | 0.7884 | 0.7547 | 0.5180 | 0.5858 | 0.8526 | 0.9513 | 1.8889 | 1.8994 |

benchmark models. More notable here is that GCN is better than the model without a relational modelling module with respect to F1-score, but the Sharpe rate of GCN is much lower than those of LSTM and other benchmark models of unrelated modelling modules, but the Sharpe rate of TGC is higher than those of all benchmark models outside of ML-GAT. Judging from the variance of the results of multiple experiments, the Sharpe rate variance of several groups of benchmark models is much larger than our model, which further verifies that our model exhibits more stable characteristics in the profitability test. In conclusion, the ML-GAT model achieves significant results for both the expected average daily returns and the Sharpe rate.

In addition, we assume that the asset value starts at 1,000, and the changes in asset value in simulated transactions for different index stocks based on different forecasting models are shown in Figure 4&5. At the end of 2018 and early 2019, the asset value predicted by our ML-GAT model exhibited a large drop, while the asset values of the other methods did not seem to change much. In fact, at the end of 2018, U.S. stocks slumped collectively mainly due to the fermentation of the Turkish financial crisis and Trump's aggressive sanctions against China; coupled with the outbreak of the new coronavirus epidemic in late 2019, U.S. stocks were also greatly affected. This also verifies the superiority of our method in capturing the characteristics of current affairs news in the financial field through BERT. After experiencing two sudden black swan events in the stock market, our method and trading

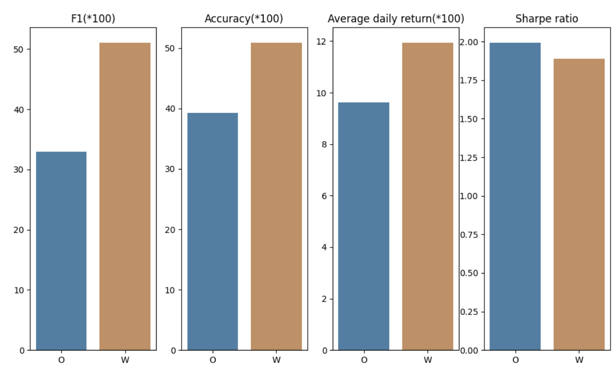


**FIGURE 3. Influence of ML-GAT with or without news information using the S&P 500 index stocks as an example. "O" indicates that there is no BERT extracting the news information feature module, and "W" indicates that there is a BERT extracting the news information feature module.**

strategy played an important role in stopping the loss and obtaining better earnings.

Finally, we analyse the above experimental results and observations and draw the following conclusions:

1) By comparing the relational modelling module and the nonrelational modelling module model, the introduction of interstock relations can usually have a positive impact on the trend forecasting task. According to [39], the research on the new model will result in more than 12% excess profit when the prediction accuracy is improved by 0.005. Based on model experimental results, our proposed method significantly outperforms other baselines.

