Applied Soft Computing Journal 84 (2019) 105747

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices

Feng Zhou [a], Qun Zhang [b,c,*], Didier Sornette [d,e], Liu Jiang [f]

[a] School of Information, Guangdong University of Finance and Economics, Guangzhou, Guangdong, China
[b] School of Finance, Guangdong University of Foreign Studies, Guangzhou, Guangdong, China
[c] Southern China Institute of Fortune Management Research, Guangdong University of Foreign Studies, Guangzhou, Guangdong, China
[d] Department of Management, Technology and Economics, ETH Zurich (Eidgenössische Technische Hochschule Zürich), Zurich, Switzerland
[e] Swiss Finance Institute, c/o University of Geneva, Geneva, Switzerland
[f] Surrey Business School, University of Surrey, Guildford, England, United Kingdom of Great Britain and Northern Ireland

ARTICLE INFO

Article history:
Received 9 November 2018
Received in revised form 30 June 2019
Accepted 27 August 2019
Available online 3 September 2019

Keywords:
Ensemble learning
Gradient boosted decision trees
Logistic regression
Stock market
Transaction cost

ABSTRACT

Forecasting the direction of the daily changes of stock indices is an important yet difficult task for market participants. Advances on data mining and machine learning make it possible to develop more accurate predictions to assist investment decision making. This paper attempts to develop a learning architecture LR2GBDT for forecasting and trading stock indices, mainly by cascading the logistic regression (LR) model onto the gradient boosted decision trees (GBDT) model. Without any assumption on the underlying data generating process, raw price data and twelve technical indicators are employed for extracting the information contained in the stock indices. The proposed architecture is evaluated by comparing the experimental results with the LR, GBDT, SVM (support vector machine), NN (neural network) and TPOT (tree-based pipeline optimization tool) models on three stock indices data of two different stock markets, which are an emerging market (Shanghai Stock Exchange Composite Index) and a mature stock market (Nasdaq Composite Index and S&P 500 Composite Stock Price Index). Given the same test conditions, the cascaded model not only outperforms the other models, but also shows statistically and economically significant improvements for exploiting simple trading strategies, even when transaction cost is taken into account.

1. Introduction

Analyzing, forecasting and trading stock indices are indispensable activities for private investors, hedge funds and proprietary trading desks, and a reasonably accurate prediction may raise the potential to yield high financial benefits and hedge against market risks. However, financial economists often question the predictability of stock indices and the existence of opportunities for profitable trading, which points to the informational efficiency and maturity of these stock markets. In the context of the Efficient Market Hypothesis [1,2], this poses the question whether stock markets are efficient engines converting information into prices. There is also the ever alluring quest to arbitrage possible market inefficiencies that could provide substantial monetary rewards to the arbitrageur.

While the efficient market hypothesis is a useful first-order representation of financial markets in normal times, unstable market regimes occur where the anchor of a fundamental price is shaky and large uncertainties characterize the future gains, which provide fertile environment for abnormal behaviors such as excess volatility, the joint occurrence of bubbles and frenzied trading [3]. In reality, it is clear that truly efficient markets do not exist, as the classic analysis by Grossman and Stiglitz [4] shows: the acquisition and analysis of information is costly, prices cannot perfectly reflect all available information since this would leave no incentive to those who spent resources to obtain it and trade on it. Moreover, traders also fail to react in due time. On the one hand, some traders exhibit over-reaction to the arrival of new information, either because they are over-confident [5,6], or have an incentive to imitate and a desire to be imitated [7]. This can result in an over- or under-valued asset, which then increases the likelihood of a rebound or crash and thus creates a negative autocorrelation in returns. On the other hand, investors may under-react due for instance to inattention, and news are slowly internalized into price, resulting in positive return autocorrelation and momentum [8–10].

