A Multi-indicator Feature Selection for CNN-Driven Stock Index Prediction

Hui Yang, Yingying Zhu(✉), and Qiang Huang

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
zhuyy@szu.edu.cn

Abstract. Stock index prediction is regarded as a challenging task due to the phenomena of non-linearity and random drift in trends of stock indices. In practical applications, different indicator features have significant impact when predicting stock index. In addition, different technical indicators which contained in the same matrix will interfere with each other when convolutional neural network (CNN) is applied to feature extraction. To solve the above problem, this paper suggests a multi-indicator feature selection for stock index prediction based on a multi-channel CNN structure, named MI-CNN framework. In this method, candidate indicators are selected by maximal information coefficient feature selection (MICFS) approach, to ensure the correlation with stock movements while reduce redundancy between different indicators. Then an effective CNN structure without sub-sampling is designed to extract abstract features of each indicator, avoiding mutual interference between different indicators. Extensive experiments support that our proposed method performs well on different stock indices and achieves higher returns than the benchmark in trading simulations, providing good potential for further research in a wide range of financial time series prediction with deep learning based approaches.

Keywords: Stock index prediction · Feature selection
Maximal information coefficient · Convolutional neural networks

1 Introduction

Stock index prediction has been an important issue in the fields of finance, engineering and mathematics due to its potential financial gain. The prediction of stock index is regarded as a challenging task of financial time series prediction. There has been so much work done on ways to predict the movements of stock price. In the past years, most research studies focused on the time series models and statistical methods to forecast future trends based on the historical data, such as ARIMA [1], ARCH [2], GARCH [4], etc. With the great development of computer science, many recent works have been proposed based on the machine learning methods, such as neural networks (NN) [8], bayesian approach [12] and support vector machine (SVM) [19], to predict the stock index trends.

L. Cheng et al. (Eds.): ICONIP 2018, LNCS 11305, pp. 35–46, 2018.
https://doi.org/10.1007/978-3-030-04221-9_4

Recently, convolutional neural network (CNN) is gradually applied in the field of stock market, and some methods [7,15,18] based on CNN have shown that CNN can be an effective tool for feature extraction whether in the task of predicting specific price level or predicting the movements of stock. These CNN-driven methods have shown state-of-the-art performance. Gunduz et al. [7] proposed a CNN architecture with a specifically ordered feature set to predict the intraday direction of stocks. In the feature set, each instance was transformed into 2D-matrix by taking into account different indicators, price and temporal information. Sezer et al. [15] proposed a CNN-TA stock trading model, and 15×15 sized 2-D images were constructed using 15 different technical indicators. However, it is worth noting that there are still some common disadvantages hindering the current CNN-driven stock index prediction methods. First, they used fixed technical indicators as the input of CNN for stock forecasting. But different stock indices represent different industries, which present different characteristics and market cycles. Therefore, the adoption of fixed technical indicators is not adaptable to the prediction of different stock indices. In addition, when convolving the indicator matrix, different indicators will interfere with each other and cause confusion information in the feature maps. Because different technical indicators are fused in the same matrix.

To solve these problems, a CNN-driven multi-indicator stock index prediction framework, named MI-CNN, is presented in this paper, which applied CNN to extract abstract features in different indicators independently. In the MI-CNN framework, we utilize maximal information coefficient feature selection (MICFS) to filter more effective technical indicators for different stock indices intelligently, instead of using fixed indicators to predict all kinds of stock indices. Then a multi-channel CNN structure is proposed to extract features from each independent technical indicator, rather than extracting all indicator features in a single matrix confusedly. Our MI-CNN framework is proved to be effective on various stock indices and numerous experiments are illustrated in this paper. The average prediction accuracy and returns achieve 60.02% and 31.07% in the experiments.

The remainder of this paper is structured as follows. In Sect. 2, we briefly review the related work in stock index prediction tasks. In Sect. 3, we describe the architecture and detailed design of the framework. Then the experiments and the corresponding analysis are shown in Sect. 4. Finally, some concluding remarks are drawn and future research directions are discussed from Sect. 5.

