AI Open 4 (2023) 165–174

Contents lists available at ScienceDirect

AI Open

journal homepage: www.keaipublishing.com/en/journals/ai-open

MONEY: Ensemble learning for stock price movement prediction via a convolutional network with adversarial hypergraph model

Zhongtian Sun [a,*], Anoushka Harit [a], Alexandra I. Cristea [a], Jingyun Wang [a], Pietro Lio [b]

[a] Department of Computer Science, Durham University, Stockton Road, DH1 3LE, Durham, UK
[b] Department of Computer Science and Technology, University of Cambridge, 15 JJ Thomson Avenue, CB3 0FD, Cambridge, UK

ARTICLE INFO

Keywords:
Semi-supervised learning
Graph neural network
Knowledge representation and application

ABSTRACT

Stock price prediction is challenging in financial investment, with the AI boom leading to increased interest from researchers. Despite these recent advances, many studies are limited to capturing the time series characteristics of price movement via recurrent neural networks (RNNs) but neglect other critical relevant factors, such as industry, shareholders, and news. On the other hand, graph neural networks have been applied to a broad range of tasks due to their superior performance in capturing complex relations among entities and representation learning. This paper investigates the effectiveness of using graph neural networks for stock price movement prediction. Inspired by a recent study, we capture the complex group-level information (co-movement of similar companies) via hypergraphs. Unlike other hypergraph studies, we also use a graph model to learn pairwise relations. Moreover, we are the first to demonstrate that this simple graph model should be applied before using RNNs, rather than later, as prior research suggested. In this paper, the long-term dependencies of similar companies can be learnt by the next RNNs, which augments their predictability. We also apply adversarial training to capture the stochastic nature of the financial market and enhance the generalisation of the proposed model. Hence, we contribute with a novel ensemble learning framework to predict stock price movement, named MONEY. It is comprised of (a) a Graph Convolution Network (GCN), representing pairwise industry and price information and (b) a hypergraph convolution network for group-oriented information transmission via hyperedges with adversarial training by adding perturbations on inputs before the last prediction layer. Real-world data experiments demonstrate that MONEY significantly outperforms, on average, the state-of-the-art methods and performs particularly well in the bear market.

1. Introduction

Stock prediction has been a crucial research topic for a long time, and investors are always interested in having a higher predictive accuracy model to gain profit. Whilst it is notoriously difficult to predict stocks when extensive uncertainty factors, such as policies or social conditions, like pandemics, can influence the financial market (Lee and Soo, 2017), evidence shows that certain factors, like industry information, can forecast the entire stock market by up to two months (Hong et al., 2007). Moreover, stocks in a similar industry will behave differently from the ones outside that industry (Feng et al., 2019). These are thus relevant relational factors necessary to integrate comprehensively. Graph neural networks (GNNs) are a powerful technology leveraging the advantage of network structure (Matsunaga et al., 2019; Sun et al., 2021a), thus promisingly suitable for such a task. Indeed, some initial solutions have been proposed (Cui et al., 2021), but are limited. Existing GNNs usually model pairwise relations of stocks and other information as simple graphs to predict if the price will rise or fall on the next trading day. However, stock prices are more likely to co-move for similar companies and, for such complex behaviour (Aghabozorgi and Teh, 2014), more advanced models are required to capture it. Therefore, we propose to build hypergraphs,[1] which allow modelling group-level relations among stocks from industry and fund-holding (mutual fund as shareholders) aspects (Cui et al., 2021).

Despite the success of GNNs on simple graphs, the study of deep learning on hypergraphs is still at an early stage, and most existing GNNs cannot exploit the high-order structure encoded by hyperedges (Bai et al., 2021a). Few studies explored the area: Sawhney

* Corresponding author.
E-mail addresses: zhongtian.sun@durham.ac.uk (Z. Sun), anoushka.harit@durham.ac.uk (A. Harit), alexandra.i.cristea@durham.ac.uk (A.I. Cristea), jingyun.wang@durham.ac.uk (J. Wang), pl219@cam.ac.uk (P. Lio).

[1] The hypergraph is a special graph, where hyperedges can connect multiple vertices at the same time (Zhang et al., 2017), and the effectiveness of convolution on hypergraphs has been demonstrated (Bai et al., 2021a).

https://doi.org/10.1016/j.aiopen.2023.10.002
Received 21 June 2023; Received in revised form 14 September 2023; Accepted 18 October 2023
Available online 20 October 2023

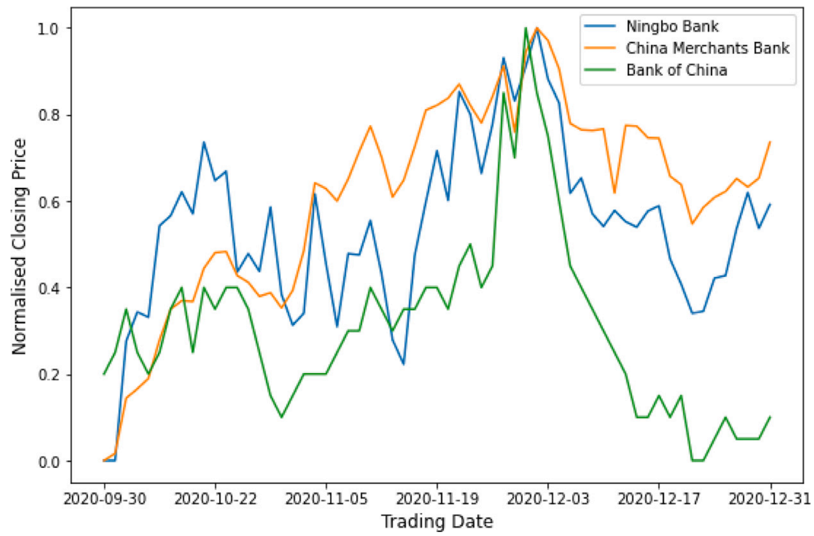

Fig. 1. Price movement of three stocks.

et al. (2020) proposed a gated temporal convolution over hypergraphs to capture stock trends and Cui et al. (2021) applied hypergraph attention networks to predict stock price movement and validated the effectiveness of using hypergraphs to capture similar patterns of stocks in the same group. However, they mainly focused on the group-level analysis and neglected the pairwise correlations between two similar companies, which indeed exist in financial markets (Qie, 2011).

For instance, Aghabozorgi and Teh (2014) suggests that stocks from the same industry may exhibit similar volatility patterns. To illustrate this point, we present Fig. 1. This figure plots stocks from Ningbo Bank (NB), China Merchants Bank (CMB), and Bank of China (BOC), using data obtained from.[2] This visual representation validates their argument. The correlation coefficient between two stocks can be calculated as:

$$R_{xy} = \frac{\sum_{k=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{k=1}^{n}(x_i - \bar{x})^2 \sum_{k=1}^{n}(y_i - \bar{y})^2}} \quad (1)$$

The correlation coefficient R_{xy} between NB and CMB is 0.72, while the correlation coefficient between NB and BOC is 0.64. The strong correlation between these two pairs can be partially attributed to them being held by the same mutual fund between 30/09/2020 and 31/12/2020. Thus, we expect other stocks in the same industry and held by the same fund to also display pairwise relations.

