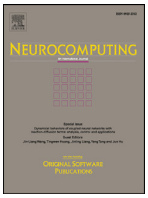
Neurocomputing 347 (2019) 46–58

Contents lists available at ScienceDirect

Neurocomputing

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



Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning

Qili Wang[a], Wei Xu[a,*], Xinting Huang[b], Kunlin Yang[a]

[a] School of Information, Renmin University of China, Beijing 100872, PR China
[b] Department of Computing and Information Systems, The University of Melbourne, Melbourne, VIC 3010, Australia

ARTICLE INFO

Article history:
Received 10 April 2018
Revised 26 February 2019
Accepted 2 March 2019
Available online 15 March 2019

Communicated by Dr. P.N. Suganthan

Keywords:
Stock price manipulation
Deep learning
Ensemble learning
Machine learning
Fraud detection

ABSTRACT

With the rapid development of the stock markets in developing countries, determining how to efficiently detect stock price manipulation activities to protect the interests of ordinary investors is really an important problem. Previous studies have introduced machine learning techniques into stock price manipulation detection and achieved better experimental results than traditional multivariate statistical techniques. Some characteristic features show statistically significant differences between manipulated and non-manipulated stocks, but this complementary information has rarely been considered in the manipulation detection model. The main contribution of our research work is the design of a novel RNN-based ensemble learning (RNN-EL) framework that combine trade-based features derived from trading records and characteristic features of the list companies to effectively detect stock price manipulation activities. Based on prosecuted manipulation cases reported by the China Securities Regulatory Commission (CSRC), we built a specific dataset containing labeled samples with trading data and characteristic information to conduct empirical experiments. The experimental results show that our proposed method outperforms state-of-the-art approaches in detecting stock price manipulation by an average of 29.8% in terms of AUC value. The managerial implication of our work is that government regulators can apply the proposed methodology to efficiently identify suspicious trading behaviors among huge amounts of trading activities in time to take action to ensure a fair trading environment.

1. Introduction

Concerns over stock price manipulation have been growing in both developed and emerging markets for the last several years [1,2]. A delicately designed manipulation scheme that aims to avoid regulation and surveillance might be intricate and elusive for detection [3,4]. Hart [5] discussed the feasibility and conditions of manipulation, which may be the first documented research that investigated manipulation from a theoretical perspective. Stock price manipulation is generally classified into three basic categories: information-based, action-based and trade-based manipulation [6]. Information-based manipulation occurs when misleading information or rumors are released intentionally to influence the stock price. Action-based manipulation is carried out by actions that influence the perceived or actual value of the listed company. In trade-based manipulation, manipulators try to manipulate the stock price by strategically buying and selling it, instead of taking publicly observable actions or releasing misleading information to influence the stock price. Among these three kinds of manipulation, trade-based manipulation is the most common type and also the most difficult one to detect.

Detecting stock price manipulation has great significance for protecting the interests of ordinary investors by ensuring a fair trading environment [7]. However, the lack of effective detection models causes challenges for regulators to efficiently identify suspicious trading behaviors among huge amounts of trading activities in time to take action. An anti-fraud supervision model with sufficient competence to provide enhanced manipulation detection accuracy can significantly reduce regulatory costs [8,9]. Meanwhile, recent developments in machine learning (ML) techniques have elevated the potential for classification in anomaly detection and solved some complex financial anti-fraud problems [10–13]. A few pioneering papers have tried to introduce ML techniques in stock market manipulation detection. Öğüt et al. [14] compared the performances of classification accuracy and sensitivity on detecting stock price manipulation for different algorithms and found that the data-mining models perform better than traditional statistical algorithms. Diaz et al. [15] applied decision trees and knowledge discovery techniques to detect stock price manipulation. Their proposed analytical model achieved better results in classifying

[*] Corresponding author.
E-mail address: weixu@ruc.edu.cn (W. Xu).

https://doi.org/10.1016/j.neucom.2019.03.006
0925-2312/© 2019 Elsevier B.V. All rights reserved.

trades and identifying new fraud patterns associated with stock price manipulation. These remarkable experimental results inspire researchers to incorporate advanced ML methods to improve manipulation detection capabilities.

However, most existing research ignored the fact that stock trading data are complex multi-dimensional time series which consist of price, trading volume, stock returns and so on. Thus, the high relevance and dependence between different points of the time series result in the inappropriateness of simply inputting data into fixed-size networks for results. To take advantage of the properties of time series, we attempt to apply a deep learning algorithm, recurrent neural network (RNN), to investigate hidden patterns of stock price manipulation activities, relying on multi-dimensional time-series trading data. Depending on their internal memory capacity, RNNs have been proved well suited for learning from experience to process, classify and predict time series and achieve outstanding results in fields such as handwriting recognition and speech recognition [16–20]. Moreover, existing stock price manipulation detection methods only utilized trading data to classify suspicious trading activities. Previous studies have investigated what characteristics would lead stocks more susceptible to successful manipulation [21–23]. Some characteristic features show statistically significant differences between manipulated and non-manipulated stocks, but this complementary information has rarely been considered in the manipulation detection model.

In this study, we propose a novel RNN-based ensemble learning (RNN-EL) framework for stock price manipulation detection with trade-based features derived from trading records and characteristic features of the list companies. Deep learning algorithms for time-series modeling and ensemble learning techniques are both incorporated in the proposed framework. We conduct empirical experiments on a specific dataset that is built based on prosecuted manipulation cases reported by the China Securities Regulatory Commission (CSRC) to address challenges related to trade-based stock price manipulation detection in China which has emerging stock markets. Our research work is able to fill the aforementioned research gaps.

The rest of this paper is organized as follows. In Section 2, we review the related work on stock price manipulation and manipulation detection method and discuss the main differences between our work and previous studies. In Section 3, we introduce a novel ensemble learning framework that combines trading data and characteristic data for stock price manipulation detection in Chinese stock markets. To assess the effectiveness and efficiency of the proposed framework, empirical analysis is performed and reported in Section 4. Finally, Section 5 offers concluding remarks and suggests future directions of research work.

2. Literature review

2.1. An overview of stock price manipulation

Stock price manipulation has attracted the attention of researchers all over the world, and the comprehensive effect of manipulation has been discussed and analyzed in past years. Allen and Gale [6] firstly classified stock price manipulation into the three basic categories mentioned above, giving clear definitions of different manipulation types and providing more specific research directions for later studies. The history of stock price manipulation is also introduced in their article. In addition, some researchers focused on the complete manipulation process and built models to investigate the realization of stock price manipulation. Van Bommel [24] proposed a model to investigate information-based manipulation and showed how manipulators earned abnormal returns through market rumors. Dissanaike and Lim [25] also used a four-phased model to analyze stock trading data from Asian markets to quantify the extent of illicit profits earned. A model was developed by Chakraborty and Yılmaz [26] to show how informed insiders can manipulate a stock, which described the trading and timing tactics according to the motivation of manipulators. Based on the assumption of trading patterns of manipulation, researchers further estimated the general effect of manipulation and the abnormal returns earned by manipulators. Aggarwal and Wu [21] applied theoretical work to conduct an empirical analysis of market manipulation cases. Their research indicated that, during the manipulation period, manipulation is typically accompanied by higher stock liquidity, volatility and abnormal returns.

2.2. Stock price manipulation detection

Realizing the harm of stock price manipulation, many researchers have devoted their efforts to stock price manipulation detection. Felixson and Pelli [27] analyzed closing price manipulation with a simple regression model in the Finnish stock market. Mahoney [28] investigated the details of prosecuted cases in the NYSE and applied a significance test to examine the features of stock price manipulation. Palshikar and Bahulkar [29] tried to manage the problem of manipulation detection and developed a fuzzy temporal logistic model for surveillance in the stock market. Pirrong [30] applied standard statistical techniques to detect manipulation and pointed out that the regulation of manipulation might be inefficient in securities and futures markets. With the expansion of the application range of computer techniques, manipulation detection methods gradually broadened and were no longer restricted to statistical and econometric models. Palshikar and Apte [31] applied graph clustering algorithms and validated the potential of introducing interdisciplinary methods into the research of manipulation detection.

With the development of artificial intelligence and big data analytics, it is a remarkable fact that prediction and detection models based on data-mining and machine-learning techniques have been widely applied in other areas of finance and achieved great success. However, the literature on attempts to detect stock price manipulation based on machine-learning is scarce. Öğüt et al. [14] are the first to introduce data-mining techniques to stock price manipulation detection. Their empirical results showed that the data-mining models perform better than traditional statistical algorithm in manipulation detection. Diaz et al. [15] applied decision trees and knowledge discovery techniques to detect stock price manipulation. The proposed analytical model achieved better results in classifying trades and identifying new fraud patterns associated with market manipulation. Cao et al. [32] proposed a model called the hidden Markov model with abnormal states to detect price manipulation activities in stock markets and to identify the specific type of the detected manipulation. The results of performance evaluation showed that all synthetic manipulation cases had been successfully detected.