Furthermore, advances in information technologies and public communication infrastructures cause us to rethink the structure

* Corresponding author at: School of Finance, Guangdong University of Foreign Studies, Guangzhou, Guangdong, China.
E-mail addresses: fengzhou@gdufe.edu.cn (F. Zhou), z.q16@mail.scut.edu.cn (Q. Zhang), dsornette@ethz.ch (D. Sornette), jl@bjbec.com (L. Jiang).

https://doi.org/10.1016/j.asoc.2019.105747
1568-4946/

of financial markets and the efficient markets hypothesis. As illustrated by the advent and development of high frequency trading (HFT), it has reduced the time scales over which prices fluctuate and had induced novel types of instabilities, such as flash crashes [11]. It is well-documented that stocks move by ∼2% a day or more, with many "jumps". Although news has some influence on price movements, only a small fraction of the volatility can be attributed to news. And there seems to be endogenous self-excited processes at work [12,13]. Many theoretical arguments suggest that bursts of volatility may be intimately related to the quasi-efficiency of financial markets, in the sense that they are hard to predict, since financial investors' adaptive behavior tends to remove price predictability [14]. As the computer power, data availability and algorithmic sophistication increased, the continuous actions of investors and of their algorithms, which are aggregated in the prices, produce a "market intelligence" more powerful than that of most of them. This can be called the "Emerging Market Intelligence Hypothesis" [3]. Given the complexity or structure of financial markets that they collectively create, this "intelligence" would make most (but not all) strategies into losing strategies.

It is in the context that one should consider the development over the years of the two tractions in the field of stock indices prediction, namely fundamental analysis and technical analysis. In the category of fundamental analysis, the fundamentalists look at the intrinsic value of stocks, performance of the industry, political events, general economic conditions and market expectations. Rather than attempting to measure all the above, the technical analysts (also known as chartists) tend to exploit potential opportunities by studying statistics generated by market activity, and using charts to derive predictable patterns and trends that may suggest how this index will behave in the future [15,16]. However, due to (1) the acceptance of the crudest form of the efficient market hypothesis [1,17,18], (2) the negative empirical evidences in several early and widely cited studies of technical analysis on the stock markets (such as [19,20]), and (3) the sense that stable forecasting patterns are unlikely to persist for long periods of time and will self-destruct when they are discovered by a large number of investors [21], the effectiveness of technical analysis is controversial.

Therefore, many studies have investigated various technical trading rules in different markets with the goal of either uncovering profitable trading rules or testing market efficiency, or both [22,23]. For example, in rejecting the Random Walk Hypothesis for weekly U.S. stock indices, some studies [24,25] have shown that past prices may be used to forecast future returns to some degree. Hsu and Kuan [26] studied a total of 39 832 technical rules or trading strategies, and concluded that there are no profitable trading rules in mature markets. Hsu, Hsu and Kuan [27] found that technical rules have predictive power on growth and emerging markets indices, at least until corresponding exchange traded funds (ETF) are introduced. Park and Irwin [22] reviewed the evidence on the profitability of technical analysis and found that "modern" empirical studies during 1988–2004 indicated that technical trading strategies consistently generate economic profits in a variety of speculative markets at least until the early 1990s. As such, if the information obtained from technical analysis is processed efficiently and appropriate algorithms are applied, then the trend of stock indices may be predicted over some future time horizons. Besides, the informational efficiency and maturity of the markets can be tested.

In the literature, some algorithms pivoted around the traditional time series forecasting are widely used, such as the moving average model (MA), the exponential smoothing (ES), the autoregressive-moving average model (ARMA) and the generalized autoregressive conditional heteroscedasticity model (GARCH) [28]. More recently, for modeling random abrupt changes in the structures and parameters of systems, the moving horizon estimation algorithm (MHE) [29] and the model-free adaptive dynamic programming algorithm (ADP) [30] have been designed for a class of nonlinear Markovian jump systems (MJSs). Besides, many algorithms and methods especially from the machine learning point of view have carved their own niche, including the logistic regression (LR), the discriminant analysis (DA), the support vector machine (SVM), the gradient boosted decision trees (GBDT), the neural network (NN), the tree-based pipeline optimization tool (TPOT) and so on [31–33]. For example, support vector machine has been used for predicting the direction of the stock index in Turkey [34] and Madrid IBEX-35 stock index [35]. Artificial neural network has been applied to predict the direction of the stock index in Japanese stock market [36], the Istanbul stock exchange [37] and American stock market [38]. In fact, the accuracies reported in these studies are not pure chance but are based solidly on the understanding that forecasting the direction of stock indices can be generally formulated as a classification problem and this kind of problem is not linearly separable. Besides, machine learning methods are particularly useful when little knowledge is available on the underlying data generating process.