Unlike prior works (Cui et al., 2021; Kim et al., 2019; Sawhney et al., 2020; Ye et al., 2021), we consider not only price information but also industry information via a graph neural network (GCN) to augment the pairwise behaviour of similar companies, before using recurrent neural networks (RNNs). Different from other studies using RNNs and GCN with historical price features (Peng, 2020), we apply GCN first so that the long-term dependencies of pairwise companies can be captured by RNNs later and avoid assuming price movement to be linear or stationary, as Aghabozorgi and Teh (2014) suggested. We do not process fund-holding information via the GCN at this stage, as it is included in the later hypergraph convolution. Moreover, compared to the industry information, fund-holding information is less influential for single stock due to the diversification investment strategy, which requires mutual funds to hold widely spread portfolios across different types of securities (Berk and Van Binsbergen, 2015) but can be a good signal of market trend. To sum up, our method, MONEY, can consider both pairwise and group-level relations between companies in the same industry and hold by the same fund, capturing dynamic historical price information.

Moreover, as the financial market is volatile and the price movement is stochastic, standard training may cause overfitting. Therefore, the research questions are:

1. Can stock price movement prediction be improved by capturing complex relational industry information?
2. How to enhance the robustness of the model while not compromising its predictive performance?

Addressing the questions, we propose to use hypergraph convolution with adversarial training by considering perturbated features and including an adversarial loss in the total loss calculation. Additionally, we use the voting ensemble learning method to classify the predicted classes yielded from standard training and adversarial training, together with the highest number of votes. The contributions of the paper are summarised as follows:

- This is a novel ensemble learning framework that firstly applies both hypergraph convolution and simple graph learning to capture complex relations among similar stocks and address the stochasticity of stocks by adversarial training, which can be used as a solid baseline for future research.
- To the best of our knowledge, *this is the first attempt* to demonstrate the effectiveness of integrating auxiliary information (e.g. industry) via GCN before using RNNs so that the long-term dependencies of pairwise relations of similar companies can be learnt by RNNs later, unlike prior approaches, which deploy RNNs first.
- Experimental results on a real-world dataset prove that our proposed model, MONEY, significantly outperforms the state-of-the-art on stock price movement prediction for most indicators (accuracy, precision, recall and F1 score) and has more stable performance.

2. Related work

2.1. GNN for price prediction

Researchers have attempted to apply GNNs to learn stock representations by modelling stock relations as graphs. For instance, Chen et al. (2018) combined a Long Short-Term Memory (LSTM) with GCN to learn shareholding information. Later, Ye et al. (2021) applied a multi-graph convolution network to predict stock movement, treating the

[2] https://uqer.datayes.com/.

embedding of shareholding, industry and news relation graphs equally in the prediction.

Recently, researchers also proposed to construct different types of graphs to model the relationships between stock prices and other relevant information. Xiong et al. (2021) built a heterogeneous graph for stock prediction by aggregating event level and contextual information. Cui et al. (2021) deployed a hypergraph convolutional network to represent the impact of industry and fund-holding on stock movements, showcasing the ability of hypergraphs to capture complex group-level relationships among stocks. It is noteworthy to mention that the prowess of hypergraph neural networks is not limited to the finance sector. Recent literature, like Guo et al. (2021), Sun et al. (2021b), Wang et al. (2022) and Li et al. (2022a), indicated their success in recommendation and learning analytics domains. For readers interested in a broader view of hypergraph modelling application across diverse fields, these studies provide valuable insights.

Despite the success of the aforementioned financial prediction methods, most of them first fed only historical price features of stocks into RNN models to obtain new price embeddings. Then, the updated embedding could be processed with the other factors, such as similar corporations (Chen et al., 2018), news (Xiong et al., 2021), industry (Hou et al., 2021) or shareholding information (Kim et al., 2019; Ye et al., 2021), by using GCN or other graph neural networks. However, this post-processing came too late. In fact, these methods were limited by assuming other historical information is linear or stationary — by using RNN first, and only capturing pairwise relations among stocks. Concomitantly, existing hypergraph models overemphasised the group-level relations and neglected instead such pairwise information. Moreover, none of these studies dealt appropriately with overfitting (here, due to continuous market fluctuations).

2.2. Adversarial training

Arguably the most effective method to enhance the generalisation of models, by 'defending' against perturbed examples, it received considerable recent attention (Bai et al., 2021b). The main purpose of adversarial training is to augment clean data with adversarial examples so that models can still deliver consistent results when facing adversarial attacks. For stock movement prediction, their stochastic and dynamic characteristics require dealing with overfitting. Addressing the problem, Feng et al. (2018) applied adversarial training to financial time-series analytics and validated the effectiveness of simulating the stochasticity of stock features. Recently, Zhang et al. (2021) proposed a sentiment-guided generative adversarial network to explore the stock prediction problem. Li et al. (2022b) then combined adversarial training with transfer learning and obtained competitive results.

2.3. Ensemble learning for price prediction

A machine-learning technique for improving classification or regression performance by considering multiple algorithms (Dong et al., 2020), it has started to be applied to price forecasting in conjunction with deep learning, where it outperformed the single models on financial time series (Jiang et al., 2020). For instance, Zhao et al. (2017) proposed a denoising autoencoders (SDAE) method with bootstrap aggregation (bagging) to model complex relationships of oil price with its factors. Li and Pan (2021) utilised a blending ensemble learning method consisting of two RNNs to predict the S&P 500 Index. Jiang et al. (2020) incorporated four types of tree-based ensemble algorithms: random forest, extremely randomised trees, XGBoost and LightGBM, with four types of RNNs: vanilla RNN, Bidirectional RNN, LSTM and gated recurrent unit (GRU) (Chung et al., 2014), into a stacking framework for stock index prediction.

3. Proposed MONEY framework

3.1. Problem formulation

This research aims to predict the movement direction of the stock for the following trading day. Given different lengths (5, 10, 20 trading days Cui et al., 2021) of past daily transaction X_s and industry features I_s, as well as fund-holding information F_s, we consider three movement directions of stock prices compared with the price P on the previous trading day $t-1$: *rise, steady, fall.* In line with previous work (Cui et al., 2021), rise (1) means the closing price of a stock on the next trading day P_t is over 0.55% higher than the closing price before P_{t-1}. If P_t is more than 0.50% lower than P_{t-1}, then the price movement is considered as fall (−1); otherwise, steady (0). These settings consider the transaction costs such as tax to find profitable trading opportunities as in practice (Harris, 2013; Lee, 2015) and also balance different types of samples. The cross-entropy loss function of stock price movement prediction can be defined as:

$$loss_{hinge} = -\sum_{c=0}^{C} y_{o,c} log(p_{o,c}) \tag{2}$$

where C is the number of classes, which, in our paper, is 3 (rise, steady, fall); y is the binary indicator, showing if the prediction of observation o is correctly classified, and p is the predicted probability of observation o as c class. If we consider adversarial training, then the total loss becomes:

$$loss_{total} = -\sum_{c=0}^{C} y_{o,c} log(p_{o,c}) - \beta \sum_{c=0}^{C} y_{o,c} log(p_{a,c}) \tag{3}$$