2.3. The main differences between our work and previous studies

Our work differs from the previous studies in four ways. First, while previous studies mainly used trading data to detect stock price manipulation, we include both trade-based features and characteristic features in our detection model. Second, though machine learning techniques were applied to detect stock price manipulation, none of the previous studies explored the recurrent neural network, which is a deep learning algorithm that is well suited for analyzing time series to investigate hidden patterns of manipulation behaviors relying on multi-dimensional time series trading data. Third, most of the previous studies were conducted based on synthetic data or manipulation cases in developed countries. Our work aims to detect price manipulation activities in Chinese stock

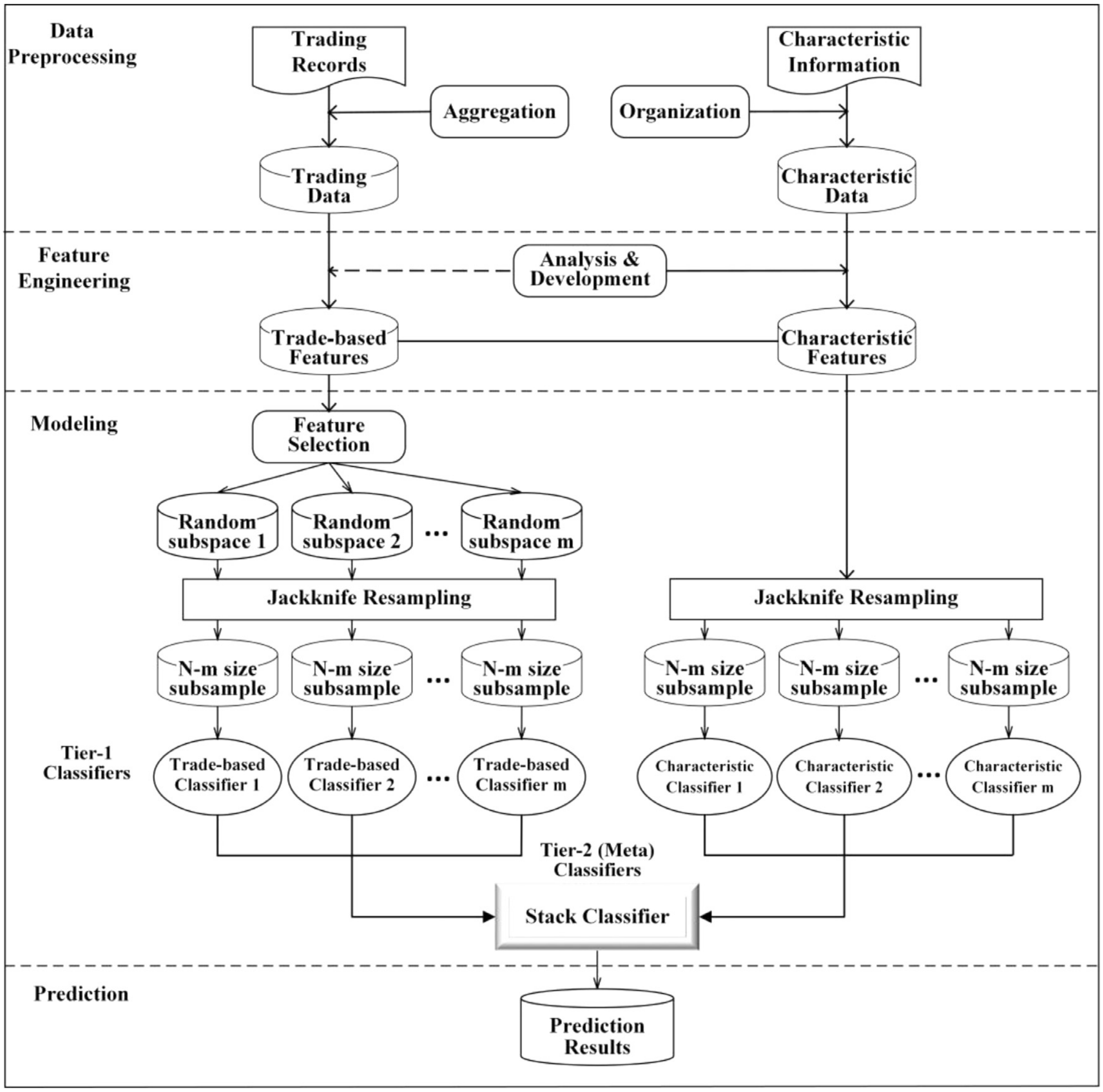


Fig. 1. An ensemble learning framework for stock price manipulation detection.

markets based on cases pursued by the CSRC. Fourth, to the best of our knowledge, this is the first successful research work that incorporates deep learning algorithms and ensemble learning techniques for stock price manipulation detection.

3. An ensemble learning framework for stock price manipulation detection

Stock price manipulation detection could be regarded as a unique type of pattern recognition. On the one hand, the manipulation behaviors are hidden in daily trading records and reflected in the stock trading data indicators. On the other hand, the characteristic information about a company would influence the possibility of its stock being manipulated, which has been studied and validated in previous research. Based on this current background, for the proposed framework, we take both trading records and characteristic information of stocks into account for stock price manipulation detection. In particular, we develop a RNN-based ensemble learning (RNN-EL) model which combines trade-based features and characteristic features for more effective mining of manipulation behaviors. The proposed framework for manipulation detection is outlined in Fig. 1. The framework consists of four main processes, namely data preprocessing, feature engineering, modeling and prediction.

3.1. Data preprocessing

According to the previous work of other researchers, stock price manipulation leads to anomalies of various stock trading data variables, including turnover rate, volatility, returns, and abnormal returns, which in a way demonstrates the feasibility of detecting manipulation using multi-dimensional stock trading data. According to a case study based on cases issued by the CSRC, stock price manipulation is usually conducted by means of intraday high-frequency trades and large-scale subscription in Chinese stock markets. To utilize the information reflected by intraday trades, daily stock trading records are collected. However, each stock has a different transaction history over a given period, so it is unfeasible to directly extract features from trading records. Through data preprocessing, we need to make the time series of trading data derived from trading records of different stocks more comparable. One common practice is to divide the whole trading period into certain time intervals so that dynamic stock trading records are transformed into stock trading data sequences according to time stamps. Diaz et al. [15] divided the six stock trading hours

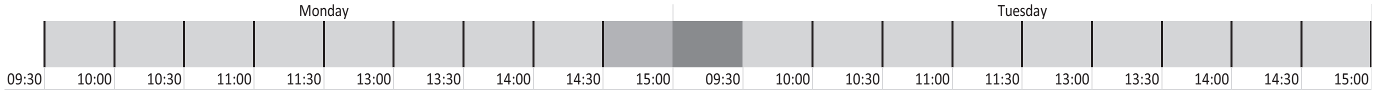


Fig. 2. Example of time period edge aggregation.

into six intervals and analyzed hourly data of manipulation and non-manipulation. Because the Chinese stock market only has four trading hours in each trading day, the sequence length of trading data would be four using hourly data, which is too short to take advantage of a recurrent neural network. Dividing the trading period into seconds is also inapplicable because time gaps between trading records may be a few seconds, which would lead to a data sparseness problem. After thorough consideration, we set the minimum time interval to one minute and dataset testing shows that all stocks have transactions during each minute.

In addition, some studies in the literature empirically investigate what characteristics would lead stocks to be more prone to being manipulated, which inspired us to add characteristic information of stocks into the manipulation detection model. Characteristics of a stock, such as market capitalization, industry, and fund holding, cannot directly show us whether the stock is being manipulated but are helpful when predicting the probability of being manipulated. Data preprocessing is also conducted on characteristic information of stocks to obtain characteristic data. For variables that can be constantly updated, such as market capitalization, we choose the data at the opening of the trading day; for other variables, we extract them from the most recently released announcement before the trading day.

3.2. Feature engineering

3.2.1. Trade-based features

All trading behaviors are hidden in the stock trading data, and manipulation behavior is no exception. Based on empirical research, Aggarwal and Wu [21] found that, during the manipulation period, stock price manipulation activities are usually associated with greater stock liquidity, greater volatility and higher returns, which implies the possibility of detecting manipulations using indicators derived from trading data.

Previous empirical studies of stock price manipulation detection were generally conducted with trading data. Öğüt et al. [14] took the differences in average daily return, volatility and trading volume between manipulated stock and indexes and built manipulation detection models with them as explanatory variables. In the study of Diaz et al. [15], four indicators representing stock returns, abnormal returns, volatility and liquidity are included in the detection model. There are also detection models focusing on stock prices and their relative changes to identify suspicious manipulated stocks [25,32–34]. Stock trading data are multi-dimensional time series, and stock price is only one representative of the stock trading data. To better explore the hidden information in trading data, other dimensions of the data are taken into account in our study.

Since we have already set the minimum time interval to one minute, a suitable feature extraction method is still needed to better describe the behavior pattern of stocks, which should achieve certain functions like better reflecting abnormal trades and determining mutation point and manipulation period. Instead of just using one time interval, we apply an aggregation strategy on trading records to obtain different time granularities, which could characterize trades from different respects. Fine granularity is suitable for recurrent neural network to detect subtle changes and help detect possible critical points of manipulation while coarse granularity could provide overall situation descriptions [35]. However, how much to accumulate while aggregating trades of a certain stock remains a critical question on account of the marginal decline during time passing. For example, setting the time interval by two hours can hardly provide more information than setting it to one hour because the information value declines with the expansion of time period. After several experimental results, we set the upper bound of time interval to 60 min and thus set the interval on a scale of 1–60. Specifically, we adopt the interval of 1 min, 2 min … 60 min and use all the 60 intervals to calculate our features.

