Information Sciences 556 (2021) 67–94

Contents lists available at ScienceDirect

Information Sciences

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

A novel graph convolutional feature based convolutional neural network for stock trend prediction

Wei Chen [a,*], Manrui Jiang [a], Wei-Guo Zhang [b], Zhensong Chen [a]

[a] School of Management and Engineering, Capital University of Economics and Business, Beijing, China
[b] School of Business Administration, South China University of Technology, Guangzhou, China

ARTICLE INFO

Article history:
Received 22 September 2020
Received in revised form 29 November 2020
Accepted 23 December 2020
Available online 29 December 2020

Keywords:
Stock trend prediction
Graph convolutional network
Convolutional neural network
Stock market information
Technical indicators

ABSTRACT

Stock trend prediction is one of the most widely investigated and challenging problems for investors and researchers. Since the convolutional neural network (CNN) was introduced to analyze financial data, many researchers have dedicated to predicting stock trend by transforming stock market data into images. However, most of the existing studies just focused on individual stock information, and ignored stock market information, such as the existing correlations between stocks. In fact, the price volatility of a stock may be affected by those of other stocks, thus, taking the stock market information into the stock trend prediction can further improve the prediction performance. In this paper, we propose a novel method for stock trend prediction using graph convolutional feature based convolutional neural network (GC–CNN) model, in which both stock market information and individual stock information are considered. Specifically, an improved graph convolutional network (IGCN) and a Dual-CNN are designed to construct GC–CNN, which can simultaneously capture stock market features and individual stock features. Six randomly selected Chinese stocks are used to demonstrate the superior performance of the proposed GC–CNN based method. The experimental analysis demonstrates that the proposed GC–CNN based method outperforms several stock trend prediction methods and stock trading strategies.

1. Introduction

Stock market prediction is always a challenge due to the high-noise, dynamic, non-linear, non-parametric and chaotic properties of stock data, and has drawn a lot of attention from researchers over the past decade. Some studies aim to predict the future prices or profits of stocks with regression methods [5,12,34], while other studies aim to predict the future trends of stock price movements with classification methods [4,1]. However, to make a profit in the stock market, investors and for-profit organizations are more concerned with the future trends of stock.

In recent years, many methods have been developed to deal with stock trend prediction problem. Initially conventional statistical methods were used. Subsequently, with the booming of artificial intelligence technology, machine learning techniques have been introduced to handle complex financial market data and proved to be useful for making stock trend predictions [30]. One type of machine learning method commonly used in stock trend prediction is neural network, which can find the non-linear and non-additive relations in data and can obtain better prediction results [9]. However, at the earlier stage, the traditional neural network has a shallow structure and cannot effectively capture data feature representation,

* Corresponding author.
E-mail address: chenwei@cueb.edu.cn (W. Chen).

https://doi.org/10.1016/j.ins.2020.12.068
0020-0255/

thereby affecting the prediction accuracy. To solve this problem, a particular type of machine learning named deep learning arises at the historic moment. Nowadays, deep learning methods, such as restricted Boltzmann machine (RBM) [18], recurrent neural network (RNN) [38], long short-term memory network (LSTM) [23,6] and convolutional neural network (CNN) [28,2], have shown great performances in predicting stock trend.

Among the above mentioned deep learning methods, CNN is a multilayer neural network structure simulating the operation mechanism of a biological vision system. Because of the great performance on extracting multi-scale localized spatial features, CNN has been used for image classification and image recognition problems. In recent years, many researchers dedicated to transforming stock market data into images, and then predicting future trends of stock by using CNN. However, to the best of our knowledge, most of the existing image transformation methods are only based on the individual stock information, such as trading data and technical indicators of target stock, while the stock market information is ignored, such as the existing correlations between stocks. In fact, in stock market, the price volatility of a stock may be affected by those of other stocks. Therefore, taking the stock market information into the stock trend prediction can further improve the prediction performance.

In this paper, we propose a novel method for stock trend prediction using graph convolutional feature based convolutional neural network (GC–CNN) model, in which both stock market information and individual stock information are considered. Inspired by the financial complex network, we construct stock market networks and the corresponding feature matrices to represent stock market information based on the correlations between stocks and the characteristic of each stock, respectively. Meanwhile, we analyze the target stock based on trading data and technical indicators to obtain the individual stock information. After that, both stock market information and individual stock information are transformed into images. Furthermore, an improved graph convolutional network (IGCN) and a Dual-CNN are designed to construct the GC–CNN model for stock trend prediction, which can simultaneously capture stock market features and individual stock features. Finally, several experiments are conducted based on six randomly selected Chinese stocks to demonstrate the superior performance of the proposed GC–CNN based method.

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

- We consider both stock market information and individual stock information for stock trend prediction. Specially, we construct stock market networks and the corresponding feature matrices to represent stock market information.
- We propose an improved GCN (IGCN) to capture stock market features based on stock market information which contains topological structure data. Moreover, we design a Dual-CNN to capture individual stock features based on individual stock information.
- We propose a graph convolutional feature based convolutional neural network (GC–CNN) model to predict stock trend by combining IGCN and Dual-CNN, in which the stock market features and individual stock features are merged into joint features.
- To demonstrate the performance of the proposed GC–CNN based method, the experimental results are evaluated from two aspects: computational performance evaluation and financial evaluation. In computational performance evaluation, we compare the proposed GC–CNN based method with several trend prediction methods. In financial evaluation, we simulated stock trading based on different predictions and several common stock trading strategies.

The structure of the paper is organized as follows. The related work is presented in Section 2. The details of the proposed method are described in Section 3. Then, we outline the experimental settings and analyze the performance evaluation of the proposed method in Section 4. Finally, the conclusion of this paper is outlined in Section 5.

2. Related work

2.1. Technical indicators

Technical indicators are the most commonly used indicators for stock trend prediction, which can summarize the behavior or trends in the time series. Rate of change (ROC) is a simple technical indicator that measures the percentage change in price over a period of time. Additionally, moving average is a most commonly used type of indicator which describes the moving average of the prices for a given period, including simple moving average (SMA), exponential moving average (EMA), weight moving average (WMA), and so on. Especially, some of the moving averages, such as EMA and WMA, give the latest data more weight while calculating. Another commonly used type of technical indicator is the oscillator, including momentum index (MTM), moving average convergence divergence (MACD), relative strength index (RSI), commodity channel index (CCI), Williams indicator (WR) and stochastic oscillators (KDJ). MTM shows the difference between hte current price and the previous price. MACD shows the differences between two EMA and is composed of DIF, DEA and MACD histogram. RSI measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock. CCI is used to assess price trend direction and strength. WR and KDJ are used to determine the overbought and oversold region based on closing price, highest price and lowest price over a period of time.

As presented in Table 1, most of the studies used a combination of several technical indicators to predict stock trend.

Table 1
Technical indicators for stock trend prediction.

Author	ROC	SMA	EMA	WMA	MTM	MACD	RSI	CCI	WR	KDJ	Other
Zhang et al. [36]		✓			✓	✓	✓			✓	✓
Chen et al. [7]	✓	✓					✓				✓
Marković et al. [20]	✓		✓			✓	✓	✓		✓	✓
Sezer et al. [28]	✓	✓	✓	✓		✓	✓	✓	✓		✓
Lei et al. [16]					✓		✓		✓	✓	✓
Picasso et al. [27]		✓	✓			✓	✓		✓	✓	✓
Tang et al. [30]							✓			✓	✓
Bisoi et al. [3]		✓				✓	✓		✓	✓	
Ananthi and Vijayakumar [1]			✓			✓	✓				✓
Zhao et al. [38]		✓				✓	✓	✓	✓	✓	

2.2. Machine learning methods

For stock trend prediction, traditional machine learning methods are quite popular. Especially, support vector machines (SVM), artificial neural networks (ANN), as well as hybrid methods are among the preferred choices. Lee [15] developed a trend prediction method by combining the SVM model and a hybrid feature selection method. Chiang et al. [8] combined particle swarm optimization (PSO) with the neural network to predict stock trend. Zhang et al. [36] constructed an ensemble method integrated with the AdaBoost algorithm, probabilistic SVM and genetic algorithm (GA) to predict stock trend. Marković et al. [20] proposed a hybrid method by combining the analytic hierarchy process and weighted kernel least squares support vector machines for stock trend prediction. Lei [16] proposed an integrated trend prediction method based on rough set and wavelet neural network (WNN). Bisoi et al. [3] proposed a hybrid stock trend prediction method, in which variational mode decomposition, extreme learning machine and differential evolution are combined. Ozorhan et al. [25] used a modified Zigzag technical indicator to discover motifs of stock trends and adopted SVM to classify the motifs. Picasso et al. [27] applied random forest (RF), SVM and ANN to predict stock trend after improving the balancing technique. Parray et al. [26] used three machine learning methods, i.e., SVM, ANN and logistic regression (LS), for predicting the next day trend of the stocks.

With increasing computational intelligence capacity, more new methods started appearing. In the following subsection, the deep learning methods which are used for stock trend prediction in literature will be mentioned.

2.3. Deep learning methods

Deep learning is a particular type of machine learning that consists of multiple layers of different contributions. Due to the provision of high-level abstraction for data modeling, deep learning methods outperform than its shallow counterparts [28,24].

In literatures, different deep learning methods are applied to predict stock trend, such as RNN, RBM, LSTM and CNN. Nelson et al. [23] adopted the LSTM network to predict stock future trends based on the price history and technical indicators. Liang et al. [18] proposed a hybrid method by combining RBM and several classifiers to predict short-term stock market trend. Chen et al. [6] analyzed text data to obtain public mood and emotion and then used the LSTM network to predict stock trend. Moews et al. [21] proposed deep feed-forward neural networks combined with exponential smoothing for financial time series trend prediction. Zhao et al. [38] introduced the attention mechanism and proposed stock trend prediction methods based on RNN, LSTM and gated recurrent unit (GRU), respectively. Naik and Mohan [22] proposed a method based on the deep neural network (DNN) to identify stock movement trend, in which candlestick data and technical indicator values are both considered. Wu et al. [32] discussed the effect of different labeling methods on prediction accuracy, and applied different methods to predict stock trend, including LSTM and GRU.

Among the above mentioned deep learning methods, CNN is recently introduced to solve the stock trend prediction problem and achieves great success. Because CNN cannot be used for financial time series data analysis directly, many researchers transform stock data into images, and then extract multi-scale localized spatial features using CNN to predict stock trend. For example, Sezer and Ozbayoglu [28] converted stock technical indicators into 2-D images and proposed a novel algorithmic trading model based on CNN to predict stock trend. Wen et al. [31] simplified noisy-filled financial temporal series via sequence reconstruction by leveraging motifs, and then utilized CNN to capture spatial structure of time series. Sezer and Ozbayoglu [29] proposed a 2-D convolutional neural network to predict stock trend by only using the 2-D stock bar chart images. Barra et al. [2] generated Gramian angular fields images from time series related to the Standard & Poor's 500 index future and exploited an ensemble of CNNs to predict the future trend of the U.S. market. Long et al. [19] constructed three matrices, i.e., transaction number matrix, buying volume matrix and selling volume matrix, to represent trading behavior pattern, and then utilized CNN to extract deep features. Hao and Gao [10] proposed a hybrid method in which the characteristics of different time scales of price series were captured through different layers of CNN.

However, CNN can only handle the data defined on regular grids and cannot be used to capture the complex topological structure [40,35]. Attempting to overcome these shortages, a novel deep learning method, named graph convolutional network (GCN), was proposed by Kipf and Welling [13]. The GCN methods perform convolution over graphs by making operations between the filters and spatial neighbors of graph nodes. Because of the superior performance, GCN methods have been successfully applied to many fields, such as brain magnetic resonance imaging classification [14], bot detection [37], building pattern classification [33], traffic forecasting [11], and gesture recognition [17]. However, to the best of authors' knowledge, GCN methods have rarely been applied to deal with stock trend prediction challenge.

3. Method

In this paper, we propose a novel method using graph convolutional feature based convolutional neural network (GC–CNN) model to predict stock trend, in which both stock market information and individual stock information are considered. As can be seen in Fig. 1, the proposed GC–CNN based method is divided into three main steps: relevant stocks discovery, image creation and trend prediction.

3.1. Relevant stocks discovery

Note that, when only the individual stock information is used to predict stock trend, the impact of stock market changes on the target stock cannot be considered. Thus, it is necessary to consider stock market information to better represent the effect of the complicated market on the target stock.

However, because of the huge size of stock market, it is unpractical to consider all the stocks to represent stock market. Therefore, in this paper, we just select several relevant stocks to represent the whole market. Considering that the price volatility of a stock may be affected by those of other stocks, we select the relevant stocks according to the relations between the target stock and the other stocks. In this paper, as in [39], Spearman rank-order correlation is used to measure the relations between stocks.

First, the logarithm return series $R_i = \{r_i(t)\}$ of stock i is defined as:

$$r_i(t) = ln(c_{i,t}) - ln(c_{i,t-1}), \tag{1}$$

where $c_{i,t}$ is the closing price of stock i at time t. Then, for stocks i and j, R_i and R_j are replaced with their rank data $Rank_i = \{Rank_i(t)\}$ and $Rank_j = \{Rank_j(t)\}$, respectively. After that, the Spearman rank-order correlation coefficient between i and j can be computed as follows:

$$\rho_{i,j} = \sum_{t=t_i-n}^{t_i}\left(Rank_i(t) - \overline{Rank_i}\right)\left(\frac{Rank_j(t) - \overline{Rank_j})}{\sqrt{\sum_{t=t_i-n}^{t_i}\left(Rank_i(t) - \overline{Rank_i}\right)^2}\sqrt{\sum_{t=t_j-n}^{t_j}\left(Rank_j(t) - \overline{Rank_j}\right)^2}}, \tag{2}$$

where n is the length of $Rank_i$ and $Rank_i$. Additionally, considering the time lag between stocks i and j, we let $t_i = t_j + l$, where l is the lag period.

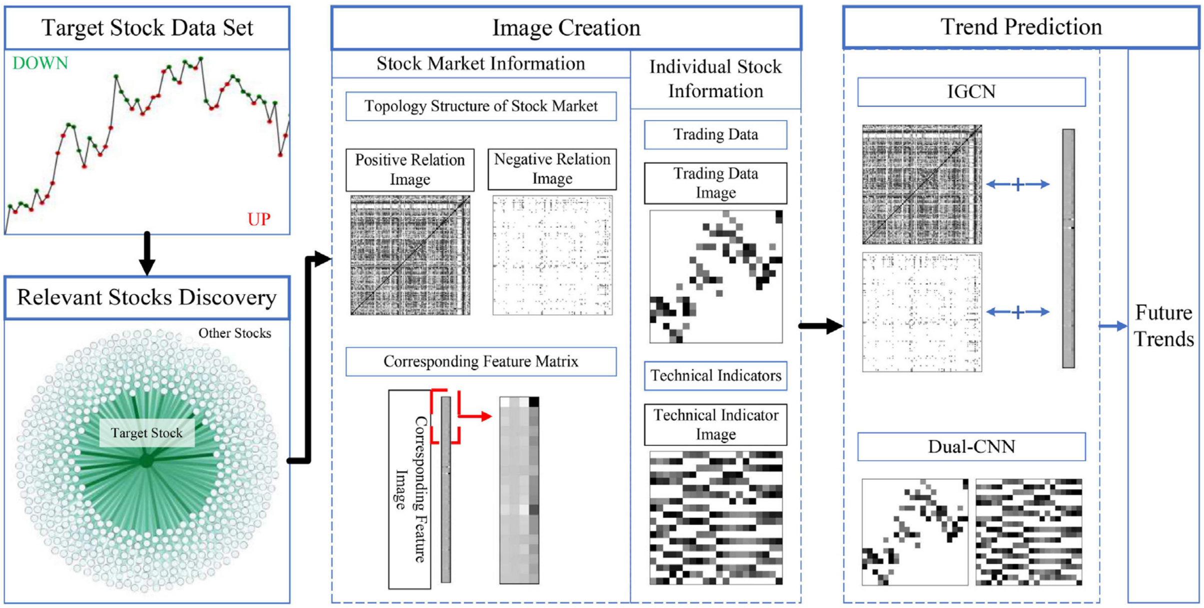


Fig. 1. The layout of the proposed GC–CNN based method for stock trend prediction.

