Applied Soft Computing Journal 91 (2020) 106205

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market

Jiawei Long [a], Zhaopeng Chen [a], Weibing He [b], Taiyu Wu [b], Jiangtao Ren [a,*]

[a] School of Data and Computer Science, Guangdong Province Key Lab of Computational Science, Sun Yat-sen University, Guangzhou, 510275, People's Republic of China

[b] GF Securities, 26 Machang Road, Tianhe District, Guangzhou, 510627, People's Republic of China

ARTICLE INFO

Article history:
Received 11 June 2019
Received in revised form 30 January 2020
Accepted 27 February 2020
Available online 4 March 2020

Keywords:
Stock trend prediction
Deep neural network
Knowledge graph

ABSTRACT

Many studies have been carried out on stock price trend prediction, but most of them focused on the public market data and did not utilize the trading behaviors owing to the unavailability of real transaction records data. In fact, trading behaviors can better reflect the market movements, and the fusion of trading information and market information can further improve the prediction accuracy. In this paper, we propose a deep neural network model using the desensitized transaction records and public market information to predict stock price trend. Considering the correlation between stocks, our method utilizes the knowledge graph and graph embeddings techniques to select the relevant stocks of the target for constructing the market and trading information. Given the considerable number of investors and the complexity of transaction records data, the investors are clustered to reduce the dimensions of the trading feature matrices, and then the matrices are fed into the convolutional neural network to unearth the investment patterns. Eventually, the attention-based bidirectional long short-term memory network can predict the stock price trends for financial decision support. The experiments on the price movement direction and trend prediction show that our method achieves the best performance in comparison with other prediction baselines.

1. Introduction

Compared to the market index or sector index, the price of a single stock fluctuates more frequently because of its dynamic, non-linear, non-parametric and chaotic properties in nature. Although the single stock forecasting is difficult considering lots of uncertainties in the stock market, countless studies still tried to discover the pattern of stock trends.

Some researchers found that all available information is reflected in prices, and through studying the pattern of the past, they could predict the movements in the future. Approximately 90% of major stock traders are using this method because they tend to capture the trend [1].

Other researchers have strived to forecast the trend of stock indexes with a wealth of sophisticated means. From their perspective, investors make their investment decisions based on the intrinsic value of stocks, performance of the industry and economy, political climate [2]. Some of them adopted the news-driven theory, mining the news for trader's sentiment regarding stock [3]. Similarly, regulatory disclosures were also used to improve short-term and especially long-term forecasting of stock indices [4]. In other papers, researchers concentrated on exploiting the time series information of stock data to improve the performance of prediction. Kraus et al. applied LSTM to achieve the best performance overall and demonstrated the potential for future applications of deep learning in finance [5].

Existing stock trend prediction approaches, however, either did not combine the transaction records and public market information, or ignored the correlation between the target stock and other relevant stocks.

To address these problems, we have referred to many studies in other tasks and tried to bring new technologies into the task of stock trend prediction. Among them, Mahmoudi et al. utilized the deep neural networks to improve the performance of investors pattern mining [6], which shows the effectiveness of CNN in the financial task. Nam et al. realized the importance of focusing on the target firm and relevant firms together and designed a method analyzing the causal relationship between companies [7], which demonstrates the significance of stock correlation. To better unearth the hidden correlation between different objects, the knowledge graph is widely utilized in the tasks like ride-hailing

* Corresponding author.
E-mail address: issrjt@mail.sysu.edu.cn (J. Ren).

https://doi.org/10.1016/j.asoc.2020.106205
1568-4946/

demand forecasting [8] and items recommendation [9], which inspires us to excavate the relevance between stocks depending on the graph techniques. Lei proposed an integrated prediction method based on Rough Set (RS) and Wavelet Neural Network (WNN), where RS is firstly introduced to reduce the feature dimensions of stock price trend [10]. Inspired by RS-WNN, we use clustering algorithm to reduce the dimensions of the trading feature matrices.

This paper proposes a model using real trading data based on deep neural network to predict stock price trend. We firstly deal with the correlation between stocks by building the companies knowledge graph, and apply the *node2vec* [11] algorithm to obtain the vector representation of the company nodes. The *node2vec* algorithm is not new, but its effectiveness in different areas has been demonstrated in many studies [12]. Then the cosine similarities between them are used as the quantitative measure of the correlation between stocks. Considering the complexity of investor information, we cluster the investors according to the profiles of trading behaviors and obtains the investor clusters for further exploration of their trading patterns. Then the transaction number matrix, the buying volume matrix, and the selling volume matrix are formed based on the trading data, which are then fed into the convolutional neural network (CNN) for deep feature extraction. After concatenating the output of CNN with the market information weighted by stock correlation, we feed them into an attention-based bidirectional long short-term memory network (BiLSTM). Depending on the investors' transaction records data and price time series data of the past m trading days, our model predicts the price movement direction of the next trading day of the target stock and its price trend in the next n trading days.

The *Deep Stock-trend Prediction Neural Network* (DSPNN) proposed in this paper has been evaluated on the real data of three Chinese A-shares stocks (CITIC Securities, GF Securities, China Pingan) for nearly seven years. The accuracies of the movement direction prediction on the target stocks for the next trading day are all about 71%, the accuracies of the trend prediction on the targets for the next five trading days are all about 65% and those are about 74% for the next seven trading days, which demonstrates the effectiveness, robustness, and practicability of the model.

The main contributions of this paper can be summarized as follows.

- We explored whether the fusion of some investors' transaction records data and public market data can further improve the accuracy of stock price trend prediction and better support the decision-making process.
- We supplemented the information of the relevant stocks to the public market data of the target stock, and the selection of the relevant stocks is based on the knowledge graph and graph embeddings technique, which shows better performance than the traditional method in the experiments.
- We selected the number of transactions, the buying volume and the selling volume as the characteristics of trading behaviors, but in view of the large number of investors, clustering and the convolutional neural network are utilized to extract features from their transaction behaviors.
- We employed the long short-term-memory network to learn the timing information and used the attention mechanism and BiLSTM to help the model pay more attention on the special trading days, which achieves the promising performance in stock trend prediction task.

The rest of the paper is organized as follows. Section 2 will briefly review relevant studies on stock price trend prediction and Section 3 will define the problem tackled in this paper. The proposed prediction model will be discussed in detail in Section 4. Then the experimental settings and results analysis are reported in Section 5. Finally, Section 6 concludes the paper and gives out research prospects for the future.

2. Related work

Predicting the future trend of a stock is essential for investors as they can reduce the risk of decision-making by appropriately determining the future movement of their investment asset. There are numerous models and strategies proposed for the stock trend predictions, which can be classified into two categories: the ones based on statistical techniques and those using machine learning techniques.

In the category of statistical approaches, there were an autoregressive integrated moving average (ARIMA), generalized autoregressive conditional heteroskedasticity (GARCH) volatility [13], and the smooth transition autoregressive model (STAR) [14]. These approaches are primarily based on the assumptions of stationarity in time series and linearity among normally distributed variables. However, the stationarity, linearity and normality assumptions are not satisfied in real stock markets. And machine learning models without these restrictive assumptions have been proposed in recent years, and they can outperform the statistical methods [15]. Therefore, machine learning methods, such as SVM [16], fuzzy system [17] and hybrid methods [18,19], have been widely employed in stock prices forecasting.