Because of the discontinuity of stock trading time between days, when constructing the aggregated features, some information of trades at the border of a trading time period cannot be captured completely. For example, at the beginning of the trading hour (such as 10:00 a.m.), we cannot get the feature of the last 60 min since the stock market begins at 9:30 a.m. We propose to resolve this limitation by assembling each days' stock trading hour to make it continuous. When dealing with the beginning of a period, we take its former period's data into consideration to complete the missing. An example is shown in Fig. 2. If the time is 10:00 on Tuesday, since the stock market begins at 9:30 on trading days, we can combine the last half trading hour on Monday and the first half trading hour on Tuesday to make up the complete 60 min data.

After the determination of time interval, we also use subtraction and division to process our feature sets. In each feature set, which has the same time interval, we use the current time period to subtract or divide its preceding one to get the difference and ratio of them. Applying preprocessing methods and aggregating trading records, we calculate primitive features like volatility, turnover rate, stock returns and abnormal returns for each time interval according to trades made during that interval to get features. For example, VOL_{01} represents the volatility in the interval of 1 min and $\mathrm{VOL}_{\mathrm{T}}(\mathrm{t})$ means the volatility at time t of time interval T. Our estimation of volatility is the standard deviation of the stock returns. Detailed definitions of trade-based features extracted from trading data are introduced in Table 1.

3.2.2. Characteristic features

In addition to trading data, characteristic data of stocks are considered in our detection model. Empirical studies have shown that some characteristic features have statistically significant differences between manipulated and non-manipulated stocks. Aggarwal and Wu [21] studied the relation between stock market and manipulations and drew a conclusion that manipulation cases tend to occur in relatively inefficient markets. Imisiker and Tas [22] collected manipulated cases from the Istanbul Stock Exchange and used some firm-specific characteristics as explanatory variables to explain which stocks are prone to be manipulated. Probit regression results of their empirical experiments showed that firms with smaller market capitalization, higher leverage ratio and a lower free float rate are more likely to be prey to manipulators. Comerton-Forde and Putniņš [23] found that stocks with less liquidity and high degree of information asymmetry are more prone to manipulation. Bid-ask spread and whether a stock is in the market index would also affect the chance to be manipulated. Apart from those features, they also pointed out that some manipulations might profit from options, especially at the time prior to expiry. The results of these previous experiments inspire us to include characteristic features of stocks in our manipulation detection model. Considering that the properties of companies and

Table 1
Trade-based features.

Primitive Feature	Features	Definition	Quantity
Volatility (*VOL*)	VOL_{01}, VOL_{02}, ... VOL_{60}	Standard deviation of the stock returns in this time interval	60
Difference	$\mathrm{VOL}_{\mathrm{T}}(t)$–$\mathrm{VOL}_{\mathrm{T}}(t-1)$	The difference between current volatility with preceding one	60
Ratio	$\mathrm{VOL}_{\mathrm{T}}(t)/\mathrm{VOL}_{\mathrm{T}}(t-1)$	The ratio of current volatility and preceding one	60
Turnover (*TO*)	TO_{01}, TO_{02}, ...TO_{60}	The ratio of volume traded in this time interval to the circulation share	60
Difference	$\mathrm{TO}_{\mathrm{T}}(t)$–$\mathrm{TO}_{\mathrm{T}}(t-1)$	The difference between current turnover with preceding one	60
Ratio	$\mathrm{TO}_{\mathrm{T}}(\mathrm{t})/\mathrm{TO}_{\mathrm{T}}(t-1)$	The ratio of current turnover and preceding one	60
Returns (*RET*)	RET_{01}, RET_{02}, ...RET_{60}	The ratio of the stock price increase in this time interval	60
Difference	$\mathrm{RET}_{\mathrm{T}}(t)$–$\mathrm{RET}_{\mathrm{T}}(t-1)$	The difference between current returns with preceding one	60
Ratio	$\mathrm{RET}_{\mathrm{T}}(t)/\mathrm{RET}_{\mathrm{T}}(t-1)$	The ratio of current returns and preceding one	60
Abnormal Returns (*AR*)	AR_{01}, $\mathrm{AR}_{02,}$ ...AR_{60}	The ratio of the stock price increase minus the ratio of index increase in this time interval	60
Difference	$\mathrm{AR}_{\mathrm{T}}(t)$-$\mathrm{AR}_{\mathrm{T}}(t-1)$	The difference between current abnormal returns with preceding one	60
Ratio	$\mathrm{AR}_{\mathrm{T}}(t)/\mathrm{AR}_{\mathrm{T}}(t-1)$	The ratio of current abnormal returns and preceding one	60
Total			720

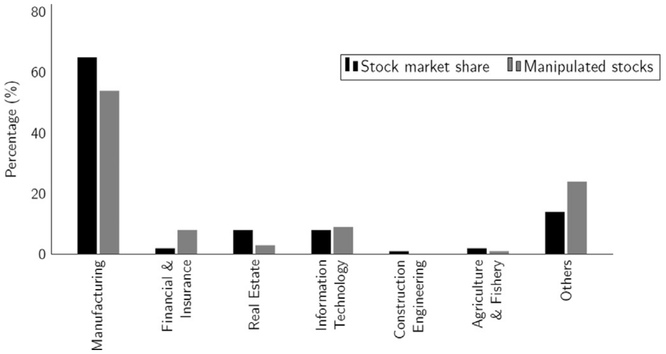


Fig. 3. Proportion of market share and manipulated stocks in different industries.

stocks may have an impact on the possibility of being chosen by manipulators, we take into account some variables that reflect the characteristics of the company and shareholdings of the stock. After preprocessing the data, we find that manipulators might have preference to stocks in certain industry on result of the inconsistence between stock market share and manipulated stocks when grouped by industry. Some industries results are shown in Fig. 3. We can tell from the graph that stocks in financial or insurance industry, for example, are more likely to be chosen by manipulators while stocks in manufacturing industry are relatively less likely to be manipulated, which inspire us to distinguish the industry differences in model settings.

Similarly, we select other characteristic features through literature review and statistical analysis. Like trade-based features, we also use subtraction and division to calculate those new features. As mentioned above, we collected data from the beginning of the trading day or released announcement so we choose the interval according to how we extract data. The specific selection and definitions of the characteristic features are listed in Table 2.

3.3. Modeling

As described above, two types of stock data, trading data and characteristic information, are combined for manipulation detection. The data of stock i at time t are organized as $S_{i,t} = (X_{i,t}, Y_i)$, where $X_{i,\ t}$ represents the $N_X \times 1$ vector of trade-based features at time t and Y_i represents the $N_Y \times 1$ vector of the characteristic features introduced in Section 3.2. N_X and N_Y represent the total number of trade-based features and characteristic features, respectively. We transform the detection problem to the problem of estimating the probability that stock i is manipulated given the relevant trading data and characteristic information:

$$\begin{aligned} &P(Stock\ i\ is\ manipulated|S_{i,t-T}, S_{i,t-T+1}, \ldots, S_{i,t-1}) \\ &\quad = P(Stock\ i\ is\ manipulated|X_{i,t-T}, X_{i,t-T+1}, \ldots, X_{i,t-1}, Y_i), \end{aligned} \tag{1}$$

where T refers to the time window length of previous inputs considered in the model. Upper-case Σ is used as a symbol for enumeration operator instead of summation operator here. According to the estimate of probability, the detection model would give an alert when a stock is at high risk of being price manipulated.

As shown in Fig. 1, each component of the modeling part is intended to enhance manipulation detection capabilities with the help of advanced machine learning algorithms and ensemble learning techniques [36,37]. The characteristic features are used as inputs for different characteristic classifiers which are built based on supervised machine learning models. As we want to investigate hidden patterns of stock price manipulation behaviors based on multi-dimensional time-series trading data, recurrent neural networks are served as trade-based classifiers in virtue of their internal memory capacity.

The architecture of RNN model used in this study is the Elman network [38], which is presented in Fig. 4. The model can also be unfolded over time as a deep network as shown in Fig. 5. The input layer vector $X(t)$ in the recurrent part consists of several trade-based features. The value of the neuron in context layer $Q(t)$ is used as an extra input for all of the neurons in the hidden layer R one time step later. Vector R is calculated by combining vectors $X(t)$ with the weight matrix W_{R-X} and $Q(t)$ with the weight matrix W_{R-Q}. The jth unit of the hidden layer vector R at time t, $R^j(t)$, is computed as follows.

$$R^j(t) = f\left(\sum_{i=1}^{N_X} W_{R-X}^{ji} X^i(t) + \sum_{i=1}^{N_Q} W_{R-Q}^{ji} Q^i(t)\right), \quad j = 1,\ 2,\ \ldots,\ N_R\ , \tag{2}$$

where W_{R-X}^{ji} and W_{R-Q}^{ji} denote the weights associated with the jth unit of R and the ith unit of $X(t)$, with length N_X, and $Q(t)$, with length N_Q, respectively. The number of neurons in hidden layer R is N_R. $R(0)$ is defined as an N_R-dimensional zero vector.We take a sigmoid function as the activation function f,

$$f(x) = \frac{1}{1+e^{-x}}. \tag{3}$$

Vector $Q(t)$ is equal to vector $R(t-1)$, so the formula above can be transformed as follows.

$$R^j(t) = f\left(\sum_{i=1}^{N_X} W_{R-X}^{ji} X^i(t) + \sum_{i=1}^{N_R} W_{R-Q}^{ji} R^i(t-1)\right), \quad j = 1,\ 2,\ \ldots,\ N_R. \tag{4}$$

Table 2
Characteristic features.