Furthermore, all the stocks are grouped by industry attributes. Then, the stocks of each group are ranked from largest to smallest according to the relations $|\rho_{S_{tar},j}|$ between target stock $\rho_{S_{tar}}$ and other stocks j. Finally, the top m stocks of each group are selected to represent the whole market.

3.2. Image creation

In the image creation phase, the stock market information and the individual stock information are respectively transformed into images.

3.2.1. Stock market information

The stock market information is composed of two parts. One is the topology structure of stock market, and the other is the characteristic of each stock in the market. Based on the selected relevant stocks, we construct stock networks to capture the topology structures of stock market, and design the corresponding feature matrices to represent the characteristic of each stock in the market. The details about generating images are presented below.

Part one. Fig. 2 shows the procedures of generating the positive and negative relation images. We first sort the selected relevant stocks by industry attributes. Then, we construct stock networks by considering the stocks as nodes and the relations between stocks as edges. Specifically, the relations between stocks are measured by Eq. (2). Due to the consideration of time lag l, the adjacent matrix of stock network for predicting target stock trend at time t can be expressed as:

$$A_M^{t,l} = \begin{bmatrix} a_{11}^{t,l} & a_{12}^{t,l} & \cdots & a_{1N_s}^{t,l} \\ a_{21}^{t,l} & a_{22}^{t,l} & \cdots & a_{2N_s}^{t,l} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_s1}^{t,l} & a_{N_s2}^{t,l} & \cdots & a_{N_sN_s}^{t,l} \end{bmatrix}, \tag{3}$$

where N_s denotes the number of selected relevant stocks and $a_{ij}^{t,l} = \rho_{i,j}$. Specially, we let $t_i = t_j = t - l$ when computing $\rho_{i,j}$.

Furthermore, the positive and negative relations are separated from the adjacent matrix $A_M^{t,l}$ to obtain matrices $A_{M,+}^{t,l}$ and $A_{M,-}^{t,l}$, respectively. Meanwhile, a threshold θ is set to simplify both positive and negative relations. Thus, $a_{ij,+}^{t,l}$, the elements of matrix $A_{M,+}^{t,l}$, are satisfied with Eq. (4), and $a_{ij,-}^{t,l}$, the elements of matrix $A_{M,-}^{t,l}$, are satisfied with Eq. (5).

$$a_{ij,+}^{t,l} = \begin{cases} a_{ij}^{t,l}, & a_{ij}^{t,l} \geqslant \theta, \\ 0, & otherwise. \end{cases} \tag{4}$$

$$a_{ij,-}^{t,l} = \begin{cases} |a_{ij}^{t,l}|, & a_{ij}^{t,l} < -\theta, \\ 0, & otherwise. \end{cases} \tag{5}$$

Then, under different t and l, several $N_s \times N_s$ positive and negative relation images are generated based on $A_{M,+}^{t,l}$ and $A_{M,-}^{t,l}$, respectively.

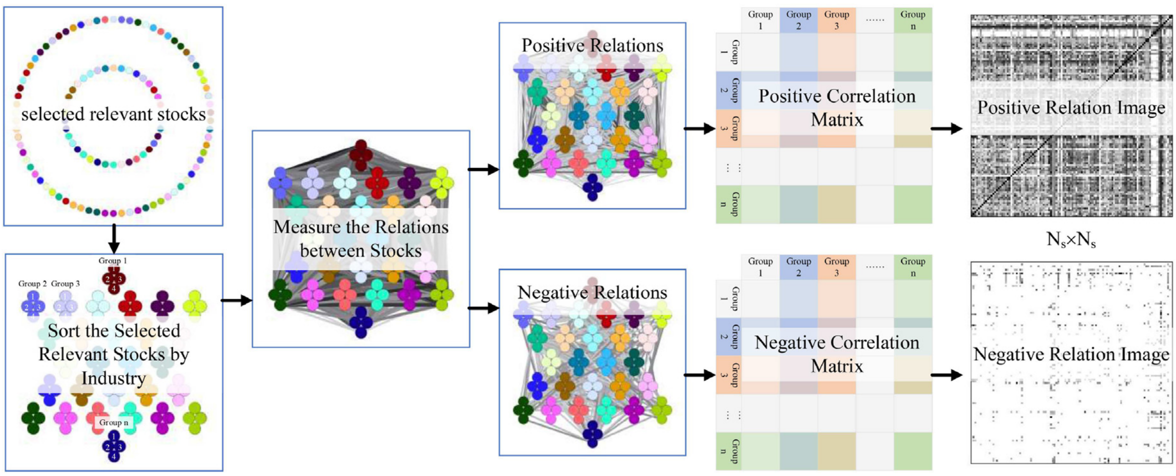


Fig. 2. The procedures of generating the positive and negative relation images.

Part two. The corresponding feature matrices are constructed by considering the features of selected relevant stocks. Fig. 3 shows the procedures of generating the corresponding feature image. We select 4 features, i.e., the relations between target stock and selected relevant stocks, the latest 1-day ROC, the average 1-day ROC in recent n days and the standard deviation of 1-day ROC in recent n days, to establish the corresponding feature matrices. The formulas of the selected features are summarized in A. For each pair of $A_{M,+}^{t,l}$ and $A_{M,-}^{t,l}$, we construct a corresponding feature matrix $M_f^{t,l} \in \mathbb{R}^{N_s \times 4}$, in which 4 features of each stock in $A_M^{t,l}$ are arranged in a row. Then, $M_f^{t,l}$ is normalized as following,

$$M_F^{t,l} = D_F^{t,l} M_f^{t,l}, \tag{6}$$

where, $D_F^{t,l} \in \mathbb{R}^{N_s \times N_s}$ is diagonal degree matrix with $D_{F,ii}^{t,l} = \left(\sum_j M_{f,ij}^{t,l}\right)^{-1}$. Finally, under different t and l, several $N_s \times 4$ corresponding feature images are obtained based on $M_F^{t,l}$.

3.2.2. Individual stock information

The individual stock information is also composed of two parts. The first part consists of trading data, and the second part consists of technical indicators. The main steps of generating images are presented in the following.

First part. Highest price series $H_{tar} = \{h_{tar,t}\}$, opening price series $O_{tar} = \{o_{tar,t}\}$, closing price series $C_{tar} = \{c_{tar,t}\}$ and lowest price series $L_{tar} = \{l_{tar,t}\}$ are transformed into four-channel trading data images according to following steps.

Step 1 To predict future trend of target stock at time t, capture subseries $H'_{tar} = \{h_{tar,t-n+1}, \ldots, h_{tar,t}\}, O'_{tar} = \{o_{tar,t-n+1}, \ldots, o_{tar,t}\}, C'_{tar} = \{c_{tar,t-n+1}, \ldots, c_{tar,t}\}$ and $L'_{tar} = \{l_{tar,t-n+1}, \ldots, l_{tar,t}\}$ of length n from $H_{tar}, O_{tar}, C_{tar}$ and L_{tar}, respectively.

Step 2 Normalize all the series according to Eq. (7), where $z \in [t-n-1, t], X_{tar} = \{H'_{tar}, O'_{tar}, C'_{tar}, L'_{tar}\}$ and $x_{tar,z} \in X_{tar}$.

$$x'_{tar,z} = n * \frac{x_{tar,z} - \min\{X_{tar}\}}{\max\{X_{tar}\} - \min\{X_{tar}\}}, \tag{7}$$

Step 3 Convert $x'_{tar,z}$ to coordinates (u, v), where $u = t - z - 1 + n$ and v is obtained by Eq. (8).

$$v = \begin{cases} \left\lceil x'_{tar,z} \right\rceil, & \text{if } x'_{tar,z} < n, \\ \left\lceil x'_{tar,z} \right\rceil - 1, & \text{if } x'_{tar,z} = n. \end{cases} \tag{8}$$

Then $x''_{tar,z}$ can be obtained by Eq. (9).

$$x''_{tar,z} = \begin{cases} x'_{tar,z} - \left\lceil x'_{tar,z} \right\rceil + 1 & \text{if } x'_{tar,z} < n, \\ 2 & \text{if } x'_{tar,z} = n. \end{cases} \tag{9}$$

Step 4 Based on coordinates (u, v) and $x''_{tar,z}$, subseries $H'_{tar}, O'_{tar}, C'_{tar}$ and L'_{tar} can be transformed into a four-channel trading data image, in which one subseries corresponds to one channel. An example is demonstrated in Fig. 4.

Second part. The series of technical indicators 1-day ROC, 5-day triple exponential moving average (TEMA), 5-day WMA, 5-day SMA, 5-day Hull moving average (HMA), 5-day EMA, 9-day CMO, 9-day WR, MACD histogram and 14-day CCI are transformed into single channel technical indicator images according to following steps, which are similar to the steps of the first part. The formulas of the technical indicators are summarized in A.

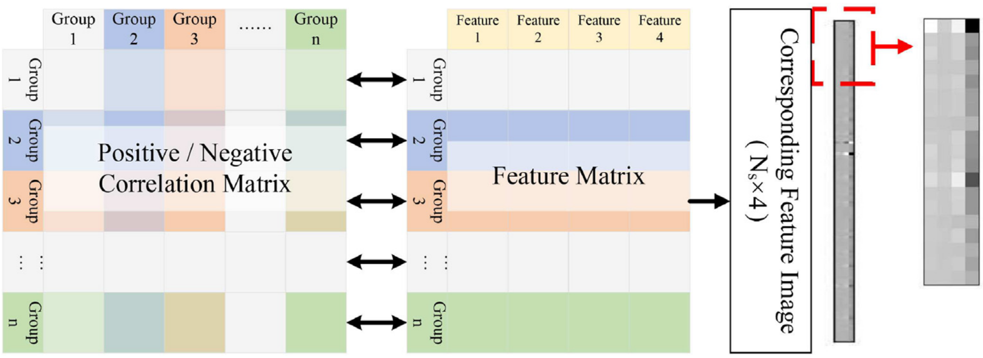


Fig. 3. The procedures of generating the corresponding feature image.

v 19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.63	0.00	1.20	0.00	0.00	1.20	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.09	0.00	0.00	0.00	0.00	0.00	0.00	1.91	1.61	0.00	0.00	1.09	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	1.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.03	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.92	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	1.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	1.64	1.85	0.00	0.00	0.00	1.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	1.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	1.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
u	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Normalized H'_{tar}	6.64	6.85	5.78	4.70	8.15	6.77	12.46	14.09	17.63	16.34	17.20	19.01	18.06	17.20	14.91	14.61	12.03	11.94	14.09	20.00
Coordinates	(0,6)	(1,6)	(2,5)	(3,4)	(4,8)	(5,6)	(6,12)	(7,14)	(8,17)	(9,16)	(10,17)	(11,19)	(12,18)	(13,17)	(14,14)	(15,14)	(16,12)	(17,11)	(18,14)	(19,19)

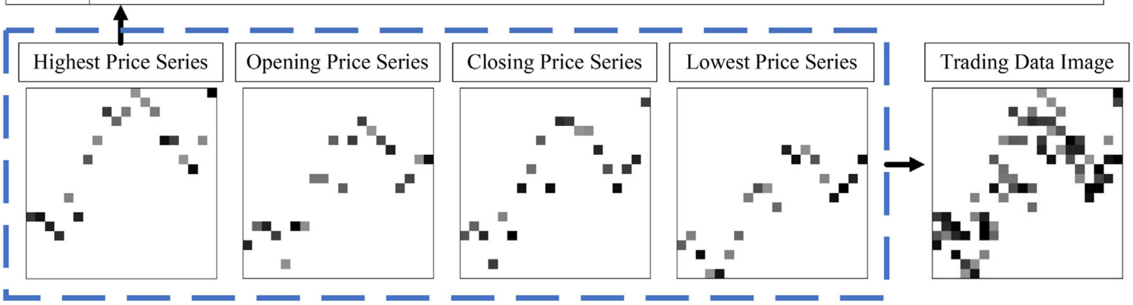


Fig. 4. An example for generating trading data image.

Step 1 To predict future trend of target stock at time t, capture subseries of length n from each original series of technical indicators.

Step 2 Normalize all the series according to Eq. (10), where $\alpha \in N^{+}$ is a parameter to limit the normalized values to $[0,\alpha], z \in [t-n-1,t], X_{tar}$ is the subseries of a technical indicator and $x_{tar,z} \in X_{tar}$.

$$x'_{tar,z} = \alpha * \frac{x_{tar,z} - \min\{X_{tar}\}}{\max\{X_{tar}\} - \min\{X_{tar}\}}, \tag{10}$$

Step 3 Convert $x'_{tar,z}$ to coordinates (u, v), where $u = t - z - 1 + n$ and v is obtained by Eq. (11).

$$v = \begin{cases} \lfloor x'_{tar,z} \rfloor, & \text{if } x'_{tar,z} < \alpha, \\ \lfloor x'_{tar,z} \rfloor - 1, & \text{if } x'_{tar,z} = \alpha. \end{cases} \tag{11}$$

Then $x''_{tar,z}$ can be obtained by Eq. (12).

$$x''_{tar,z} = \begin{cases} x'_{tar,z} - \lfloor x'_{tar,z} \rfloor + 1 & \text{if } x'_{tar,z} < \alpha \\ 2 & \text{if } x'_{tar,z} = \alpha \end{cases} \tag{12}$$

Step 4 Based on coordinates (u, v) and $x''_{tar,z}$, each subseries can be transformed into a $\alpha \times n$ single channel image. Then, by combining all the images vertically, a single channel technical indicator image $(10\alpha \times n)$ can be obtained. An example is demonstrated in Fig. 5.

3.3. Trend prediction

In the trend prediction phase, as can be seen in Fig. 6, we propose a graph convolutional feature based convolutional neural network (GC–CNN) model to predict future trends of target stock. The GC–CNN model consists of two parts: an improved GCN (IGCN) and a Dual-CNN, where stock market information is processed by IGCN and individual stock information is processed by Dual-CNN. The details about IGCN and Dual-CNN are presented as follows.

v	1	0.000	0.000	0.000	0.000	0.000	1.254	1.416	1.482	1.215	0.000	0.000	0.000	0.000	1.083	0.000	1.297	1.668	1.856	2.000	1.490
	0	1.265	1.000	1.208	1.765	1.869	0.000	0.000	0.000	0.000	1.940	1.842	1.547	1.711	0.000	1.918	0.000	0.000	0.000	0.000	0.000
u		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Normalized subseries		0.265	0.000	0.208	0.765	0.869	1.254	1.416	1.482	1.215	0.940	0.842	0.547	0.711	1.083	0.918	1.297	1.668	1.856	2.000	1.490
Coordinates		(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,1)	(6,1)	(7,1)	(8,1)	(9,0)	(10,0)	(11,0)	(12,0)	(13,1)	(14,0)	(15,1)	(16,1)	(17,1)	(18,1)	(19,1)

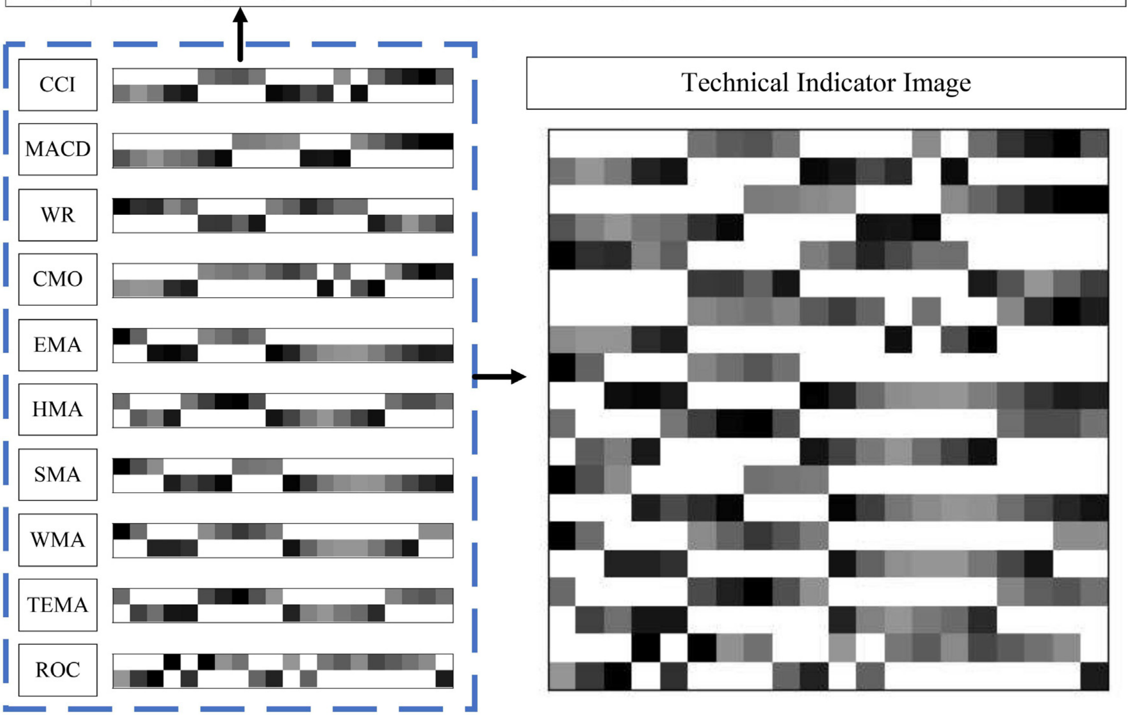


Fig. 5. An example for generating technical indicator image.