In essence, stock data are time-series multidimensional vectors and more appropriate approaches should be taken in processing this special type of data. Many recent studies use deep neural networks for stock price prediction. Ding et al. were in favor of that the stock market is event-driven and utilized the deep convolutional neural network to the events extracted from news [20]. Considering LSTMs perform well in time-series problem, they might yield better advancement in stock prediction than conventional ones. Thus Chen et al. proposed an LSTM-based approach for stock returns prediction with a 13% improvement of accuracy [21]. The model of Luca Di al. indicated satisfying performance could be achieved for five-day prediction interval with LSTM and dropout [22], but they trained only on price series data which leaves room for improvement. In order to achieve non-stationary data analysis, Zhou et al. designed EMD2FNN, a 2-stage, end-to-end model for the prediction of the stock market trend [23]. Given the complexity of the stock market, Bisoi et al. developed a new robust kernel extreme learning machine (RKELM) for the stock price and trend prediction, which used a differential evolution algorithm to optimize the kernel functions for accurate prediction and applied a variational mode decomposition to remove non-linearity from the stock closing price data [24]. Also using the evolutionary algorithms, Ramezanian et al. proposed an integrated framework consisting of GNP, MLP and time series models to forecast the stock return, where different modules are responsible for optimizing different problems [25].

People influence the stock market by trading behaviors. However, few of the current studies are aware of the importance of trading patterns of people and the necessity of focusing on the target stocks and relevant stocks together. Thus this paper utilizes the trading data of investors and the market data of relevant stocks to predict the stock price trend. The fusion of investors' trading information and public market information and the reasonable integration of CNN and LSTM will further improve the performance.

3. Problem definition

This paper aims to predict stock price movement direction of the next trading day and the trend of the next n trading days. It can be divided into two subproblems: (1) what is the direction and the trend, (2) how does the predictions being made. The following sections concentrate on them, where we first formally quantify the direction and trend of price changes and then define the prediction problem.

3.1. Direction and trend definition

We use the difference between the *close price* of the current trading day and that of previous trading day to define the price movement direction y.

$$y_i = \begin{cases} 0, \text{if } s_i \le s_{i-1} \\ 1, \text{if } s_i > s_{i-1} \end{cases} \tag{1}$$

where y_i and s_i is the price movement direction and *close price* of the current trading day, and s_{i-1} is the *close price* of the previous trading day.

However, intraday trading is not profitable because of the transaction cost. Therefore, people pay more attention to the trend of the next n trading days rather than the direction of change.

In this work, we take the *close price* as the price of a trading day and then a price sequence for several trading days is formed, which will be labeled with t according to the definition of trend. Zhao et al. have put forward a way to quantify trend in stock movement [26], but it does not limit the length of the sequence, which is not in line with our situation.

Following is a brief introduction to the definition of the trend proposed in this study. Firstly, for the price sequence $S = \{s_1, s_2, s_3, s_4, \ldots, s_n\}$ where n is the number of trading days, if s_i satisfies $s_i > s_{i-1}$ and $s_i > s_{i+1}$, the point is a peak and then will be added into set P. Correspondingly, if s_i satisfies $s_i < s_{i-1}$ and $s_i < s_{i+1}$, the point will be considered as a trough and added into the set T. The set P and set T satisfy $|len(P) - len(T)| \le 1$ and $len(P) + len(T) \le len(S)$.

The existing of retracement and reversal increases the difficulty of trend labeling. The retracement refers to the situation where the movement amplitude after the inflection point is less than the original amplitude, and the trend is still the same as the original trend, as shown in the (a) and (c) subgraphs in Fig. 1. The reversal refers to the situation where the movement amplitude is larger than the original amplitude, and the trend is contrary to the original trend (see (b) and (d) subgraphs). Then we define two hyper-parameters used in the trend labeling, the retracement parameter α and the reversal parameter β. In this paper, α and β are set to 0.9 and 0.2 respectively.

When the stock starts to rise from trough T_i and reaches the second trough T_{i+1} after the peak P_i, the formulas for calculating the sub-trend q_i are as follows:

$$return_1 = \frac{P_i - T_{i+1}}{P_i - T_i} \tag{2}$$

$$q_i = \begin{cases} 0, \text{if } return_1 < \alpha \\ 1, \text{if } \alpha \le return_1 \le 1 + \beta \\ 2, \text{if } return_1 > 1 + \beta \end{cases} \tag{3}$$

When the stock starts to fall from the peak P_i and reaches the second peak P_{i+1} after the trough T_{i+1}, the formulas for calculating the sub-trend q_i are as follows:

$$return_2 = \frac{P_{i+1} - T_{i+1}}{P_i - T_{i+1}} \tag{4}$$

$$q_i = \begin{cases} 0, \text{if } return_2 > 1 + \beta \\ 1, \text{if } \alpha \le return_2 \le 1 + \beta \\ 2, \text{if } return_2 < \alpha \end{cases} \tag{5}$$

After having the sub-trends, we can then calculate the trend during this period.

$$t = \begin{cases} 2, \text{every sub-trend } q_i \text{ is } 2 \\ 0, \text{every } q_i \text{ is } 0 \\ 1, \text{otherwise} \end{cases} \tag{6}$$

The trend of S will be 'UP' when $t = 2$, 'DOWN' when $t = 0$ or 'NO' when $t = 1$.

3.2. Prediction problem

Given the time series data $[x_i, x_{i+1}, \ldots, x_{i+m}]$ where m is the window size of the input data, the prediction problem is to predict the direction of price movement $\hat{y}_{i+m+1}$, and the trend $\hat{t}_{i+m+n}$, where n is the number of days that need to predict the trend.

We want to build and train a model that can minimize the following loss function:

$$\boldsymbol{L} = \sum_{i=1}^{l} (Loss(\hat{y}_{i+m+1}, y_{i+m+1}) + Loss(\hat{t}_{i+m+n}, t_{i+m+n})), \tag{7}$$

where y_{i+m+1} is the real price movement of the next trading day, t_{i+m+n} is the real trend of the next n days and l is the size of the training data.

4. Proposed model

This section will introduce our methodology, the *Deep Stock-trend Prediction Neural Network* (DSPNN). The framework of the DSPNN is shown in Fig. 2, which consists of three components.

In the DSPNN, the Stock Market Information Module constructs a market indicator vector for each stock, finds the relevant stocks to the target, and their indicator vectors are weighted and added up to produce the ***Market Information Vector*** of the target stock. Afterward, in the Stock Trading Information Module, the investors are clustered according to their profiles of trading behaviors and the trading feature matrices are constructed based on the transaction behaviors of the investor clusters on the target and the relevant stocks. The matrices are subsequently fed into the CNN to obtain the ***Group Trading Vector*** of the target stock. Following this, the two vectors are concatenated and fed into the Attention-based BiLSTM Prediction Module to predict the target's price movement direction and its trend.

4.1. Stock market information

For each stock, we construct its market information v_{id} based on the indicators, consisting of *open price*, *close price*, *high price*, *low price*, *daily change*, *transaction volume*, *turnover*, *number of transactions*, *rate of change* and *amplitude*. And then the relevant stocks to the target stock are selected by using knowledge graph and graph embeddings techniques. Finally the indicator vectors of stocks are weighted by the similarities between them and added up to produce the ***Market Information Vector*** v_M.

4.1.1. Relevant stocks discovery

Only using the information of the target stock cannot comprehensively reflect the market changes, so it is necessary to integrate the market information of several relevant stocks to better represent the complicated market impact to the target, which will bring higher accuracy and stronger robustness.