Features	Definition	Quantity
Market capitalization	Market price of one share multiplied by the number of ordinary shares in issue. The amount of shares is updated after a capital change.	3
Difference	The difference of market capitalization between the last two capital changes.	
Ratio	The ratio of market capitalization between the last two capital changes.	
Share proportion of large shareholders	The proportion of the total shareholdings of the top ten shareholders among all the shareholders. The large shareholders list is updated every three months.	3
Difference	The difference of share proportion of large shareholders between the last two updates.	
Ratio	The ratio of share proportion of large shareholders between the last two updates.	
Stock circulation share ratio	The proportion of circulation shares in the total share capital. This indicator is updated whenever the total amount of tradable shares is changed.	3
Difference	The difference of stock circulation share ratio between the last two updates.	
Ratio	The ratio of stock circulation share ratio between the last two updates.	
Fund holding	Percentage of shares held by funds and institutions. Calculated for each stock annually based on funds' and institutions' compulsory bulletins.	3
Difference	The difference of fund holding percentage between the last two updates.	
Ratio	The ratio of fund holding percentage between the last two updates.	
Bid-ask spread	The difference between the ask and best bid in the former day and it is updated every trading day.	3
Difference	The difference of bid-ask spread between the last two trading days.	
Ratio	The ratio of bid-ask spread between the last two trading days.	
State-owned enterprise	Dummy variable that indicates whether this company is a state-owned enterprise.	1
Market index stock	Dummy variable that indicates whether this stock is included in market index.	1
Listed options	Dummy variable that indicates whether this stock has listed options.	1
Expiry	Dummy variable that indicates whether it is the last trading day before its expiry if the stock has listed options.	1
Industry type	Industry ID to represent the industry of a stock (one-hot encoding is used and assume there are n types).	n
Stock market type	Stock market ID to represent the market of a stock (one-hot encoding is used and assume there are m types).	m
Total		$n+m+19$

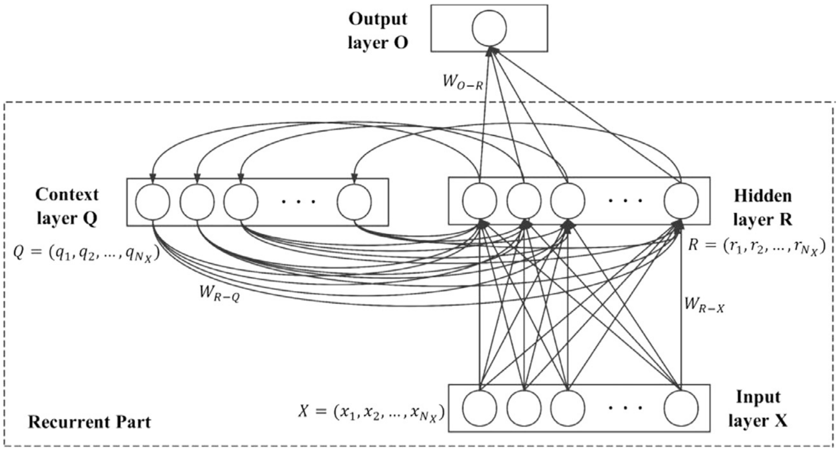


Fig. 4. The architecture of recurrent neural networks.

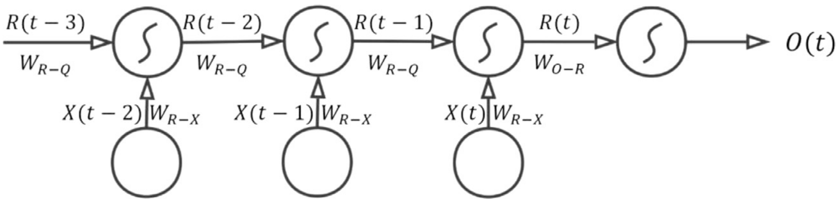


Fig. 5. Unfolding the RNN model as a deep network.

Vectors $R(t)$ is connected to the output layer O through the weight matrix W_{O-R}. At time t, output of the RNN model, $O(t)$, can be obtained as follows.

$$O(t) = f\left(\sum_{i=1}^{N_R} W_{O-R}^{i} R^{i}(t)\right), \tag{5}$$

where W_{O-R}^{i} denote the weights associated with the unit of the output layer O and the ith unit of $R(t)$.

Backpropagation is performed during the process of backward pass, which computes the gradients from the output to the input using the chain rule. For model training, stochastic gradient descent is applied using the backpropagation through time (BPTT) algorithm according to the characteristics of the proposed model structure. The training process continues until all weight matrices reach convergence. Denoting the actual judgment of a training sample at time t as $V(t)$, the error value $E_O(t)$ is defined as the difference between $V(t)$ and $O(t)$,

$$E_O(t) = V(t) - O(t)\ . \tag{6}$$

In order to avoid model over-fitting and improve the generalization ability of the proposed method, we add the L2-norm regularization terms to update weight matrices of the network. The weight matrix W_{O-R} is updated as

$$W_{O-R}(t+1) = \alpha E_O(t) \times R(t)^T + (1-\beta)W_{O-R}(t), \tag{7}$$

where α is the learning rate and β is the regularization parameter.

Then, the gradient of the error vector in the hidden layer R at time t, $E_R(t)$, are computed with the error value of the previous step as

$$E_R(t) = W_{O-R}(t+1)^T \times E_O(t) \odot ((1-R(t)) \odot R(t)), \tag{8}$$

where $\odot$ denotes element-wise vector multiplication. Next, we use the backpropagation through time learning algorithm to update W_{R-X} and W_{R-Q}. The formulas are as follows.

$$W_{R-X}(t+1) = \alpha \sum_{\tau=0}^{T} E_R(t-\tau)X(t-\tau)^T + (1-\beta)W_{R-X}(t), \tag{9}$$

$$W_{R-Q}(t+1) = \alpha \sum_{\tau=0}^{T} E_R(t-\tau)R(t-1-\tau)^T + (1-\beta)W_{R-Q}(t), \tag{10}$$

where T is the number of unfolded back steps.

After the network has been trained and established, the output, $O(t)$, is defined as the estimate of the probability that the stock is manipulated given the trading records and characteristic information.

In the process of feature selection, bootstrap method is applied to randomly sample N trade-based features ($N < N_X$) to construct a random subspace. For each random subspace, we train a RNN model and build a deep learning classifier for time-series learning. Moreover, each tier-1 classifier, whether trade-based or characteristic, is trained on a unique $N - m$ size subsample (m refers to the number of trade-based classifiers and characteristic classifiers) generated through jackknife resampling technique. This double disturbance strategy is employed to keep more differences in the ensemble system to improve procedural bias through disturbance of both the samples and the feature spaces. Then the classification results from the tier-1 classifiers are used as inputs for the tier-2 classifier, or the stack classifier.

3.4. Prediction

Our work aims to investigate hidden patterns of manipulation behaviors relying on multi-dimensional stock data and predict the possibility of being manipulated to help detect stock price manipulation. For our case dataset, each case is manually marked with a class label “1” if manipulation occurs; otherwise, it will be marked with a different class label “0”. These labels for each case represent the ground truths of the predictive tasks. In order to avoid serious learning biases caused by an imbalanced training set, a balanced class distribution is maintained [39,40]. Considering that there may exist historical dependencies in data, we split the dataset into a training-validation set and a test set by time, taking the latest cases for model testing. In this way, we can better investigate the generalization ability of the fraud detection models. After we have finished the whole training process, a RNN-based ensemble learning classifier is established to predict whether a case from the test set has been manipulated. Then, we compare the prediction results with the pre-marked labels to evaluate the effectiveness of our proposed framework.

4. Empirical analysis

4.1. Data description

We collected stock price manipulation cases reported on the CSRC website from 2012 to 2016. To be specific, all reports of administrative penalties that contain the word “manipulation” were manually identified. The announcements of penalty decisions for manipulation cases contain not only the basic information of the manipulation, such as the trading venue, the stock name and code, and the relevant dates, but also a statement about how the manipulation process was conducted, which helps us categorize the type of manipulation and determine the time window of manipulation realization. After classifying the manipulation types for all price manipulation cases, we selected all of the trade-based cases, which account for more than 90% of all of the CSRC reported cases. Moreover, given that our research focuses on intraday manipulation detection, some cases for which the precise manipulation dates are not reported on the announcements from CSRC are eliminated. The data of all selected manipulation cases formed the manipulated cases dataset. The total number of included CSRC reports is 40, involving 33 single manipulators or groups, 64 stocks and 257 manipulated cases.