3.3.1. IGCN

In the proposed IGCN, as can be seen in Fig. 7, there are two steps to obtain stock market features based on the positive and negative relation images, and the corresponding feature images.

In step one, the stock market information is integrated according to the positive and negative relation images and the corresponding feature images. First of all, based on the positive and negative relation images, two Laplacian matrixes $L_+ \in \mathbb{R}^{N_s \times N_s}$ and $L_- \in \mathbb{R}^{N_s \times N_s}$ are respectively defined as

$$L_+ = D_+^{-1/2} A_{M,+}^{t,l} D_+^{-1/2}, \tag{13}$$

and

$$L_- = D_-^{-1/2} A_{M,-}^{t,l} D_-^{-1/2}, \tag{14}$$

where $D_+ \in \mathbb{R}^{N_s \times N_s}$ and $D_- \in \mathbb{R}^{N_s \times N_s}$ are diagonal degree matrixes with $D_{+,ii} = \sum_j A_{M,+,ij}^{t,l}$ and $D_{-,ii} = \sum_j A_{M,-,ij}^{t,l}$, respectively.

Then, L_+ and L_- are merged into $L_{+,-} \in \mathbb{R}^{N_s \times N_s \times 2}$. After that, according to industry attributes, $L_{+,-}$ goes through a convolutional layer to obtain $L'_{+,-} \in \mathbb{R}^{\frac{N_s}{m} \times \frac{N_s}{m} \times N_1}$, in which the number of $m \times m$ filters is set to N_1 and the stride is set to $m \times m$. Suppose H^h is the input matrix, W^h is the filter, b is the bias parameter and $\sigma(\cdot)$ is an activation function, the formula of convolutional operation is shown as following:

$$H^{h+1} = \sigma\left(\left(H^h * W^h\right)(i,j) + b\right). \tag{15}$$

In order to maintain the original correspondence between the positive (negative) relation images and the corresponding feature images, $M_F'^{t,l} \in \mathbb{R}^{\frac{N_s}{m} \times \frac{N_s}{m} \times 1}$ is obtained based on the corresponding feature image by convolution operation, in which the number of $m \times m$ filters is set to 1, and the stride is set to $m \times 1$.

Furthermore, $L'_{+,-}$ and $M_F'^{t,l}$ go through a graph convolution layer, and the outputs can be expressed as:

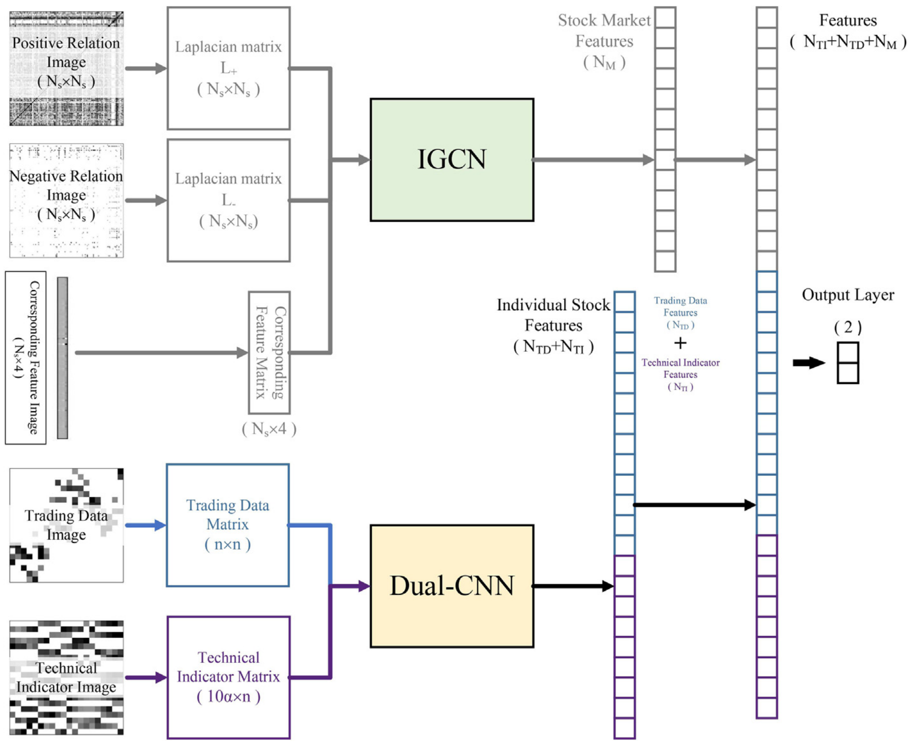


Fig. 6. The framework for proposed GC–CNN model.

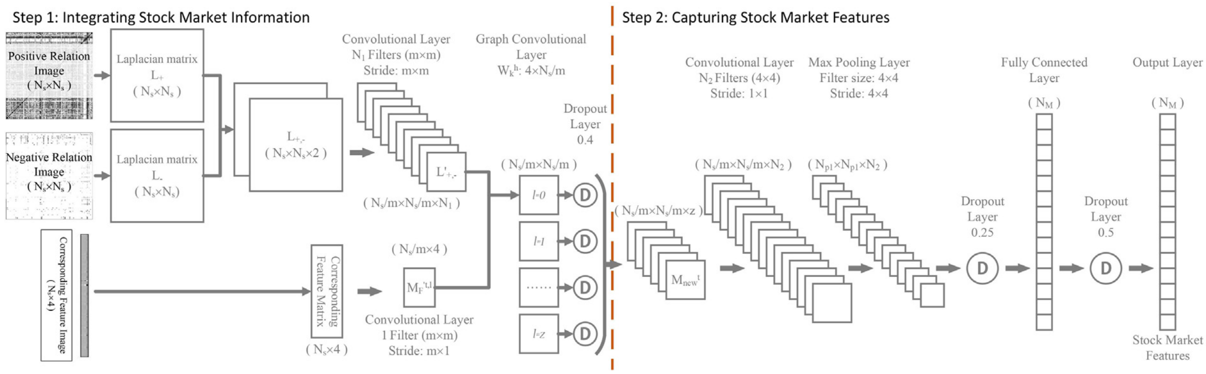


Fig. 7. The framework for proposed IGCN.

$$H^{h+1} = \sigma\left(\sum_{k=1}^{N_1} l'_k M_F'^{t,l} \Theta_k^h w_k^h + b\right), \tag{16}$$

where $l'_k \in \mathbb{R}^{\frac{N_s}{m}\times\frac{N_s}{m}}$ is the kth channel of $L'_{+,-}$, $\Theta_k^h \in \mathbb{R}^{4\times\frac{N_s}{m}}$ is a matrix of filter parameters, $w_k^h \in \mathbb{R}^{\frac{N_s}{m}\times\frac{N_s}{m}}$ is the kth layer-specific trainable weight matrix. Note that, the product of $\Theta_k^h \in \mathbb{R}^{4\times\frac{N_s}{m}}$ and $W_k^h \in \mathbb{R}^{\frac{N_s}{m}\times\frac{N_s}{m}}$ can be learned as one matrix $W_k^h \in \mathbb{R}^{4\times\frac{N_s}{m}}$, therefore, Eq. (16) can be simplified as:

$$H^{h+1} = \sigma\left(\sum_{k=1}^{N_1} l'_k M_F'^{t,l} W_k^h + b\right). \tag{17}$$

Moreover, under different time lag l, several $\frac{N_s}{m}\times\frac{N_s}{m}$ outputs can be obtained after above mentioned operations. The dropout layer (0.4) is added to prevent over-fitting.

Finally, we merge all the outputs to obtain a new matrix $M_{new}^t \in \mathbb{R}^{\frac{N_s}{m}\times\frac{N_s}{m}\times z}$, where z is the number of merged outputs.

In step two, we aim to capture stock market features based on matrix M_{new}^{t}. Six layers are used and listed as follows: convolutional layer, max pooling layer, two dropout layers (0.25, 0.5), fully connected layer and output layer.

For convolutional layer, N_2 filters (4×4) are used, and the stride is set to 1×1. Thus, after convolution operation, $\frac{N_s}{m} \times \frac{N_s}{m} \times N_2$ image can be obtained. For max pooling layer, the filter size and stride are both set to 4×4. Thus, after max pooling operation, $N_{p1} \times N_{p1} \times N_2$ image can be obtained, where $N_{p1} = \left[\frac{N_s}{4m}\right] + 1$. For fully connected layer, suppose H^h is the input matrix, W^h is the weight matrix and b is the bias parameter, the outputs can be expressed as:

$$H^{h+1} = \sigma\left(H^h * W^h + b\right). \tag{18}$$

Finally, the stock market features $F_M \in \mathbb{R}_{\mathbb{M}}^{\mathbb{N}}$ can be obtained.

3.3.2. Dual-CNN

In the designed Dual-CNN, as can be seen in Fig. 8, trading data images and technical indicator images are processed in parallel to obtain individual stock features, which consist of trading data features and technical indicator features.

For trading data images, input layer, two convolutional layers, two max pooling layers, two dropout layers (0.25, 0.5), fully connected layer and output layer are used. For convolutional layers and max pooling layers, different size of filters, i.e., $3 \times 3, 5 \times 5$ and 7×7, is adapted. Smaller size of filters can catch more details of the images. Therefore, in our study, during capturing trading data features, the filter sizes of convolutional layers and max pooling layers are set to 3×3. As can be seen in Fig. 8, after the first convolution operation and max pooling operation, the original trading data image $(n \times n)$ turns to $N_{p2} \times N_{p2} \times 8$ image, where $N_{p2} = n - 2$. Then, we continue the second convolution operation and max pooling operation, $N_{p3} \times N_{p3} \times 16$ image can be obtained, where $N_{p3} = \left[\frac{N_{p2}}{3}\right] + 1$. Finally, $N_{p3} \times N_{p3} \times 16$ image goes through the dropout layers (0.25, 0.5) and the fully connected layer to obtain the trading data features $F_{TD} \in \mathbb{R}_{\mathbb{TD}}^{\mathbb{N}}$.

For technical indicator image, the processes of capturing technical indicator features are similar to those of capturing trading data features. As can be seen in Fig. 8, after the first convolution operation and max pooling operation, the original technical indicator image $(10\alpha \times n)$ turns to $N_{p4} \times N_{p5} \times 4$ image, where $N_{p4} = 10\alpha - 2$ and $N_{p5} = n - 2$. Then, we continue the second convolution operation and max pooling operation, $N_{p6} \times N_{p7} \times 8$ image can be obtained, where $N_{p6} = \left[\frac{N_{p4}}{3}\right] + 1$ and $N_{p7} = \left[\frac{N_{p5}}{3}\right] + 1$. Finally, $N_{p6} \times N_{p7} \times 8$ image goes through the dropout layers (0.25, 0.5) and fully connected layer to obtain the technical indicator features $F_{TI} \in \mathbb{R}_{\mathbb{TI}}^{\mathbb{N}}$.

In the end, the individual stock features $F_T \in \mathbb{R}^{N_{TD}+N_{TI}}$ can be obtained by combining trading data features F_{TD} and technical indicator features F_{TI}.

3.3.3. GC–CNN

In this paper, we propose a graph convolutional feature based convolutional neural network (GC–CNN) model to predict stock trend by combining IGCN and Dual-CNN. After processing images by IGCN and Dual-CNN, features $F_M \in \mathbb{R}_{\mathbb{M}}^{\mathbb{N}}$ and

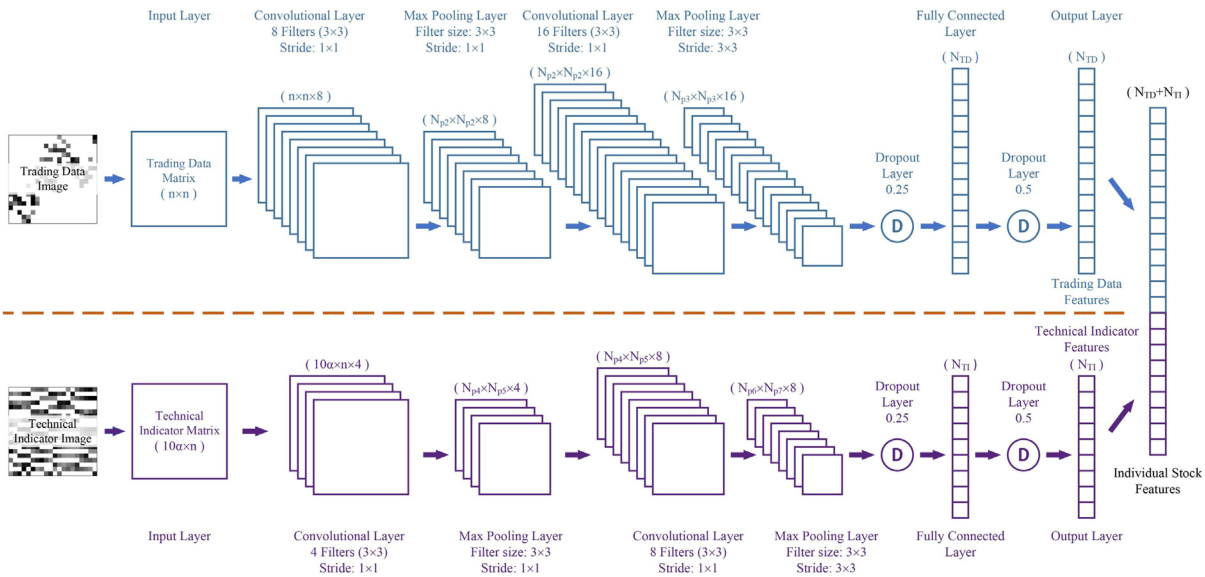


Fig. 8. The framework for designed Dual-CNN.

$F_T \in \mathbb{R}^{N_{TD}+N_{TI}}$ are merged into joint features $F_{ALL} \in \mathbb{R}^{N_M+N_{TD}+N_{TI}}$. Then, softmax function is used to get outputs. Suppose W^O is the weight matrix and b is the bias parameter, the outputs H^O can be expressed as:

$$H^O = softmax\left(F_{ALL} * W^O + b\right), \tag{19}$$

where $H^O \in \mathbb{R}^2$ and $W^O \in \mathbb{R}^{(N_M+N_{TD}+N_{TI})\times 2}$.