where the addition term is the adversarial loss and $p_{a,c}$ is the predicted probability of a perturbed observation o classified correctly as c class; β is a hyperparameter balancing the two types of loss, so that the model is encouraged to classify both original objects and perturbed samples, correctly. The overall structure of our proposed framework is shown in Fig. 2. It consists of two models;

model A: industry information first passes stock price (a) through a GCN (b); then a GRU and temporal attention layer (c) are applied to capture the time-series characteristics of stocks followed by a hypergraph construct (d); then separate convolution networks (e) for industry and fund-holding information; and, finally, a linear classification neural network (f);

model B: As we have considered the pairwise industry information via GCN in model A, model B starts with a GRU, to capture price-time dependency, following existing research, and without the GCN (b), to process industry information at the beginning but includes instead an extra input of adversarial samples to the classification layer (g). The predictions yielded by Models A and B, which comprise modules *a,b,c,d,e,f* and *a,c,d,e,f,g*, respectively, will be integrated into a new embedding, denoted as $\hat{y}_s$. $\hat{y}_{adv}$ is the prediction yielded from model B with adversarial examples. $\hat{y}_f$ is the final prediction, which considers both $\hat{y}_s$ and $\hat{y}_{adv}$, via a voting method.

The differences between our MONEY model and state-of-the-art HGTAN (Cui et al., 2021) are threefold: (1) we use GCN first to augment the pairwise industry relation before using RNN to capture the long-term dependency of price and demonstrating the effectiveness of using GCN first; (2) we consider the stochastic characteristics of the stock market and apply adversarial training to enhance the generalisation ability of model; (3) we do not use triple attention mechanisms among hyperedges and hypergraphs, but still outperform the HGTAN as shown in Tables 2–4. Details are explained in the following subsections, and we will explain how to construct hypergraphs first.

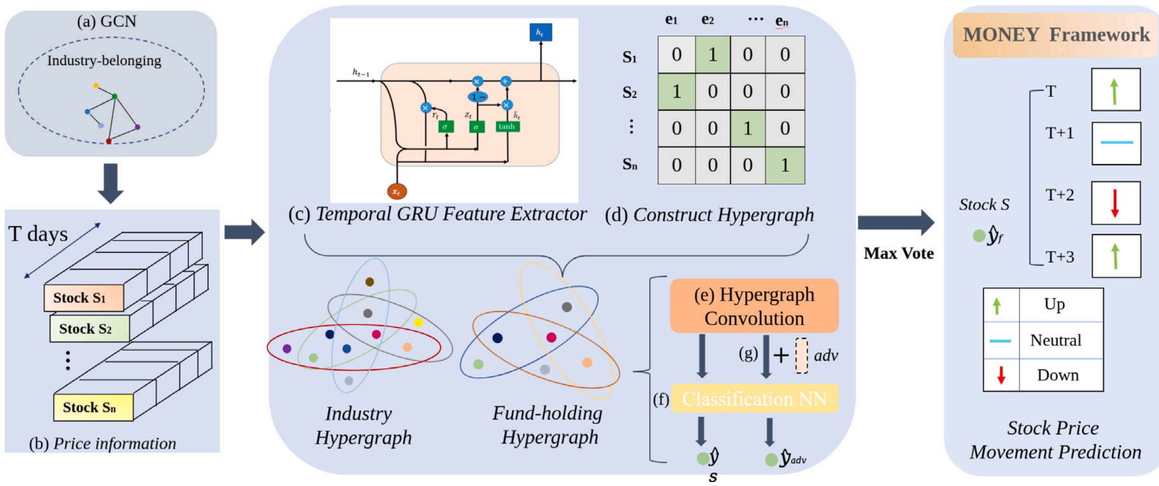

Fig. 2. Illustration of MONEY framework.

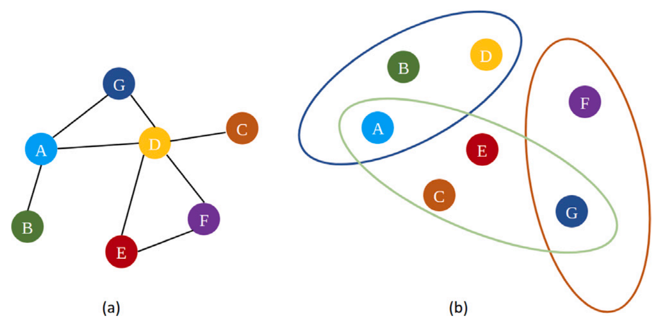

Fig. 3. Simple graph and hypergraph.

3.2. Hypergraph construction

Definition 1 (*Hypergraph*). An undirected hypergraph can be denoted as $G = (V, E)$ where V is the set of N nodes and E is the set of M hyperedges. In a hypergraph, any of the edges e can join any number of vertices v, to describe more complex relationships between entities, shown as (b) in Fig. 3; while in simple graphs, one edge can only link two vertices, representing a pairwise relation between nodes as (a) in Fig. 3. A hypergraph is often denoted as an incidence matrix $H \in R^{N\times M}$:

$$H(i,j) = \begin{cases} 1 \text{ if node } i \text{ is included in hyperedge } j \\ 0 \text{ otherwise} \end{cases}$$

Same as Cui et al. (2021), we construct two hypergraphs to model the industry and fund-holding relations among stocks, separately. In the industry hypergraph, all the companies in the same industry are connected by one hyperedge and each node $v \in V$ has features with 10 dimensions; this is similar to the fund-holding hypergraph. We apply graph convolutions to the two hypergraphs to update embeddings of each stock (node) to be later used in the price movement classification.

3.3. GCN for industry information

We feed the daily stock prices $x_{s,t}$ of stock s as stock features and the related industry information I_s into a GCN $f(\theta)$ to augment the pairwise patterns of similar companies and obtain an updated embedding x_i^t for every trading day t:

$$x_i^t = f(\theta) \cdot x_s \tag{4}$$

We build undirected edges between stocks that are in the same industry and use two layers of convolution, as shown in Fig. 4. We did not use GCN to process fund-holding information, which is considered at a later stage. Moreover, compared to the industry information, fund-holding information is less influential for single stock due to the diversification investment strategy, which requires mutual funds to hold widely spread portfolios across different types of securities (Berk and Van Binsbergen, 2015), but can be a good signal of market trend. After the GCN layer, the new embedding x_i^t will be passed to GRU to capture the time-series information.

3.4. Gated recurrent unit for long term dependency

Following Cui et al. (2021), we use a GRU to learn the embedding of stock features, due to its ability to capture long-term dependency for sequential data. Note that the input stock feature has been augmented by considering the influence of industry. The primary purpose of GRU is to deploy a gated process, to manage and update the flow of information between cells of neural network units, which can be formulated as follows:

$$\begin{aligned} z &= \sigma(W_z \cdot x_i^t + U_z \cdot h_{t-1} + b_z), \\ r &= \sigma(W_r \cdot x_i^t + U_r \cdot h_{t-1} + b_r), \\ \hat{h} &= tanh(W_h \cdot x_i^t + r * U_h \cdot h_{t-1} + b_z), \\ h_t &= z * h_{t-1} + (1-z) * \hat{h}. \end{aligned} \tag{5}$$

where x_i^t is obtained from Eq. (4) and W_z, W_r, W_h are the weight matrices, which need to be trained. h_{t-1} is the hidden state, which includes historical information from the previous trading day and U_z, U_r, U_h are also parameters to train. z is the reset gate to decide how much historical information should be disregarded and r is the update gate to decide how much past information to pass on for the future. The update and reset gates could alleviate the vanishing gradient problem during the backpropagation of time-series data. $\hat{h}$ denotes the current information to be considered for the current hidden state h_t. The new embedding h_t will feed into the temporal attention layer.