In order to complete the training of the detection model, it is necessary to find control samples of non-manipulated stocks. Considering that the characteristics may vary greatly from stock to stock in terms of industry, size or market capitalization, using stocks that are similar to the manipulated stocks as comparison data will improve the ability of the detection model to use context information and reverse bias. For each manipulated stock, we select a similar stock in the same exchange and industry following a set of criteria, which require that the differences in market capitalization and circulation shares between the similar stock and manipulated stock are both less than 5% and that there is no suspension during the manipulation process. In addition to similar stocks, we use SSE 50 Index constituent stocks for comparisons between manipulated and non-manipulated cases, which may not have similar characteristics. SSE 50 Index constituent stocks are the largest stocks of good liquidity and low risk of being manipulated from the Shanghai security market, which demonstrate the trading pattern in a sufficiently large market. By introducing the heavyweight as a supplement to the control sample, the overall market situation of every trading day is included in the analysis scope of the model. The more comprehensive and in-depth comparison ensures the ability of the detection model to identify cases of stock price manipulation. We randomly selected 10 SSE 50 Index constituent stocks as additional stocks. The number of selected similar and additional stocks is 74, and we collected the trading records and characteristic information of these stocks for the same 187 trading days when manipulations were conducted. The total numbers of similar cases and additional cases are 257 and 1870, respectively.

We use high-frequency intraday trading data of all stocks to construct a unique dataset, which contains more than 1 million trades from 2384 cases of 64 manipulated stocks, 64 similar stocks and 10 additional stocks, as mentioned above. Specifically, we only intercept trading records for the manipulation time period for each case, which is given in CSRC reports. The source of high-frequency trading data is the RESSET High-frequency Data System. All of the quote and transaction records of stocks are available for each trading day. As discussed in Section 3, we have 60 different time intervals. For each manipulated case, according to the certain operation time period (for example, if the manipulator conducted

Table 3
Summary statistics of trade-based features at an interval of 1 min.

Cases	Mean	Median	Std. dev	Skewness	Kurtosis	n
Manipulated cases						
Volatility	0.295%	0.157%	0.450%	7.46	172.19	40,909
Turnover	0.0182%	0.00641%	5.49	20.52	712.31	40,909
Returns	0.00381%	0.00178%	0.423%	4.01	194.60	40,909
Abnormal returns	0.00569%	0	2.31%	41.57	1793.78	40,909
Similar cases						
Volatility	0.201%	0.106%	0.333%	6.22	76.10	40,909
Turnover	0.00798%	0.00304%	2.15	15.37	405.60	40,909
Returns	0.000121%	0.00134%	0.306%	0.03	66.08	40,909
Abnormal returns	0.00201%	0	2.12%	45.77	2152.17	40,909
Additional cases						
Volatility	0.185%	0.156%	0.132%	1.51	5.26	1,480,364
Turnover	0.00311%	0.00122%	0.64	7.93	112.03	1,480,364
Returns	−0.00334%	0	0.159%	0.24	4.05	1,480,364
Abnormal returns	−0.00126%	0	3.64%	27.30	745.97	1,480,364

Table 4
Summary statistics of characteristic features (excluding the dummy and one-hot encoding variables).

Cases	Mean	Median	Std. dev	Skewness	Kurtosis
Manipulated cases					
Market capitalization (log)	3.933	3.882	0.294	0.772	0.822
Share proportion of large shareholders	57.73%	60.15%	15.32	−0.792	0.399
Stock circulation share ratio	79.33%	85.77%	21.73	−0.675	−0.809
Fund holding	35.07%	33.29%	26.62	0.767	1.066
Bid-ask spreads	2.84%	2.61%	1.32	0.755	0.851
Similar cases					
Market capitalization (log)	6.93	4.21	3.62	0.383	−1.86
Share proportion of large shareholders	59.72%	58.87%	16.95	0.165	−0.433
Stock circulation share ratio	75.28%	77.24%	20.66	−0.430	−0.899
Fund holding	42.58%	42.66%	25.69	0.254	−0.750
Bid-ask spreads	2.38%	0.91%	3.71	0.432	0.514
Additional cases					
Market capitalization (log)	11.20	11.14	0.365	0.617	−0.044
Share proportion of large shareholders	67.37%	70.33%	19.02%	−0.355	−0.485
Stock circulation share ratio	93.48%	100%	14.94	−2.863	8.621
Fund holding	58.79%	63.50%	25.49	−0.374	−0.868
Bid-ask spreads	2.23%	0.84%	0.94	0.367	0.421

manipulation actions between 9:55 a.m. and 10:30 a.m., this 35-min period is the manipulation time period), a multi-dimensional time series is built by aggregating corresponding trading records for each minute, and the data of the comparison stocks for the same time period are also collected and preprocessed accordingly. Thus, the total length of the derived labeled multi-dimensional time series of trading data is 377,890 (some of the manipulation time windows coincided), and these labeled time series are used for the training and testing of the recurrent part. Table 3 shows the statistics of trade-based features for the manipulated cases, similar cases and additional cases at an interval of 1 min.

Besides trading data, characteristic data are also analyzed in our model for comprehensive pattern recognition. Thus, we collect the characteristic information introduced in Section 3 from the Wind Database for each manipulated, similar and additional stock. Considering that a few features of characteristic information are only updated in quarterly or semiannual reports, we choose the data presented in the most recent specific report before the time period with which we are concerned (for example, if the manipulation date is May 7th, 2015, we will collect data on the share proportion of large shareholders from the company's first quarterly report in 2015 for characteristic information). As for bid-ask spreads, we calculate mean daily spread in the former day as the unweighted average. With adequate preparation of data for the recurrent and non-recurrent parts of the model, complete multi-dimensional data are obtained through carefully merging minutely time series data with characteristic data for each stock. The statistics of characteristic features for the manipulated cases, similar cases and additional cases are shown in Table 4.

4.2. Evaluation metrics

The test results are evaluated by a commonly-used classification performance measure, *F1* score, based on the confusion matrix.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

$$precision = \frac{TP}{TP + FP} \tag{12}$$

$$recall = \frac{TP}{TP + FN} \tag{13}$$

$$F1 = \frac{2 \ \times precision \times recall}{precision + recall} \tag{14}$$

where true positive (*TP*) refers to the number of manipulated samples that are correctly predicted, false positive (*FP*) is the number of non-manipulated samples that are judged as manipulated ones, and false negative (*FN*) is the number of manipulated samples that are predicted as non-manipulated ones. A predictive probability, which ranges from 0 to 1, of being manipulated is given for each sample by each classifier. Then, we set a certain cutoff

Table 5
Performances of trade-based classifiers.

Classifier	AUC	F1	Prec.	Rec.	Classifier	AUC	F1	Prec.	Rec.
ANN	0.657	0.219	13.6%	55.0%	Naïve Bayes	0.619	0.228	14.9%	48.9%
BayesNet	0.693	0.282	18.1%	64.2%	RF	0.721	0.246	15.3%	63.0%
C5.0 tree	0.670	0.233	16.3%	40.9%	RNN	0.760	0.264	16.2%	**70.3%**
KNN	0.643	0.214	13.4%	53.6%	RNN-RSE	**0.804**	**0.303**	**19.3%**	70.1%
LogitReg	0.699	0.266	16.7%	65.6%	SVM	0.611	0.215	14.0%	46.4%
					Improvement over the best baseline	11.5%	7.4%**	6.6%**	6.9%*

* p-value is less than 0.05 ** p-value is less than 0.01.

value, 0.5, to classify the test samples into a manipulated class or non-manipulated class. The *F1* score is the harmonic combination of precision and recall, which are two quantities that describe the predictive accuracy. The higher the *F1* score, the more accurate is the prediction. Besides, AUC (area under ROC curve) is adopted as an overall performance measure. The ROC (receiver operating characteristic) curve is a graphical plot that created by plotting the true positive rate (TPR) against the false positive rate (FPR) at different possible thresholds. The AUC is considered as a better overall performance measure than accuracy because it is independent of cutoff value [41]. The higher the AUC value, the better prediction performance a model achieves.

4.3. Experimental results

As introduced in Section 3, trade-based features are coupled with RNN models as theirs recurrent structure can remember former inputs of time-series data. Previous studies have mostly applied linear regression and logistic regression (LR) to detect stock price manipulation. Recently, some machine learning methods have obtained better results for financial fraud detection than traditional statistical techniques, including Naïve Bayesian, SVM and random forest (RF). Given the lack of consensus on the best manipulation detection method, we incorporated several popular classifiers in addition to linear regression and logistic regression. The selected classifiers are as follows: LR, Bayesian Network (BayesNet), ANN, SVM, KNN, C5.0 tree, RF and Naïve Bayes. These 8 classifiers were run as baseline classifiers in comparative experiments. In order to avoid overfitting classifiers, we employ the popular 10-fold cross-validation approach for model evaluation and model selection [42]. Ten equal-sized subsets of the training-validation set are generated. Then, nine subsets are used to train a classifier while the remaining one is used to test it. This procedure is repeated for 10 times, with each subset used nine times for training and once for testing. Automated tuning with grid search in parameter space is employed for parameter tuning to ensure the objectivity and fairness of the results of comparative experiments.