By using softmax function, the probabilities of classes 0 and 1 can be obtained. Then, the final results are expressed as:

$$Label_t = \begin{cases} 0, & P_{0,t} > 0.6, \\ 1, & P_{1,t} > 0.6, \\ 2, & \text{otherwise}, \end{cases} \tag{20}$$

where $P_{0,t}$ is the probability of class 0 at time t, $P_{1,t}$ is the probability of class 1 at time t, and class 2 represents the uncertain results.

4. Performance evaluation

The performance of the proposed GC–CNN based method for stock trend prediction was evaluated from two aspects: computational performance evaluation and financial evaluation. In computational performance evaluation, the proposed GC–CNN based method was compared with several trend prediction methods. In financial evaluation, we simulated stock trading based on different predictions and some common stock trading strategies.

The methods are implemented by using TensorFlow and Sklearn. Then, the experiments are conducted by using Python 3.7 and run on a computing system with an Intel i7-8650U and 16 GB RAM.

4.1. Data gathering and preparation

The stocks for our experiments are from the Chinese stock market. To preprocess the original data, the following two steps were used. First, we calculated the values of technical indicators which were used to generate images. Second, considering the missing data caused by suspension, we filled the missing data with zero to represent the suspended situation. According to a well-known industry classification method, i.e., Shenwan first-level industry classification, all the stocks are divided into 28 groups. Table 2 shows some basic attributes of target stocks which are randomly selected from the Chinese stock market.

For each target stock, the label is determined according to Eq. (21), where c_t is the closing price at time t, label 1 represents the up trend and label 0 represents the down trend.

$$L_t = \begin{cases} 1, & c_{t+1} > c_t, \\ 0, & \text{otherwise}. \end{cases} \tag{21}$$

In addition, the sliding window method (See Fig. 9) is used for training and testing. For each target stock, the training period is set to 5 years and the testing period is set to 1 year. Note that, stock 603808 doesn't have enough data for training and testing, thus, the data from 2015–07-01 to 2018–12-31 is used for training, and the data from 2019–01-01 to 2019–12-31 is used for testing.

4.2. Evaluation metrics

To compare the performances of different methods, several evaluation metrics are needed to be selected in advance. In computational performance evaluation, Accuracy, Precision, Recall and F-measure (F_1) are used. The corresponding formulas are listed in Eqs. (22)–(28):

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Ngative}}, \tag{22}$$

Table 2
Basic attributes of target stocks.

Code	Industry	Listing time	Experimental data segment	Trading days	Missing data
600809	Food & Beverage	1994-01-06	[2010-01-01,2020-01-01)	2431	16
300330	Computer	2012-06-19	[2013-01-01,2020-01-01)	1702	52
603808	Textile & Apparel	2015-04-22	[2015-07-01,2020-01-01)	1100	81
002580	Electrical Equipment	2011-05-06	[2012-01-01,2020-01-01)	1945	160
601318	Non-bank Financial	2007-03-01	[2010-01-01,2020-01-01)	2431	54
603123	Commerce	2012-05-03	[2013-01-01,2020-01-01)	1702	79

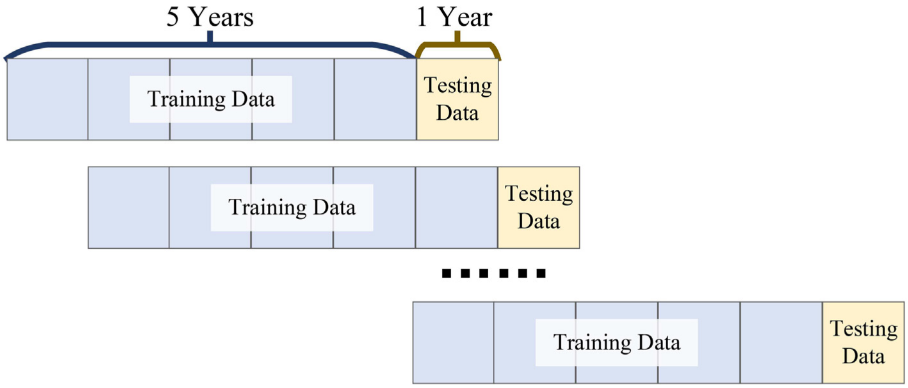


Fig. 9. The sliding window method for training and testing.

$$\text{Precision}_{pos} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}, \tag{23}$$

$$\text{Precision}_{neg} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Negative}}, \tag{24}$$

$$\text{Recall}_{pos} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Ngative}}, \tag{25}$$

$$\text{Recall}_{neg} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}, \tag{26}$$

$$F_{1,pos} = \frac{2 \times \text{Precision}_{pos} \times \text{Recall}_{pos}}{\text{Precision}_{pos} + \text{Recall}_{pos}}, \tag{27}$$

$$F_{1,neg} = \frac{2 \times \text{Precision}_{neg} \times \text{Recall}_{neg}}{\text{Precision}_{neg} + \text{Recall}_{neg}}. \tag{28}$$

In financial evaluation, we simulate stock trading and mainly focus on trading returns and risk. Thus, total money (TMoney), average annual return (AAR) and sharp ratio (SR) are used to evaluate simulated trading results. The corresponding formulas are listed in Eqs. (29)–(31), where annual risk free rate is 2.5%.

$$\text{TMoney} = \text{Available Capital} + \text{Closing Price} \times \text{Number of Stocks}, \tag{29}$$

$$\text{AAR} = \left[\left(\frac{\text{TMoney}}{\text{Start Money}} \right)^{1/\text{Number of Years}} - 1 \right] \times 100\%, \tag{30}$$

$$\text{SR} = \frac{\text{Rate of Return} - \text{Risk Free Rate}}{\text{Standard Deviation of Return}}. \tag{31}$$

It should be noted that, the results for the trading days filled with zero are ignored while calculating evaluation metrics.

4.3. Computational performance evaluation

4.3.1. Comparative methods

To show the computational performance of the proposed method, we compared the results of the proposed method with those obtained by other methods, including Dual-CNN, CNN-TA, SVM, MLP, DT. Each method was executed 20 times to reduce randomness.

(1) **Dual-CNN**. The Dual-CNN method is a part of the proposed GC–CNN based method, in which only the individual stock information are considered. Trading data images and technical indicator images are considered as inputs for Dual-CNN, and the parameters are set to the same as those in GC–CNN based method.
(2) **CNN-TA**. The CNN-TA method is proposed by [28], which contains an input layer, two convolutional layers, a max pooling layer, two dropout layers, a fully connected layer and an output layer. For each day, the image is generated by using 15 technical indicators and 15 different intervals of technical indicators.

(3) **CNN-LSTM**. The CNN-LSTM method is proposed by Hao and Gao [10], in which CNN is used to extracted features of different time scales, and LSTM is used to learn time dependencies in features. CNN-LSTM consists of three models with different structures. For the first model that contains a LSTM layer and a dropout layer, 40 days closing price data are considered as inputs. For the second model that contains a convolutional layer, a max pooling layer, a LSTM layer and a dropout layer, 40 days 5-day SMA data are considered as inputs. For the third model that contains two convolutional layers, two max pooling layers, a LSTM layer and a dropout layer, 40 days 10-day SMA data are considered as inputs. After combining outputs of three models, two fully connected layers are utilized to obtain the final results.
(4) **GAF-CNN**. The GAF-CNN method is proposed by Barra et al. [2], in which time series are transformed into Gramian angular fields (GAF) images, and then fed to ensemble CNNs for stock trend prediction. GAF-CNN is composed of 20 CNNs which have the same model structure with different weight initialization method. For each CNN, an input layer, five convolutional layers, three max pooling layers, two dropout layers and a fully connected layer are used. In the experiments, the time series of close-open with different intervals, i.e., 1 day, 2 days, 3 days and 5 days, are transformed into GAF images for stock trend prediction.
(5) **SVM**. The support vector machine (SVM) is one of the commonly used methods for stock trend prediction. In the experiments, the values of 1-day ROC, 5-day TEMA, 5-day WMA, 5-day SMA, 5-day HMA, 5-day EMA, 9-day CMO, 9-day WR, MACD histogram and 14-day CCI time t are used to predict stock trend at time $t+1$.
(6) **MLP**. The multi-layer perceptron (MLP) is one of the most widely used neural networks for stock trend prediction. In the experiments, the values of 1-day ROC, 5-day TEMA, 5-day WMA, 5-day SMA, 5-day HMA, 5-day EMA, 9-day CMO, 9-day WR, MACD histogram and 14-day CCI time t are used to predict stock trend at time $t+1$.
(7) **DT**. The decision tree (DT) is one of the basic methods to solve the classification problem. In the experiments, the values of 1-day ROC, 5-day TEMA, 5-day WMA, 5-day SMA, 5-day HMA, 5-day EMA, 9-day CMO, 9-day WR, MACD histogram and 14-day CCI time t are used to predict stock trend at time $t+1$.

4.3.2. Parameter setting

The parameters employed for the proposed GC–CNN based method and other comparative methods are presented in Table 3.

4.3.3. Comparison results of different methods

In this part, Table 4 shows the computation efficiencies of training different methods, and the comparison results of different methods for six target stocks are summarized in Table 5. In order to clearly observe the performance of the proposed method, we marked the best results in bold and also drew bar charts based on the results (See Fig. 10). It is observed that:

(1) The computation efficiencies of SVM, MLP and DT methods are high, but for GC–CNN, Dual-CNN, CNN-TA, CNN-LSTM and GAF-CNN methods, the computation efficiencies are low. Especially, the proposed GC–CNN based method and GAF-CNN method require longer training time.
(2) The proposed GC–CNN based method outperforms Dual-CNN in nearly all evaluation metrics. This indicates that, with the consideration of stock market information, the prediction performance can be improved.
(3) The proposed GC–CNN based method outperforms most comparison methods, but does not always stand out in all the evaluation metrics. It is clear that when predicting future trends of six target stocks, none of the methods always remain high values of Accuracy, Precision, Recall or F_1. In addition, we find that, although the same prediction method is used, the predictive performances may be quite different. For instance, by using the DT method, the Precision$_{pos}$ values of stocks 603808 and 600809 are 3.15% and 64.63%, respectively. By using the CNN-TA method, the Precision$_{pos}$ values of stocks 002580 and 600809 are 25.56% and 43.38%, respectively. These indicate that same features have different effects on different stocks, and the same method has different predictive ability for different stocks.
(4) The performance of the proposed GC–CNN based method is more stable than other comparative methods. As can be seen clearly from Fig. 10, unlike other comparative methods, GC–CNN based method consistently performs well for all the stocks in all evaluation metrics. That is to say, the computational performance of the proposed GC–CNN based method is more stable.

Additionally, for the proposed GC–CNN based method, Fig. 11 shows the drop-column importances of the individual stock features and the stock market features. It is clear that:

(1) The individual stock features and the stock market features do not necessarily contribute positively to the predicted results.
(2) The individual stock features have more contributions to the predicted results than the stock market features in most cases, especially for $F_{1,pos}$ and $F_{1,neg}$.
(3) The stock market features have positive contributions to the results in most cases.

Table 3
The parameter settings of the methods.

Methods	Parameters and Description	Value
GC–CNN	n: The length of series for relevant stocks discovery and image creation.	20
	m: The number of selected stocks for each group.	4
	N_s: The number of selected stocks.	112
	θ: A threshold to simplify relations in stock networks.	0.2
	l: Time lag.	0 and 1
	z	2
	N_1: The number of filters at convolutional layer in step 1 of GCN module.	For 600809: 6. For 300330: 4. For 603808: 4. For 002580: 12. For 601318: 8. For 603123: 3.
	N_2: The number of filters at convolutional layer in step 2 of GCN module.	8
	N_M: The length of stock market features.	For 600809: 16. For 300330: 1. For 603808: 16. For 002580: 3. For 601318: 64. For 603123: 32.
	N_{TD}: The length of trading data features.	32
	α: A parameter to limit the normalized values.	2
	N_{TI}: The length of technical indicator features.	16
	σ: Activation function.	ReLU
	Learning rate.	0.001
	Max iterations.	500
Dual-CNN	N_{TD}: The length of trading data features.	32
	α: A parameter to limit the normalized values.	2
	N_{TI}: The length of technical indicator features.	16
	σ: Activation function.	ReLU
	Learning rate.	0.001
	Max iterations.	500
CNN-TA	Model structure.	Input layer: 15×15 Convolutional layer: 32 filters (3×3) Convolutional layer: 64 filters (3×3) Max pooling layer: 2×2 Dropout layer: 0.25 Fully connected layer: 128 Dropout later: 0.5 Output layer: 2
	Activation function.	ReLU
	Learning rate.	0.001
	Max iterations.	200
CNN-LSTM	Model structure. (First model)	Input layer: 40 LSTM layer: 10 Dropout layer: 0.25 Output layer: 10
	Model structure. (Second model)	Input layer: 40 Convolutional layer: 10 filters (3×3) Max pooling layer: 2×2 LSTM layer: 10 Dropout layer: 0.25 Output layer: 10
	Model structure. (Third model)	Input layer: 40 Convolutional layer: 10 filters (3×3) Max pooling layer: 2×2 Convolutional layer: 20 filters (2×2) Max pooling layer: 2×2 LSTM layer: 10 Dropout layer: 0.25 Output layer: 10
	Model structure.	Input layer: 30 (Outputs 1 + Outputs 2 + Outputs 3) Fully connected layer: 10 Fully connected layer: 1
	Activation function.	Convolutional layers: LeakyReLU Fully connected layers: Sigmoid
	Learning rate.	Decreasing
	Max iterations.	80

Table 3 (*continued*)

Methods	Parameters and Description	Value
GAF-CNN	Model structure.	Input layer: $40 \times 40 \times 3$ Convolutional layer: 32 filters (3×3) Max pooling layer: 2×2 Dropout layer: 0.25 Convolutional layer: 64 filters (3×3) Convolutional layer: 64 filters (3×3) Max pooling layer: 2×2 Dropout layer: 0.25 Convolutional layer: 128 filters (3×3) Convolutional layer: 128 filters (3×3) Max pooling layer: 2×2 Fully connected layer: 1024 Output layer: 2
	Activation function.	ReLU
	Learning rate.	0.01
	Max iterations.	30
	Threshold for determine final results.	0.55
SVM	Kernel.	rbf
	Gamma	10
MLP	Model structure.	Input layer: 10 Hidden layers: 5,5 Output layer: 1
	Activation function.	ReLU
DT	Max depth.	4

Table 4
Training time comparison (in seconds).

Methods	Average Training Time
GC–CNN	1110.42
Dual-CNN	193.02
CNN-TA	47.01
CNN-LSTM	38.45
GAF-CNN	1470.07
SVM	0.01
MLP	0.91
DT	0.17

These results further indicate that (1) the individual stock features have important contributions to the predicted results, especially for metrics $F_{1,pos}$ and $F_{1,neg}$; (2) the prediction performance can be improved by considering the stock market features.

4.4. Financial evaluation

In financial evaluation, we simulated stock trading based on different predictions and some common stock trading strategies. The initial capital for financial evaluation is 50000 RMB.

According to the predictions of different methods, each stock is bought, sold or held. If the predicted label is 1, the stock is bought at the closing price of that day with all of the current available capital. If the predicted label is 0, the stock is sold at the closing price of that day. Otherwise, no action is taken at that day. It should be noted that, if the same label comes consecutively, only the first label is activated. Repeated labels are ignored until the label changes.

Additionally, some other commonly used stock trading strategies were compared with the proposed methods.