Moreover, in statistics and machine learning communities (or even machine learning competition like Kaggle[1]), there are different ensemble learning algorithms that aim to reduce variance and bias associated with prediction, help to avoid overfitting, and better improve the prediction performance than those obtained from any of the constituent learning algorithms alone [39,40]. For instance, Kaucic [41] proposed a variable length evolutionary learning ensemble approach, where a wide range of different types of technical indicators are considered as input variables, Facebook used the GBDT for non-linear feature transformation and fed them to the LR for the final prediction [42], besides, Microsoft Bing presented eight ensemble methods, shared their experience and learning on designing and optimizing model ensembles to improve the click-through rate prediction [43]. As a quantile regression based hybrid and a feedforward neural network, the quantile regression neural network (QRNN) trained by particle swarm optimization (namely PSOQRNN) outperformed the QRNN for forecasting the volatility of S&P 500 Stock Index and NSE India Stock Index, since it overcame the problems such as large computational time, slow convergence rate and entrapment in local minima [44]. Besides, sparsity techniques were suggested to be complemented by advanced algorithms in machine learning to discover governing equations underlying a nonlinear dynamical system from noisy measurement data [45].

Inspired by the latest trends in academia and industry, we propose a cascaded ensemble LR2GBDT in order to construct a new class of learning architecture to predict the direction of the daily change of stock price indices. Our learning architecture is achieved by first training the LR model on all raw features and then training the GBDT model on both the results of LR and the raw features. The application of the hybrid learning model on the financial domain is based on the assumption that a single model may not be totally sufficient to interpret the information contained in the stock indices data [46,47]. We also hypothesize that, given the superiority in other domains, the cascaded model would outperform the baseline models. To illustrate the performance of the proposed model, we attempt to predict the direction of movement of three daily stock price indices (Nasdaq Composite Index, S&P 500 Composite Index, and Shanghai Stock Exchange Composite Index), and compare the results with those respectively obtained from the widely used models, such as LR, GBDT,

[1] The website is www.kaggle.com.

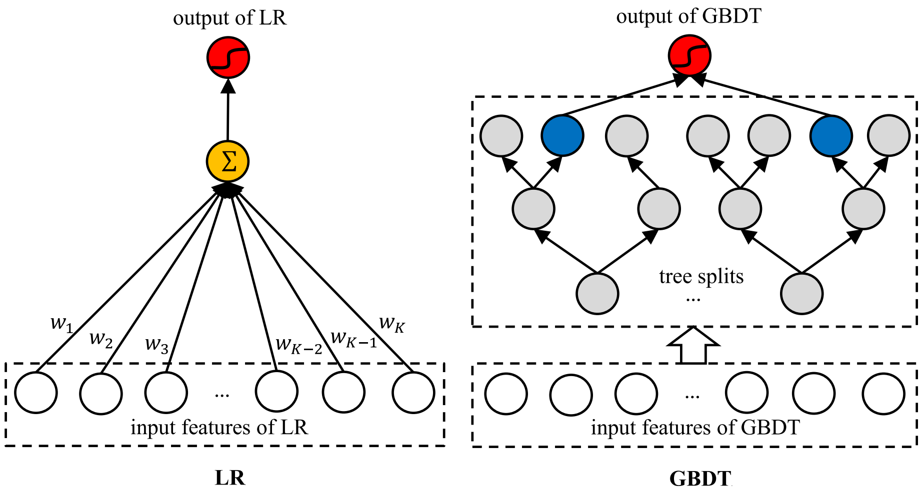

Fig. 1. Graphical illustration of two baseline models: LR and GBDT.