2 Related Work

Financial time series modeling is regarded as one of the most challenging forecasting problems. In [17], Kevin indicated that the change in the stock price was better forecasted by the non-linear methods when compared with linear regression models. In [5], it has shown that forecasting price movements can often result in more trading results. Oriani et al. [13] evaluated the impact of technical indicators on stock forecasting and concluded that lagging technical indicators can improve the accuracy of the stock forecasts compared to that made with the original series of closing price.

The purpose of feature selection method is to reduce data complexity and improve prediction accuracy. Feature subset selection methods can be classified into two categories: the filter approach and the wrapper approach [9]. Lee proposed a F-score and supported sequential forward search feature selection method, which combined the advantages of filter methods and wrapper methods to select the optimal feature subset from the original feature set [11]. Su et al. proposed an integrated nonlinear feature selection method to select the important technical indicators objectively in forecasting stock price [16]. The results showed that the proposed method outperforms the other models in accuracy, profit evaluation and statistical test.

Recently, more and more practice shows promising performance in different ways of combining CNN and stock prediction tasks together. Tsantekidis et al. [18] proposed a deep learning methodology, based on CNN, that predicted the price movements of stocks, using as input large-scale, high-frequency time-series derived from the order book of financial exchanges. Results showed that CNN is better suited for this kind of task in finance. In [6], researchers extracted commonly used indicators from financial time series data and used them as their features of artificial neural network (ANN) predictor. They generated 28×28 images by taking snapshots that were bounded by the moving window over a daily period.

3 Proposed Framework

The architecture of the MI-CNN framework is first briefly described in Fig. 1. More specifically, the system selects several effective indicators through MICFS from given stock index data. Then potential features of each indicator are extracted using a special CNN structure. After that, extracted features are input into the ANN model to provide prediction results. Finally, a straight trading strategy [3] is applied according to the final prediction results. We will introduce the details of the trading strategy in the experimental section.

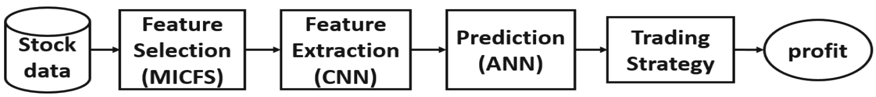


Fig. 1. Block diagram of complete framework

3.1 Stock Feature Selection

Stock market data and several common-use technical indicators are employed as input features in this study. Because most of the technical indicators are calculated from basic stock market data, the redundancy of information is unavoidable between different technical indicators. Therefore, we apply the MICFS approach to filter out indicators which are most relevant to the movements of given stock index, while the correlation between the selected indicators is minimal.

Maximal information coefficient (MIC) is a measure of dependence for two-variable relationships that captures a wide range of associations both functional and not [14]. MIC belongs to a larger class of maximal information-based nonparametric exploration (MINE) statistics for identifying and classifying relationships. Let D be a set of ordered pairs. For a grid G, let $D|_G$ denote the probability distribution induced by the data D on the cells of G. And I denote mutual information (MI):

$$I(x;y) = \iint p(x,y)log\frac{p(x,y)}{p(x)p(y)}dxdy \tag{1}$$

Let $I^*(D,x,y) = \max_G I(D|G)$, where the maximum is taken over all x-by-y grids G (possibly with empty rows/columns). MIC is defined as

$$MIC(D) = \max_{xy<B(|D|)} \frac{I^*(D,x,y)}{log_2 min\{x,y\}} \tag{2}$$

where B is a growing function satisfying $B(n) = o(n)$, and the authors heuristically suggest $B(n) = n^{0.6}$ as a default setting.

In this study, we don't limit the number of selected indicators until the best feature subset is constructed. For each indicator f, the MICFS is described by:

$$J(f) = \frac{1}{n}\sum_{i=1}^{n} MIC(C;f_i) - \frac{1}{n}\beta\sum_{s\in S}\sum_{i=1}^{n} MIC(s_i:f_i) \tag{3}$$

where $MIC(C;f_i)$ is the MIC of labels C and feature f, and f_i is the ith series of feature f, $MIC(s_i;f_i)$ is the MIC of candidate feature f and the selected feature s in feature subset S, and coefficient $\beta \in [0,1]$ indicates the effect of selected feature redundancy to the result. n is the number of series contained in each indicator, and $n = 3$ in our study. The basic steps of feature selection are summarized by the pseudo code listed below.