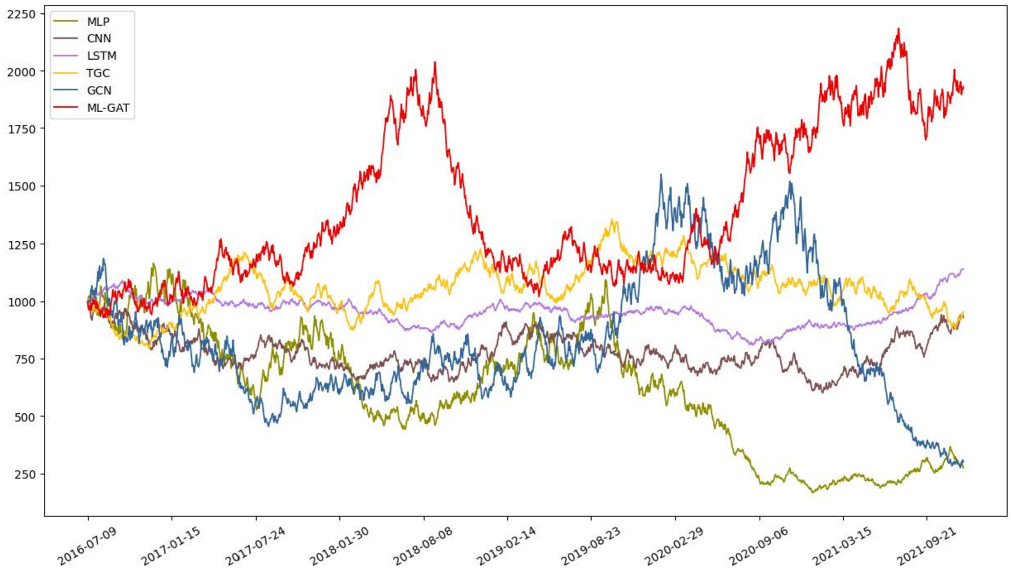


**FIGURE 4.** Comparison of the change in the value of S&P 500 assets, assuming assets are invested from 1000.

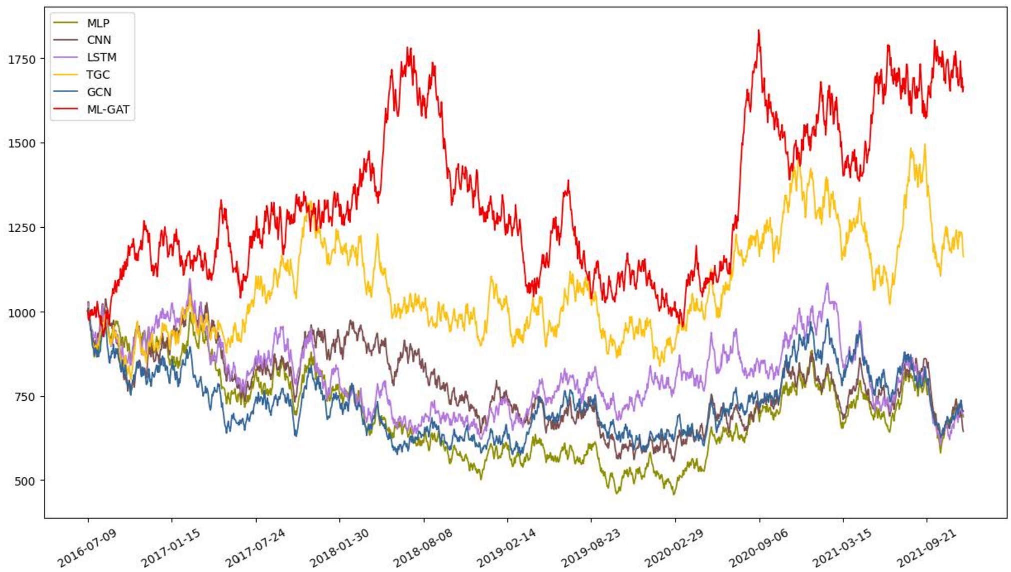


**FIGURE 5.** Comparison of the change in the value of CSI 300 assets, assuming assets are invested from 1000.

**TABLE 7.** List of the first-order Wiki relations.

| | Relation Index | Relation Name | Description |
|---|---|---|---|
| 1 | R=P127 | Owend by | owner of the subject. |
| 2 | R=P155 | Follows | immediately prior item in a series of which the subject is a part. |
| 3 | R=P156 | Followed by | immediately following item in a series of which the subject is a part. |
| 4 | R=P355 | Subsidiary | subsidiary of a company or organization, opposite of parent organization. |
| 5 | R=P361 | Part of | object of which the subject is a part. |
| 6 | R=P414 | Stock Exchange | exchange on which this company is traded. |
| 7 | R=P749 | Parent organization | parent organization of an organization, opposite of subsidiaries. |
| 8 | R=P1830 | Owner of | entities owned by the subject. |
| 9 | R=P1889 | Different from | item that is different from another item, with which it is often confused |

2) To improve the accuracy of model prediction, it is necessary not only to consider modelling stock relationships, but also to selectively aggregate different types of relationship information. During the experiment, we found that the GCN