3.5. Temporal attention

The attention mechanism has been increasingly applied to predict stock price trends (Yu and Wu, 2019; Li et al., 2020; Cui et al., 2021). Attention can capture the different influences of the hidden representations on the overall learned embedding at different time steps (Feng et al., 2018). Due to the recency bias hypothesis (Hao et al., 2016), stating that the most recent price has a stronger correlation with

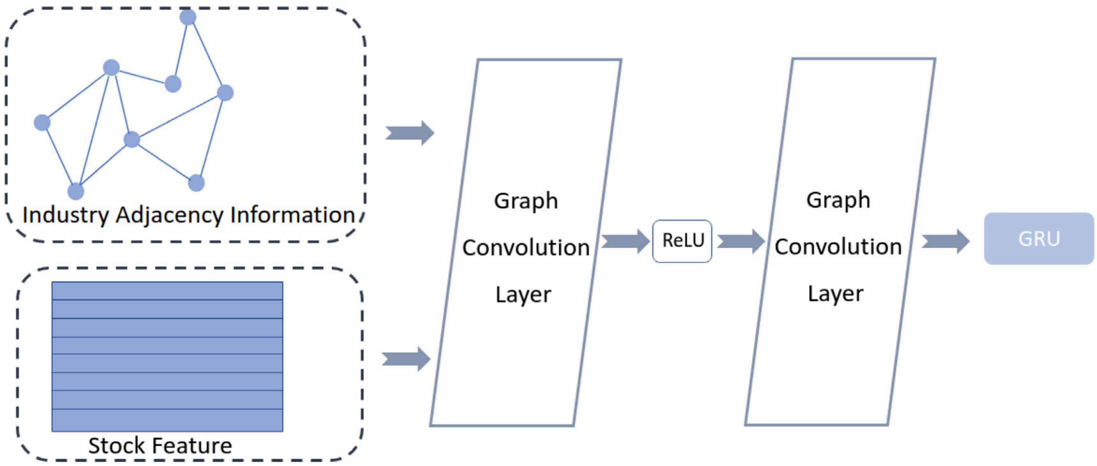

Fig. 4. Overview of GCN.

the future movement, it is reasonable to use a temporal attention layer to yield the aggregated temporal dynamics representations of stocks s_d:

$$s_d = \sum_{t=1}^{T} \alpha_s^t h_s^t$$
$$\alpha_s^t = \frac{\exp(\hat{\alpha}_s^t)}{\sum_{t=1}^{T} \exp(\hat{\alpha}_s^t)} \tag{6}$$
$$\hat{\alpha}_s^t = \tanh(W_\alpha h_s^t + b_\alpha) U_\alpha^T$$

where α_s^t is the weight of the hidden state at time t of stock s. $W_\alpha h_s^t$, b_α and U_α^T are parameters needed to be learned. The aggregated temporal representation s_d will then feed into the hypergraph convolution layer, explained in Section 3.6.

3.6. Hypergraph convolution with adversarial training

To model the group-level relationships among stocks, we apply hypergraph convolution to both the industry graph G_i and the fund-holding hypergraph G_f. For instance, in the industry graph G_i, each stock connects via the same edge e_i and will aggregate messages passed from its neighbours. One step of convolution on the hypergraph can be formulated as:

$$h_s^{l+1} = \sigma(\sum_{j=1}^{M} \sum_{i=1}^{N} \cdot h^l F) \tag{7}$$

where h^{l+1} is the feature matrix of stock s at the next layer and σ is a nonlinear activation LeakyReLU. $\sum_{j=1}^{M} \sum_{i=1}^{N}$ refers to applying the convolution to the set of M hyperedges, starting from hyperedge j, where all the contained nodes of different hyperedges M will update based on their local neighbours, starting from node i. F is a weight matrix to be learnt between the l and $l+1$ layer. After the convolution, we have updated the industry embedding h_i and the fund-holding embedding h_f, which will be simply concatenated, to obtain a new embedding h_s^m for the prediction layer and further adversarial training in Eq. (8).

Adversarial Training is to augment each minibatch of clean data with adversarial examples (AEs), which are generated by adding adversarial perturbation (Shafahi et al., 2019). Feng et al. (2018), validated the effectiveness of adversarial training to address the stochastic property of stock price. Therefore, we add adversarial perturbation to stock embedding at a higher level after the temporal attention layer, but before the prediction layer.

$$h_s^{adv} = h_s^m + p_s^{adv}, \; p_s^{adv} = argmax \; loss_{adv}(y, y^{adv}) \tag{8}$$

$$p_s^{adv} = \epsilon \frac{G^s}{\|G^s\|_2}, G^s = \frac{\partial l(y, y^{adv})}{\partial h_s} \tag{9}$$

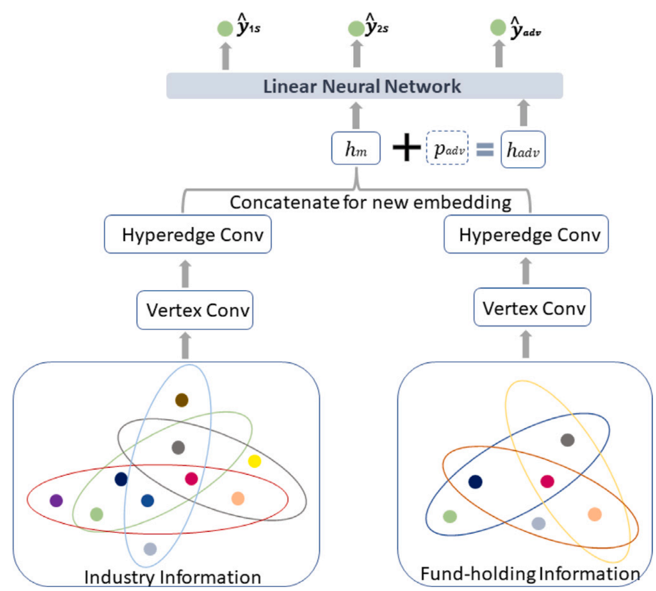

Fig. 5. Hypergraph convolution with adversarial training in MONEY framework.

where h_{adv}^s is the latent representation of an adversarial example and h^s is the concatenated embedding obtained from the hypergraph convolution above. p_s^{adv} is the optimal max-norm constrained perturbation, updated by the fast gradient approximation method (Goodfellow et al., 2014) in Eq. (9), where G is the gradient of the adversarial loss with L2 norm constraint of stock s and ϵ is to change the scale of perturbation. Feng et al. (2018) empirically demonstrated that adversarial training enforced the decision boundary to be close to original objects so that the model is able to capture stochasticity and classify the perturbed samples correctly.

Fig. 5 describes the process of using a hypergraph convolution network for the industry and fund-holding hypergraphs and the h_s and its adversarial example h_{adv}^s will be fed into a linear prediction layer for classification. $\hat{y_{1s}}$ and $\hat{y_{2s}}$ is the prediction generated by model A and B separately. These two predictions will be concatenated and $\hat{y_{adv}}$ is the adversarial prediction generated from model B as explained in Section 3.1. The ensemble learning of these three predictions is further explained in Section 3.7.