Algorithm 1. Maximal information coefficient feature selection algorithm

Input:
  Training dataset $D = \{F, C\}, F = (f_1, f_2, ..., f_n)$
Output:
  Selected feature subset $S, S \subseteq F$
Begin Set $S = \emptyset$, $\beta = 1$, $B(n) = n^{0.6}$
for $i = 1 : n$ **do**
  Calculate J for each feature f and $f \notin S$
  Select the maximum J and put corresponding feature f into subset S_i
  Train the model using S_i, and obtain the accuracy $P(S_i)$
end for
Select the best $P(S)$ and output the feature subset S
End

Eight popular indicators are chosen as candidate features for MICFS and each indicator contains three common-use series. An open-source library Technical Analysis Library (TA-Lib) is utilized to calculate technical indicators above. The details of all the eight technical indicators are depicted in Table 1.

Table 1. Candidate technical indicators

Technical indicator	Concrete series	Description
$Price$	$high$, low,$close$	Highest price, lowest price and close price
Vol	Vol, Vol_5, Vol_{10}	Trading volume of stock index
$MACD$	$MACD$, $MACD_{hist}$, $MACD_{signal}$	Moving Average Convergence and Divergence
RSI	RSI_5, RSI_{10}, EMA_n	Relative Strength Index
KD	$slow_K$, $slow_D$, $fast_k$	Stochastic Index
WR	WR_5, WR_{10}, WR_{20}	Larry Williams R
ROC	ROC_5, ROC_{10}, ROC_{20}	Rate of Change
CCI	CCI_5, CCI_{10}, CCI_{20}	Commodity Channel Index

3.2 CNN-Driven Stock Feature Extraction

In normal conditions, CNN is principally utilized to deal with image-related problem because of the ability to discern the spatial correlation of neighboring pixels. CNN can automatically extract the characteristic relationship between adjacent data elements and reconstruct the feature vectors [10]. The financial time series forecasting problem can be implicitly converted into an image classification problem when technical analysis data is shaped into two-dimension matrices [15]. As for the task of stock prediction in this study, market data and technical indicators are continuous-discrete time. As a consequence, vectors in each indicator series are relevant while different indicators are independent of each other. What's more, there are potential characteristics between the series of the same technical indicator with different computing periods. It raises the difficulty of extracting features from the stock data for predicting stock movements. Considering the above issues and the characters of CNN, a more applicable multi-channel CNN-driven framework is proposed to extract features from technical indicators in stock prediction task.

Figure 2 illustrates the sketch of the CNN architecture when three indicators are selected. Continuous time series of each indicator are formed as the shape of two-dimension matrices:

$$Indicator = \begin{bmatrix} f_{11} & \cdots & f_{1n} \\ f_{21} & \cdots & f_{2n} \\ f_{31} & \cdots & f_{3n} \end{bmatrix}_{3\times n} \tag{4}$$

where n represents a continuous date span of indicator and we fix n =10. For example, as for the indicator RSI, three series RSI_5, RSI_{10} and RSI_{20} are formed as a $3 \times n$ matrix. For each indicator, we use the same convolution architecture, two convolutional layers without subsampling, to extract features, which can reflect the extracted abstract features in a high level. Finally, the extracted features are integrated and put into the final prediction model, fully connected ANN. The purpose of our design is to make CNN automatically extract underlying features of each indicator. At the same time, it will not be subject to any other indicator when it extracts features of a certain indicator, owing to the independent CNN structure. The number of CNN channels depends on how many indicators are selected in feature subset S.