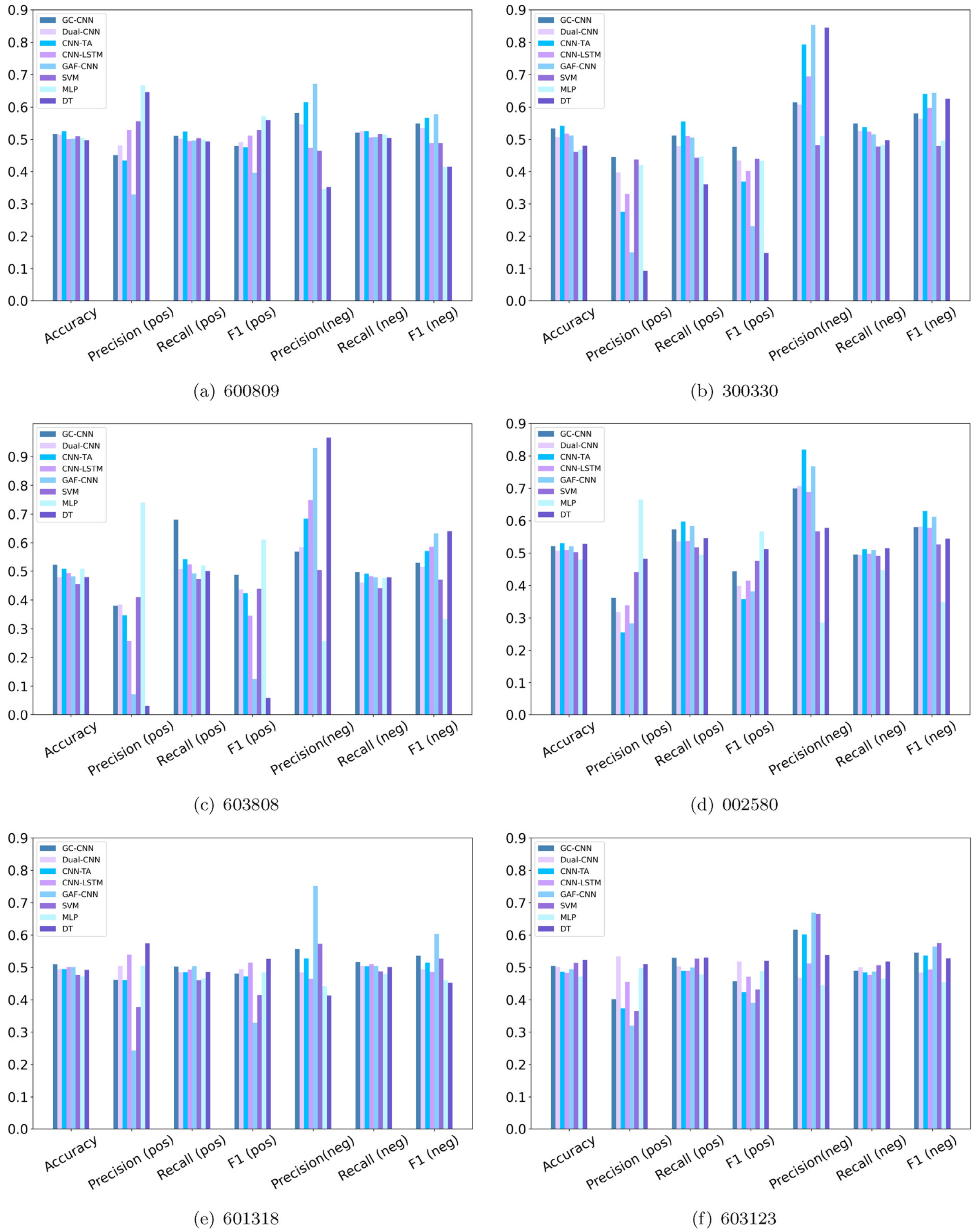
(1) **RSI signal**. RSI signal is a method for determining trading signals according to the value of 12-day relative strength index (RSI). The formula is given in A. 12-day RSI value is less than 30, the stock is bought at the closing price of that day. If 12-day RSI value is more than 70, the stock is sold at the closing price of that day. Otherwise, no action is taken at that day. Repeated labels are ignored until the label changes.
(2) **SMA short-term signal**. SMA short-term signal is a method for determining trading signals according to the value of 5-day SMA. If 5-day SMA value is less than the current closing price, the stock is bought at the closing price of that day. If 5-day SMA value is more than the current closing price, the stock is sold at the closing price of that day. Otherwise, no action is taken at that day. Repeated labels are ignored until the label changes.
(3) **SMA long-term signal**. SMA long-term signal is a method for determining trading signals according to the value of 20-day SMA. If 20-day SMA value is less than the current closing price, the stock is bought at the closing price of that day. If 20-day SMA value is more than the current closing price, the stock is sold at the closing price of that day. Otherwise, no action is taken at that day. Repeated labels are ignored until the label changes.

Table 5
Comparison results for computational performance evaluation.

Code	Methods	Accuracy	Precision_{pos}	Recall_{pos}	$F_{1,pos}$	Precision_{neg}	Recall_{neg}	$F_{1,neg}$	Average Rank
600809	GC–CNN	51.66% (2)	45.04% (6)	51.13% (2)	47.89% (6)	58.11% (3)	52.08% (3)	54.93% (3)	3.57
	Dual-CNN	51.45% (3)	48.09% (5)	50.22% (4)	49.13% (5)	54.66% (4)	52.53% (2)	53.57% (2)	3.86
	CNN-TA	**52.53% (1)**	43.38% (7)	**52.43% (1)**	47.48% (7)	61.48% (2)	**52.59% (1)**	**56.69% (1)**	**3.00**
	CNN-LSTM	50.04% (7)	52.87% (4)	49.53% (7)	51.14% (4)	47.27% (5)	50.62% (7)	48.89% (7)	5.57
	GAF-CNN	50.23% (6)	32.87% (8)	49.66% (6)	39.55% (8)	**67.15% (1)**	50.74% (6)	57.81% (6)	5.14
	SVM	50.95% (4)	55.61% (3)	50.38% (3)	52.87% (3)	46.39% (6)	51.64% (4)	48.88% (4)	4.14
	MLP	50.46% (5)	**66.67% (1)**	49.94% (5)	**57.10% (1)**	34.59% (8)	51.46% (5)	41.37% (5)	4.71
	DT	49.76% (8)	64.63% (2)	49.40% (8)	56.00% (2)	35.21% (7)	50.43% (8)	41.47% (8)	6.00
300330	GC–CNN	53.37% (2)	**44.50% (1)**	51.20% (2)	**47.62% (1)**	61.43% (5)	**54.89% (1)**	**57.98% (1)**	**2.43**
	Dual-CNN	50.74% (5)	39.69% (4)	47.83% (5)	43.38% (3)	60.75% (6)	52.63% (3)	56.40% (3)	4.57
	CNN-TA	**54.21% (1)**	27.54% (6)	**55.56% (1)**	36.83% (6)	79.28% (3)	53.78% (2)	64.09% (2)	3.00
	CNN-LSTM	51.81% (3)	33.07% (5)	51.08% (3)	40.15% (5)	69.42% (4)	52.40% (4)	59.72% (4)	4.00
	GAF-CNN	51.23% (4)	14.99% (7)	50.56% (4)	23.12% (7)	**85.35% (1)**	51.57% (5)	64.29% (5)	4.14
	SVM	46.00% (8)	43.64% (2)	44.21% (7)	43.92% (2)	48.21% (8)	47.64% (8)	47.92% (8)	6.14
	MLP	46.61% (7)	41.95% (3)	44.59% (6)	43.23% (4)	51.00% (7)	48.30% (7)	49.61% (7)	5.86
	DT	48.05% (6)	9.32% (8)	36.07% (8)	14.81% (8)	84.46% (2)	49.77% (6)	62.63% (6)	5.86
603808	GC–CNN	**52.20% (1)**	37.96% (4)	**56.94% (1)**	45.56% (2)	68.04% (5)	**49.62% (1)**	**57.39% (1)**	**2.57**
	Dual-CNN	47.87% (7)	38.38% (3)	50.67% (5)	43.68% (4)	58.43% (6)	46.02% (7)	51.49% (7)	5.43
	CNN-TA	50.82% (2)	34.65% (5)	54.32% (2)	42.31% (5)	68.38% (4)	49.08% (2)	57.14% (2)	3.57
	CNN-LSTM	49.28% (4)	25.75% (6)	52.46% (3)	34.54% (6)	74.83% (3)	48.21% (3)	58.64% (3)	4.00
	GAF-CNN	48.16% (5)	7.17% (7)	49.12% (7)	12.51% (7)	93.01% (2)	47.90% (4)	63.24% (4)	4.86
	SVM	45.49% (8)	40.94% (2)	47.27% (8)	43.88% (3)	50.43% (7)	44.03% (8)	47.01% (8)	6.14
	MLP	50.82% (2)	**74.02% (1)**	51.93% (4)	**61.04% (1)**	25.64% (8)	47.62% (6)	33.33% (6)	4.29
	DT	47.95% (6)	3.15% (8)	50.00% (6)	5.93% (8)	**96.58% (1)**	47.88% (5)	64.02% (5)	5.00
Code	Methods	Accuracy	Precision_{pos}	Recall_{pos}	$F_{1,pos}$	Precision_{neg}	Recall_{neg}	$F_{1,neg}$	Average Rank
002580	GC–CNN	52.13% (3)	36.15% (4)	57.32% (3)	44.34% (4)	69.96% (4)	49.54% (5)	58.01% (5)	3.86
	Dual-CNN	50.69% (6)	31.80% (6)	53.55% (6)	39.90% (6)	70.73% (3)	49.43% (6)	58.19% (6)	5.14
	CNN-TA	**53.03% (1)**	25.56% (8)	**59.70% (1)**	35.79% (8)	**81.88% (1)**	51.15% (2)	62.97% (2)	**3.14**
	CNN-LSTM	50.92% (5)	33.88% (5)	53.68% (5)	41.54% (5)	68.83% (5)	49.73% (4)	57.74% (4)	4.86
	GAF-CNN	52.08% (4)	28.34% (7)	58.31% (2)	38.14% (7)	76.73% (2)	51.00% (3)	61.27% (3)	3.86
	SVM	50.25% (7)	44.09% (3)	51.69% (7)	47.59% (3)	56.71% (7)	49.13% (7)	52.65% (7)	5.86
	MLP	47.95% (8)	**66.45% (1)**	49.41% (8)	**56.68% (1)**	28.52% (8)	44.74% (8)	34.84% (8)	6.00
	DT	52.86% (2)	48.24% (2)	54.51% (4)	51.19% (2)	57.72% (6)	**51.50% (1)**	**54.43% (1)**	3.29
601318	GC–CNN	**51.01% (1)**	46.17% (5)	50.21% (2)	48.11% (5)	55.70% (3)	**51.68% (1)**	**53.61% (1)**	**2.71**
	Dual-CNN	49.45% (5)	50.51% (3)	48.53% (5)	49.50% (3)	48.43% (5)	50.41% (3)	49.40% (3)	4.14
	CNN-TA	49.47% (4)	46.08% (6)	48.51% (6)	47.26% (6)	52.74% (4)	50.31% (5)	51.50% (5)	5.00
	CNN-LSTM	50.09% (3)	53.89% (2)	49.29% (3)	51.49% (2)	46.42% (6)	51.02% (2)	48.61% (2)	3.43
	GAF-CNN	50.13% (2)	24.37% (8)	**50.36% (1)**	32.85% (8)	**75.16% (1)**	50.38% (4)	60.33% (4)	3.57
	SVM	47.66% (7)	37.73% (7)	46.03% (8)	41.47% (7)	57.26% (2)	48.76% (7)	52.67% (7)	5.86
	MLP	47.25% (8)	50.42% (4)	46.60% (7)	48.44% (4)	44.19% (7)	47.99% (8)	46.01% (8)	6.43
	DT	49.22% (6)	**57.43% (1)**	48.59% (4)	**52.64% (1)**	41.29% (8)	50.10% (6)	45.27% (6)	4.86
603123	GC–CNN	50.52% (3)	40.20% (5)	52.98% (2)	45.71% (5)	61.62% (3)	48.93% (4)	54.55% (4)	3.57
	Dual-CNN	50.13% (4)	**53.44% (1)**	50.25% (4)	51.79% (2)	46.81% (7)	50.00% (3)	48.35% (3)	4.00
	CNN-TA	48.64% (6)	37.34% (6)	48.91% (6)	42.35% (7)	60.17% (4)	48.46% (6)	53.69% (6)	5.57
	CNN-LSTM	48.31% (7)	45.52% (4)	48.88% (7)	47.14% (4)	51.17% (6)	47.67% (7)	49.36% (7)	5.86
	GAF-CNN	49.44% (5)	32.01% (8)	49.96% (5)	39.02% (8)	**66.93% (1)**	48.72% (5)	56.39% (5)	4.86
	SVM	51.36% (2)	36.51% (7)	52.69% (3)	43.14% (6)	66.53% (2)	50.65% (2)	57.51% (2)	3.29
	MLP	47.17% (8)	49.79% (3)	47.81% (8)	48.78% (3)	44.49% (8)	46.46% (8)	45.45% (8)	6.57
	DT	**52.41% (1)**	51.04% (2)	**53.02% (1)**	**52.01% (1)**	53.81% (5)	**51.84% (1)**	**52.81% (1)**	**2.29**

(4) **MACD signal**. MACD signal is a method for determining trading signals according to MACD. MACD is composed of DIF, DEA and MACD histogram. If DIF value is more than DEA value, the stock is bought at the closing price of that day. If the DIF value is less than the DEA value, the stock is sold at the closing price of that day. Otherwise, no action is taken at that day. Repeated labels are ignored until the label changes.

For purposes of comparison, Table 6 shows the holding gains and losses (P_H) and the maximum profit (P_M) of different stocks. The P_H is calculated based on holding the stock without selling, and the P_M is calculated based on the true trend. It is observed that the P_H of stocks 600809 and 601318 are positive, and the P_H of stocks 300330, 002580, 603808 and 603123 are negative. But if the trend predictions are accurate, excess returns can be achieved. The simulated trading results of the proposed GC–CNN based method and the comparative methods for six target stocks are listed in Table 7, and the best results are marked in bold. It is obvious that, for all the stocks, the proposed GC–CNN based method is able to gain more profits than the comparative methods. For stock 600809, all the methods can achieve positive returns. Specially, six methods' AAR values are more than 10%, and seven methods' SR values are more than 1. For stock 300330, the proposed GC–CNN, CNN-TA and RSI signal are the only methods with positive annualized returns, and only GC–CNN and RSI signal methods'



(a) 600809 (b) 300330 (c) 603808 (d) 002580 (e) 601318 (f) 603123

Fig. 10. Bar charts of comparison results for computational performance evaluation.