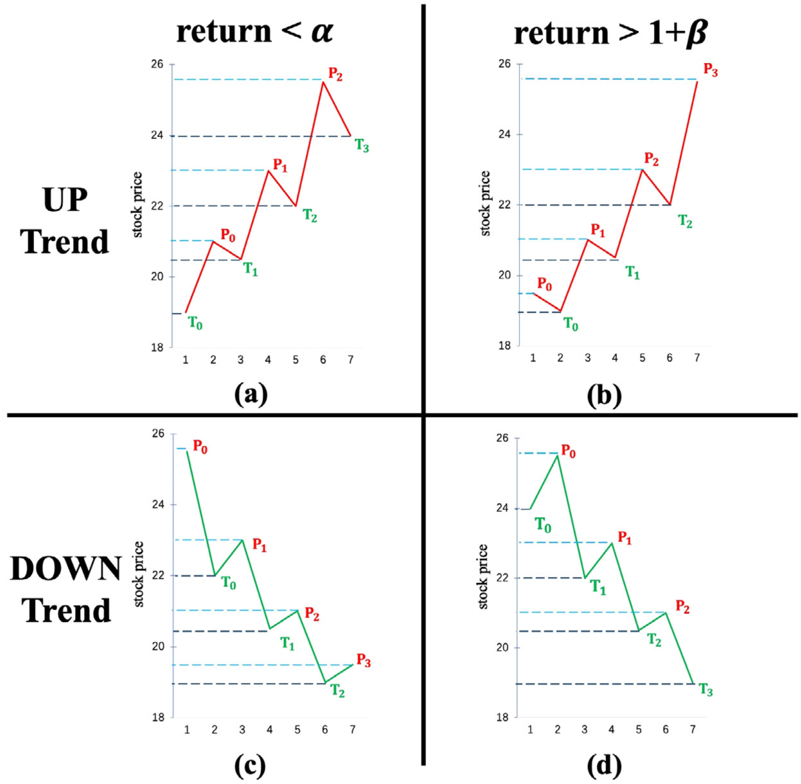

Fig. 1. The definitions of UP trend and DOWN trend.

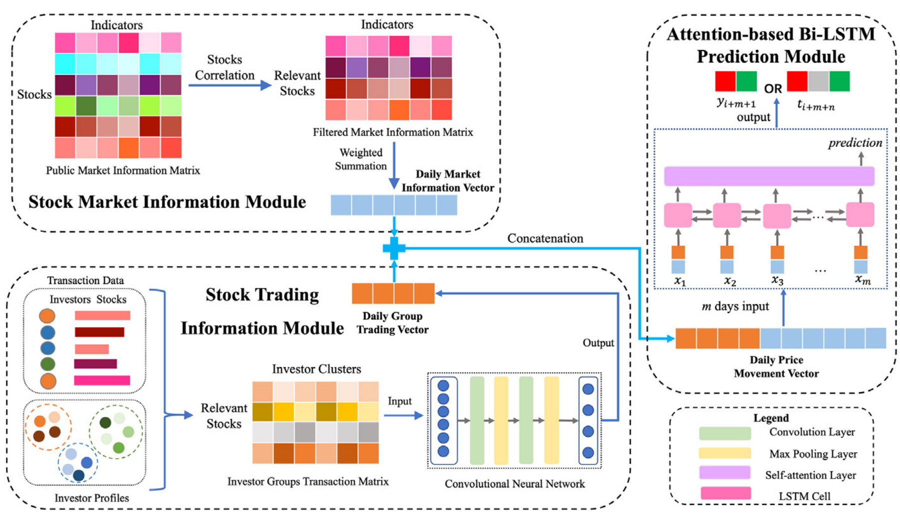

Fig. 2. The framework of DSPNN model.

However, traditional methods of finding relevant stocks are based on simple co-occurrence relationship and their performance are far from satisfactory. In order to better unearth the hidden correlation between stocks and find the relevant stocks of the target, the companies knowledge graph having multiple entities and relations is designed and built in this work, which describes the securities market situation.

The companies knowledge graph is defined as a set of triples (i, r, j) where $i, j \in E$, $r \in R$, E is the set of entities and R is a set of relations among them. Entity types include *company*, *concept*, *industry* and relation types include *shareholding*, *relevance* and *affiliation*. Concepts are the hottest keywords like blockchain, 5G, etc., and an industry is a group of companies that are related based on their primary business activities.

After building the graph, we can apply the *node2vec* algorithm to sampling the nodes and use node embeddings to represent the companies. With the mapping function $f : V \rightarrow R^d$ obtained by *node2vec*, we can have the node embeddings of the stocks s_1 and s_2. Then the cosine similarity between them can be calculated using Eq. (8).

$$cos(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1| \times |v_2|}, \tag{8}$$

where v_1 and v_2 are the graph embeddings of the stocks, that is to say, $v_1 = f(s_1)$ and $v_2 = f(s_2)$.

We can use the above method to calculate the similarity between any pair of stocks and the relevant stocks are those having top-K similarities to the target stock.

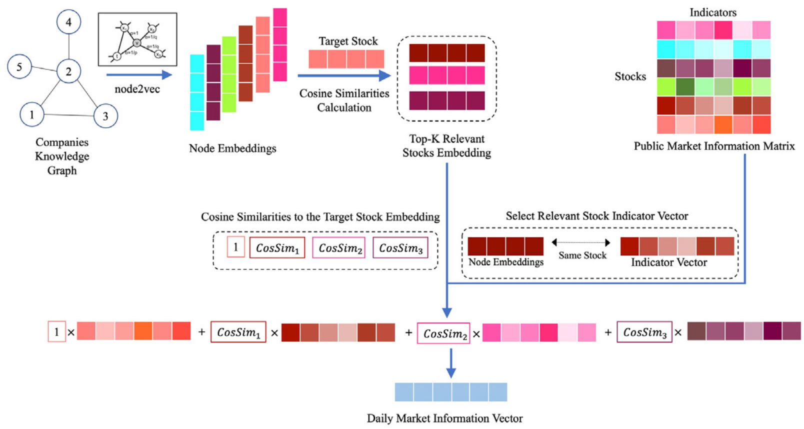

Fig. 3. The procedure of constructing market information vector based on the relevant stocks and their similarities to the target.

4.1.2. Market information integration

The procedure of utilizing the graph to obtain the ***Market Information Vector*** is shown in Fig. 3.

e_t^k is the cosine similarity between the target stock and a relevant stock s_k. α_t^k is the weight measuring the relevance of s_k to the target s_t. A softmax function is applied to e_t^k to ensure all the weights sum to 1.

$$e_t^k = cos(f(s_k), f(s_t)) \tag{9}$$

$$a_t^k = \frac{exp(e_t^k)}{\sum_{i=1}^{K} exp(e_t^i)}, \tag{10}$$

where K is the size of relevant stocks.

The relevance weights are used to sum up the v_{id} of each relevant stock to obtain the ***Market Information Vector*** v_M with Eq. (11).

$$v_M = v_{id}^t + \sum_{j=1}^{K} (\alpha_t^j \times v_{id}^j), \tag{11}$$

where v_{id}^t is the indicator vector of the target stock and v_{id}^i is the indicator vector of a relevant stock.

4.2. Stock trading information

We select the number of transactions, buying volume, and selling volume on the target stock and relevant stocks as the indicators of investors' trading behaviors. However, due to the large number of investors, we need to cluster the investors at the beginning and then construct behavior matrices of the clusters. The convolutional neural network will extract the trading features from the matrices and then generate the ***Group Trading Vector*** v_T.

4.2.1. Investor clustering

Since the investors influence the stock market by their trading behaviors, we select their *number of transactions*, *buying volume*, and *selling volume* on the target and relevant stocks as the indicators.

However, due to the large number of investors, the sizes of the feature matrices will become too large to be processed under reasonable computing resources, so we need to aggregate the investors by using clustering algorithm, and then just focus on the trading indicators of these investor clusters. Moreover, considering the hidden correlation between stocks, using only the investor trading operations on the target stock is not enough for prediction, so it is necessary to consider the investor clusters' operations on the relevant stocks which have been found by the previous module.

With the trading data, we can construct investor profiles of trading behaviors according to four types of indicators, including *Sector Selection Ability*, *Trading Time Preference*, *Holding Control Capability* and *Trading Method*, and then K-means algorithm is applied to the investors clustering based on their profiles. The description of the selected indicators used to cluster the investors is shown in Table 1.

4.2.2. Trading behavior pattern

The trading feature matrices of investor clusters can be constructed based on the *number of transactions*, *buying volume*, and *selling volume* on the target and relevant stocks.