**TABLE 8. List of the second-order Wiki relations.**

| | Relation Index | Relation Name | Description |
|---|---|---|---|
| 1 | R1=P17 | Country | sovereign state of this item; don't use on humans |
| | R2=P159 | Headquarters location | specific location where an organization's headquarters is or has been situated. |
| 2 | R1=P17 | Country | sovereign state of this item; don't use on humans |
| | R2=P495 | Country of origin | country of origin of this item (creative work, food, phrase, product, etc.) |
| 3 | R1=P17 | Country | sovereign state of this item; don't use on humans |
| | R2=P740 | Location of formation | location where a group or organization was formed |
| 4 | R1=P31 | Instance of | that class of which this subject is a particular example and member |
| | R2=P452 | Industry | industry of company or organization |
| 5 | R1=P31 | Instance of | that class of which this subject is a particular example and member |
| | R2=P1056 | Product or material produced | material or product produced by a government agency, business, industry, facility, or process |
| 6 | R1=P31 | Instance of | that class of which this subject is a particular example and member |
| | R2=P1454 | Legal form | legal form of an organization |
| 7 | R1=P112 | Founded by | founder or co-founder of this organization, religion or place |
| | R2=P112 | Founded by | founder or co-founder of this organization, religion or place |
| 8 | R1=P112 | Founded by | founder or co-founder of this organization, religion or place |
| | R2=P749 | Parent organization | parent organization of an organisation, opposite of subsidiaries (P355) |
| 9 | R1=P121 | Item operated | equipment, installation or service operated by the subject |
| | R2=P121 | Item operated | equipment, installation or service operated by the subject |
| 10 | R1=P127 | Owned by | owner of the subject |
| | R2=P127 | Owned by | owner of the subject |
| 11 | R1=P127 | Owned by | owner of the subject |
| | R2=P355 | Subsidiary | subsidiary of a company or organization, opposite of parent organization |
| 12 | R1=P127 | Owned by | owner of the subject |
| | R2=P749 | Parent organization | parent organization of an organisation, opposite of subsidiaries (P355) |
| 13 | R1=P127 | Owned by | owner of the subject |
| | R2=P1830 | Owner of | entities owned by the subject |
| 14 | R1=P131 | Located in the administrative territorial entity | the item is located on the territory of the following administrative entity. |
| | R2=P159 | Headquarters location | specific location where an organization's headquarters is or has been situated. |
| 15 | R1=P155 | Follows | immediately prior item in a series of which the subject is a part |
| | R2=P155 | Follows | immediately prior item in a series of which the subject is a part |
| 16 | R1=P155 | Follows | immediately prior item in a series of which the subject is a part |
| | R2=P355 | Subsidiary | subsidiary of a company or organization, opposite of parent organization |
| 17 | R1=P159 | Headquarters location | specific location where an organization's headquarters is or has been situated. |
| | R2=P17 | Country | sovereign state of this item; don't use on humans |
| 18 | R1=P159 | Headquarters location | specific location where an organization's headquarters is or has been situated. |
| | R2=P131 | Located in the administrative territorial entity | the item is located on the territory of the following administrative entity. |
| 19 | R1=P159 | Headquarters location | specific location where an organization's headquarters is or has been situated. |
| | R2=P159 | Headquarters location | specific location where an organization's headquarters is or has been situated. |
| 20 | R1=P159 | Headquarters location | specific location where an organization's headquarters is or has been situated. |
| | R2=P740 | Location of formation | location where a group or organization was formed |
| 21 | R1=P166 | Award received | award or recognition received by a person, organisation or creative work |
| | R2=P166 | Award received | award or recognition received by a person, organisation or creative work |
| 22 | R1=P176 | Manufacturer | manufacturer or producer of this product |
| | R2=P452 | Industry | industry of company or organization |
| 23 | R1=P355 | Subsidiary | subsidiary of a company or organization, opposite of parent organization |
| | R2=P127 | Owned by | owner of the subject |
| 24 | R1=P355 | Subsidiary | subsidiary of a company or organization, opposite of parent organization |
| | R2=P155 | Follows | immediately prior item in a series of which the subject is a part |
| 25 | R1=P355 | Subsidiary | subsidiary of a company or organization, opposite of parent organization |
| | R2=P355 | Subsidiary | subsidiary of a company or organization, opposite of parent organization |
| 26 | R1=P355 | Subsidiary | subsidiary of a company or organization, opposite of parent organization |
| | R2=P1830 | Owner of | entities owned by the subject |
| 27 | R1=P361 | Part of | object of which the subject is a part |
| | R2=P361 | Part of | object of which the subject is a part |
| 28 | R1=P361 | Part of | object of which the subject is a part |
| | R2=P414 | Stock Exchange | exchange on which this company is traded |
| 29 | R1=P361 | Part of | object of which the subject is a part |
| | R2=P463 | Member of | organization or club to which the subject belongs |
| 30 | R1=P414 | Stock Exchange | exchange on which this company is traded |
| | R2=P361 | Part of | object of which the subject is a part |
| 31 | R1=P452 | Industry | industry of company or organization |
| | R2=P31 | Instance of | that class of which this subject is a particular example and member |
| 32 | R1=P452 | Industry | industry of company or organization |
| | R2=P176 | Manufacturer | manufacturer or producer of this product |
| 33 | R1=P452 | Industry | industry of company or organization |
| | R2=P452 | Industry | industry of company or organization |
| 34 | R1=P452 | Industry | industry of company or organization |
| | R2=P1056 | Product or material produced | material or product produced by a government agency, business, industry, facility, or process |