Table 1
Dataset summary.

Data	Number
Node (Stock)	758
Attributes	6
Industry-belonging relationships	104
Fund-holding relationships	61
Percentage of steady	24.7%
Percentage of rising	38%
Percentage of dropping	37.3%

3.7. Ensemble learning

Various ensemble strategies can be utilised, including bagging, boosting, stacking, blending, averaging, weighted average, AdaBoost and so on Dietterich (2000). In this paper, we propose two base models, as shown in Fig. 2: (A) The stock price information will be processed with industry information via GCN and then the updated embeddings of the stock will be fed into a GRU with a temporal attention layer, a hypergraph convolution network and a linear neural network, to obtain a prediction. (B) Stock features will directly pass through a GRU, temporal attention layer and hypergraph convolution network, in order to obtain another prediction, which will be concatenated with the one generated by model A as $\hat{y}_s$. Additionally, adversarial examples after the hypergraph convolution will also be fed into the linear prediction layer and yield a $\hat{y}_{adv}$. Then we use the max voting here to classify the sample as the class $\hat{y}_s$ and $\hat{y}_{adv}$ with the highest votes.

4. Experiments

4.1. Experimental setup

Dataset In this paper, we use the dataset from Cui et al. (2021), containing 758 frequently traded stocks collected from the A-share market in China between 01/04/2013 and 12/31/2019, to be fairly compared with Cui et al. (2021). In the future, we will collect data from other markets like US or Europe to further validate our framework. Each stock in the dataset has six attributes: the opening price, high price, low price, close price, trading amount, and value. All the input features of stocks have been min–max normalised. If some stocks lack trading data during a temporary suspension period, the price attributes of the most recent day before the suspension will be used. We split the dataset into three parts: 60% for training, 20% for validation, and 20% for testing in line with Cui et al. (2021). The validation set is utilised to optimise the hyperparameters of our model. The industry information groups stocks into 104 industry categories, defined by the Shenwan Industry Classification Standard,[3] and 'fund-holding' information is learnt from quarterly portfolio reports of the 61 mutual funds established before 2013 in the A-share market, similar as Cui et al. (2021). Details of the dataset are shown in Table 1.

Baselines We compare against prior works, from one trading method, mean reversion (MR), one conventional LSTM model, one dual attention LSTM and five recently proposed graph neural network algorithms as baselines for stock movement prediction.

- MR: the model applied a mean reversion indicator to predict the local trend reversion by assuming extreme changes in the price will revert back to its previous state (Serban, 2010).
- LSTM: the model applied LSTM to predict future movements of stock prices (Nelson et al., 2017).
- DARNN: the model proposed a dual stage attention with recurrent neural network to selectively harness relevant features for efficient prediction (Qin et al., 2017).
- GCN+LSTM: the model applied GCN to learn relationships among stocks via feeding embeddings of stocks to an LSTM network (Chen et al., 2018).
- HATS: the model used a hierarchical attention network to adaptively learn the importance of different relation types for stock prediction (Kim et al., 2019).
- TGC: applied a novel temporal graph convolution to model the temporal evolution jointly, and relational embeddings of stocks for prediction (Feng et al., 2019).
- STHGCN: proposed a novel Spatio-Temporal Hypergraph Convolution Network to learn stock price evolution over stock industry relations (Sawhney et al., 2020).
- HGTAN: proposed a novel hypergraph tri-attention network to predict the stock price movement and considered both industry-belonging and fund-holding information (Cui et al., 2021). This represents the current state of the art

Parameters We implement the proposed ensemble learning framework with PyTorch 1.10.2 and CUDA 10.2. According to Ye et al. (2021), the length of historical information impacts model performance; here, we thus test the model with different lengths of trading information: the past 5, 10 and 20 trading days. Hyperparameters are borrowed from Cui et al. (2021), optimised with the same validation set, for a fair comparison: the feature dimension of a stock is set as 16 and batch size as 32; the hidden units' size of GRU is 32 and the dimensions of d_k and d_v in the temporal attention mechanism are both 8. The maximum number of epochs is 600 and the dropout rate is 0.5. β in the loss function 3 is set as $1e{-}2$, as in Feng et al. (2018). We use the same settings for the baseline models as their public implementations.

Evaluation We use accuracy, precision, recall, and F1 score to evaluate the classification performance of the proposed model, which are calculated as:

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\ Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \\ F1-score &= 2 * \frac{Precision * Recall}{Precision + Recall} \end{aligned} \tag{10}$$

where TP and TN are the correctly predicted positive classes and negative classes, respectively. FP and FN denote the falsely predicted positive classes and negative classes, respectively. As we implement multi-label classification (3 movement directions), the metric is calculated in macro-setting using the scikit-learn library. For instance, precision is calculated as:

$$Precision = \left(\frac{TP_r}{TP_r+FP_r} + \frac{TP_s}{TP_s+FP_s} + \frac{TP_f}{TP_f+FP_f}\right) * \frac{1}{3} \tag{11}$$

where r, s and f refers to rise, steady and fall, respectively.

4.2. Effectiveness results on the real world dataset

Tables 2, 3, and 4 show the future stock price movement prediction performance of different models on the real-world benchmark datasets, with the past 5, 10 and 20 trading days as lengths of the look-back window. MONEY mostly significantly outperforms all baselines in precision, recall and F1 score metrics, where it exceeds the second best by an average of 1.14%, 3.63% and 2.45% for three trading lengths. F1 value considers both precision and recall measurements, which enhances models' sensitivity and generalisation ability for performance evaluation (Sokolova et al., 2006). In stock investment, we prefer to seize every chance to earn profits (high recall: proportion of actual positives correctly identified) and expect to earn more each time (high precision: proportion of positive identification being correct). Therefore we believe that the F1 score is the most important metric of these four metrics. In such a case, our proposed MONEY model significantly

[3] http://www.swsindex.com/idx0530.aspx.

Table 2
Classification performance of baselines and the proposed model with 5 trading days as record.

Method	Accuracy	Precision	Recall	F1
MR	35.59%	39.37%	33.77%	36.36%
LSTM	34.92%	35.34%	33.91%	34.27%
DARNN	37.68%	37.81%	35.17%	36.43%
GCN+LSTM	37.24%	37.23%	33.54%	35.22%
HATS	38.74%	36.92%	34.29%	35.52%
TGC	37.43%	38.28%	34.05%	36.01%
STHGCN	38.53%	37.35%	34.65%	35.89%
HGTAN	**39.51%**	38.90%	36.96%	37.89%
MONEY	38.67%	**42.06%**	**41.80%**	**41.93%**

Table 3
Classification performance of baselines and the proposed model with 10 trading days as record.

Method	Accuracy	Precision	Recall	F1
MR	34.73%	29.34%	31.79%	30.52%
LSTM	35.09%	38.09%	34.37%	35.90%
DARNN	38.89%	38.59%	35.22%	36.82%
GCN+LSTM	37.44%	39.07%	34.49%	36.62%
HATS	38.05%	39.23%	34.52%	36.67%
TGC	38.42%	39.35%	35.72%	37.44%
STHGCN	38.81%	36.57%	35.11%	35.75%
HGTAN	39.83%	**41.72%**	37.32%	39.37%
MONEY	**41.04%**	39.83%	**41.79%**	**40.79%**

Table 4
Classification performance of baselines and the proposed model with 20 trading days as record.