In this section, three groups of comparative experiments were conducted to assess the effectiveness of our proposed stock price manipulation detection framework, each of which evaluated the utility of one facet of the framework. Experiment group 1 (presented in Section 4.3.1) evaluated the performance of RNN model versus baseline models in detecting stock price manipulation with only trade-based features. We also tested the efficacy of using random subspace ensembles of RNNs (RNN-RSE). Experiment group 2 (presented in Section 4.3.2) assessed the effectiveness of including characteristic features as supplementary information in the manipulation detection model. Experiment group 3 (presented in Section 4.3.3) assessed the overall performance of the proposed RNN-based ensemble learning (RNN-EL) framework in comparison with state-of-the-art stock manipulation detection methods.

4.3.1. Evaluating trade-based classifiers

In the proposed framework, RNNs are served as trade-based classifiers since they have done well on many sequence learning problems by virtue of their internal memory [20,43]. Previous studies mainly included trade-based features in the manipulation detection model. With only trade-based features, which classifier can better detect stock price manipulation? Traditional statistical algorithms, general machine learning techniques, or RNNs? We used paired t-tests to compare the performance of RNN classifier against the baseline models and test the significance of the improvements over the best baseline. The statistical significance of an improvement is expressed as a p-value [44]. The lower the p-value, the more statistically significant is the difference between two evaluation results. To ensure that the feature information used by different classifiers is the same, time-series of trade-based features are expanded into one-dimensional vectors for non-time-series modeling. For example, if the number of trade-based features is N_X and the time window length of previous inputs is T in the RNN modeling, then the number of features is $N_X \times T$ for other classifiers. We evaluate the predictive performances of RNN models with the parameter T ranging from 1 to 10 to search for a suitable time window length. According to results of cross-validation, T is set to 7 to make the best use of internal memory. For the RNN-RSE classifier, the prediction results were established by plurality voting of the outputs of sub RNN models.

The evaluating results are shown in Table 5. AUC and F1 values reflect the overall performance of the classifiers. Precision represents the proportion of the actual manipulated samples to the predicted manipulated samples, while recall indicates the proportion of the actual manipulated samples that have been predicted manipulated. In comparison, recall weighed more in manipulation detection since the missing warning of possible illegal transactions will irreparably damage the interests of ordinary investors. But precision is also an important evaluation metric. An anti-fraud model with higher precision can significantly reduce regulatory costs. RNN classifier outperformed the eight baseline classifiers on recall and AUC value, which indicated that RNN was better suited to detect stock price manipulation from stock trading data than other models. By using random subspace ensembles of RNNs, RNN-RSE classifier reached the highest AUC value, F1 value and precision, outperforming the baseline classifiers on any evaluation metric. Specifically, as denoted in Table 5, RNN-RSE improves over the RF classifier by 11.5% on AUC value, over the BayesNet classifier by 7.4% on F1 score, over the BayesNet classifier by 6.6% on precision and over the LogitReg classifier by 6.9% on recall. The improvements of four evaluation metrics over the best baseline were all statistically significant. RNN-RSE had nearly the same recall as RNN and was superior to the latter in the performance of other indicators. This suggests that ensemble learning techniques do help improve manipulation detection capabilities by keeping more differences in the ensemble system to improve procedural bias through disturbance of both the samples and the feature spaces.

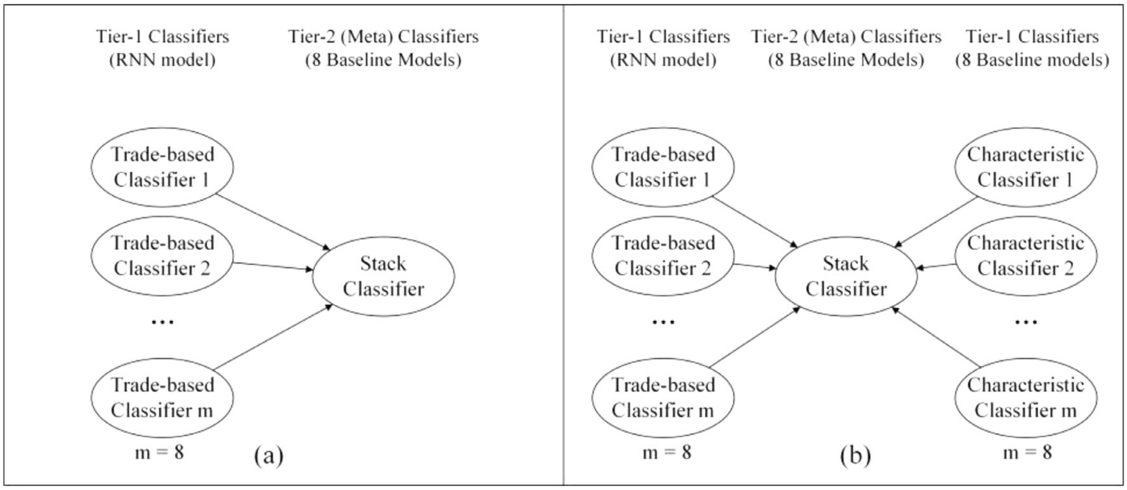


Fig. 6. The final structure of models reported in Table 6.

Table 6
Performances of stack classifiers using different feature sets.

Stack classifiers	Trade-based features			
	AUC	F1	Prec.	Rec.
ANN	0.801	0.302	19.4%	68.5%
BayesNet	0.794	0.292	18.4%	69.7%
C5.0 tree	**0.845**	**0.365**	**24.2%**	74.6%
KNN	0.789	0.290	18.3%	69.5%
LogitReg	0.840	0.345	22.5%	73.6%
Naïve Bayes	0.825	0.319	20.4%	73.0%
RF	0.838	0.322	20.3%	**77.0%**
SVM	0.831	0.328	21.1%	73.1%
Stack classifiers	Trade-based features and characteristic features			
	AUC	F1	Prec.	Rec.
ANN	0.857	0.334	21.0%	81.6%
BayesNet	0.859	0.484	36.2%	73.5%
C5.0 tree	0.892	0.437	29.1%	88.2%
KNN	0.864	0.328	20.3%	85.5%
LogitReg	0.895	0.469	32.7%	82.4%
Naïve Bayes	0.852	**0.507**	**39.2%**	71.8%
RF	**0.907**	0.470	31.7%	**90.2%**
SVM	0.876	0.387	25.2%	84.0%

4.3.2. Evaluating the effectiveness of including characteristic features

Why the proposed manipulation detection framework should consider characteristic features as a supplement have been discussed above. In this section, we want to empirically investigate whether the inclusion of characteristic information can indeed improve the prediction accuracy. Firstly, each of the 8 baseline classifiers was run using the characteristic feature sets on a subsample, resulting in 8 characteristic classifiers in total. In the same way, 8 trade-based classifiers were trained based on RNN model. Secondly, all the 8 aforementioned classifiers were run as stack classifiers, resulting in 8 different stack arrangements. We used 8 trade-based classifiers as inputs for each stack classifier. The training set for the stack classifiers was generated by running the pre-trained tier-1 classifiers on the training samples. The experimental results of stack classifiers using only trade-based features are shown in Table 6 (Fig. 6(a) denotes the model structure). Thirdly, similarly, we utilized all 16 tier-1 classifiers, including 8 trade-based classifiers and 8 characteristic classifiers, as inputs for each of the 8 stack classifiers. Table 6 also demonstrates the results of stack classifiers using both trade-based and characteristic features (the model structure is presented in Fig. 6(b)).

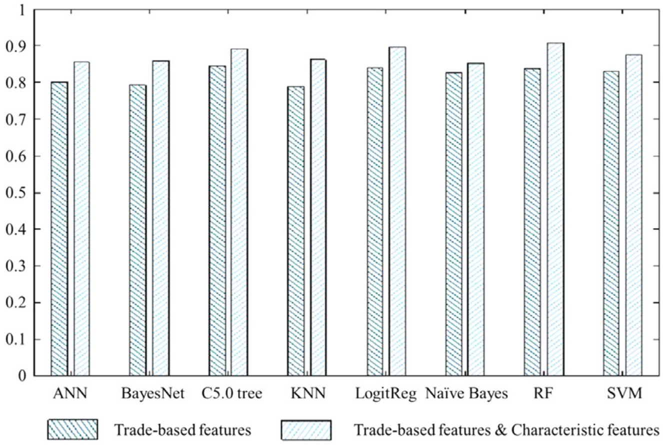


Fig. 7. The comparative AUC value: stack classifiers using different feature sets.

Table 6 presents the classification performance of different stack classifiers using different feature sets. By comparing the evaluation results of trade-based classifiers in Table 5 with those of stack classifiers using trade-based features in Table 6, we can see that the combination of deep learning methods and stacking techniques dramatically improves the prediction accuracy under different evaluation metrics. The C5.0 tree stack classifier that incorporated predictions of 8 RNN classifiers as input features achieved the highest AUC value of 0.845 which improved over that of the best baseline RF by 17.2%.