In the field of image, subsampling plays the role of dimensionality reduction and invariance. But in this study, the series of each indicator takes on specific meanings, and the different series, such as RSI_5 and RSI_{10}, are independent of each other. If subsampling, such as the maxpool layer, is adopted, the information in the indicator series would be lost and cause information loss probably. To avoid the loss of extracted features, there are two convolutional layers without subsampling applied in this study.

In the two convolutional layers, the sizes of the convolution kernels are 3×3 and 3×2, and the stride length is 1. Zero-padding strategy is applied in the first convolutional layer. In this way, when extracting features, it can not only extract momentum features between horizontally adjacent data points in an identical series, but also extract other underlying features between every two series and between total three series that are vertically adjacent. The number of feature maps in the two convolutional layers are 16 and 32, respectively.

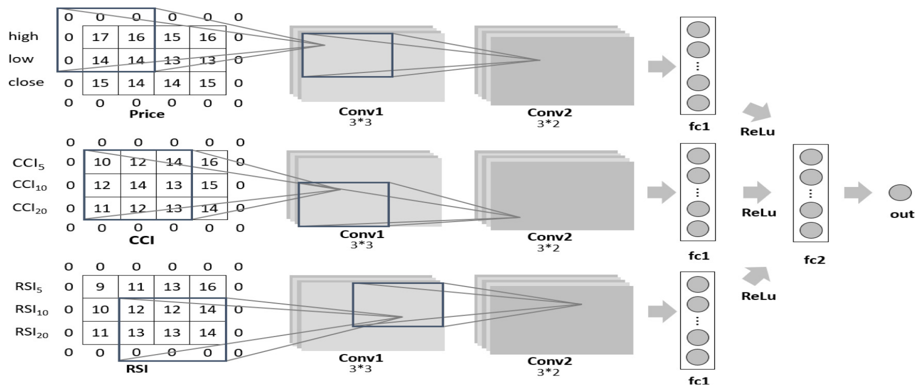


Fig. 2. Architecture of CNN when 3 indicators are selected

Fully connected ANNs are deployed after the second convolutional layers in each feature extraction channel. Then the ANNs are merged in the next hidden layer. The outputs are the movements of a given stock index. Given 1 shows the stock index price will rise next day. By contrast, given 0 indicates the opposite way. Specifically, the rectified linear units (ReLu) activation function is applied in ANN and the learning rate α is set to 0.01. The back-propagation algorithm is used to train the model. The parameters of the model are learned by minimizing the categorical cross entropy loss:

$$C(W) = -\sum_{i=1}^{L} y_i \cdot log y_i^{'} \tag{5}$$

where L is the number of different labels and the notation W refers to the parameters of weights. The ground truth vector is denoted by y, while $y^{'}$ is the predicted label distribution. In order to overcome the over fitting phenomenon, we punish the weights W with L2 regularization, and the L2 regularization is calculated as:

$$C^{'} = C + \frac{\lambda}{2n} \sum_{w} w^2 \tag{6}$$

where C denotes the original cost function, λ is regularization coefficient and n is the scale of training set. w is the weights need to be punished.

4 Experiment

4.1 Experiments on SPY

The basic experiments are developed on the S&P 500 Index ETF (SPY), from January 2008 to December 2017. The stock market data is published on Yahoo Finance. Afterwards, the dataset is divided into two sets. Data with eight years is served as training dataset while the remaining data with two years is used as the test dataset. The input data is scaled to [0, 1], using the min-max normalization approach. We employ accuracy and returns as our evaluation metrics:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$Returns = \frac{C_{final} - C_{initial}}{C_{initial}} \times 100\% \tag{8}$$

TP, FP, TN and FN represent respectively the true positive, false positive, true negative and false negative [7]. $C_{initial}$ and C_{final} present the capital at the beginning and at the end of transaction respectively.

The process of feature selection is analyzed in accordance with the execution phase of the MICFS approach. Table 2 illustrates the J values of candidate indicators in each iteration of MICFS approach, where $J(f)$ is calculated by Eq. 3. In each iteration, the indicator with the largest J value will be selected into the feature subset S. In Table 3, each row illustrates the selected feature subset S and the J value of the selected feature at this iteration. It also illustrates the prediction accuracy $P(S)$ on the training set when the current feature subset S is used as input data for the MI-CNN model. As shown in the table, when the feature subset contains Price, CCI and MACD, the prediction accuracy reaches the highest value.