Method	Accuracy	Precision	Recall	F1
MR	35.32%	38.03%	33.60%	35.68%
LSTM	35.03%	36.43%	34.23%	35.20%
DARNN	38.41%	37.99%	39.24%	38.60%
GCN+LSTM	37.30%	39.28%	34.16%	36.54%
HATS	38.85%	38.70%	35.06%	36.78%
TGC	37.81%	36.96%	34.49%	35.67%
STHGCN	38.45%	37.22%	32.82%	34.87%
HGTAN	**40.02%**	41.77%	39.03%	40.32%
MONEY	39.90%	**43.92%**	**40.61%**	**42.20%**

improves the F1 score by 7.34% in total using the Wilcoxon signed-rank test and is more advanced, compared with the state-of-the-art HGTAN, particularly without using complex triple attention among hyperedges and hypergraphs. This demonstrates the effectiveness of capturing group-level and pairwise information via hypergraph convolution and GCN on stock price movement. For the accuracy metric, our model obtains the highest value for the ten trading days window and performs relatively competitively with knowledge of the past 20 and 5 trading days. The improvement of using adversarial training is analysed in Section 4.4.

4.3. Investment simulation and profitability

We compare our MONEY model with competitors from 04/23/2019 to 05/09/2019, ten trading days in total, following the same forecasting window size as in Cui et al. (2021), and depict it in Table 5. It gains the highest accuracy in five days and the crucial advantage of our proposed framework, MONEY, is its significantly stable performance in prediction.

It achieves over 70% accuracy (all accuracies higher than 70% underlined) in seven days, while the state-of-the-art, HGTAN, only has such performance in three days, and the remaining models mostly have accuracy higher than 70% in one of the ten trading days. We normalised four important market indexes: Hushen 300, Shenzheng Zongzhi, Shangzheng 50 and Zhongxiao 300 to visualise in Fig. 6, and they all plummeted into a so-called 'bear market' during the ten trading days. More specifically, the most representative index Hushen 300, consisting of the 300 largest market capitalisation and most liquid stocks, fell from 4019 on 04/23/2019 to 3599.7 on 05/09/2019 rapidly, and the investors all panicked. Fig. 7 shows falling stocks are the majority for seven days (04/23, 04/25, 04/26, 04/29, 05/06, 05/08, 05/09), and our method constantly obtains over 70% accuracy, except for one day, which illustrates the effectiveness of our method in comparison to other models, and its capacity to avoid loss.

Similar to Cui et al. (2021), we compare the profitability of different models from 08/31/2018 to 10/31/2019, shown in Table 6. CR is the cumulative return rate, frequently used in profitability analysis (Jacobs and Levy, 1988). SR denotes the Sharpe ratio, which measures investment return with the associated risk (Sharpe, 1966; Magdon-Ismail and Atiya, 2004):

$$Sharpe\ ratio = \frac{E(R_s) - R_f}{\sigma_s} \tag{12}$$

where E is the expected value, R_s is the return rate of stock s and R_f is the risk-free rate (1.5% as one-year deposit interest rate in 2019), and σ_s denotes the standard deviation of the stock excess return. Our method reaches the highest cumulative return rate, 22.33%, with competitive performance at risk management with the highest Sharpe ratio of 0.775, which indicates it returns more profit compared with other models under the same risk situation.

4.4. Ablation study

We compare the classification performance of our MONEY framework, its separate components and the second best model HGTAN in Table 7 on the dataset with knowledge of the 10 past trading days. We thus demonstrate the improvement brought by each module, first separately and then as a whole via our MONEY framework:

- GCN+GRU+TA+HGCN (Module A): GCN is firstly applied to process historical price features with industry information augmenting the *pairwise relations*, followed by a GRU network with a temporal attention layer (TA) and the generated embedding is then fed into a hypergraph convolution network (HGCN) to learn the industry and fund-holding *group-level* relations of stocks.
- GRU+TA+HGCN (Module B): a variant without pairwise relations using GCN; otherwise it is the same as variant A.
- GRU+TA+HGCN+Adv (Module C): this variant does not consider pairwise relations using GCN but extends HGCN with adversarial training to enhance the generalisation ability to deal with the *stochasticity* of stock features, as Feng et al. (2018) suggested, compared with variant A.
- GRU+TA (Module D): this is the base component, which solely considers *historical* price information.
- GRU+TA+GCN+HGCN (Module E): this is to compare with Module A and demonstrates the effectiveness of applying GCN to capture pairwise relations before RNN models.
- GCN + GRU + TA + Adv (Module F): this variant contrasts with the MONEY method to validate the value of HGCN.

Overall, each module improves the performance compared to the baseline (Module D) and our model outperforms all variants in terms of recall and F1 score by using the synergistic power of all the modules together. MONEY also significantly outperforms the state-of-the-art competitor, HGTAN, in terms of accuracy, recall and F1 score and delivers significantly stable performance in a bear market, as addressed in Section 4.3. To be specific, when comparing modules A and B, GCN improves all four metrics in total 3.14%, particularly 1.03% in the most critical metric, F1 value, as discussed before. This validates our point that pairwise relations should not be neglected, even if we have already considered group-level information. Compared with Module E, Module A significantly improves three measurements in total 5.11% (1.49%

Table 5
Accuracy of stock movement prediction between 04/23/2019 and 05/09/2019.

Date	04/23	04/24	04/25	04/26	04/29	04/30	05/06	05/07	05/08	05/09
MR	16.23%	**52.77%**	9.76%	17.68%	19.13%	64.78%	7.12%	73.48%	25.33%	17.81%
LSTM	24.80%	46.44%	14.38%	22.43%	22.03%	66.49%	8.31%	69.92%	30.87%	20.32%
DARNN	35.36%	43.27%	20.19%	30.74%	24.14%	61.87%	14.12%	57.26%	30.61%	25.86%
GCN+LSTM	22.43%	48.02%	15.44%	21.50%	23.22%	63.72%	11.35%	71.11%	28.36%	18.73%
HATS	37.86%	38.26%	31.40%	38.13%	43.67%	50.53%	48.55%	50.26%	44.59%	33.25%
TGC	69.39%	38.26%	31.40%	38.13%	43.67%	50.53%	48.55%	50.26%	44.59%	33.25%
STHGCN	21.24%	27.44%	15.04%	23.88%	14.78%	66.62%	12.67%	**75.20%**	26.65%	19.26%
HGTAN	57.78%	22.03%	**88.26%**	60.16%	**72.43%**	16.62%	**76.52%**	31.27%	44.99%	56.07%
MONEY	**71.37%**	28.76%	70.84%	**72.03%**	27.18%	**71.90%**	71.77%	27.84%	**70.84%**	**71.77%**