SVM, NN and TPOT models. Furthermore, our study addresses the five following questions: (1) Can the cascaded LR2GBDT-based model be trusted? (2) Can the best classifier be selected with good confidence (i.e. stability)? (3) Is the LR's predicting score feature relevant? (4) Can the LR2GBDT-based strategies make profits? (5) Does transaction cost remove the vested profits?

Given the available literature, our contributions in this paper can be concluded as follows:

- First, to our knowledge, this study is the first in deploying the cascaded ensemble learning architecture called LR2GBDT to forecast and trade stock indices. And we compare the strategy performance of LR2GBDT model to its baseline models and the SVM, NN and TPOT, thereby deriving relevant insights for academics and practitioners alike.
- Second, we evaluate the prediction accuracy and trading performance of above models for the stock indices data of two different stock markets, which are an emerging market (Shanghai Stock Exchange Composite Index) and a mature stock market (Nasdaq Composite Index and Standard & Poor's 500 Composite Stock Price Index). It offers the prospect of studying the informational efficiency of stock markets.
- Third, we investigate the trading strategies in a daily investment horizon instead of monthly or yearly frequencies, with the consideration of transaction cost and buy–sell thresholds, contributing to exploit short-term strategies for more stock indices data.

The rest of this paper is organized as follows. Section 2 gives a brief introduction to the logistic regression (LR) and the gradient boosted decision trees (GBDT). Section 3 then proposes the cascaded ensemble LR2GBDT model and the trading strategy. Section 4 describes research data and computation of technical indicators, which serve as input features of above models. This section also presents the chosen metrics of prediction accuracy and trading performance. The empirical results and comprehensive evaluations of all these models are discussed in Section 5. Finally, several concluding remarks are presented in Section 6.

2. LR and GBDT

2.1. LR: Logistic regression

The logistic regression is a widely used classification technique [48], which provides a very powerful discriminative model based on the well-known logistic function. The LR models the conditional probability that $y_i = \pm 1$ conditional on $\mathbf{x}_i$ according to Eqs. (1) and (2):

$$\hat{y}_{LR}(y_i = 1|\mathbf{x}_i) = sigmoid(\mathbf{w}^T\mathbf{x}_i + c) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}_i + c)}}\ , \tag{1}$$

$$\hat{y}_{LR}(y_i = -1|\mathbf{x}_i) = 1 - \hat{y}_{LR}(y_i = 1|\mathbf{x}_i) = \frac{1}{1 + e^{\mathbf{w}^T\mathbf{x}_i + c}}\ , \tag{2}$$

where the ith sample $\mathbf{x}_i \in \mathbf{R}^K$ has K features and is part of the data set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ with N samples, and each sample $\mathbf{x}_i$ has an observed class label $y_i \in \{-1, 1\}$ whose union is the set $\mathbf{Y} = \{y_i\}_{i=1}^N$. Finally, $\mathbf{w} \in \mathbf{R}^K$ is the weight vector and $c \in \mathbf{R}$ is the global bias. These two equations can be combined as

$$\hat{y}_{LR} = \frac{1}{1 + e^{-y_i(\mathbf{w}^T\mathbf{x}_i + c)}}\ . \tag{3}$$

We fit this logistic function using our training data, which consists of feature vectors, to obtain $\mathbf{w}$ and c, and then use the fitted model to perform predictions. The fitting process normally uses the maximum likelihood estimation method. In addition, to avoid over-fitting, the L^2 regularized term $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ is generally added in the cost function to avoid too much concentration on just a few weights. Thus, we solve the following optimization problem

$$\min_{\mathbf{w},c}\ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\log(1 + e^{-y_i(\mathbf{w}^T\mathbf{x}_i + c)}), \tag{4}$$

where the parameter C is the inverse of the regularization strength, and smaller C's specify stronger regularization.