The transaction records on the stocks are classified according to the trading date, the investor cluster, and the stocks involved. Afterward, three matrices of each trading day are formed including the transaction number matrix, the buying volume matrix and the selling volume matrix. The dimensions of these matrices are $(K+1)\times C$, where K is the number of relevant stocks and C is the number of the investor clusters. The procedure for constructing the matrices is shown in Fig. 4.

Subsequently, the matrices will be fed into the convolutional neural network for deep features extraction, and a d-dimension vector is obtained for each trading day to represent the investor clusters' trading patterns on the target and relevant stocks.

4.2.3. CNN based feature extraction

In this section, we will introduce the reason why we use CNN to extract the trading features and how to construct the structure for this application.

Although people have their own trading behaviors, there will be some common patterns among them. The reason of using CNN to extract trading features is that convolutional neural network can find and learn local and global space features in the input matrices. In our application, the input matrices are formed by various trading features of different types of traders, such as buying volume, selling volume, buying price and selling price. With the convolutional operations, the important similarities and differences of trading patterns can be detected by various filters. And the pooling operations can discard unimportant trading features. The dropout operations can alleviate the over fitting problem of trading feature extraction and improve the generalization of the model.

The CNN for investor features extraction, following Kim's study [27], consists of four layers, including input, convolution layer, max-pooling layer, and fully connected layer (see Fig. 5). Following, we will discuss each layer step by step.

Table 1
Description of the selected indicators used for investors clustering.

Indicator	Illustration
Sector concentration	The ratio of the number of transaction on the sector that accounts for the largest proportion of investment to the total number of transaction.
Hot sector preference	The ratio of the number of transaction on the popular sectors to the total number of transaction.
Bull market preference	The ratio of the number of transaction on a bull market to the total number of transaction.
Bear market preference	The ratio of the number of transaction on a bear market to the total number of transaction.
Bull market holding	The ratio of the value of holding on a bull market to the total asset.
Bear market holding	The ratio of the value of holding on a bear market to the total asset.
Trading intensity	The rate of the average number of transaction to the average daily total assets for several trading days.
Trading frequency	The rate of trading days having transaction to the total trading days.
Trading enthusiasm	The rate of trading days holding shares to the total trading days.
Return on assets	The ratio of total number of profit to the average daily total asset for several trading days.

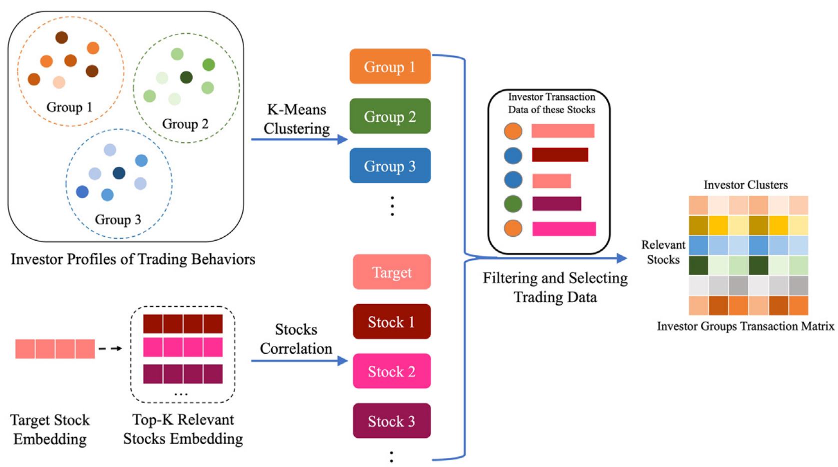

Fig. 4. The procedure of constructing trading feature matrix for investor clusters on the target and relevant stocks.

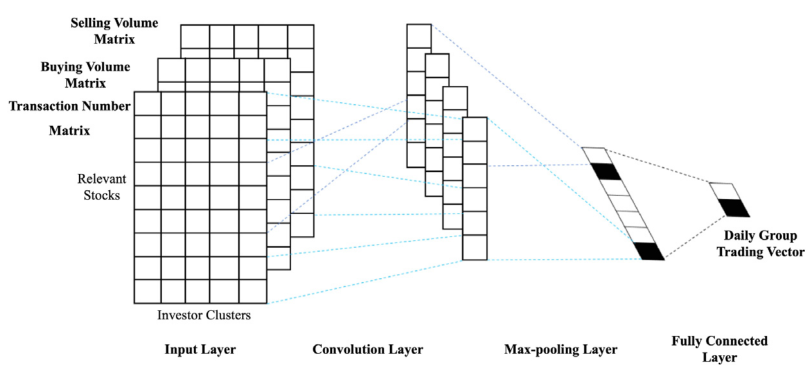

Fig. 5. The procedure of obtaining the Group Trading Vector with the help of the CNN.

Input layer treats the input matrix as a sequence of C groups, each of which is represented by a $(K+1)$-dimensional vector: $[x_1, x_2, x_3, \dots, x_C]$ where C is the number of investor clusters, $K+1$ is the number of stocks and $x_i \in \mathbb{R}^{K+1}$, $\forall i = 1, 2, 3, \dots, C$.

Convolution layer aims to capture local features that concurrently appear in the previous layer by a set of learnable filters called convolution kernels. Mathematically, the weight matrix for the convolution filter is $w \in \mathbb{R}^{h \times d}$, which will be applied to the window of h features with an embeddings dimension of d. After convolving every possible window of features, the feature map c becomes:

$$c = [c_1, c_2, c_3, \dots, c_{n-h+1}], \tag{12}$$

where $c \in \mathbb{R}^{n-h+1}$ and the convolutional filter c_i for position i in the matrix is calculated by:

$$c_i = f(\mathbf{w} : x_{i:i+h-1} + b), \tag{13}$$

where $b \in \mathbb{R}$ is the bias and f is a non-linear activation function.

Max-pooling layer addresses the most important features by pooling over every feature map bearing a close resemblance to the process of feature selection in finding investments patterns. Thus, the pooled feature map, p, will be calculated by:

$$p = [max(c_1, c_2, c_3, \dots, c_{n-h+1})] \tag{14}$$

Fully connected layer as a high dimensional dense layer, receives the concatenated and flattened pooled feature maps and fed the maps into the output layer whose output is the class probabilities. The output layer computes these probabilities by softmax activation as follows:

$$P(y=j|\mathbf{x},\mathbf{w},b) = softmax_j(\mathbf{x}^T\mathbf{w}+b) = \frac{e^{\mathbf{x}^T\mathbf{w}_j+b_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^T\mathbf{w}_k+b_k}}, \tag{15}$$

where w_k and b_k are the weight vector and bias of the kth class.

The output of the CNN is the daily ***Group Trading Vector*** v_T, which is then concatenated with the daily ***Market Information Vector*** v_M, resulting the daily ***Price Movement Vector*** v_P.

4.3. Attention-based BiLSTM prediction

To forecast the stock price movement direction and trend, we leverage Bidirectional Long Short-term Memory network (BiLSTM) on stock market information and stock trading information to generate the stock price movement representation for trading days.

The v_P of m past trading days are fed into the attention-based BiLSTM and the models will predict the direction of price change in the next trading day and the trend in the next n trading days of the target stock (see Fig. 6).