Table 2. Execution detail of MICFS approach on SPY

Iteration	1	2	3	4	5	6	7
J(Price)	**0.1038**	-	-	-	-	-	-
J(Vol)	0.0950	−0.4485	−0.5219	−0.6817	−0.3031	**−0.4245**	-
J(MACD)	0.0943	−0.0184	**−0.1491**	-	-	-	-
J(KDJ)	0.0996	−0.0283	−0.1791	−0.6812	**−0.2811**	-	-
J(RSI)	0.0958	−0.0577	−0.2730	−0.5399	−0.5112	−0.7858	−0.9583
J(WR)	0.0987	−0.0157	−0.3337	−0.5030	−0.4199	−0.6485	**−0.7802**
J(ROC)	0.0959	−0.0462	−0.2261	**−0.3928**	-	-	-
J(CCI)	0.1010	**0.0212**	-	-	-	-	-

Table 3. Results of MICFS approach on SPY

Selected feature subset S	Maximum J(f)	P(S)
Price	0.1038	59.75%
Price, CCI	0.0212	63.32%
Price, CCI, MACD	−0.1491	**66.91%**
Price, CCI, MACD, ROC	−0.3928	64.51%
Price, CCI, MACD, ROC, KD	−0.2811	63.34%
Price, CCI, MACD, ROC, KD, Vol	−0.4245	60.30%
Price, CCI, MACD, ROC, KD, Vol, WR	−0.7802	59.65%
Price, CCI, MACD, ROC, KD, Vol, WR, RSI	−1.5551	57.65%

We apply the MICFS method to conduct experiments on several stock indices introduced in [3]. Figure 3 presents the experimental results of MICFS on different stock indices. Since different stock indices represent different industries and have their own characteristics, the number of features in the optimal feature subset is different when forecasting stock trends. It shows that the MICFS method can select effective features for different stock indices.

The signals produced from the final prediction model are applied to implement trading simulations. A simple trading strategy is conducted with the given signals.

$$Action = \begin{cases} buy & signal_c = 1 \text{ and } signal_{c-1} = 0 \\ sell & signal_c = 0 \text{ and } signal_{c-1} = 1 \\ sit & others \end{cases} \tag{9}$$

where $signal_c$ represents the signal of current date and $signal_{c-1}$ represents the signal of the last date. We adopt the T + 1 trading system in the trading simulations. The capital curves of trading results are shown in Fig. 4; it displays the accumulated capitals trading on SPY index during the year of 2016 and 2017. In order to simplify the transaction process, the transaction fees are not took into account. As shown in this picture, the red line presents the capital trading with our MI-CNN framework, while the blue line indicates the capital trading with Buy&Hold strategy, where we simply buy the index at the beginning of transaction and sell it at the end. The Buy&Hold strategy is usually considered as a benchmark to compare with different trading strategies. Comparing with

Buy&Hold strategy, our method obtains higher returns, exactly 59.16%. Overall, our system performs well on the SPY stock index.