**TABLE 8.** (*Continued.*) List of the second-order Wiki relations.

| | | | |
|---|---|---|---|
| 35 | R1=P452 | Industry | industry of company or organization |
| | R2=P1454 | Legal form | legal form of an organization |
| 36 | R1=P463 | Member of | organization or club to which the subject belongs |
| | R2=P361 | Part of | object of which the subject is a part |
| 37 | R1=P463 | Member of | organization or club to which the subject belongs |
| | R2=P463 | Member of | organization or club to which the subject belongs |
| 38 | R1=P495 | Country of origin | country of origin of this item (creative work, food, phrase, product, etc.) |
| | R2=P17 | Country | sovereign state of this item; don't use on humans |
| 39 | R1=P495 | Country of origin | country of origin of this item (creative work, food, phrase, product, etc.) |
| | R2=P740 | Location of formation | location where a group or organization was formed |
| 40 | R1=P625 | Coordinate location | geocoordinates of the subject. |
| | R2=P625 | Coordinate location | geocoordinates of the subject. |
| 41 | R1=P740 | Location of formation | location where a group or organization was formed |
| | R2=P17 | Country | sovereign state of this item; don't use on humans |
| 42 | R1=P740 | Location of formation | location where a group or organization was formed |
| | R2=P159 | Headquarters location | specific location where an organization's headquarters is or has been situated. |
| 43 | R1=P740 | Location of formation | location where a group or organization was formed |
| | R2=P495 | Country of origin | country of origin of this item (creative work, food, phrase, product, etc.) |
| 44 | R1=P740 | Location of formation | location where a group or organization was formed |
| | R2=P740 | Location of formation | location where a group or organization was formed |
| 45 | R1=P749 | Parent organization | parent organization of an organisation, opposite of subsidiaries (P355) |
| | R2=P112 | Founded by | founder or co-founder of this organization, religion or place |
| 46 | R1=P749 | Parent organization | parent organization of an organisation, opposite of subsidiaries (P355) |
| | R2=P127 | Owned by | owner of the subject |
| 47 | R1=P749 | Parent organization | parent organization of an organisation, opposite of subsidiaries (P355) |
| | R2=P749 | Parent organization | parent organization of an organisation, opposite of subsidiaries (P355) |
| 48 | R1=P749 | Parent organization | parent organization of an organisation, opposite of subsidiaries (P355) |
| | R2=P1830 | Owner of | entities owned by the subject |
| 49 | R1=P793 | Significant event | significant or notable events associated with the subject |
| | R2=P793 | Significant event | significant or notable events associated with the subject |
| 50 | R1=P1056 | Product or material produced | material or product produced by a government agency, business, industry, facility, or process |
| | R2=P31 | Instance of | that class of which this subject is a particular example and member |
| 51 | R1=P1056 | Product or material produced | material or product produced by a government agency, business, industry, facility, or process |
| | R2=P452 | Industry | industry of company or organization |
| 52 | R1=P1056 | Product or material produced | material or product produced by a government agency, business, industry, facility, or process |
| | R2=P1056 | Product or material produced | material or product produced by a government agency, business, industry, facility, or process |
| 53 | R1=P1344 | Participant of | event a person or an organization was/is a participant in, |
| | R2=P1344 | Participant of | event a person or an organization was/is a participant in, |
| 54 | R1=P1454 | Legal form | legal form of an organization |
| | R2=P31 | Instance of | that class of which this subject is a particular example and member |
| 55 | R1=P1454 | Legal form | legal form of an organization |
| | R2=P452 | Industry | industry of company or organization |
| 56 | R1=P1454 | Legal form | legal form of an organization |
| | R2=P1454 | Legal form | legal form of an organization |
| 57 | R1=P1830 | Owner of | entities owned by the subject |
| | R2=P127 | Owned by | owner of the subject |
| 58 | R1=P1830 | Owner of | entities owned by the subject |
| | R2=P355 | Subsidiary | subsidiary of a company or organization, opposite of parent organization |
| 59 | R1=P1830 | Owner of | entities owned by the subject |
| | R2=P749 | Parent organization | parent organization of an organisation, opposite of subsidiaries (P355) |
| 60 | R1=P1830 | Owner of | entities owned by the subject |
| | R2=P1830 | Owner of | entities owned by the subject |
| 61 | R1=P5009 | Complies with | the product or work complies with a certain norm or passes a test |
| | R2=P5009 | Complies with | the product or work complies with a certain norm or passes a test |
| 62 | R1=P6379 | Has works in the collection | collection that have works of this artist |
| | R2=P6379 | Has works in the collection | collection that have works of this artist |