The LSTM is a special recurrent neural network model which cannot only store and access a longer range of contextual information in the sequential input, but also handle the vanishing gradient problem in the meanwhile. A single cell in LSTM has a cell state and three gates, i.e., input gate i, forget gate f and output gate o. Formally, the LSTM can be formulated as follows:

$$i_t = \sigma(W_i x_t + U_i \overrightarrow{h}_{t-1} + b_i) \tag{16}$$

$$f_t = \sigma(W_f x_t + U_f \overrightarrow{h}_{t-1} + b_f) \tag{17}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t \tag{18}$$

$$o_t = \sigma(W_o x_t + U_o \overrightarrow{h}_{t-1} + b_o) \tag{19}$$

$$\hat{c}_t = W_c x_t + U_c h_{t-1} + b_c \tag{20}$$

$$h_t = o_t \odot tanh(c_t) \tag{21}$$

where $X = x_1, x_2, \ldots, x_m$ and m denote the input vector and the length of X respectively. And $W_i, W_f, W_c, W_o, b_i, b_f, b_c, b_o$ are the parameters as weight matrices and biases, $\odot$ is the sigmoid function, and $h_1, h_2, \ldots, h_m$ represents a sequence of semantic features. The above formulas can be represented in short as:

$$h_t = LSTM(x_t, h_{t-1}) \tag{22}$$

In order to make full use of the time information, BiLSTM is utilized in this work, which consists of forward and backward LSTM networks [28]. After the hidden states in different directions are obtained, they are concatenated to form the final hidden state at t.

The hidden states in different directions are obtained, and then they are concatenated to obtain the final output at t.

$$\overrightarrow{h}_t = LSTM(x_t, \overrightarrow{h}_{t-1}) \tag{23}$$

$$\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t-1}) \tag{24}$$

$$h_t = [\overrightarrow{h}_t : \overleftarrow{h}_t] \tag{25}$$

The above formulas can be represented in short as:

$$h_t = BiLSTM(x_{1:m}, t) \tag{26}$$

where $x_{1:m}$ denotes the sequence of concatenated trading information and market information $x_1, x_2, \ldots, x_m$.

Because stock price movements are often related to some specific time points, such as the time when the government issues new policies and the periods when some enterprises publish their financial reports. In order to focus on these important moments, we add the attention mechanism into the BiLSTM. The attention mechanism is a variant of attention mechanism for some tasks where is no extra information to guide the extraction of import features like classification [29]. It gives different weights according to the state of time series data at different moments, so it is able to pay more attention to the information of key trading days.

$$u_t = tanh(Wh_t) \tag{27}$$

$$\alpha_{t,i} = \frac{exp(u_t^T u)}{\sum_{t=1}^{T} exp(u_t^T u)}, \tag{28}$$

$$\hat{y} = \sum_{t=1}^{T} \alpha_t h_t \tag{29}$$

where u is a trainable parameter matrix used to represent context information and $\alpha_{t,i}$ is the allocation coefficient of input states, both of which are randomly initialized and optimized during the training procedure.

Since our task is formulated as a classification problem, we use negative log likelihood loss function.

$$L(y_t, \hat{y}_t) = -y_t log \hat{y}_t \tag{30}$$

Note that Eq. (30) is the loss for one instance. In this paper, the adaptive moment estimation (Adam) algorithm is utilized to minimize the above loss values over all instances in the training data. The parameters of CNN and BiLSTM will be optimized during minimizing the above loss function.

5. Experiments

This section compares the performance of DSPNN and other baselines as well as the performance based on different features. The evaluation demonstrates that the fusion of transaction records data and public market information helps improve the prediction performance. At the same time, investors clustering and knowledge-graph-based techniques can better mine the features of the investors and the market.

We utilized a computing system consisting of an Intel i7-7700k with four cores running at 4.2 GHz and 16 GB RAM. The experiments with deep learning are implemented by TensorFlow.

To prove the performance of the proposed model, we experiment with direction classification, where the direction of the stock price movement of the next trading day is predicted, and trend classification, where the trend of the stock price for the next several trading days is predicted.

5.1. Experimental settings

In our experiments, about nine million clients' transaction records data and market information on stocks are used to train and test the proposed model and other baselines. The data are provided by a top-five Chinese securities company from March 2012 to June 2018 and subjected to rigorous desensitization to ensure the privacy of investors. To better introduce the dataset, examples of the desensitized transaction record, the market data for a stock on a trading day and the profile for a client are shown in Table 2, Table 3 and Table 4 respectively. The CITIC Securities, GF Securities and China Pingan are selected as the target stocks to evaluate the performance of the models.

Not only the accuracy is reported but also the balanced accuracy and AUC metrics for comparing the performance of the prediction task. The balanced accuracy is defined as $c \times \frac{TP}{P} + (1-c) \times \frac{TN}{N}$, where $c \in [0,1]$ is the cost associated with the misclassification of a positive example [30], in order to account for unbalanced classes of the trends in our dataset.

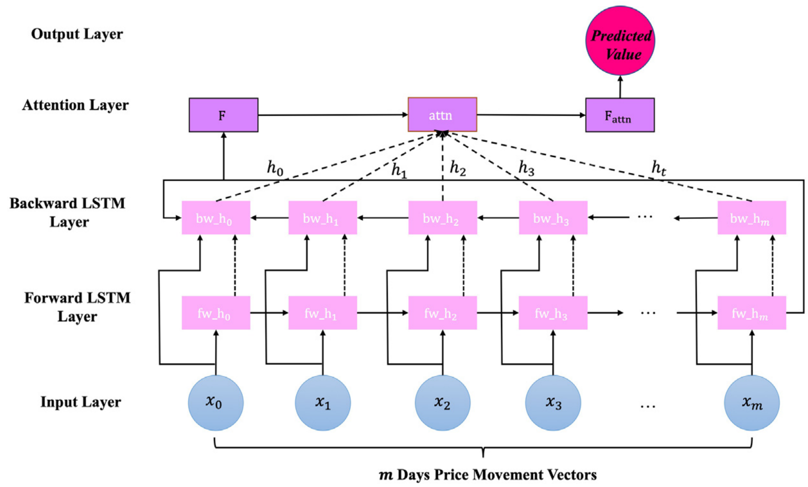

Fig. 6. The procedure of prediction using the Attention-based BiLSTM. The predicted value may be the price direction or trend of the target stock.

Table 3
An example of market data for a stock on a trading day.

Stock name	Present price	Price change	Turnover amount	Net inflow
CITIC Securities	23.75	0.62	2 765 566 088	110 068 967
Volume ratio	Amplitude	Highest	Lowest	Volume
1.08	2.59	23.77	23.17	118 201 492
60 days change	Circulation value	Total market value	P/E ratio	P/E ratio TTM
1.18	233 098 215 375	267 690 135 772	30.65	22.84
MA(5)	MA(10)	MA(20)	MA(60)	MA(120)
23.34	23.44	22.74	22.33	22.58
MA(250)	Surged limit	Decline limit	Highest in 52 weeks	Lowest in 52 weeks
21.84	25.44	20.82	27.45	15.64
1 min change	3 min change	5 min change	1 min turnover	3 min turnover
0.04	0.04	0.09	109 425 750	109 425 750

Table 4
An example of profile for a client.

Client ID	Purchase proportion of hot spots	Purchase proportion of personal sectors	Proportion of bull market purchase	Proportion of bear market purchases
30400XXXXXXX	0.22724	0.244837	0.231261	0.4510937
Bull market holding	Bear market holding	Proportion of holding time	Trading frequency	Profitability
0.8481	0.97707	1	0.6477	−2
Trading intensity	Clustering	Total assets	Registration time	Transactions
0.531939	1	212 798	2011-07-XX	21 140

Table 2
An example of client transaction flow data after desensitization (The Client ID has been desensitized).

Trading date	Client ID	Stock code	Volume	Amount
2018-07-13	30400XXXXXXX	600030.SS	300	5013

In addition to comparing the prediction performance of different models, we also conducted the experiments with or without transaction records data, whether the investors were clustered, whether the relevant stocks were weighted, to verify the promotion of transaction records data, investors clustering, and knowledge graph techniques.