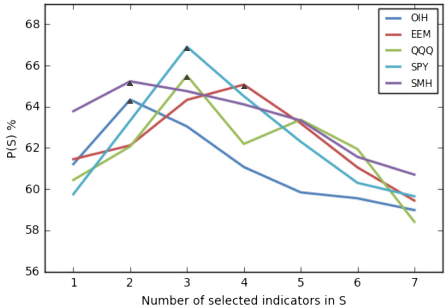


Fig. 3. MICFS results with different stock indices

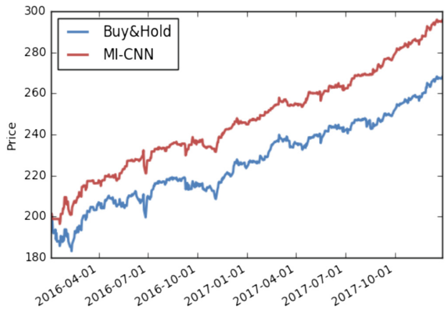
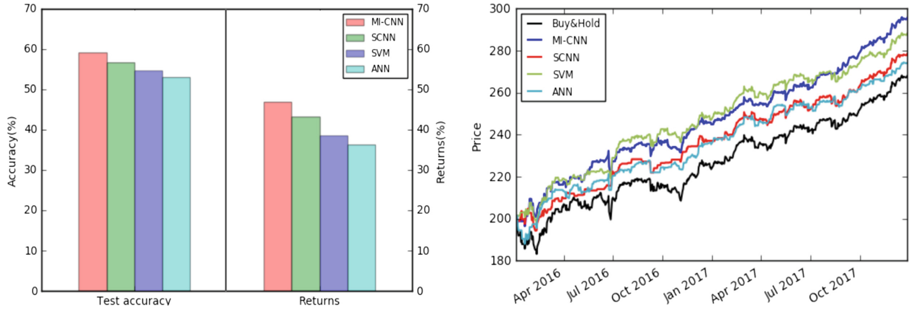


Fig. 4. Trading simulation on SPY

4.2 Comparison with Other Models

In this part, several popular prediction models are compared with our MI-CNN framework. The classifier that often involved in other decision support systems contains SVM, ANN and so on. These methods are used for contrasting with MI-CNN and the simulations are conducted on SPY with confirmed indicators. The structure of ANN is designed as a three-layer network, and the number of neurons in the hidden layer is determined at $2N + 1$ [3], where N is the dimension of input vectors. Finally, a single CNN (SCNN) structure is used to extract features in the indicator matrix, where the selected indicators are converted into a 2-D matrix together [15]. The contrast results are illustrated in Fig. 5(a). The accumulated capital curves are displayed in Fig. 5(b).

The histogram shows the proposed MI-CNN presents the best performance with both the test accuracy and returns. Then the SCNN model ranks in the



(a) Results comparation (b) Trading simulations

Fig. 5. Comparing results of different models

second and SVM ranks in the following. ANN performs worst that may caused by the vector-ordered representation of input data. Moreover, the high prediction accuracy, in general, conducts more fruitful returns in the real stock market. So the MI-CNN conducts the highest returns in the trading simulation.

4.3 Experiments on Other Stock Indices

To prove the adaptability of MI-CNN method with various stock indices, several stock indices in different industries are tested in this section. These stock indices are a good mix of large, mid, and small stocks, which has been introduced in [3]. In order to compare with the existing work in the latest research, the trading simulations are conducted on the dataset in [3]. The testing result with different stock indices are shown in Table 4. In terms of prediction accuracy, our MI-CNN is generally higher than their experiment results. The returns of trading simulation are better than theirs partly. We can see the accumulated returns

Table 4. Experiment results on different indices

Stock index	Accuracy in [3]	Accuracy of MI-CNN	Returns of Buy&Hold	Returns in [3]	Returns of MI-CNN
SPY	61.11%	**61.94%**	13.69%	24.75%	**27.37%**
EEM	55.56%	**60.31%**	17.07%	**36.48%**	33.04%
EFA	54.76%	**61.50%**	8.13%	30.09%	**32.37%**
FXI	52.78%	**58.71%**	4.54%	28.07%	**35.75%**
IWM	53.79%	**59.11%**	25.60%	**31.48%**	29.01%
OIH	53.57%	**57.90%**	21.71%	24.17%	**25.68%**
QQQ	58.33%	**60.69%**	19.29%	29.09%	**32.31%**
SMH	52.78%	**60.07%**	18.74%	29.66%	**33.09%**
Average	55.33%	**60.02%**	16.09%	29.22%	**31.07%**

conducted with our MI-CNN framework are universally higher than that with Buy&Hold strategy. The average prediction accuracy and returns of MI-CNN are 60.02% and 31.07% respectively.