fairly considers all relational information when modelling relations, so the GCN obtains a higher F1-score but lower accuracy than LSTM. On the other hand, ML-GAT assigns different weights to different relationship types according to the current state of the node through a multilayer attention mechanism, so it achieves better results.

3) Predictions are made by vector representations extracted with BERT from financial news and then embedded in ML-GAT. Our model can find stock trends from news events, which further validates the good performance enabled by including financial news in stock trend prediction.

### G. THE EFFECT OF NEWS COMPONENT

In Figure 3, we show the experimental results with and without the BERT feature module for extracting news information. From the figure, it can be determined that the F1-score, accuracy, and average daily return of ML-GAT after capturing news information features are improved by 18.08%, 11.54%, and 2.3%, respectively, compared to the feature extraction module without news information. Regrettably, in terms of the Sharpe ratio, the model without the BERT extraction feature module performs better. We have deeply considered this result, and after many experiments, we hypothesize that it occurs because the news text data are sparse. Sometimes there is a lot of daily news, and sometimes there is no important news in a day, so the news in the market is still not sufficient to accurately infer stock trends and investment returns.

### H. WIKIDATA COMPANY-BASED RELATIONS

In this paper, we obtain 9 types of first-order ($A \xrightarrow{R} B$) and 62 types of second-order ($A \overset{R_1}{\rightarrow} B \overset{R_2}{\leftarrow} C$)relations between companies corresponding to the selected stocks in S&P500 as shown in the table 5&6. A, B, and C represent entities in Wikidata; $R$, $R_1$,and $R_2$ denotes different relations types defined in Wikidata. The details of the obtained relations are in the table 7&8.

## V. CONCLUSION

In this study, we propose a multilayer graph attention neural network model for stock trend prediction. We incorporate data describing financial markets, news, and corporate relations into a graph neural attention network-based model through a specific feature extraction module to compensate for the lack of prior knowledge of existing stock forecasting methods. ML-GAT aims to selectively filter different types of information to form an aggregated graph through multiple layers of attention mechanisms at different levels to learn the feature representation of nodes that are useful for prediction tasks. It is hoped that prediction accuracy can be improved through graph-based learning. We compare ML-GAT with common benchmark models based on public datasets to evaluate the effectiveness of the proposed method.

The results also demonstrate the importance of using financial news and relational data. GCN, TGC, and ML-GAT involve different aggregation methods for relational data that lead to different prediction accuracies.

We will continue to improve our future work mainly from the following aspects. First, we will explore different coding tools suitable for different characteristics in the financial field, such as processing financial news texts with more advanced natural language technology, to more accurately capture different dimensions' feature of the stock changing process. Second, we will perform factor optimization analysis on the obtained multiple characteristics and select the most important technical indicators to forecast the stock trend to achieve higher accuracy. Then, we will improve the method of constructing relational networks between stocks based on large knowledge bases such as Wikidata but not limited to Wikidata. We should try more ways to build a network of relationships between stocks, such as use a multi-Hawkes Process to initial a correlation graph between stocks [43]. Finally, we plan to apply the model to stock market risk prevention and quantitative investment and generalize it to forecasting tasks in other financial fields.

## REFERENCES

[1] S. Feng, C. Xu, Y. Zuo, G. Chen, F. Lin, and J. XiaHou, "Relation-aware dynamic attributed graph attention network for stocks recommendation," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108119.