Next we compared performances of classifiers in the upper and lower parts of Table 6. The stack classifiers using both trade-based and characteristic features had considerably better AUC values (Fig. 7.) and F1 scores than those using only trade-based features, outperforming them by nearly 0.055 on AUC value and 0.107 on F1 score on average. Moreover, these improvements were coupled with both increased precision and recall for seven of the classifiers. Naïve Bayes classifier was the only exception, decreasing its recall by 1.2%, but it attained a great improvement on precision by 18.8% and achieved better overall performances. The overall AUC values of these stack classifiers using both trade-based and characteristic features ranged from 0.852 to 0.907, with a best result attained by the RF classifier. It also had a highest recall of 90.2% which indicated that the detection model could identify most of the suspicious trading behaviors. These comparative experimental

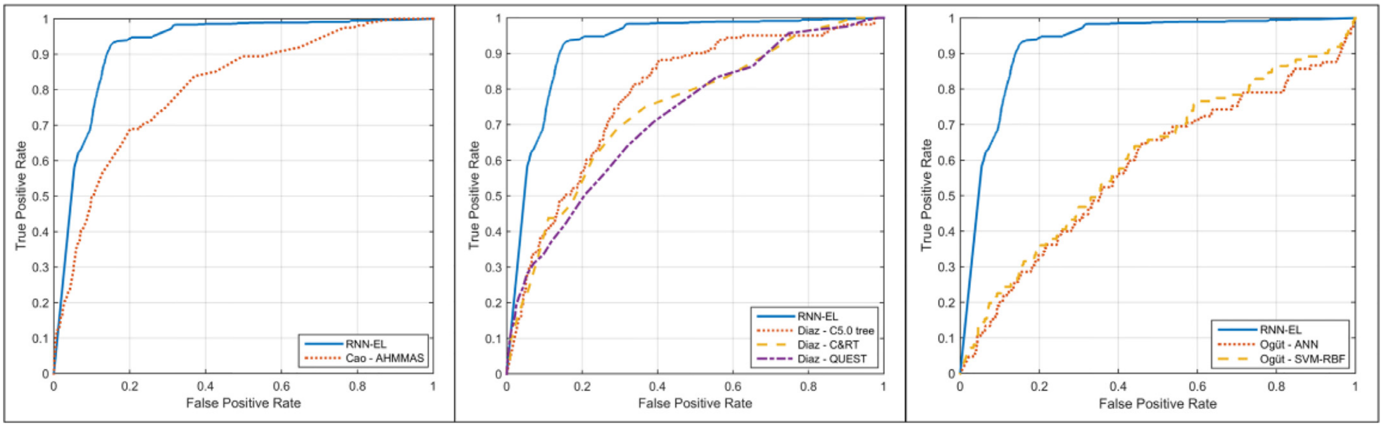


Fig. 8. ROC curves for RNN-EL and comparison methods.

Table 7
Results for RNN-EL and comparison methods.

Method	AUC	F1	Prec.	Rec.
RNN-EL	**0.907**	**0.470**	**31.7%**	**90.2%**
Cao – AHMMAS	0.796	0.293	18.7%	67.5%
Diaz – C5.0 tree	0.788	0.263	16.2%	70.0%
Diaz – C&RT	0.734	0.256	15.9%	65.7%
Diaz – QUEST	0.719	0.264	16.5%	66.4%
Öğüt – SVM-RBF	0.581	0.155	9.3%	46.3%
Öğüt – ANN	0.572	0.162	10.1%	40.6%

results suggest that the enhanced performance of combined stack classifiers was attributable to the inclusion of complementary information provided by characteristic features.

4.3.3. Evaluating RNN-EL in comparison with existing manipulation detection methods

In order to further assess the effectiveness of the proposed framework, we evaluated RNN-EL approach in comparison with three prior manipulation detection methods that had attained state-of-the-art results: Öğüt et al. [14], Diaz et al. [15] and Cao et al. [34]. Based on the experimental results presented in Table 6, RF is selected as the stack classifier for RNN-EL approach since it provides the best results. Each of these method was run on our specific dataset, with only trade-based features considered. Öğüt et al. have attained better results than multivariate statistical techniques by using ANN and SVM with radial basis function (SVM-RBF) to detect stock price manipulation in Turkey by observing daily volatility, volume and return. Diaz et al. applied a set of 10 features, including indicators that reflected changes in trading liquidity, volatility, returns and abnormal returns, in combination with three decision trees algorithms: QUEST, C5.0 and C&RT. Cao et al. attained excellent results on synthetic exploratory financial data by a novel approach call adaptive hidden Markov model with anomaly states (AHMMAS). All comparisons were run on the same trading data used in the prior experiments. The data preprocessing procedures were performed according to the original literature. Optimal values of parameters were adopted during comparative experiments.

Table 7 shows the experimental results for RNN-EL and comparison methods. As can be seen from the table, RNN-EL have attained considerably better results than all the comparison models associated with the three methods, outperforming them by over 13.9% on AUC value, 60.4% on F1 score, 69.5% on precision and 33.6% on recall. These performance gains suggest that RNN-EL have sufficient competence to provide enhanced manipulation detection capabilities. ROC curves were generated and shown in Fig. 8. The curve of RNN-EL was located closest to the top left corner of the three figures, suggesting that the proposed method have attained more outstanding overall performances. Overall, these results illustrate the efficacy of RNN-EL as a viable mechanism for manipulation detection.

5. Discussion and conclusions

Given the consensus that protecting ordinary investors from stock price manipulation activities is really an important problem, much effort has been dedicated to researching effective detection methods that can efficiently identify suspicious trading behaviors among huge amounts of trading activities in time to take action. The regulatory costs can be significantly reduced if there is an anti-fraud supervision model with sufficient competence to provide enhanced manipulation detection accuracy.

Meanwhile, machine learning techniques are taking an increasingly important role in combating financial fraud. A few pioneering papers have tried to introduce ML techniques in stock market manipulation detection and have achieved prominent results. Along with better learning abilities and generalization abilities of advanced models, however, financial fraud detection is an untypical case and, especially requires built-in domain knowledge from practical business process for model construction. While previous studies have only utilized trading data to classify suspicious trading activities, characteristic features have rarely been considered in the manipulation detection model. Moreover, most existing research ignored the fact that stock trading data are complex multi-dimensional time series which consist of price, trading volume, stock returns and so on. Thus, the high relevance and dependence between different points of the time series result in the inappropriateness of simply inputting data into fixed-size networks for results.

In this study, we propose a novel RNN-based ensemble learning (RNN-EL) framework for stock price manipulation detection with trade-based features derived from trading records and characteristic features of the list companies. Deep learning algorithms for time-series modeling and ensemble learning techniques are both incorporated in the proposed framework. To the best of our knowledge and belief, this is the first successful research work that incorporates deep learning algorithms and ensemble learning techniques for stock price manipulation detection.

Our empirical tests have revealed that the proposed RNN-EL framework for stock price manipulation detection is effective. Based on prosecuted manipulation cases reported by the CSRC, we built a real-world dataset from related trading records and characteristic information. Comparative experimental results on this dataset have indicated that each facet of the framework facilitates the improvements of manipulation detection capabilities. More specifically, experiment group 1 revealed that RNN was better suited to detect stock price manipulation from stock trading data than traditional statistical and general machine learning baselines in virtue of its internal memory capacity. It suggests that monitoring suspicious trading behaviors within a period of several minutes could provide more clues on manipulation activities. It was also proved that random subspace ensemble was more effective than using a single classifier. In experiment group 2, stack classifiers using both trade-based and characteristic features had attained considerably better results which was attributable to the inclusion of complementary information provided by characteristic features. Experiment group 3 showed that RNN-EL framework was able to outperform state-of-the-art approaches.

Collectively, RNN-EL have attained considerably better results than all the state-of-the-art methods, outperforming them by over 13.9% on AUC value, 60.4% on F1 score, 69.5% on precision and 33.6% on recall. These performance gains suggest that RNN-EL have sufficient competence to provide enhanced manipulation detection capabilities and indicate the efficacy of RNN-EL as a viable mechanism for manipulation detection.

In future work, one possible research direction is to apply other suitable methods to analyze stock trading data time series. For example, Long Short Term Memory (LSTM) is a popular deep learning technique in the field of pattern recognition, whose successful application in tasks such as handwriting and speech recognition show us the potential of adopting LSTM for manipulation detection. It is believed that with the development of advanced pattern recognition techniques, a higher-level approach to the stock price manipulation problem could be found. Another possible research direction is the implementation of the detection model in real scenes. The time complexity of RNN-based model is much higher than other existing detection models. How to effectively train and deploy the proposed model to meet actual needs is an important issue.

It is also a meaningful research direction to determine how to integrate more data sources, such as social relationships of executives of the listed company and announcement content, into the detection framework. It is believed that the performance of the manipulation detection system can improve with the help these types of information.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (Grant no. U1711262, 71771212), Humanities and Social Sciences Foundation of the Ministry of Education (No. 14YJA630075, 15YJA630068), Hebei Social Science Fund (HB13GL021), and the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (No. 15XNLQ08).