The left part of Fig. 1 depicts the structure of this model. As shown in the figure, the LR model is trained based on the input features. For the training and testing, the implementation of the logistic regression classifier model is realized with the L^2 penalty as already mentioned, while the model parameters are determined by the tool (LogisticRegression) included in the scikit-learn Python library.

2.2. GBDT: Gradient boosted decision trees

The classification task can also be completed based on a collection of base learners (i.e., decision tree classifiers) and their combination through a technique called gradient boosting. GBDT model, as described in the right part of Fig. 1, is widely used by data scientists to achieve state-of-the-art results in many machine learning challenges [49]. Different from the linear models like logistic regression, gradient boosted decision trees are more flexible to implement non-linear and crossing transformations on the input features. Specifically, they are robust to outliers, scalable, and able to naturally model non-linear decision boundaries thanks to their hierarchical structure. This kind of model is suitable for handling numerical features and categorical features with tens of categories.

The detail algorithm for GBDT given below provides more details about the implementation of GBDT model. Please note that M is the maximum number of iterations for training (i.e., the number of trees), η is the step size used for combining the weight of individual trees in updates to prevents over-fitting (i.e., learning rate), γ_m is the minimum loss reduction required to make a further partition on a leaf node of the tree. In Step 3, the term $\nu T + \frac{1}{2}\beta\|\gamma\|^2$ penalizes the complexity of the model (i.e., the regression tree functions), where T is the number of leaves in the tree, β is the regularization parameter. This generates a final classifier whose misclassification rate can be reduced by combining many classifiers whose misclassification rate may be high. However, it sometimes fails to improve the performance of the base inducer due to over-fitting.

As illustrated by the following GBDT algorithm, given a sample X, GBDT uses M additive functions to give the output

$$\hat{y}_{GBDT} = \sum_{m=1}^{M} \eta_m b(X; \gamma_m). \tag{5}$$

For the training and testing, the implementation of GBDT model is realized by the XGBoost package.[2]

Algorithm GBDT
Input: training set $D = (X, Y)$, feature vector $X = \{x_{k,i}, k = 1, 2, \ldots, K;\ i = 1, 2, \ldots, N\}$ and target $Y = \{y_i, i = 1, 2, \ldots, N\}$ and $y_i \in \{-1, 1\}$.
Output: a binary classifier $F(X) = sign[f_M(X)] = sign[\sum_{m=1}^{M} \eta_m b(X; \gamma_m)] \in \{-1, 1\}$ in a form of a weighted sum of tree $b(X; \gamma_m)$ where γ_m parametrizes the splits.
(1) Initialize the model with a constant value: $f_0(X) = 0$.
(2) **for** $m = 1$ to M **do**
(3) Compute $(\gamma_m, \eta_m) = \underset{\gamma,\eta}{\text{argmin}} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_{k,i}) + \eta b(x_{k,i}; \gamma)) + \nu T + \frac{1}{2}\beta\|\gamma\|^2$, where T is the number of leaves in the tree, the loss function $L(\cdot)$ not only measures how well the model fit the training data, but also measures the model complexity mainly through the term $\eta b(x_{k,i}; \gamma)$.
(4) Update the model: $f_m(X) = f_{m-1}(X) + \eta_m b(X; \gamma_m)$.
(5) **end for**
Return: $f_m(X)$.

3. Our proposed approaches

3.1. LR2GBDT: Cascading LR onto GBDT

Different models can complement each other. As a linear model, the output of the LR has a nice probabilistic interpretation, and can be updated easily with new data using stochastic gradient descent. Moreover, the algorithm can be regularized to avoid over-fitting. Compared with linear models, GBDT can process multiple types of features as well as collinearity simultaneously. It does not need extra feature preprocessing such as normalization, and it enables us to characterize the importance of different features to enhance the interpretability of the obtained models.