5.2. Comparison with other models

In order to show the effectiveness of DSPNN model, groups of experiments are conducted among commonly used models including Random forest, AdaBoost and LSTM. In the experiments, Random forest was implemented according to a recent stock trend prediction study, which trains models from historical data using random forest with feature selection [31]. For AdaBoost, we refer to the implementation of the baseline in Zhao's work [26]. The LSTM based prediction model in Yao's work [32] was also taken as baseline. All the baseline models are implemented according to the references. Additionally, there are some experiments where the prediction models were fed with different input features. In order to distinguish them more clearly, the suffixes are used to denote the training data of the models. For example, DSPNN-m represents the DSPNN model trained with only simple market information and DSPNN-M represents the DPSNN model trained with only weighted market information (see Table 5).

Table 5
Models and their abbreviations used in this paper.

Abbreviation	Model
RF-MT	The Random Forest trained with weighted market information and comprehensive trading information on multiple stocks.
AB-MT	The Adaboost model trained with weighted market information and comprehensive trading information on multiple stocks.
LSTM-mt	The LSTM model trained with simple market information and simple trading information on a single stock.
LSTM-MT	The LSTM model trained with weighted market information and comprehensive trading information on multiple stocks.
DSPNN-m	The DPSNN model trained with only simple market information.
DSPNN-M	The DPSNN model trained with only weighted market information.
DSPNN-t	The DPSNN model trained with only simple trading information on a single stock.
DSPNN-T	The DPSNN model trained with only comprehensive trading information on multiple stocks.
DSPNN-mt	The DPSNN model trained with simple market information and simple trading information on a single stock.
DSPNN-MT	The DPSNN model trained with weighted market information and comprehensive trading information on multiple stocks.

Table 6
The similarities of the 14 stocks relevant to the CITIC Securities.

Stock code	Company name	Cosine similarity
600280.SH	Nanjing Central Emporium (Group) Stocks Co., Ltd.	0.5957
600748.SH	Shanghai Industrial Development Co., Ltd.	0.5936
002476.SZ	Shandong Polymer Biochemicals Co., Ltd.	0.5710
600837.SH	Haitong Securities Co., Ltd.	0.5481
300002.SZ	Beijing Ultrapower Software Co., Ltd.	0.5447
000712.SZ	Guangdong Golden Dragon Development Inc.	0.5356
600999.SH	China Merchants Securities Co., Ltd.	0.5355
601788.SH	Everbright Securities Co., Ltd.	0.5178
600900.SH	China Yangtze Power Co., Ltd.	0.4867
601318.SH	Pingan Insurance (Group) Company of China, Ltd.	0.4856
000776.SZ	GF Securities Co., Ltd.	0.4836
000783.SZ	Changjiang Securities Co., Ltd.	0.4773
601628.SH	China Life Insurance Co., Ltd.	0.4760
601601.SH	China Pacific Insurance (Group) Co., Ltd.	0.4758

5.3. Results analysis

We forecast the direction and trend of stock price movement respectively.

In the classification of movement direction, we mainly focus on predicting the movement of CITIC Securities for comparative experiments, using the data of the first m trading days to forecast the direction of movement of the next trading day. Then we use the best model to predict the movements of CITIC Securities, GF Securities, and China Pingan.

In the classification of trend, the five-day trend and seven-day trend of CITIC Securities are predicted using different models and the trends of three target stocks are forecasted using the best model to ensure its generalization.

5.3.1. Classification: the direction of price movement

In order to illustrate the experiment more simply, CITIC Securities is taken as the target stock when comparing with other baselines and then we will report the best prediction performance on other two target stocks. Through grid search, we found that the model has the best prediction performance when $K = 14$, $C = 10$ and $m = 30$, that is, clustering investors into 10 groups, using 14 relevant stocks for market information integration and using the data of previous 30 days can get the best prediction results.

The accuracies corresponding to different values of K and C are shown in Fig. 7. K represents the number of related stocks. When the number of related stocks is too small, the useful information obtained by the model is not enough. When there are too many related stocks, more noise will be introduced and the noise will affect the judgment of the model. C represents the number of user clusters. When C is small, the differentiation between clusters is small, and their trading patterns are relatively similar, so there are less differences can be found by our model. When C is too large, some clusters are not representative. The transaction behavior of these users may be special cases, and more outlying data may be introduced.

Table 7
Results from classifying the price movement direction of CITIC Securities.

Models	Accuracy	AUC
Proposed methods		
DSPNN-MT	**0.7359**	**0.7704**
DSPNN-mt	0.6718	0.7110
DSPNN-T	0.6666	0.7142
DSPNN-t	0.5795	0.5865
DSPNN-M	0.5555	0.5415
DSPNN-m	0.5338	0.5365
Deep learning		
LSTM-MT	0.7089	0.7491
LSTM-mt	0.6513	0.7074
Machine learning		
RF-MT	0.4887	0.4881
AB-MT	0.4945	0.4934

The similarities between CITIC Securities and other stocks are shown in Table 6.

We now proceed to evaluate the models for predicting the direction of stock price movement. The corresponding results are in detailed in Table 7. Since the classes of price movement direction are relatively balanced, we did not use balanced accuracy for evaluation in this part.

Among the baseline models from traditional machine learning, we find the Adaboost has a better performance, which yields an improvement of 1.2 percent points compared to the Random Forest. This is due to its more complex model structure and longer training time.

Deep learning outperforms traditional machine learning. For instance, the LSTM-MT yields an improvement of 43.4 percent points over the Adaboost.

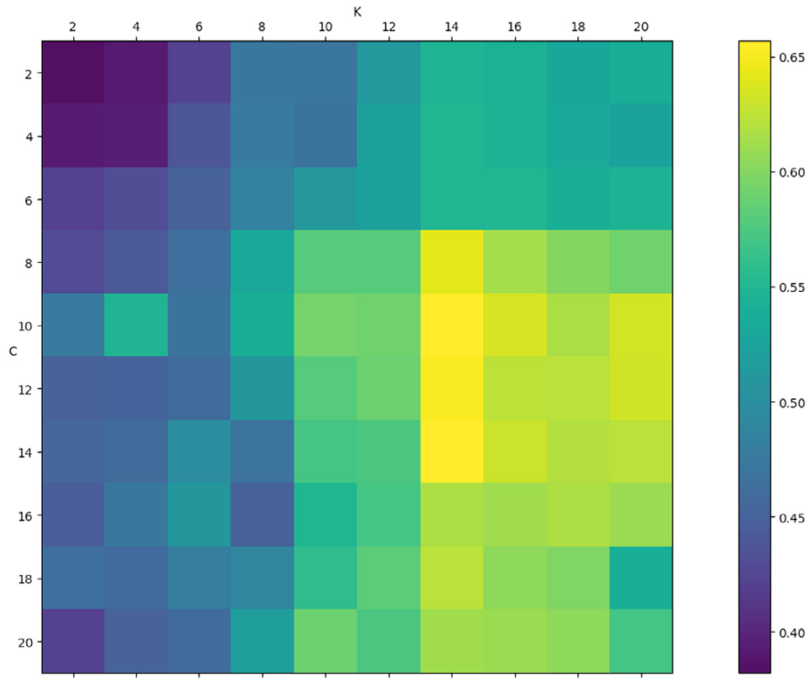

Fig. 7. The prediction accuracies corresponding to different values of K and C.

Table 8
Results from classifying the directions of the target stocks using DSPNN-MT.

Stock	Accuracy	AUC
CITIC Securities	0.7359	0.7704
GF Securities	0.7076	0.7649
China Pingan	0.7282	0.7797

Table 9
The trend classes distribution of the target stocks.

Stock	Five days trend			Seven days trend		
	UP	DOWN	NO	UP	DOWN	NO
CITIC Securities	344	376	825	208	252	1085
GF Securities	372	430	743	220	267	1058
China Pingan	425	368	752	287	238	1019

The proposed model DPSNN outperforms the LSTM under the situations using different training data and yields an improvement of 3.8 percent points over the LSTM because of the bidirectional structure and the attention mechanism of DSPNN, which can pay more attention to the features of specific trading days, so as to better learn the time series information of stock price.