SR values are more than 1. For stock 603808, the proposed GC–CNN, Dual-CNN, RSI signal and MACD signal are the only methods with positive annualized returns, and only GC–CNN, Dual-CNN, RSI signal methods' SR values are more than 1. For stock 002580, the proposed GC–CNN is the only method with positive annualized returns, but none of the methods'

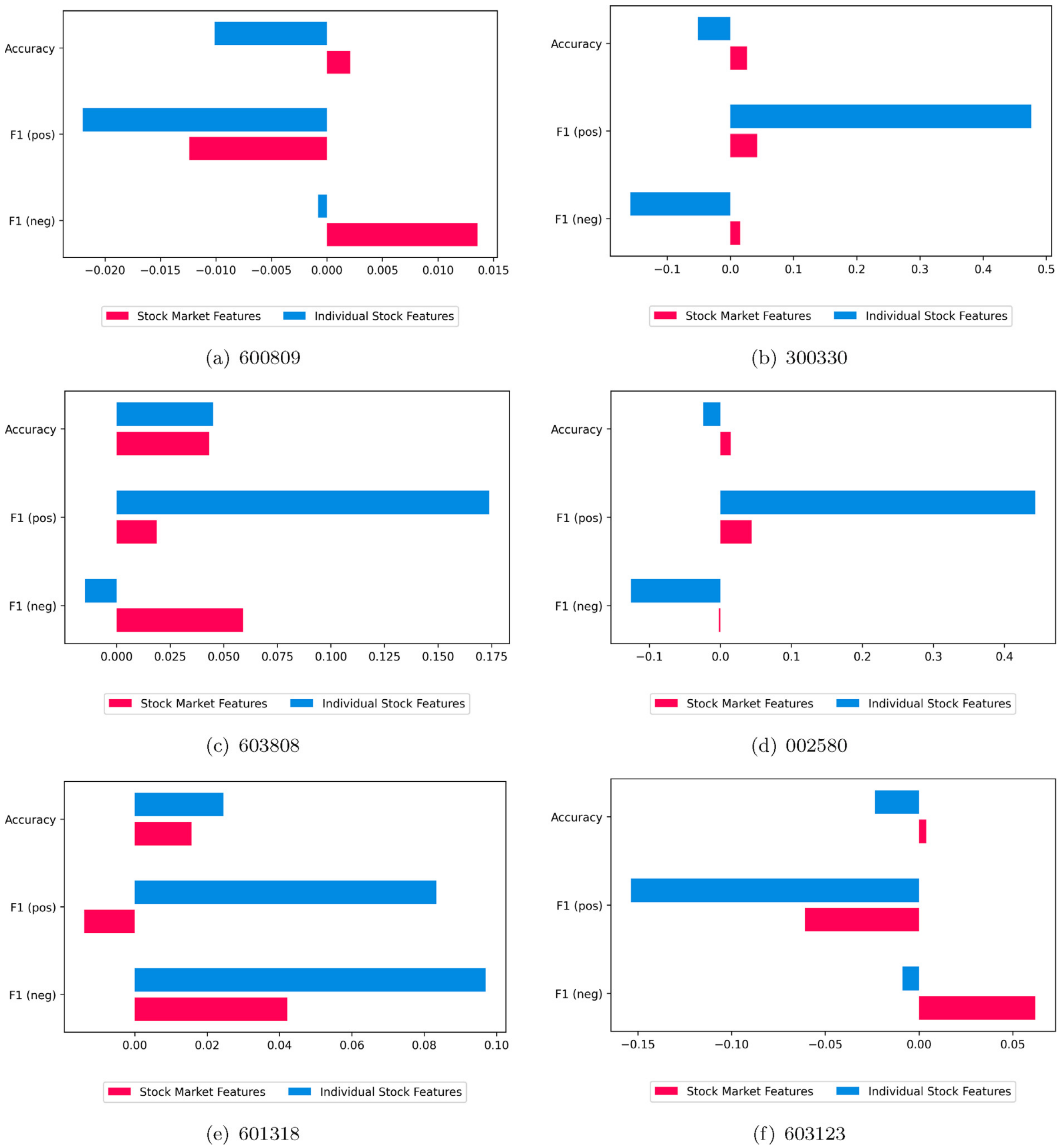


Fig. 11. The drop-column importances of features for the proposed GC–CNN based method.

Table 6
The holding gains and losses and the maximum profit of different stocks.

Code	Holding Gains and Losses (P_H)	Maximum Profit (P_M)
600809	129430.08	14897932089.51
300330	−4252.93	5503839.45
002580	−26536.82	10147989.03
603808	−3273.32	260181.89
601318	61696.65	160287819.61
603123	−5574.92	578124.70

Table 7
Comparison results for financial evaluation.

Code	Methods	TMoney (Ignore trading commission)	TMoney (0.1% trading commission)	AAR	SR
600809	GC–CNN	**283689.00**	**211416.32**	**33.42%**	**3.30**
	Dual-CNN	170722.00	133494.95	21.70%	2.35
	CNN-TA	162229.00	111272.28	17.35%	1.60
	CNN-LSTM	88726.30	68999.07	6.65%	0.52
	GAF-CNN	133242.25	122376.68	19.60%	1.63
	SVM	120937.00	93869.10	13.43%	1.64
	MLP	105746.00	78857.11	9.54%	1.06
	DT	83083.85	59454.03	3.52%	0.15
	RSI signal	74645.00	74506.28	8.30%	0.81
	SMA short-term signal	96605.00	70287.58	7.05%	0.52
	SMA long-term signal	100209.00	86856.21	11.68%	1.05
	MACD signal	83416.00	77829.02	9.25%	0.81
300330	GC–CNN	**84445.00**	**75948.00**	**23.25%**	1.44
	Dual-CNN	46707.00	42816.85	−7.46%	−1.01
	CNN-TA	62519.00	56440.02	6.25%	0.35
	CNN-LSTM	51211.95	47736.97	−2.29%	−0.39
	GAF-CNN	49598.90	46470.30	−3.59%	−0.42
	SVM	33911.00	28217.26	−24.88%	−2.80
	MLP	36631.00	33732.17	−17.86%	−3.39
	DT	34057.00	32277.34	−19.65%	−2.05
	RSI signal	76271.00	75360.71	22.77%	**1.78**
	SMA short-term signal	51210.00	45095.15	−5.03%	−0.68
	SMA long-term signal	32368.00	30077.30	−22.44%	−2.14
	MACD signal	46639.00	45151.99	−4.97%	−0.57
603808	GC–CNN	**64045.00**	**60856.52**	**21.71%**	**3.23**
	Dual-CNN	59291.00	56371.20	12.74%	1.33
	CNN-TA	45483.00	42880.19	−14.24%	−1.69
	CNN-LSTM	49024.30	48046.30	−3.91%	−0.78
	GAF-CNN	46448.15	45271.97	−9.46%	−1.28
	SVM	40771.00	39175.40	−21.65%	−4.05
	MLP	41672.00	39894.66	−20.21%	−5.10
	DT	49171.00	48273.87	−3.45%	−0.65
	RSI signal	58439.00	58287.83	16.58%	1.91
	SMA short-term signal	44568.00	41643.13	−16.71%	−2.86
	SMA long-term signal	49947.00	48666.89	−2.67%	−1.05
	MACD signal	52329.00	51212.52	2.43%	−0.01

Code	Methods	TMoney (Ignore trading commission)	TMoney (0.1% trading commission)	AAR	SR
002580	GC–CNN	**62911.00**	**56100.20**	**3.91%**	**0.21**
	Dual-CNN	34331.40	30540.39	−15.15%	−2.00
	CNN-TA	27537.20	25056.24	−20.57%	−3.15
	CNN-LSTM	48212.83	44480.16	−3.82%	−0.81
	GAF-CNN	44065.15	42516.41	−5.26%	−1.20
	SVM	40133.00	33886.39	−12.16%	−1.60
	MLP	21790.40	19485.53	−26.96%	−5.73
	DT	34537.00	32359.70	−13.50%	−1.95
	RSI signal	41057.00	40727.93	−6.61%	−1.19
	SMA short-term signal	34085.00	29016.83	−16.59%	−3.09
	SMA long-term signal	37158.00	34578.65	−11.57%	−2.82
	MACD signal	42031.00	40026.65	−7.15%	−1.55
601318	GC–CNN	**225190.21**	**165637.00**	**27.07%**	**2.30**
	Dual-CNN	73852.72	59764.84	3.63%	0.23
	CNN-TA	123132.27	89370.99	12.32%	0.97
	CNN-LSTM	74871.30	61582.57	4.26%	0.35
	GAF-CNN	85303.70	79980.49	9.85%	1.10
	SVM	46240.00	33003.28	−7.97%	−1.78
	MLP	58309.00	48263.55	−0.70%	−0.70
	DT	65060.00	52962.07	1.16%	−0.27
	RSIsignal	60629.00	60230.30	3.79%	0.34
	SMAshort-termsignal	72273.00	54109.03	1.59%	−0.16
	SMAlong-termsignal	59515.00	50501.65	0.20%	−0.33
	MACDsignal	73614.00	65063.49	5.41%	0.49
603123	GC–CNN	**64388.00**	**58445.32**	**8.12%**	0.73
	Dual-CNN	63684.00	57820.26	7.54%	0.81
	CNN-TA	37128.00	33740.05	−17.85%	−4.15
	CNN-LSTM	39580.65	36605.30	−14.44%	−3.71

(*continued on next page*)

Table 7 (*continued*)

Code	Methods	TMoney	TMoney	AAR	SR
	GAF-CNN	43000.15	42240.57	−8.09%	−2.39
	SVM	51226.00	45429.76	−4.68%	−1.68
	MLP	31357.00	28145.80	−24.97%	−5.17
	DT	47384.00	42566.04	−7.73%	−2.33
	RSIsignal	41325.00	41177.48	−9.25%	−2.89
	SMAshort-termsignal	47286.00	41814.27	−8.55%	−2.50
	SMAlong-termsignal	48227.00	45582.41	−4.52%	−1.71
	MACDsignal	57586.00	55952.53	5.79%	**0.97**

SR values are more than 1. For stock 601318, most of the methods can achieve positive returns, but only GC–CNN and GAF-CNN methods' SR values are more than 1. For stock 603123, the proposed GC–CNN, Dual-CNN and MACD signal are the only methods with positive annualized returns, but none of the methods' SR values are more than 1. In summary, among all the methods, only the proposed GC–CNN based method can keep AAR values positive and SR values more than 0. Especially, for stocks 600809, 300330, 603808 and 601318, the AAR values are more than 20% and the SR values are more than 1.5 by using the proposed method, indicating that the proposed GC–CNN based method is able to achieve more stable, higher and consistent returns.

Moreover, Figs. 12–17 show monthly profits of target stocks by using different methods during simulated trading. It is obvious that the trades generated by the proposed GC–CNN based method are more successful most of the time. This proves again that the proposed method is more stable and can achieve consistent returns.

4.5. Discussion

By comparing the results based on computational performance evaluation and financial evaluation, it is interesting to see that the methods with better computational performance may not generate higher profits when we simulated stock trading. For example, the proposed GC–CNN based method does not always outperform other comparative methods in terms of computational performance evaluation, however, GC–CNN achieves the highest returns in financial evaluation. Especially, for stock 002580, although CNN-TA is the method with the best computational performance, only GC–CNN based method can achieve positive returns. The reasons for this phenomenon are: (1) When we simulated trading based on predictive labels, repeated labels are ignored until the label changes. That is to say, not all the correct predictive labels can generate trading signals. (2) The positive return depends on the correct buying signal and the relatively effective selling signal. In summary, the methods with better computational performance do not necessarily generate better trading signals.

To better illustrate the returns after each pair of buying and selling operations, Figs. 18–23 are presented. For each stock, the final return is equal to the total positive returns minus the total negative returns. It is obvious that all the methods have the possibility of generating negative returns, and the methods with better computational performance do not guarantee that the total positive returns are more than the total negative returns. For example, for stock 603808, as can be seen from Fig. 20, the MLP method outperforms other methods in Precision_{pos} and $F_{1,pos}$, but it generates more negative returns, thereby the final return is negative. For stock 002580, as can be seen from Fig. 21, CNN-TA method outperforms other methods in $\text{Accuracy}, \text{Recall}_{pos}, \text{Precision}_{neg}$ and $F_{1,neg}$, but it generates more negative returns, thereby the final return is negative. In

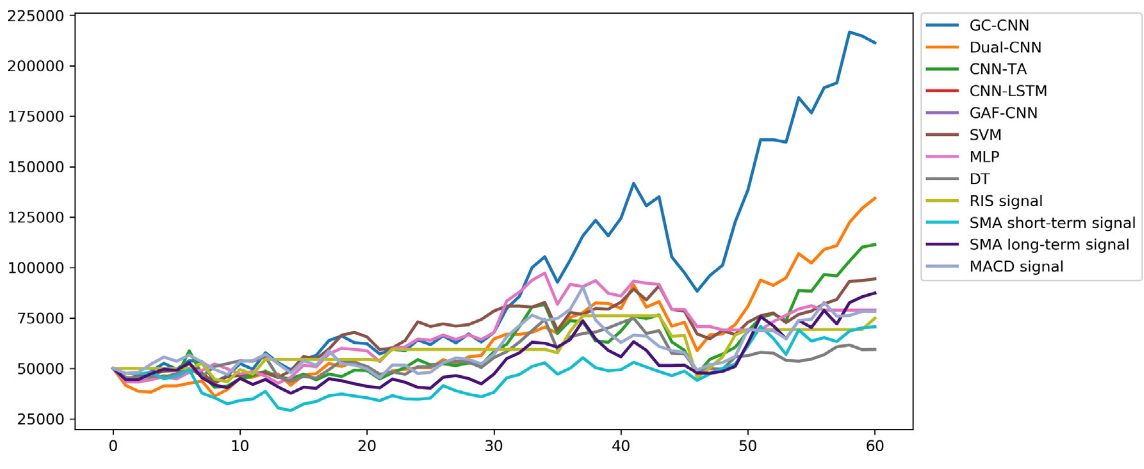


Fig. 12. Profits of stock 600809 over the months.

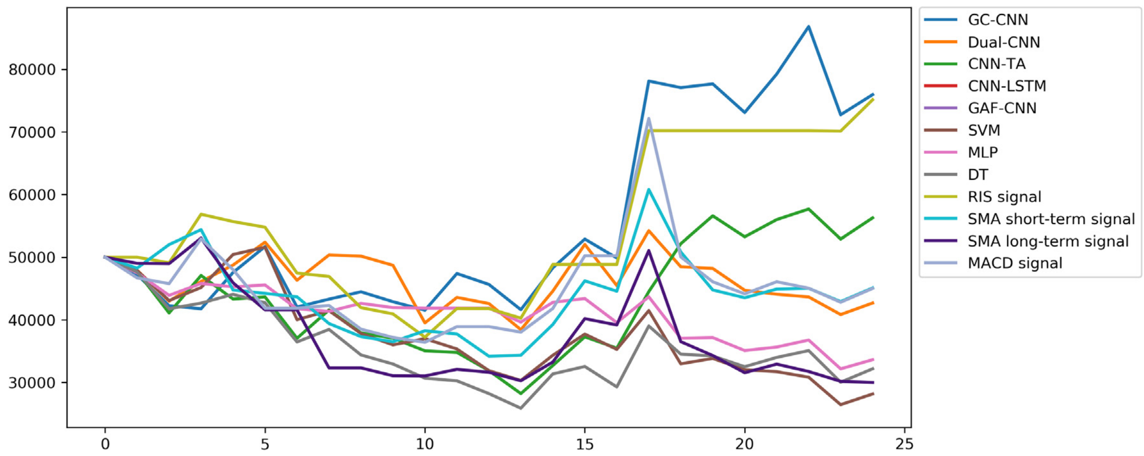


Fig. 13. Profits of stock 300330 over the months.

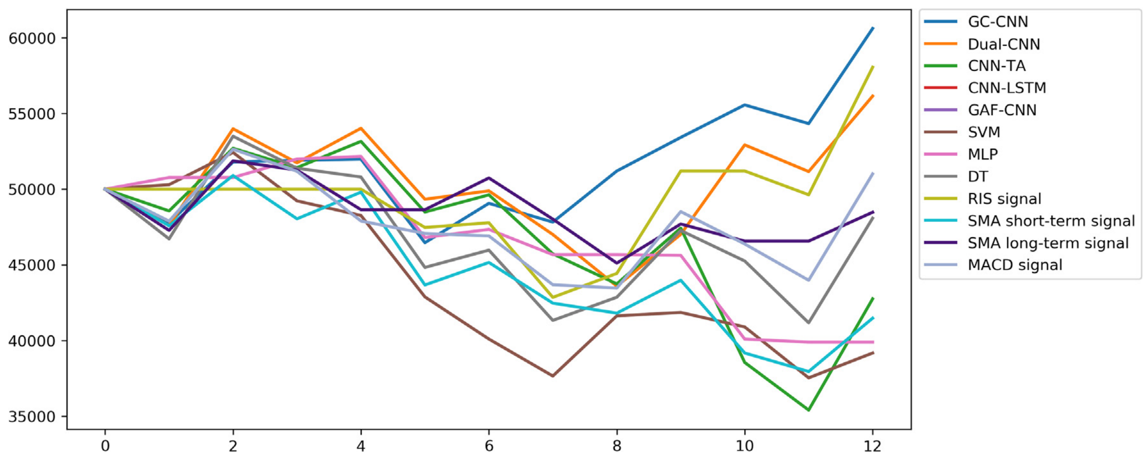


Fig. 14. Profits of stock 603808 over the months.