[2] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 4741–4753, May 2021.

[3] E. Hoseinzade and S. Haratizadeh, "CNNpred: CNN-based stock market prediction using a diverse set of variables," *Expert Syst. Appl.*, vol. 129, pp. 273–285, Sep. 2019.

[4] J. M.-T. Wu, Z. Li, G. Srivastava, M.-H. Tasi, and J. C.-W. Lin, "A graph-based convolutional neural network stock price prediction with leading indicators," *Softw., Pract. Exp.*, vol. 51, no. 3, pp. 628–644, Mar. 2021.

[5] J. Zhao, D. Zeng, S. Liang, H. Kang, and Q. Liu, "Prediction model for stock price trend based on recurrent neural network," *J. Ambient Intell. Hum. Comput.*, vol. 12, no. 1, pp. 745–753, Jan. 2021.

[6] J. M.-T. Wu, Z. Li, N. Herencsar, B. Vo, and J. C.-W. Lin, "A graph-based CNN-LSTM stock price prediction algorithm with leading indicators," *Multimedia Syst.*, pp. 1–20, Feb. 2021.

[7] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *J. Finance*, vol. 25, no. 2, pp. 383–417, May 1970.

[8] R. K. Dash, T. N. Nguyen, K. Cengiz, and A. Sharma, "Fine-tuned support vector regression model for stock predictions," *Neural Comput. Appl.*, pp. 1–15, Mar. 2021.

[9] W. Chen, H. Zhang, M. K. Mehlawat, and L. Jia, "Mean–variance portfolio optimization using machine learning-based stock price prediction," *Appl. Soft Comput.*, vol. 100, Mar. 2021, Art. no. 106943.

[10] Q. Zhang, L. Yang, and F. Zhou, "Attention enhanced long short-term memory network with multi-source heterogeneous information fusion: An application to BGI genomics," *Inf. Sci.*, vol. 553, pp. 305–330, Apr. 2021.

[11] Y. Chen, Z. Wei, and X. Huang, "Incorporating corporation relationship via graph convolutional neural networks for stock price prediction," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag.*, Oct. 2018, pp. 1655–1658.

[12] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su, "Modeling the stock relation with graph network for overnight stock movement prediction," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 4541–4547.

[13] H. Wang, S. Li, T. Wang, and J. Zheng, "Hierarchical adaptive temporal-relational modeling for stock trend prediction," in *Proc. IJCAI*, 2021, pp. 3691–3698.

[14] D. Matsunaga, T. Suzumura, and T. Takahashi, "Exploring graph neural networks for stock market predictions with rolling window analysis," 2019, *arXiv:1909.10660*.

[15] W. Chen, M. Jiang, W.-G. Zhang, and Z. Chen, "A novel graph convolutional feature based convolutional neural network for stock trend prediction," *Inf. Sci.*, vol. 556, pp. 67–94, May 2021.

[16] D. Kahneman, "Maps of bounded rationality: Psychology for behavioral economics," *Amer. Econ. Rev.*, vol. 93, no. 5, pp. 1449–1475, Nov. 2003.

[17] R. K. Nayak, D. Mishra, and A. K. Rath, "A Naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices," *Appl. Soft Comput.*, vol. 35, pp. 670–680, Oct. 2015.

[18] C. Lohrmann and P. Luukka, "Classification of intraday S&P500 returns with a random forest," *Int. J. Forecasting*, vol. 35, no. 1, pp. 390–407, Jan. 2019.

[19] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–7.

[20] H. Wang, T. Wang, and Y. Li, "Incorporating expert-based investment opinion signals in stock prediction: A deep learning framework," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 971–978.

[21] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, and D. C. L. Ngo, "Text mining for market prediction: A systematic review," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7653–7670, 2014.

[22] J. Liu, H. Lin, X. Liu, B. Xu, Y. Ren, Y. Diao, and L. Yang, "Transformer-based capsule network for stock movement prediction," in *Proc. 1st Workshop Financial Technol. Natural Lang. Process.*, 2019, pp. 66–73.

[23] J. Tang and X. Chen, "Stock market prediction based on historic prices and news titles," in *Proc. Int. Conf. Mach. Learn. Technol.*, 2018, pp. 29–34.