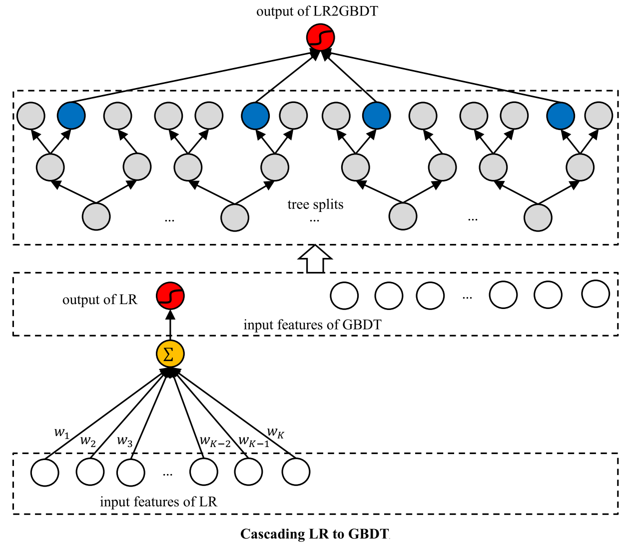

Fig. 2. Graphical illustration of the cascaded model: LR2GBDT.

We propose to combine these two models by cascading LR onto GBDT, with the goal of predicting the directions of financial market price changes. We refer to this combination as LR2GBDT for simplicity. As shown in Fig. 2, the LR2GBDT model is built by first training the LR model, and then looking at the resulting LR's predicting score as an additional feature, and feeding it together with the original features as the inputs of the GBDT model. Given a sample X, the specific scoring formula is described as follows:

$$\hat{y}_{LR2GBDT} = sigmoid(\bar{z})\ , \qquad \bar{z} = \sum_{m=1}^{M} \eta_m b(\{X, \hat{y}_{LR}\}; \gamma_m), \tag{6}$$

where $\hat{y}_{LR2GBDT}$ is the prediction score derived from the prediction value $\bar{z}$ of GBDT, which is fed by the original input features X and the predicting score of the LR (i.e., $\hat{y}_{LR}$).

From the architecture of LR2GBDT, we can see how the LR and GBDT complement each other, with the potential to achieve better prediction accuracy of the classifiers. LR is a generalized linear model with the advantages of simplicity, interpretability and scalability. However, since it can only model linear relations between features, non-linear relations have to be included manually. In contrast, the GBDT model is a powerful and convenient way to implement non-linear transformations of input features, and interpretable results can be obtained without extra feature preprocessing [43]. As such, once suitable features are picked, the performance of the model would be improved significantly.

3.2. Trading strategy

Since the return is not proportional to predicted accuracy. For instance, there could be large rare losses that would have minimal impact on the prediction accuracy but would degrade the trading performance. The next issue to be addressed would thus be discussing the profitability of a trading strategy under the predictions obtained from the LR2GBDT approach. However, rather than developing an optimal trading strategy for personal investment, this academic article is more concerned about examining the effect of learning architectures on the predictive ability of earnings. Hence, we use the simplest possible trading strategy, also name it long–short strategy, to assist our analysis. The long–short trading strategy is not presented at random, it has certain theory and practice foundation. First, the long–short strategy is complied with the idea, presented by Zhou et al. [50], regarding

[2] https://github.com/dmlc/xgboost.