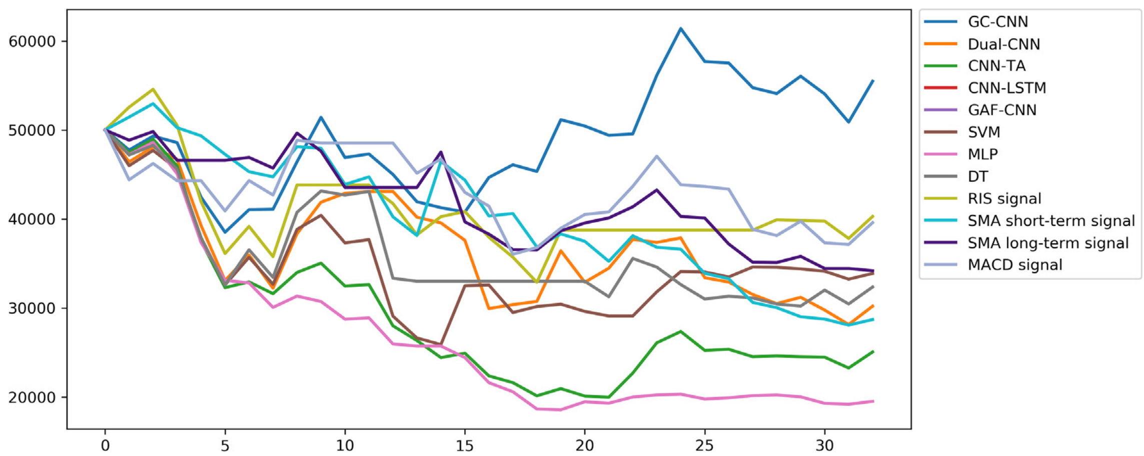


Fig. 15. Profits of stock 002580 over the months.

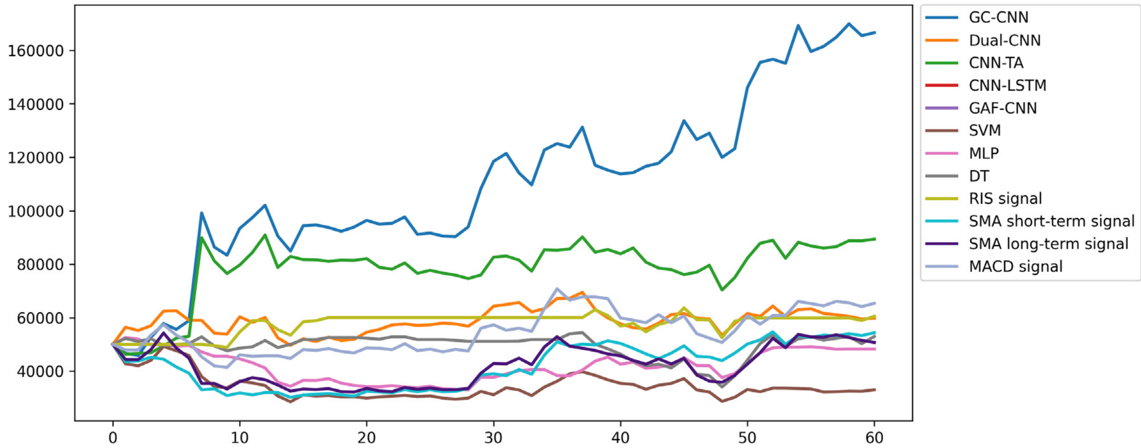


Fig. 16. Profits of stock 601318 over the months.

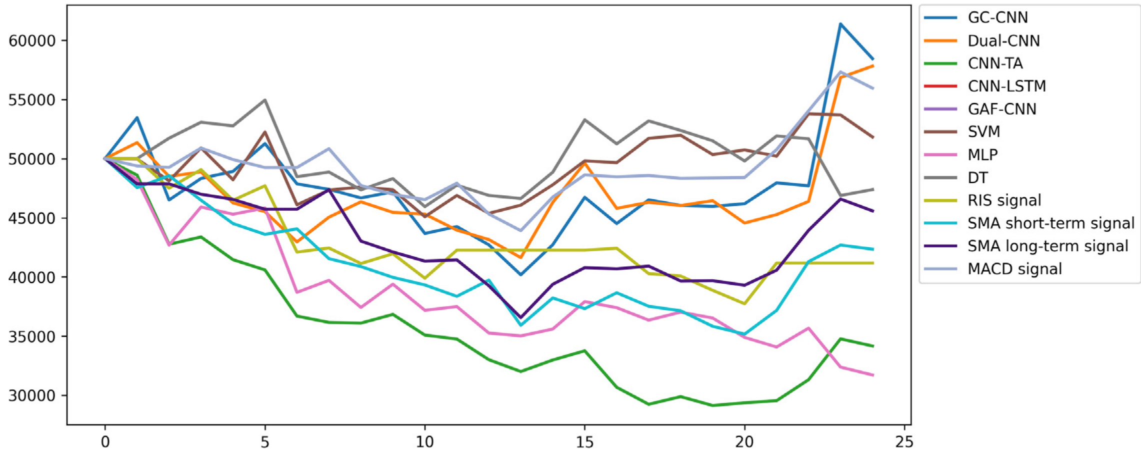
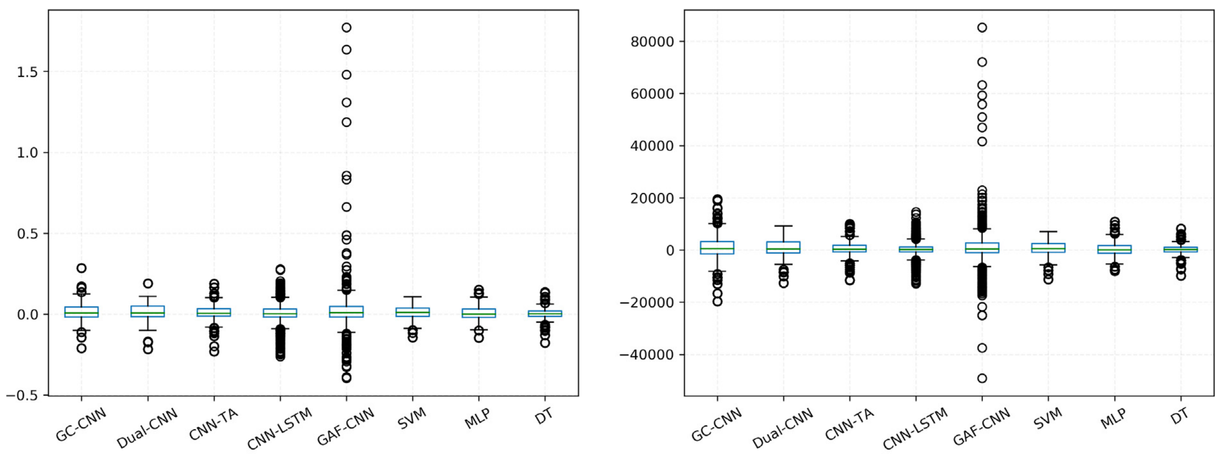
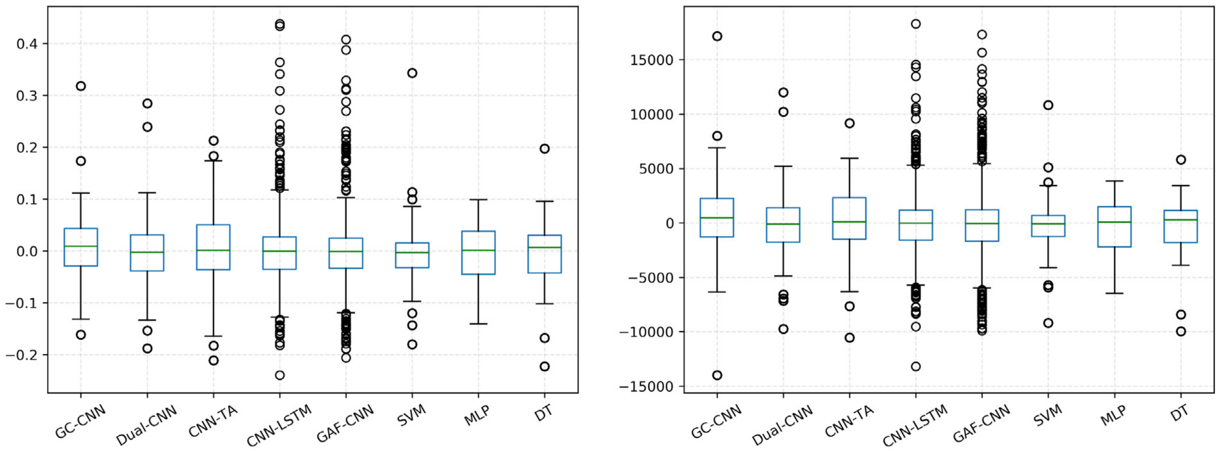


Fig. 17. Profits of stock 603123 over the months.



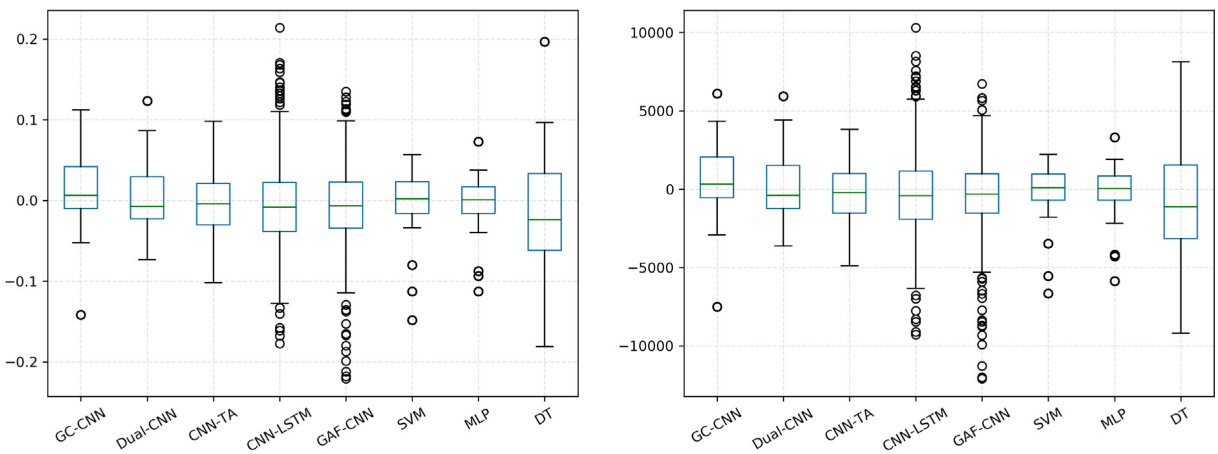
(a) 600809 - Return rate distribution (b) 600809 - Return distribution

Fig. 18. The boxplots for stock 600809.



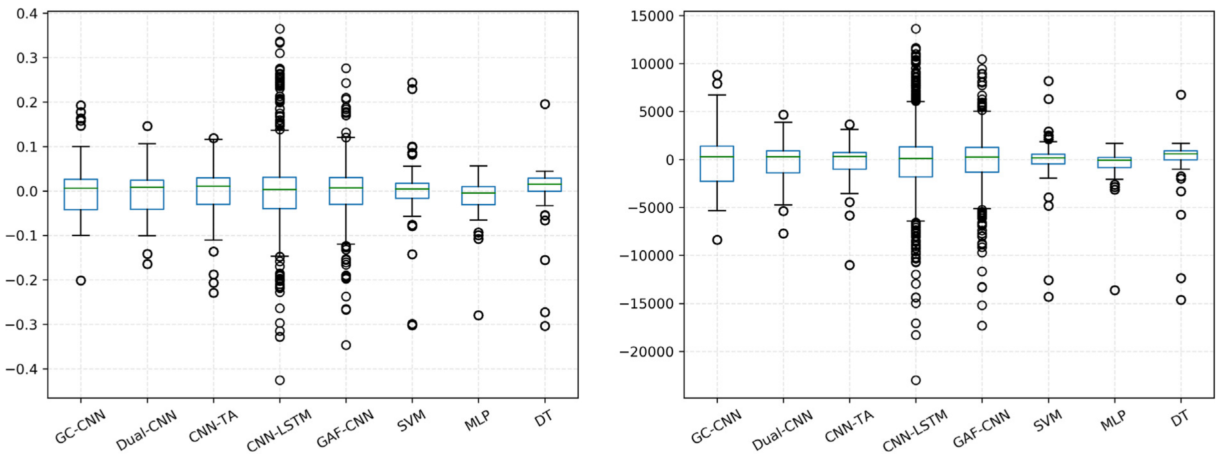
(a) 300330 - Return rate distribution (b) 300330 - Return distribution

Fig. 19. The boxplots for stock 300330.



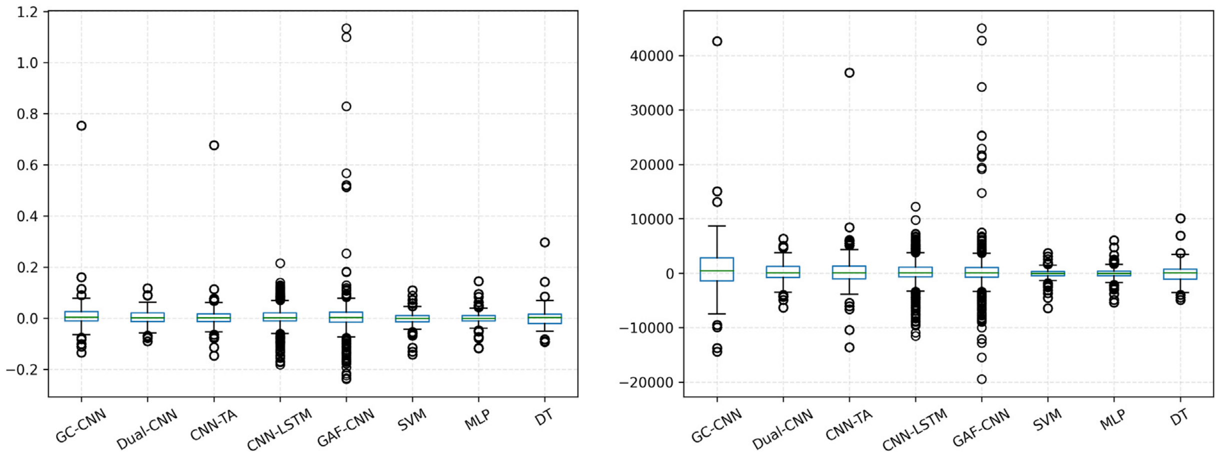
(a) 603808 - Return rate distribution (b) 603808 - Return distribution

Fig. 20. The boxplots for stock 603808.



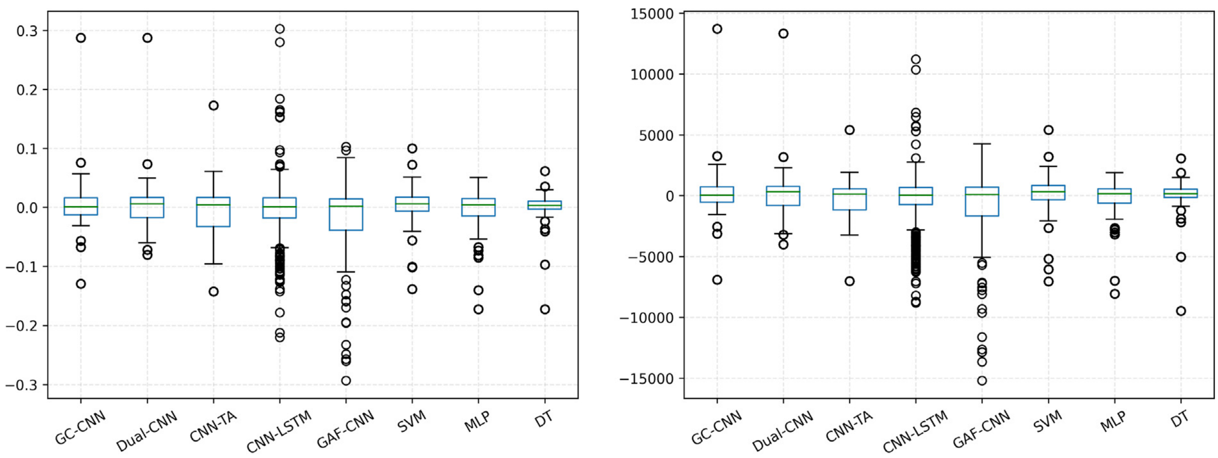
(a) 002580 - Return rate distribution (b) 002580 - Return distribution

Fig. 21. The boxplots for stock 002580.