Investors clustering and knowledge graph techniques yield consistent improvements for the proposed model and deep learning methods. As a result, the DSPNN-MT model performs best among all models, amounting to a total improvement of 50.5 percent points over the Random Forest. When using only the market information, the DSPNN-M yields an improvement of 4 percent points over the DSPNN-m, which is due to the correlation between stocks found in the knowledge graph. The addition of trading data further enhances the performance of the models, the DSPNN-t has better performance compared with the DSPNN-m, and the DSPNN-MT has an 32.5% improvement compared with the DSPNN-M, which shows that the addition of trading data can promote the performance, and user clustering and graph-based market information weighting can further improve the model.

The results in Table 8 show that DSPNN-MT can achieve satisfying performance in the price movement direction prediction of the three target stocks, not just in the prediction of one stock.

The reasons why the traditional machine learning algorithms plus our ideas do not perform well are as follow:

1. Since the numbers of parameters of traditional machine learning algorithms are less than deep learning algorithms, they have weaker fitting abilities to deal with the high frequency and nonlinearity of stock price change.

2. The methods based on neural network have become the mainstream in different tasks. The LSTM plus the features we put forward is more competitive and is a more suitable baseline of DSPNN.

Although the performance of traditional methods are not satisfying, their training time is the advantage over the deep learning algorithms. When we increase the number of trading days used for training, the advantages of RF-MT and AB-MT begin to appear, with relatively stable performance but less training time. But when the window size of training days exceeds 20, the performance of deep learning method is better than the traditional machine learning.

From the experiments on RF-mt and AB-mt which use only simple market features and trading features, we notice that they have worse performance instead of RF-MT and AB-MT, which use knowledge-graph based market feature integration and CNN-based trading feature extraction. And when we repeatedly sample trading days from the dataset and train RF-mt, AB-mt, RF-MT and AB-MT with the data on the same trading days, we obtain the results in Fig. 9. The accuracy ranges of MT-models are narrower than those of mt-models. This shows that our ideas make the models more competitive and stable.

5.3.2. Classification: the trend of price movement

Table 9 shows the classes of the five-day trend and seven-day trend of the target stocks, and the number of NO-Trend examples is much larger than the other two classes, which explains why we focus on the balanced accuracy in the following analysis.

Table 10 reports the results for predicting the trends of CITIC Securities. The balanced accuracies of the Random Forest and

Table 10
Results from classifying the price trend of CITIC Securities.

Models	Five days trend			Seven days trend		
	Accuracy	Balanced accuracy	AUC	Accuracy	Balanced accuracy	AUC
DSPNN-MT	**0.6564**	**0.6048**	0.7003	**0.7589**	**0.6191**	**0.7310**
DSPNN-mt	0.6051	0.5564	0.6437	0.6872	0.6022	0.6978
DSPNN-T	0.6153	0.5413	0.5889	0.7538	0.5441	0.6806
DSPNN-t	0.5436	0.4974	0.5398	0.6410	0.5052	0.5863
DSPNN-M	0.5384	0.5125	0.5203	0.6564	0.4859	0.5539
DSPNN-m	0.4871	0.4583	0.5027	0.6256	0.4883	0.5105
LSTM-MT	0.6256	0.5750	**0.7012**	0.7487	0.5884	0.6769
LSTM-mt	0.6000	0.5611	0.6386	0.6615	0.5693	0.6684
RF-MT	0.3875	0.3505	0.5165	0.4409	0.3808	0.5264
AB-MT	0.3839	0.3479	0.5175	0.3556	0.3837	0.5218

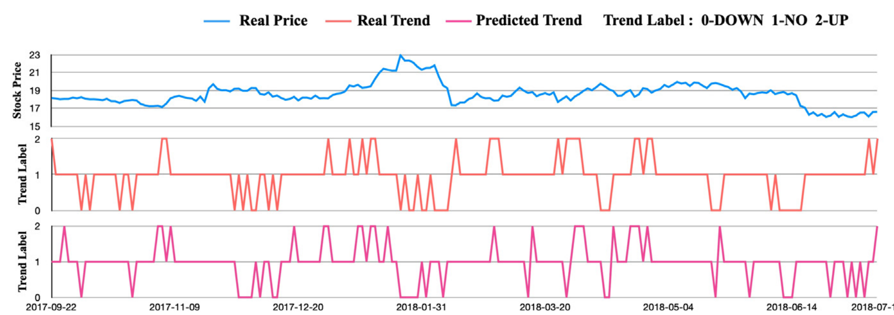

Fig. 8. Comparison between the real prices and trends of CITIC Securities with the trends predicted by DSPNN-MT. The blue line is the real close price series of CITIC Securities from 2017 to 2018. The orange line conforms to the trends of real prices while the red line is the trend series predicted by the DSPNN-MT.

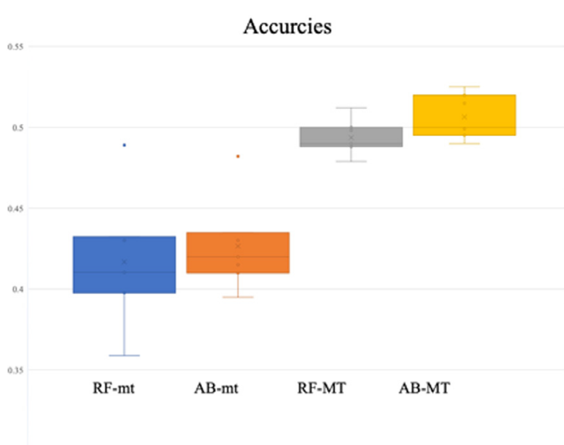

Fig. 9. The boxplot of the accuracies of RF-mt, AB-mt, RF-MT and AB-MT.

Adaboost are no more than 40%, which are far away from the LSTM. The LSTM-MT achieves a five-day balanced accuracy that is 65.3% higher than that of the Adaboost and its seven-day balanced accuracy is 53.3% higher than that of the Adaboost.

With regard to deep learning, both the weighted market information and comprehensive trading information on stocks improve performance beyond the simple market and trading information. Similarly, the performance of DSPNN is also improved.

We can see that the DSPNN with only trading information, performs better than the DSPNN with only market information, and the promotion of comprehensive trading information on multiple stocks is more obvious than that of the weighted market information, which is due to the trading features extraction of CNN.

The best performance is achieved by DSPNN-MT in predicting the five-day trend and seven-day trend, which are about 5.1% and 5.2% over the LSTM-MT and about 73.8% and 61.4% over the Adaboost. Such obvious improvement is not only owing to the reasonable integration of the CNN and BiLSTM but also the features construction by the investor clustering and graph embeddings.

The results of the prediction on the three target stocks are shown in Table 11, which demonstrates the generalization and effectiveness of DSPNN-MT.

Fig. 8 shows the true stock price and trend of CITIC Securities from 2017 to 2018 and the trend predicted using DSPNN-MT. The blue line is the curve of the real price, and the orange line is the corresponding trend. The price rises and falls directly corresponds to the change in the trend. When the stock price rises or falls sharply, the trend will change significantly. When the stock price is in a period of relatively complex fluctuations, the trend will also fluctuate. The red curve is the trend predicted by DSPNN-MT, which is very similar to the orange curve and can be well fitted with the orange one at key positions. In fact, the changes of these two curves between the up-trend and down-trend are very synchronous, and many differences occur when there is a change between the no-trend and the other trends. No-trend tends to continue the trend of the previous days and the stock price during no-trend does not change greatly. It proves that the DSPNN model can effectively predict the trend of the stock price.