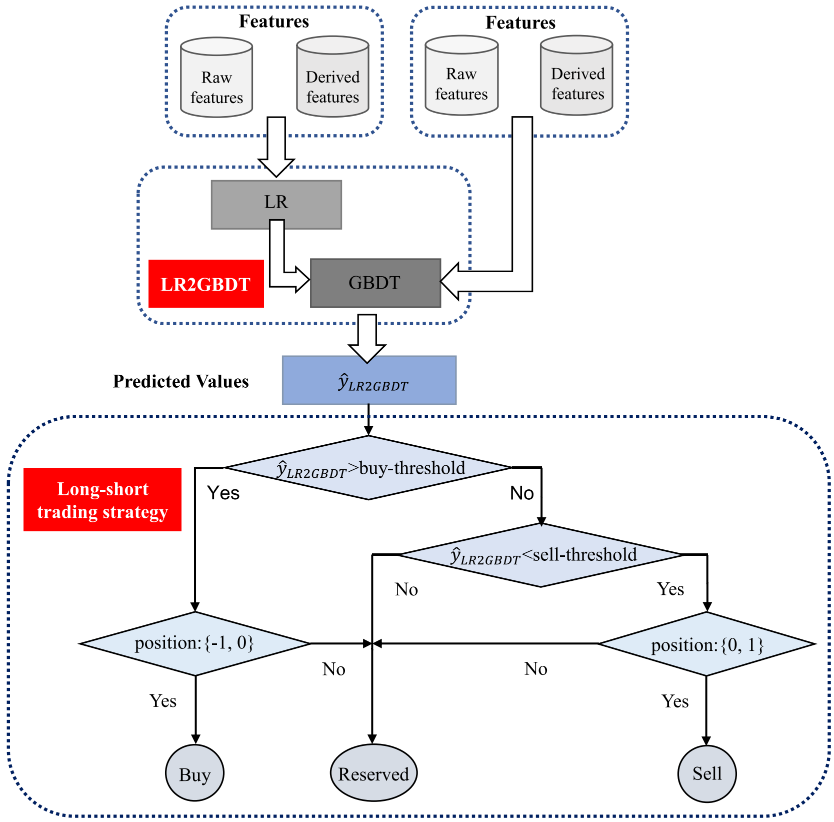

Fig. 3. Graphical illustration of the trading strategy.

the underlying strategies as nonlinear interactions of the input time series that is able to uncover the natures of the stock index generating process. Second, it has been used to compute the profitability by our collaborator Zhou in [38] .

Fig. 3 clearly shows not only the processes of the LR2GBDT but also the rules of the so-called long–short strategy. Note that the buy-threshold and the sell-threshold are introduced in the long–short strategy to decide when to enter into or exit from the market. In this flow chart, we first get the predicted value according to the LR2GBDT model, where the value is ranged from [0, 1], and represent the probability of the next day's price going up. Then, it decides to buy (sell) if the predicted probability is higher than the buy-threshold (lower than the sell-threshold); otherwise no deal will be make.

In addition to the thresholds, the trading activity in the long–short strategy is also associated with positions. We limit the unit of short position and long position to no more than one. Specifically, if the first predicted value is larger than the buy-threshold, the strategy consists in buying one unit of the index. At this point, the position is $+1$, and it will be kept until the next predicted probability lower than the sell-threshold, at which time the position is changed to -1, i.e., the existing position is sold and further one additional unit of the stock index is borrowed. Similarly, we change the position -1 to $+1$ when the next predicted value is larger than the buy-threshold. This process is repeated until the end.

In our experiments, we will analyze the profitability with the long–short trading strategy under two scenarios with taking or not taking the transaction cost into consideration. In the absence of transaction cost, the values of buy-threshold and sell-threshold are set as 0.5; otherwise, they are tuned as maximizing the benefit on the training and validation datasets by grid search on interval (0, 1) and satisfied that buy-threshold is larger than the sell-threshold.

4. Data, features and measures

4.1. Description of the financial data

To examine the performance of our proposed approaches, the Shanghai Stock Exchange Composite Index (SSEC) is chosen as the stock index data of the Chinese stock market (an emerging stock market) in our experiments, while the Nasdaq Composite Index (NCI) and the Standard & Poor's 500 Composite Stock Price Index (S&P 500) are together used as the stock indices data for the stock market in the United States (a mature stock market). These data were downloaded from Yahoo Finance Website,[3] including the daily opening price, high price, low price, closing price and trading volume for each index.

The data of NCI and S&P 500 cover the period from January 3, 2012 to December 23, 2016, while SSEC covers from January 4, 2010 to December 31, 2014. In our experiments, they will be divided into training, validation and testing datasets as the ratio of 7 : 2 : 1. The descriptions of the data are reported in Table 1.