(a) 601318 - Return rate distribution (b) 601318 - Return distribution

Fig. 22. The boxplots for stock 601318.



(a) 603123 - Return rate distribution (b) 603123 - Return distribution

Fig. 23. The boxplots for stock 603123.

comparison, although the proposed GC–CNN based method does not always outperform other methods in terms of computational performance evaluation, the trading signals obtained by GC–CNN are better than other methods, thereby better simulated trading results are obtained.

5. Conclusion

In this study, we propose a GC–CNN based method for stock trend prediction by considering both stock market information and individual stock information. In the proposed method, the stock market information, i.e., the constructed stock networks and the corresponding feature matrices, and the individual stock information, i.e., trading data and technical indicators of target stock, are transformed into images. Then, we predict future trends of target stock by using the proposed GC–CNN model, in which the stock market features are captured by the proposed IGCN, and the individual stock features are captured by the designed Dual-CNN.

We randomly select six stocks from the Chinese stock market to demonstrate the superior performance of the proposed method from two aspects: computational performance evaluation and financial evaluation. In computational performance evaluation, Dual-CNN, CNN-TA, SVM, MLP and DT methods are compared with the proposed method. In financial evaluation, we simulated stock trading based on different predictions and other common stock trading strategies, such as RSI signal, SMA short-term signal, SMA long-term signal and MACD signal. The comparison results show that: (1) The proposed GC–CNN based method requires longer training time; (2) With the consideration of stock market information, the prediction performance can be improved; (3) The proposed GC–CNN method does not always outperform other methods in terms of computational performance evaluation; (4) The computational performance of the proposed GC–CNN based method is more stable than other comparative methods; (5) The proposed GC–CNN based method is able to achieve more stable, higher and consistent returns; (6) The methods with better computational performance does not necessarily generate better trading signals, and the trading signals obtained by the proposed GC–CNN based method are better than those obtained by other methods. These results indicate that although the proposed GC–CNN based method requires longer training time, it outperforms other stock trend prediction methods and common stock trading strategies.

For future improvement of the trend prediction method, there are some possible directions. Firstly, the metaheuristic algorithms such as particle swarm optimization (PSO), firefly algorithm (FA) and differential evolution (DE) can be utilized to optimize the structural parameters of the GC–CNN. Secondly, many other types of images can be generated in the image creation phase, such as bull patterns and 3D images. Thirdly, the proposed method might perform better if investor sentiment is considered. Fourthly, the proposed method can be modified and used for long-term stock trend prediction by considering more features, such as enterprise association networking information, industry background, shareholder structure, initial public offering (IPO) positioning, etc.

CRediT authorship contribution statement

Wei Chen: Conceptualization, Methodology, Investigation, Writing - review & editing. **Manrui Jiang:** Software, Data curation, Writing - original draft, Writing - review & editing. **Wei-Guo Zhang:** Software, Visualization, Writing - review & editing. **Zhensong Chen:** Validation, Writing - review & editing.

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.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (Nos. 71720107002, 72071134), the Special fund for basic scientific research operating expenses of Beijing Municipal Colleges and Universities of Capital University of Economics and Business (No. QNTD202002), the Project of High-level Teachers in Beijing Municipal Universities in the Period of 13th Five-year Plan (No. CIT&TCD20190338), and the Humanity and Social Science Foundation of Ministry of Education of China (No. 19YJAZH005).

Appendix A

A.1. Rate of change (ROC)

The ROC indicator can be calculated as follows:

$$ROC_t = \frac{p_t - p_{t-m}}{p_{t-m}} \times 100\%, \tag{1}$$

where p_t is stock price at time t, m is the length of time period. In general, ROC is calculated based on closing prices.

The relevant indicators and their formulas that we used in this paper are listed as follows:

- Latest ROC: $ROC_t = \frac{p_t - p_{t-1}}{p_{t-1}} \times 100\%$.
- The average ROC in recent n days: $\overline{ROC_t(n)} = \frac{1}{n}\sum_{k=0}^{n-1} ROC_{t-k}$.
- The standard deviation of ROC in recent n days: $\sigma_{ROC_t(n)} = \sqrt{\frac{1}{n}\sum_{k=0}^{n-1}\left(ROC_{t-k} - \overline{ROC_t(n)}\right)^2}$.

A.2. Simple moving average (SMA)

The formula of SMA is given in Eq. (2). In general, SMA is calculated based on closing prices.

$$SMA_t(P, n) = \frac{1}{n}\sum_{k=t-n+1}^{t} p_k. \tag{2}$$

A.3. Exponential moving average (EMA)

The formula of EMA is shown as follows:

$$EMA_t(P,n) = \alpha \times p_t + (1-\alpha) \times EMA_{t-1}(n-1), \tag{3}$$

where $\alpha = \frac{2}{n+1}$ is the smoothness index and $EMA_0 = p_0$. In general, EMA is calculated based on closing prices.

A.4. Weight moving average (WMA)

The formula of WMA is illustrated in Eq. (4). In general, WMA is calculated based on closing prices.

$$WMA_t(P,n) = \frac{\sum_{k=t-n+1}^{t} [(n-t+k) \times p_k]}{\sum_{m=1}^{n} m}. \tag{4}$$

A.5. Hull moving average (HMA)

HMA can be calculated as follows:

$$HMA_t(P,n) = WMA_t\left(2 \times WMA_t\left(P, \frac{n}{2}\right) - WMA_t(P,n), [\sqrt{n}]\right). \tag{5}$$

A.6. Triple exponential moving average (TEMA)

The calculation of TEMA is expressed as follows:

$$TEMA_t(P,n) = 3 \times EMA_t(P,n) - 3 \times EMA_t(EMA_t(P,n),n) + EMA_t(EMA_t(EMA_t(P,n),n),n). \tag{6}$$

A.7. Moving average convergence and divergence (MACD)

Moving average convergence and divergence (MACD) is composed of DIF, DEA and MACD histogram. Eqs. (7)–(9) show the calculations of MACD.

$$DIF_t = EMA_t(p,12) - EMA_t(p,26). \tag{7}$$

$$DEA_t = EMA_t(DIF_t, 9). \tag{8}$$

$$\text{MACD histogram} = 2 \times (DIF_t - DEA_t). \tag{9}$$

A.8. Relative strength index (RSI)

The RSI indicator can be calculated as follows:

$$RSI_t(n) = 100 - \frac{100}{1 + \frac{Average\ Gain}{Average\ Loss}}. \tag{10}$$

A.9. Commodity channel index (CCI)

The formula of CCI is expressed as follows:

$$CCI_t(n) = \frac{TP - SMA_t(TP,n)}{0.015 \times SMA_t(TP - SMA_t(TP,n),n)}, \tag{11}$$

where $TP = \frac{h_t + l_t + c_t}{3}$.

A.10. Williams indicator (WR)

The calculation of WR is expressed as follows:

$$WR_t(n) = \frac{\max\{H\} - c_t}{\max\{H\} - \min\{L\}} \times -100, \tag{12}$$

where c_t is the closing price at time t, $H = \{h_{t-n+1}, h_{t-n+2}, \ldots, h_t\}$ is a set of highest prices in recent n days and $L = \{l_{t-n+1}, l_{t-n+2}, \ldots, l_t\}$ is a set of lowest prices in recent n days.

A.11. Chande momentum oscillator (CMO)

The formula of CMO is illustrated in Eq. (13), where S_u is the sum of the momentum of up days and S_d is the sum of the momentum of down days.

$$CMO_t(n) = \frac{S_u - S_d}{S_u + S_d} \times 100. \tag{13}$$

References

[1] M. Ananthi, K. Vijayakumar, Stock market analysis using candlestick regression and market trend prediction (CKRM), J. Ambient Intell. Humaniz. Comput. (2020), https://doi.org/10.1007/s12652-020-01892-5.
[2] S. Barra, S.M. Carta, A. Corriga, A.S. Podda, D.R. Recupero, Deep learning and time series-to-image encoding for financial forecasting, IEEE-CAA J. Automatica Sin. 7 (3) (2020) 683–692, https://doi.org/10.1109/JAS.2020.1003132.
[3] 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.
[4] L. Cagliero, P. Garza, G. Attanasio, E. Baralis, Training ensembles of faceted classification models for quantitative stock trading, Computing 102 (5) (2020) 1213–1225.
[5] M.-Y. Chen, B.-T. Chen, A hybrid fuzzy time series model based on granular computing for stock price forecasting, Inf. Sci. 294 (2015) 227–241.
[6] M.-Y. Chen, C.-H. Liao, R.-P. Hsieh, Modeling public mood and emotion: Stock market trend prediction with anticipatory computing approach, Comput. Hum. Behav. 101 (2019) 402–408, https://doi.org/10.1016/j.chb.2019.03.021.
[7] T. Chen, F. Chen, An intelligent pattern recognition model for supporting investment decisions in stock market, Inf. Sci. 346–347 (2016) 261–274, https://doi.org/10.1016/j.ins.2016.01.079.
[8] W.-C. Chiang, D. Enke, T. Wu, R. Wang, An adaptive stock index trading decision support system, Expert Syst. Appl. 59 (2016) 195–207.
[9] L. de Haan, C. Mercadier, C. Zhou, Adapting extreme value statistics to financial time series: dealing with bias and serial dependence, Financ. Stoch. 20 (2) (2016) 321–354.
[10] Y. Hao, Q. Gao, Predicting the trend of stock market index using the hybrid neural network based on multiple time scale feature learning, Appl. Sci-Basel 10 (11) (2020) 3961.
[11] M. Jiang, W. Chen, X. Li, S-GCN-GRU-NN: A novel hybrid model by combining a Spatiotemporal Graph Convolutional Network and a Gated Recurrent Units Neural Network for short-term traffic speed forecasting, J. Data Inform. Manag. (2020), https://doi.org/10.1007/s42488-020-00037-9.
[12] M. Jiang, L. Jia, Z. Chen, W. Chen, The two-stage machine learning ensemble models for stock price prediction by combining mode decomposition, extreme learning machine and improved harmony search algorithm, Ann. Oper. Res. (2020), https://doi.org/10.1007/s10479-020-03690-w.
[13] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, (2016). arXiv:1609.02907..
[14] S.I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, D. Rueckert, Metric learning with spectral graph convolutions on brain connectivity networks, NeuroImage 169 (2018) 431–442.
[15] M.-C. Lee, Using support vector machine with a hybrid feature selection method to the stock trend prediction, Expert Syst. Appl. 36 (8) (2009) 10896–10904.
[16] L. Lei, Wavelet neural network prediction method of stock price trend based on rough set attribute reduction, Appl. Soft. Comput. 62 (2018) 923–932.
[17] Y. Li, Z. He, X. Ye, Z. He, K. Han, Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition, EURASIP J. Image Video Process. 2019 (2019) 78.
[18] Q. Liang, W. Rong, J. Zhang, J. Liu, Z. Xiong, Restricted boltzmann machine based stock market trend prediction, in: 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1380–1387.
[19] J. Long, Z. Chen, W. He, T. Wu, J. Ren, An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market, Appl. Soft. Comput. 91 (2020), https://doi.org/10.1016/j.asoc.2020.106205.
[20] I. Marković, M. Stojanović, J. Stanković, M. Stanković, Stock market trend prediction using AHP and weighted kernel LS-SVM, Soft Comput. 21 (18) (2017) 5387–5398.
[21] B. Moews, J.M. Herrmann, G. Ibikunle, Lagged correlation-based deep learning for directional trend change prediction in financial time series, Expert Syst. Appl. 120 (2019) 197–206.
[22] N. Naik, B.R. Mohan, Intraday stock prediction based on deep neural network, Natl. Acad. Sci. Lett.-India 43 (3) (2020) 241–246.
[23] D.M.Q. Nelson, A.C.M. Pereira, R.A. de Oliveira, Stock market's price movement prediction with LSTM neural networks, in: 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1419–1426.
[24] A.M. Ozbayoglu, M.U. Gudelek, O.B. Sezer, Deep learning for financial applications: A survey, Appl. Soft. Comput. 93 (2020), https://doi.org/10.1016/j.asoc.2020.106384 106384.
[25] M.O. Ozorhan, I.H. Toroslu, O.T. Sehitoglu, Short-term trend prediction in financial time series data, Knowl. Inf. Syst. 61 (1) (2019) 397–429.
[26] I.R. Parray, S.S. Khurana, M. Kumar, A.A. Altalbe, Time series data analysis of stock price movement using machine learning techniques, Soft Comput. (2020), https://doi.org/10.1007/s00500-020-04957-x.
[27] A. Picasso, S. Merello, Y. Ma, L. Oneto, E. Cambria, Technical analysis and sentiment embeddings for market trend prediction, Expert Syst. Appl. 135 (2019) 60–70.
[28] O.B. Sezer, A.M. Ozbayoglu, Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach, Appl. Soft. Comput. 70 (2018) 525–538.
[29] O.B. Sezer, A.M. Ozbayoglu, Financial trading model with stock bar chart image time series with deep convolutional neural networks, (2019). arXiv:1903.04610..
[30] H. Tang, P. Dong, Y. Shi, A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points, Appl. Soft. Comput. 78 (2019) 685–696.
[31] M. Wen, P. Li, L. Zhang, Y. Chen, Stock market trend prediction using high-Order information of time series, IEEE Access 7 (2019) 28299–28308.
[32] D. Wu, X. Wang, J. Su, B. Tang, S. Wu, A Labeling Method for Financial Time Series Prediction Based on Trends, Entropy 22 (1162) (2020). doi:10.3390/e22101162..
[33] X. Yan, T. Ai, M. Yang, H. Yin, A graph convolutional neural network for classification of building patterns using spatial vector data, ISPRS-J. Photogramm. Remote Sens. 150 (2019) 259–273.
[34] J. Zhang, L. Li, W. Chen, Predicting stock price using two-Stage machine learning techniques, Comput. Econ. (2020), https://doi.org/10.1007/s10614-020-10013-5.
[35] Q. Zhang, J. Chang, G. Meng, S. Xu, S. Xiang, C. Pan, Learning graph structure via graph convolutional networks, Pattern Recognit. 95 (2019) 308–318.
[36] X. Zhang, A. Li, R. Pan, Stock trend prediction based on a new status box method and AdaBoost probabilistic support vector machine, Appl. Soft. Comput. 49 (2016) 385–398.
[37] J. Zhao, X. Liu, Q. Yan, B. Li, M. Shao, H. Peng, Multi-attributed heterogeneous graph convolutional network for bot detection, Inf. Sci. 537 (2020) 380–393.

[38] J. Zhao, D. Zeng, S. Liang, H. Kang, Q. Liu, Prediction model for stock price trend based on recurrent neural network, J. Ambient Intell. Humaniz. Comput. (2020), https://doi.org/10.1007/s12652-020-02057-0.
[39] T. Zhong, Q. Peng, X. Wang, J. Zhang, Novel indexes based on network structure to indicate financial market, Physica A 443 (2016) 583–594.
[40] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, M. Sun, Graph neural networks: a review of methods and applications, (2018). arXiv:1812.08434..