[24] J. Liu, C. Xia, X. Li, H. Yan, and T. Liu, "A BERT-based ensemble model for Chinese news topic prediction," in *Proc. 2nd Int. Conf. Big Data Eng.*, May 2020, pp. 18–23.
[25] Q. Li, J. Tan, J. Wang, and H. Chen, "A multimodal event-driven LSTM model for stock prediction using online news," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3323–3337, Oct. 2021.
[26] C.-H. Chen, P.-Y. Chen, and J. C.-W. Lin, "An ensemble classifier for stock trend prediction using sentence-level Chinese news sentiment and technical indicators," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 7, no. 3, SI, pp. 53–64, Mar. 2022, doi: 10.9781/ijimai.2022.02.004.
[27] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.
[28] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–30, Apr. 2019.
[29] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108218.
[30] R. Kim, C. Ho So, M. Jeong, S. Lee, J. Kim, and J. Kang, "HATS: A hierarchical graph attention network for stock movement prediction," 2019, *arXiv:1908.07999*.
[31] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, 2018.
[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[33] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 135–144.
[34] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Commun. ACM*, vol. 57, no. 10, pp. 78–85, 2014.
[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[37] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[38] M. L. De Prado, *Advances in Financial Machine Learning*. Hoboken, NJ, USA: Wiley, 2018.
[39] S. Bouktif, A. Fiaz, and M. Awad, "Augmented textual features-based stock market prediction," *IEEE Access*, vol. 8, pp. 40269–40282, 2020.
[40] S. M. Idrees, M. A. Alam, and P. Agarwal, "A prediction approach for stock market volatility based on time series data," *IEEE Access*, vol. 7, pp. 17287–17298, 2019.
[41] M.-E. Wu, J.-H. Syu, J. C.-W. Lin, and J.-M. Ho, "Effective fuzzy system for qualifying the characteristics of stocks by random trading," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 8, pp. 3152–3165, Aug. 2022.
[42] C.-H. Chen, C.-Y. Lu, T.-P. Hong, J. C.-W. Lin, and M. Gaeta, "An effective approach for the diverse group stock portfolio optimization using grouping genetic algorithm," *IEEE Access*, vol. 7, pp. 155871–155884, 2019.
[43] T. Yin, C. Liu, F. Ding, Z. Feng, B. Yuan, and N. Zhang, "Graph-based stock correlation and prediction for high-frequency trading systems," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108209.
[44] J. C.-W. Lin, Y. Shao, Y. Djenouri, and U. Yun, "ASRNN: A recurrent neural network with an attention model for sequence labeling," *Knowl.-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106548.

**KUN HUANG** received the bachelor's degree in engineering, in 2021. He is currently pursuing the master's degree in applied statistics with the Zhejiang University of Science and Technology. His current research interests include financial market forecasting and natural language processing.

**XIAOMING LI** received the master's degree from Xinjiang University, in 2008, and the Ph.D. degree from Tianjin University, in March 2020. He is currently pursuing the Ph.D. degree with the College of International Business, Zhejiang Yuexiu University. He is also an Associate Professor with the College of International Business, Zhejiang Yuexiu University, and the Deputy Director of the Shaoxing Key Laboratory of Intelligent Monitoring and Prevention of Smart City. His research interests include human behavior dynamics and multi-layer network local community detection.

**FANGYUAN LIU** received the B.S. degree in international economics and trades from the East China University of science and technology, Shanghai, China, in 2018, and the M.S. degree in international accounting and finance from the University of Birmingham, Birmingham, U.K., in 2019. She is currently pursuing the Ph.D. degree in business and management with UCSI University, Kuala Lumpur, Malaysia.

**XIAOPING YANG** is currently pursuing the Ph.D. degree with the School of International Business, Lead researcher of Institute of Modern Economics and Management, Zhejiang Yuexiu University of Foreign Languages. He is also a Professor and the Dean of the School of International Business, Lead Researcher of Institute of Modern Economics and Management, Zhejiang Yuexiu University of Foreign Languages. His main research interests include business big data, corporate finance, and governance.

**WEI YU** received the B.S. degree from Jinggangshan University, Ji'an, China, in 2012, the M.S. degree from Henan Normal University, Xinxiang, China, in 2015, and the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2020. He is currently a Lecturer with Zhejiang Yuexiu University, Shaoxing, China. His research interests include dynamic complex network analysis, large-scale data mining, and machine learning.