5 Conclusion

In this paper, we propose a CNN-driven multi-indicator framework for stock index movement prediction. We employ MICFS to select effective indicators for different stock indices by considering the correlation between indicators and labels, and considering the redundancy between different indicators simultaneously. Feature extraction is carried out for several selected indicators respectively, which avoids interference between different indicators. For multi-indicator, we design a multi-channel CNN structure to extract underlying features from different indicators. In the experiments, the average prediction accuracy and average returns achieve 60.02% and 31.07% respectively. Experimental results demonstrate that our MI-CNN framework can successfully select features and effectively improve the prediction and transaction results.

There is much room for improvement in the future works. On the one hand, the source and representation of input data are potentially diverse. Finance data and news text could be taken into account. On the other hand, the features extracted by CNN can be considered as an important reference for different trading strategies.

Acknowledgements. This work was supported by: (i) National Natural Science Foundation of China (Grant No. 61602314); (ii) the Natural Science Foundation of Guangdong Province of China (Grant No. 2016A030313043); (iii) Fundamental Research Project in the Science and Technology Plan of Shenzhen (Grant No. JCYJ20160331114551175).

References

1. Bollerslev, T.: Generalized autoregressive conditional heteroskedasticity. J. Econom. **31**(3), 307–327 (1986)
2. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control. Wiley, Hoboken (2015)
3. Chiang, W.C., Enke, D., Wu, T., Wang, R.: An adaptive stock index trading decision support system. Expert Syst. Appl. **59**, 195–207 (2016)
4. Donaldson, R.G., Kamstra, M.: An artificial neural network-garch model for international stock return volatility. J. Empirical Finance **4**(1), 17–46 (1997)
5. Enke, D., Thawornwong, S.: The use of data mining and neural networks for forecasting stock market returns. Expert Syst. Appl. **29**(4), 927–940 (2005)
6. Gudelek, M.U., Boluk, S.A., Ozbayoglu, A.M.: A deep learning based stock trading model with 2-D CNN trend detection. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8, November 2017. https://doi.org/10.1109/SSCI.2017.8285188

7. Gunduz, H., Yaslan, Y., Cataltepe, Z., Gunduz, H., Yaslan, Y., Cataltepe, Z.: Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. Knowl.-Based Syst. **137**, 138–148 (2017)
8. Guresen, E., Kayakutlu, G., Daim, T.U.: Using artificial neural network models in stock market index prediction. Expert Syst. Appl. **38**(8), 10389–10397 (2011)
9. Huang, C.L., Tsai, C.Y.: A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. Expert Syst. Appl. **36**(2), 1529–1539 (2009)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
11. Lee, M.C.: Using support vector machine with a hybrid feature selection method to the stock trend prediction. Expert Syst. Appl. **36**(8), 10896–10904 (2009)
12. Malagrino, L.S., Roman, N.T., Monteiro, A.M.: Forecasting stock market index daily direction: a Bayesian network approach. Expert Syst. Appl. **105**, 11–22 (2018)
13. Oriani, F.B., Coelho, G.P.: Evaluating the impact of technical indicators on stock forecasting. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8. IEEE (2016)
14. Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M., Sabeti, P.C.: Detecting novel associations in large datasets. Science **334**(6062), 1518 (2011)
15. Sezer, O.B., Ozbayoglu, A.M.: Algorithmic financial trading with deep convolutional neural networks: time series to image conversion approach. Appl. Soft Comput. **70**, 525–538 (2018)
16. Su, C.H., Cheng, C.H.: A Hybrid Fuzzy Time Series Model Based on ANFIS and Integrated Nonlinear Feature Selection Method for Forecasting stock. Elsevier Science Publishers B. V., Amsterdam (2016)
17. Swingler, K.: Applying Neural Networks: A Practical Guide. Morgan Kaufmann, Burlington (1996)
18. Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Forecasting stock prices from the limit order book using convolutional neural networks. In: Business Informatics, pp. 7–12 (2017)
19. Wen, Q., Yang, Z., Song, Y., Jia, P.: Automatic stock decision support system based on box theory and SVM algorithm. Expert Syst. Appl. Int. J. **37**(2), 1015–1022 (2010)