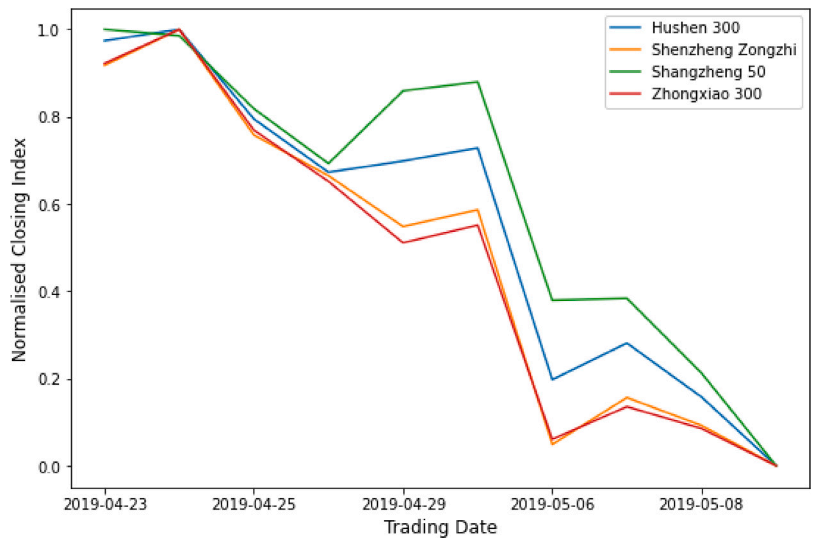

Fig. 6. China a share market index trend.

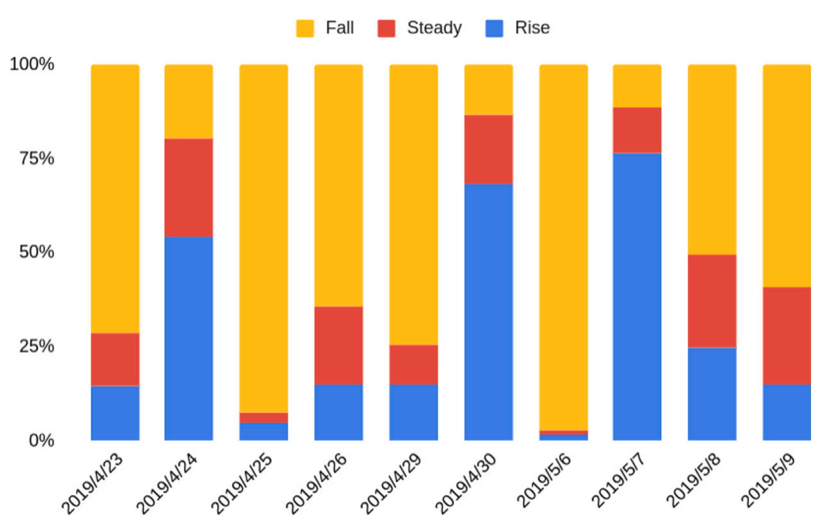

Fig. 7. Different movement of stocks.

in F1 value) and performs competitively in terms of precision. The result illustrates that price movement can be learnt more effectively by using GCN before RNNs. GCN will first capture similar volatility patterns of stocks in the same industry, enhancing the prediction ability of the following RNN models. Moreover, adversarial training can significantly enhance the model's robustness by considering the stock price's stochasticity, as the precision, recall and F1 score of module C are all improved, compared with module B, at a reasonable cost of downgrading the accuracy. Adding a hypergraph convolution to module D, the fundamental function would improve by 2.41% in terms of accuracy, shown in module B. The precision and F1 score can also increase by 2.27% and 1.03%. In addition, the comparison between MONEY and module F also demonstrates the need to use hypergraphs to deal with the group-level information for stock price movement prediction.

5. Conclusion and future directions

In this paper, we point out that existing work for stock movement prediction suffers from insufficiently capturing both group-wise and pairwise relations of relevant information rather than solely historical price features, leading to a weak generalisation ability due to the

Table 6
Profitability of different models during back test.

Method	CR	SR
MR	4.89%	0.123
LSTM	4.73%	0.147
DARNN	3.23%	0.083
GCN+LSTM	6.51%	0.217
HATS	11.55%	0.697
TGC	8.23%	0.513
STHGCN	6.23%	0.248
HGTAN	19.78%	0.699
MONEY	22.33%	0.775

Table 7
An ablation study of MONEY on dataset with 10 trading days as record.

Method	Accuracy	Precision	Recall	F1
Module A	**41.38**%	42.07%	39.35%	40.66%
Module B	41.28%	41.29%	38.11%	39.64%
Module C	40.01%	**42.87%**	38.56%	40.60%
Module D	38.86%	39.02%	38.20%	38.61%
Module E	40.37%	42.39%	36.42%	39.18%
Module F	39.38%	40.04%	37.34%	38.66%
HGTAN	39.83%	41.72%	37.32%	39.37%
MONEY	41.04%	39.83%	**41.79%**	**40.79%**

stochastic characteristics of stocks. We also demonstrate that pairwise relation learning should be applied before RNN models rather than later, as stocks display similar volatility patterns and using the GCN model first to capture the patterns can enhance the prediction ability of the following RNN models. Addressing these problems, we propose a novel ensemble learning framework, MONEY, to better assist investors in predicting future trends of stocks. To effectively capture *pairwise information* of industry, a graph convolution network is applied before RNN models. To capture the *group-level information* of both industry and fund-holding, a hypergraph convolution network is implemented after the GRU model with a temporal attention layer. Adversarial training is introduced before the final prediction layer in the model (B) to simulate the *stochastic movement* during training. Ensemble learning allows the two models to complement each other, keeping their benefits and enhancing learning about these relations and robustness. All components are jointly trained on real-world stock market datasets. Our model significantly outperforms the state-of-the-art for most of the indicators without using complex triple attention mechanisms among hyperedges and hypergraphs and provides much more stable performance, particularly when facing a bear market.

As aforementioned, we only test our model in one market, and the time period does not include Covid-19 to be comparable with Cui et al. (2021). It is a limitation and in future, we will explore the method in different markets. More advanced deep learning methods, including graph contrastive learning (Sun et al., 2022b) and graph dynamic attention (Brody et al., 2021; Sun et al., 2022a) can also be applied for stock prediction tasks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aghabozorgi, S., Teh, Y.W., 2014. Stock market co-movement assessment using a three-phase clustering method. Expert Syst. Appl. 41 (4), 1301–1314.

Bai, T., Luo, J., Zhao, J., Wen, B., Wang, Q., 2021b. Recent advances in adversarial training for adversarial robustness. arXiv preprint arXiv:2102.01356.

Bai, S., Zhang, F., Torr, P.H., 2021a. Hypergraph convolution and hypergraph attention. Pattern Recognit. 110, 107637.

Berk, J.B., Van Binsbergen, J.H., 2015. Measuring skill in the mutual fund industry. J. Financ. Econ. 118 (1), 1–20.

Brody, S., Alon, U., Yahav, E., 2021. How attentive are graph attention networks? arXiv preprint arXiv:2105.14491.

Chen, Y., Wei, Z., Huang, X., 2018. 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. pp. 1655–1658.

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

Cui, C., Li, X., Du, J., Zhang, C., Nie, X., Wang, M., Yin, Y., 2021. Temporal-relational hypergraph tri-attention networks for stock trend prediction. arXiv preprint arXiv:2107.14033.