Conflict of interest statement

The authors declare that there are no conflict of interest statement.

References

[1] M. Punniyamoorthy, J. Joy Thoppan, ANN-GA based model for stock market surveillance, J. Financ. Crime 20 (1) (2012) 52–66.
[2] Y.C. Huang, Y.J. Cheng, Stock manipulation and its effects: pump and dump versus stabilization, Rev. Quant. Financ. Acc. 44 (4) (2013) 1–25.
[3] A.R. Admati, P. Pfleiderer, A theory of intraday patterns: volume and price variability, Rev. Financ. Stud. 1 (1) (1988) 3–40.
[4] D. Kong, M. Wang, The manipulator's poker: order-based manipulation in the Chinese stock market, Emerg. Mark. Financ. Trade 50 (2) (2014) 73–98.
[5] O.D. Hart, On the profitability of speculation, Q. J. Econ. 91 (4) (1977) 579–597.
[6] F. Allen, D. Gale, Stock-price manipulation, Review of Financ. Stud. 5 (3) (1992) 503–529.
[7] N. Khanna, R. Sonti, Value creating stock manipulation: feedback effect of stock prices on firm value, J. Financ. Mark. 7 (3) (2004) 237–270.
[8] A. Abbasi, C. Albrecht, A. Vance, J. Hansen, Metafraud: a meta-learning framework for detecting financial fraud, MIS Q. 36 (4) (2012) 1293–1327.
[9] J.Z. Lei, A.A. Ghorbani, Improved competitive learning neural networks for network intrusion and fraud detection, Neurocomputing 75 (1) (2012) 135–145.
[10] C.C. Lin, A.A. Chiu, S.Y. Huang, D.C. Yen, Detecting the financial statement fraud: the analysis of the differences between data mining techniques and experts' judgments, Knowl.-Based Syst. 89 (9) (2015) 459–470.
[11] C. Di Ciccio, H. Van der Aa, C. Cabanillas, J. Mendling, J. Prescher, Detecting flight trajectory anomalies and predicting diversions in freight transportation, Decis. Support Syst. 88 (2016) 1–17.
[12] S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly detection for streaming data, Neurocomputing 262 (2017) 134–147.
[13] J. Zhai, Y. Cao, Y. Yao, X. Ding, Y. Li, Computational intelligent hybrid model for detecting disruptive trading activity, Decis. Support Syst. 93 (2016) 26–41.
[14] H. Öğüt, M. Mete Doğanay, R. Aktaş, Detecting stock-price manipulation in an emerging market: the case of Turkey, Exp. Syst. Appl. 36 (9) (2009) 11944–11949.
[15] D. Diaz, B. Theodoulidis, P. Sampaio, Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices, Exp. Syst. Appl. 38 (10) (2011) 12757–12771.
[16] S. Lawrence, C.L. Giles, S. Fong, Natural language grammatical inference with recurrent neural networks, IEEE Trans. Knowl. Data Eng. 12 (1) (2000) 126–140.
[17] R. Kumar, S. Srivastava, J.R.P. Gupta, A. Mohindru, Diagonal recurrent neural network based identification of nonlinear dynamical systems with Lyapunov stability based adaptive learning rates, Neurocomputing 287 (2018) 102–117.
[18] X. Chen, X. Liu, Y. Wang, M.J.F. Gales, P.C. Woodland, Efficient training and evaluation of recurrent neural network language models for automatic speech recognition, IEEE/ACM Trans. Audio Speech Lang. Process. 24 (11) (2016) 2146–2157.
[19] Y. Chherawala, P.P. Roy, M. Cheriet, Feature set evaluation for offline handwriting recognition systems: application to the recurrent neural network model, IEEE Trans. Cybern. 46 (12) (2015) 2825–2836.
[20] M.E. Jalal, M. Hosseini, S. Karlsson, Forecasting incoming call volumes in call centers with recurrent neural networks, J. Bus. Res. 69 (11) (2016) 4811–4814.
[21] R.K. Aggarwal, G. Wu, Stock market manipulations, J. Bus. 79 (4) (2006) 1915–1953.
[22] S. Imisiker, B.K.O. Tas, Which firms are more prone to stock market manipulation? Emerg. Mark. Rev. 16 (3) (2013) 119–130.
[23] C. Comerton-forde, T.J. Putniņš, Stock price manipulation: prevalence and determinants, Rev. Financ. 18 (1) (2014) 23–66.
[24] J. Van Bommel, Rumors, J. Financ. 58 (4) (2003) 1499–1519.
[25] G. Dissanaike, K.H. Lim, Detecting and quantifying insider trading and stocks manipulation in Asia markets, Asian Econ. Pap. 14 (3) (2015) 1–20.
[26] A. Chakraborty, B. Yılmaz, Informed manipulation, J. Econ. Theory 114 (1) (2004) 132–152.
[27] K. Felixson, A. Pelli, Day end returns—stock price manipulation, J. Multinatl. Financ. Manag. 9 (2) (1999) 95–127.
[28] P.G. Mahoney, The stock pools and the securities exchange act, J. Financ. Econ. 51 (3) (1999) 343–369.
[29] G.K. Palshikar, A. Bahulkar, Fuzzy temporal patterns for analyzing stock market databases, in: Proceedings of the International Conference on Advances in Data Management, 2000, pp. 135–142.
[30] C. Pirrong, Detecting manipulation in futures markets: the Ferruzzi soybean episode, Am. Law Econ. Rev. 6 (1) (2004) 28–71.
[31] G.K. Palshikar, M.M. Apte, Collusion set detection using graph clustering, Data Min. Knowl. Discov. 16 (2) (2008) 135–164.
[32] Y. Cao, Y. Li, S. Coleman, A. Belatreche, T.M. Mcginnity, A hidden Markov model with abnormal states for detecting stock price manipulation, in: Proceedings of the International Conference on Systems, Man, and Cybernetics, 2013, pp. 3014–3019.
[33] K. Golmohammadi, O.R. Zaiane, D. Díaz, Detecting stock market manipulation using supervised learning algorithms, in: Proceedings of the International Conference on Data Science and Advanced Analytics, 2015, pp. 435–441.
[34] Y. Cao, Y. Li, S. Coleman, A. Belatreche, T.M. Mcginnity, Adaptive hidden Markov model with anomaly states for price manipulation detection, IEEE Trans. Neural Netw. Learn. Syst. 26 (2) (2015) 318–330.
[35] A.M. Rather, A. Agarwal, V.N. Sastry, Recurrent neural network and a hybrid model for prediction of stock returns, Exp. Syst. Appl. 42 (6) (2015) 3234–3241.
[36] Y. Ren, L. Zhang, P.N. Suganthan, Ensemble classification and regression: recent developments, applications and future directions, IEEE Comput. Intel. Mag. 11 (1) (2016) 41–53.
[37] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, P.N. Suganthan, Ensemble of differential evolution variants, Inf. Sci. 423 (2018) 172–186.
[38] J.L. Elman, Finding structure in time, Cognit. Sci. 14 (2) (1990) 179–211.
[39] Y. Wang, X. Li, X. Ding, Probabilistic framework of visual anomaly detection for unbalanced data, Neurocomputing 201 (2016) 12–18.

[40] M.A.H. Farquad, I. Bose, Preprocessing unbalanced data using support vector machine, Decis. Support Syst. 53 (1) (2012) 226–233.

[41] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Proces. Manag. 45 (4) (2009) 427–437.

[42] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of International Joint Conference on Artificial Intelligence, 1995, pp. 1137–1143.

[43] M. Seera, C.P. Lim, K.S. Tan, W.S. Liew, Classification of transcranial Doppler signals using individual and ensemble recurrent neural networks, Neurocomputing 249 (2017) 337–344.

[44] D. Hull, Using statistical testing in the evaluation of retrieval experiments, in: Proceedings of the Sixteenth International Conference on Research and Development in Information Retrieval, 2001, pp. 329–338.

Mr. Qili Wang is a Ph.D. student at School of Information, Renmin University of China. He got his bachelor degree in Computer Sciences at School of Information, Renmin University of China. His research interests include market prediction, big data analytics, and decision support systems. He has published research papers in international journals and conferences, such as Geoinformatica and Neurocomputing.

Dr. Wei Xu is an associate professor at School of Information, Renmin University of China. He is a research fellow at Department of Information Systems, City University of Hong Kong. He got his bachelor and master degree in Mathematics at Xi'an Jiaotong University and doctor degree in Management Science at Chinese Academy of Sciences. His research interests include big data analytics, business intelligence and decision support systems. He has published over 100 research papers in international journals and conferences, such as Annals of Operations Research, Decision Support Systems, European Journal of Operational Research, IEEE Trans. Systems, Man and Cybernetics, International Journal of Production Economics and Production and Operations Management.

Mr. Xinting Huang is currently a Ph.D. student at School of Computing and Information Systems, The University of Melbourne. He got his bachelor degree in Mathematics at School of Information, Renmin University of China. His research interests include information retrieval, data mining, and machine learning.

Mr. Kunlin Yang is currently an undergraduate student at School of Information, Renmin University of China. His research interests include business intelligence, and decision support systems.