6. Conclusion

Deep learning has made lots of achievements in natural language processing, computer vision, and other fields. However, in the financial field, especially in financial time series prediction, little progress has been made recently. The prediction accuracy of a single stock is only about 60% because of the immaturity of the Chinese A-share market, the non-linear instability of the time series, and the absence of open reliable dataset.

Therefore, to further improve the performance of trend prediction, the following studies are carried out in this paper:

(1) Considering the fusion of trading information and public market information can improve the performance, this paper

Table 11
Results from classifying the price trend of the target stocks using DSPNN-MT.

Stock	Five days trend			Seven days trend		
	Accuracy	Balanced accuracy	AUC	Accuracy	Balanced accuracy	AUC
CITIC Securities	0.6564	0.6048	0.7003	0.7589	0.6191	0.7310
GF Securities	0.6410	0.5962	0.6831	0.7487	0.6150	0.6869
China Pingan	0.6615	0.6119	0.6911	0.7333	0.5837	0.6891

utilizes the investors' trading records data to model their trading patterns, applies the graph-based techniques to construct the market features, and fuses the transaction information and market information.

(2) In view of the fact that a single stock will be affected by other market factors, this paper builds the companies knowledge graph to unearth the stock correlation and utilizes *node2vec* to obtain the graph embeddings, which are then used to calculate the similarities of stocks and find the relevant stocks of the target.

(3) In order to extract useful trading information, this paper first clusters investors according to the profiles of investors and obtains different investor clusters. Based on these clusters, the transaction records data is integrated to obtain the transaction matrices, which are then fed into CNN to further extract the trading features.

(4) Considering the importance of the key trading days, the Attention-based BiLSTM is employed to forecast the direction of price changes and the trend of price movements.

In the experiments, traditional machine learning, classical neural network, and the proposed DSPNN model are evaluated on three target stocks in the Chinese A-share market. The accuracy of trend prediction has reached 70% or more, which has been greatly improved compared with the current accuracy of about 65% [26]. The results demonstrated the performance, robustness, and practicability of our model.

More attention should be paid to further improve the efficiency of the model. At the same time, it is necessary to keep on improving the performance by combining the text and sentiment information from financial news and other social media in future work.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2020.106205.

CRediT authorship contribution statement

Jiawei Long: Conceptualization, Methodology, Software, Writing - original draft. **Zhaopeng Chen:** Data curation, Software, Writing - original draft. **Weibing He:** Resources, Supervision. **Taiyu Wu:** Resources, Supervision. **Jiangtao Ren:** Project administration, Conceptualization, Writing - review & editing.

Acknowledgment

This research is partially supported by National Natural Science Foundation of China (No. U1811462).

References

[1] J. Agrawal, V. Chourasia, A. Mittra, State-of-the-art in stock prediction techniques, Int. J. Adv. Res. Electr. Electron. Instrum. Eng. 2 (4) (2013) 1360–1366.
[2] M. Ballings, D. Van den Poel, N. Hespeels, R. Gryp, Evaluating multiple classifiers for stock price direction prediction, Expert Syst. Appl. 42 (20) (2015) 7046–7056.
[3] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, X. Deng, Exploiting topic based twitter sentiment for stock prediction, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vol. 2, 2013, pp. 24–29.
[4] S. Feuerriegel, J. Gordon, Long-term stock index forecasting based on text mining of regulatory disclosures, Decis. Support Syst. 112 (2018) 88–97.
[5] M. Kraus, S. Feuerriegel, Decision support from financial disclosures with deep neural networks and transfer learning, Decis. Support Syst. 104 (2017) 38–48.
[6] N. Mahmoudi, P. Docherty, P. Moscato, Deep neural networks understand investors better, Decis. Support Syst. 112 (2018) 23–34.
[7] K. Nam, N. Seong, Financial news-based stock movement prediction using causality analysis of influence in the Korean stock market, Decis. Support Syst. 117 (2019) 100–112.
[8] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, Y. Liu, Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting, in: 2019 AAAI Conference on Artificial Intelligence, AAAI'19, 2019.
[9] H. Wang, F. Zhang, X. Xie, M. Guo, Dkn: Deep knowledge-aware network for news recommendation, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2018, pp. 1835–1844.
[10] L. Lei, Wavelet neural network prediction method of stock price trend based on rough set attribute reduction, Appl. Soft Comput. 62 (2018) 923–932.
[11] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–864.
[12] E. Palumbo, G. Rizzo, R. Troncy, E. Baralis, M. Osella, E. Ferro, Knowledge graph embeddings with node2vec for item recommendation, in: European Semantic Web Conference, Springer, 2018, pp. 117–120.
[13] P.H. Franses, H. Ghijsels, Additive outliers, GARCH and forecasting volatility, Int. J. Forecast. 15 (1) (1999) 1–9.
[14] N. Sarantis, Nonlinearities, cyclical behaviour and predictability in stock markets: international evidence, Int. J. Forecast. 17 (3) (2001) 459–482.
[15] M. Ture, I. Kurt, Comparison of four different time series methods to forecast hepatitis A virus infection, Expert Syst. Appl. 31 (1) (2006) 41–46.
[16] K. jae Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (1) (2003) 307–319.
[17] L. Shen, H.T. Loh, Applying rough sets to market timing decisions, Decis. Support Syst. 37 (4) (2004) 583–597.
[18] J.-Z. Wang, J.-J. Wang, Z.-G. Zhang, S.-P. Guo, Forecasting stock indices with back propagation neural network, Expert Syst. Appl. 38 (11) (2011) 14346–14355.
[19] J. Patel, S. Shah, P. Thakkar, K. Kotecha, Predicting stock market index using fusion of machine learning techniques, Expert Syst. Appl. 42 (4) (2015) 2162–2172.
[20] X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.
[21] K. Chen, Y. Zhou, F. Dai, A LSTM-based method for stock returns prediction: A case study of China stock market, in: 2015 IEEE International Conference on Big Data, Big Data, IEEE, 2015, pp. 2823–2824.
[22] L. Di Persio, O. Honchar, Recurrent neural networks approach to the financial forecast of Google assets, Int. J. Math. Comput. Simul. 11 (2017).
[23] F. Zhou, H. min Zhou, Z. Yang, L. Yang, EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction, Expert Syst. Appl. 115 (2019) 136–151.
[24] R. Bisoi, P. Dash, A. Parida, Hybrid Variational Mode Decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis, Appl. Soft Comput. 74 (2019) 652–678.
[25] R. Ramezanian, A. Peymanfar, S.B. Ebrahimi, An integrated framework of genetic network programming and multi-layer perceptron neural network for prediction of daily stock return: An application in Tehran stock exchange market, Appl. Soft Comput. (2019) 105551.
[26] Z. Zhao, R. Rao, S. Tu, J. Shi, Time-weighted LSTM model with redefined labeling for stock trend prediction, in: 2017 IEEE 29th International Conference on Tools with Artificial Intelligence, ICTAI, IEEE, 2017, pp. 1210–1217.

[27] Y. Kim, Convolutional neural networks for sentence classification, 2014, arXiv preprint arXiv:1408.5882.
[28] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Netw. 18 (5) (2005) 602–610.
[29] Z. Lin, M. Feng, C.N.d. Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding, 2017, arXiv preprint arXiv:1703.03130.
[30] K.H. Brodersen, S.O. Cheng, K.E. Stephan, J.M. Buhmann, The Balanced accuracy and its posterior distribution, in: International Conference on Pattern Recognition, 2010.
[31] J. Zhang, S. Cui, Y. Xu, Q. Li, T. Li, A novel data-driven stock price trend prediction system, Expert Syst. Appl. 97 (2018) 60–69.
[32] S. Yao, L. Luo, H. Peng, High-frequency stock trend forecast using LSTM model, in: 2018 13th International Conference on Computer Science & Education, ICCSE, IEEE, 2018, pp. 1–4.