Dietterich, T.G., 2000. Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems. Springer, pp. 1–15.

Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q., 2020. A survey on ensemble learning. Front. Comput. Sci. 14 (2), 241–258.

Feng, F., Chen, H., He, X., Ding, J., Sun, M., Chua, T.-S., 2018. Enhancing stock movement prediction with adversarial training. arXiv preprint arXiv:1810.09936.

Feng, F., He, X., Wang, X., Luo, C., Liu, Y., Chua, T.-S., 2019. Temporal relational ranking for stock prediction. ACM Trans. Inf. Syst. (TOIS) 37 (2), 1–30.

Goodfellow, I.J., Shlens, J., Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Guo, L., Yin, H., Chen, T., Zhang, X., Zheng, K., 2021. Hierarchical hyperedge embedding-based representation learning for group recommendation. ACM Trans. Inf. Syst. (TOIS) 40 (1), 1–27.

Hao, Y., Chu, H.-H., Ho, K.-Y., Ko, K.-C., 2016. The 52-week high and momentum in the Taiwan stock market: Anchoring or recency biases? Int. Rev. Econ. Finance 43, 121–138.

Harris, L., 2013. What to do about high-frequency trading. Financ. Anal. J. 69 (2), 6–9.

Hong, H., Torous, W., Valkanov, R., 2007. Do industries lead stock markets? J. Financ. Econ. 83 (2), 367–396.

Hou, X., Wang, K., Zhong, C., Wei, Z., 2021. St-trader: A spatial-temporal deep neural network for modeling stock market movement. IEEE/CAA J. Autom. Sin. 8 (5), 1015–1024.

Jacobs, B.I., Levy, K.N., 1988. Disentangling equity return regularities: New insights and investment opportunities. Financ. Anal. J. 44 (3), 18–43.

Jiang, M., Liu, J., Zhang, L., Liu, C., 2020. An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms. Physica A 541, 122272.

Kim, R., So, C.H., Jeong, M., Lee, S., Kim, J., Kang, J., 2019. Hats: A hierarchical graph attention network for stock movement prediction. arXiv preprint arXiv:1908.07999.

Lee, E.J., 2015. High frequency trading in the Korean index futures market. J. Futures Mark. 35 (1), 31–51.

Lee, C.-Y., Soo, V.-W., 2017. Predict stock price with financial news based on recurrent convolutional neural networks. In: 2017 Conference on Technologies and Applications of Artificial Intelligence (TAAI). IEEE, pp. 160–165.

Li, Y., Dai, H.-N., Zheng, Z., 2022b. Selective transfer learning with adversarial training for stock movement prediction. Connect. Sci. 1–19.

Li, Y., Li, L., Zhao, X., Ma, T., Zou, Y., Chen, M., 2020. An attention-based LSTM model for stock price trend prediction using limit order books. In: Journal of Physics: Conference Series, Vol. 1575. IOP Publishing, 012124.

Li, Y., Pan, Y., 2021. A novel ensemble deep learning model for stock prediction based on stock prices and news. Int. J. Data Sci. Anal. 1–11.

Li, M., Zhang, Y., Li, X., Cai, L., Yin, B., 2022a. Multi-view hypergraph neural networks for student academic performance prediction. Eng. Appl. Artif. Intell. 114, 105174.

Magdon-Ismail, M., Atiya, A.F., 2004. Maximum drawdown. Risk Mag. 17 (10), 99–102.

Matsunaga, D., Suzumura, T., Takahashi, T., 2019. Exploring graph neural networks for stock market predictions with rolling window analysis. arXiv preprint arXiv:1909.10660.

Nelson, D.M., Pereira, A.C., De Oliveira, R.A., 2017. Stock market's price movement prediction with LSTM neural networks. In: 2017 International Joint Conference on Neural Networks (IJCNN). Ieee, pp. 1419–1426.

Peng, S., 2020. Stock Forecasting using Neural Network with Graphs (Ph.D. thesis). University of York.

Qie, W., 2011. Market Volatility's Relationship with Pair-Wise Correlation of Stocks and Portfolio Man-ager's Performance. Stockholm University.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G., 2017. A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint arXiv:1704.02971.

Sawhney, R., Agarwal, S., Wadhwa, A., Shah, R.R., 2020. Spatiotemporal hypergraph convolution network for stock movement forecasting. In: 2020 IEEE International Conference on Data Mining (ICDM). IEEE, pp. 482–491.

Serban, A.F., 2010. Combining mean reversion and momentum trading strategies in foreign exchange markets. J. Bank. Financ. 34 (11), 2720–2727.

Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T., 2019. Adversarial training for free!. Adv. Neural Inf. Process. Syst. 32.

Sharpe, W.F., 1966. Mutual fund performance. J. Bus. 39 (1), 119–138.

Sokolova, M., Japkowicz, N., Szpakowicz, S., 2006. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In: Australasian Joint Conference on Artificial Intelligence. Springer, pp. 1015–1021.

Sun, Z., Harit, A., Cristea, A.I., Yu, J., Al Moubayed, N., Shi, L., 2022a. Is unimodal bias always bad for visual question answering? A medical domain study with dynamic attention. In: 2022 IEEE International Conference on Big Data (Big Data). IEEE, pp. 5352–5360.

Sun, Z., Harit, A., Cristea, A.I., Yu, J., Shi, L., Al Moubayed, N., 2022b. Contrastive learning with heterogeneous graph attention networks on short text classification. In: 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–6.

Sun, Z., Harit, A., Yu, J., Cristea, A.I., Al Moubayed, N., 2021a. A generative Bayesian graph attention network for semi-supervised classification on scarce data. In: 2021 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–7.

Sun, Z., Harit, A., Yu, J., Cristea, A.I., Shi, L., 2021b. A brief survey of deep learning approaches for learning analytics on MOOCs. In: Intelligent Tutoring Systems: 17th International Conference, ITS 2021, Virtual Event, June 7–11, 2021, Proceedings 17. Springer, pp. 28–37.

Wang, X., Ma, W., Guo, L., Jiang, H., Liu, F., Xu, C., 2022. Hgnn: Hyperedge-based graph neural network for mooc course recommendation. Inf. Process. Manage. 59 (3), 102938.

Xiong, K., Ding, X., Du, L., Liu, T., Qin, B., 2021. Heterogeneous graph knowledge enhanced stock market prediction. AI Open 2, 168–174.

Ye, J., Zhao, J., Ye, K., Xu, C., 2021. Multi-graph convolutional network for relationship-driven stock movement prediction. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, pp. 6702–6709.

Yu, M.-H., Wu, J.-L., 2019. Ceam: A novel approach using cycle embeddings with attention mechanism for stock price prediction. In: 2019 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, pp. 1–4.

Zhang, C., Hu, S., Tang, Z.G., Chan, T.H., 2017. Re-revisiting learning on hypergraphs: confidence interval and subgradient method. In: International Conference on Machine Learning. PMLR, pp. 4026–4034.

Zhang, Y., Li, J., Wang, H., Choi, S.C., 2021. Sentiment-guided adversarial learning for stock price prediction. Front. Appl. Math. Stat. 7, 8.

Zhao, Y., Li, J., Yu, L., 2017. A deep learning ensemble approach for crude oil price forecasting. Energy Econ. 66, 9–16.