4.2. Technical indicators and derived features

Since we attempt to forecast the direction of daily closing price changes in stock indices, it is necessary to introduce technical indicators (or named hand-designed features) to improve the accuracy of prediction. As stock indices are subject to many factors of influence, we cannot expect to achieve good result by applying a single technical indicator. According to the review of domain scholars and prior researches [51–53], we notice that the combination of technical indicators summarized in Table 2 have been widely used as the input variables. We thus select these 12 technical indicators to add to the initial variables.

[3] Yahoo Finance website: https://finance.yahoo.com.

Table 1
Data and description, where N_{all} denotes total number of trading days.

Name	N_{all}	Data range	Testing dataset
Nasdaq Composite Index (NCI)	1255	2012.01.03–2016.12.23	2016.07.01–2016.12.23
Standard and Poor's 500 Composite Stock Price Index (S&P 500)	1254	2012.01.03–2016.12.23	2016.06.29–2016.12.23
Shanghai Stock Exchange Composite Index (SSEC)	1210	2010.01.04–2014.12.31	2014.07.08–2014.12.31

Table 2
Initially selected technical features and their formulas. C_t is the closing price at time t, L_t is the low price on the day indexed by time t, H_t and O_t are the corresponding high and opening price, and MA_n is the moving average of the price over n days.

Feature name	Description	Formula	Reference
Stochastic $\%K_t$	It compares where a security's price closed relative to its price range over a given time period.	$\frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} * 100$, LL_t and HH_t mean lowest low and highest high in the last n days, respectively.	[54]
Stochastic $\%D_t$	Moving average of Stochastic $\%K_t$.	$\frac{\Sigma_{i=0}^{n-1} \%K_{t-i}}{n}$	[54]
Stochastic slow $\%D_t$	Moving average of Stochastic $\%D_t$.	$\frac{\Sigma_{i=0}^{n-1} \%D_{t-i}}{n}$	[55]
$Momentum_t$	It measures how much a security's price has changed over a given time span.	$C_t - C_{t-n-1}$	[56]
ROC_t	Price rate-of-change quantifies the variation between the current price and the price n days ago.	$\frac{C_t}{C_{t-n}} * 100$	[57]
Larry Williams' $\%R_t$	It measures overbought/oversold levels.	$\frac{H_n - C_t}{H_n - L_n} * 100$	[54]
$A/D\ Oscillator_t$	Accumulation/distribution oscillator associates changes in price with volume.	$\frac{H_t - C_{t-1}}{H_t - L_t} * volume_t$	[56]
n-$day\ Disparity_t$	It is the distance between the current price and the moving average of n days.	$\frac{C_t}{MA_n} * 100$	[58]
$2n$-$day\ Disparity_t$	It is the distance between the current price and the moving average of $2n$ days.	$\frac{C_t}{MA_{2n}} * 100$	[58]
$OSCP_t$	Price oscillator displays the relative difference between two moving averages of a security's price.	$\frac{MA_n - MA_{2n}}{MA_n}$	[54]
CCI_t	Commodity channel index measures the variation of a security's price from its statistical mean.	$\frac{M_t - SM_t}{0.015 * D_t}$, where $M_t = \frac{H_t + L_t + C_t}{3}$, $SM_t = \frac{\Sigma_{i=1}^{n} M_{t-i+1}}{n}$ and $D_t = \frac{\Sigma_{i=1}^{n} \lvert M_{t-i+1} - SM_t \rvert}{n}$	[54,56]
RSI_t	Relative strength index is a momentum oscillator that measures the speed and change of price movements over a given time span, which ranges from 0 to 100.	$100 - \frac{100}{1 + (\frac{\Sigma_{i=0}^{n-1} UP_{t-i}}{n})/(\frac{\Sigma_{i=0}^{n-1} DW_{t-i}}{n})}$, where upward-price-change $UP_t = \max\{0, C_t - C_{t-1}\}$ and downward-price-change $DW_t = \max\{0, C_{t-1} - C_t\}$.	[54]

4.3. Metrics of prediction accuracy

We define I_i as the indicator of the success of the prediction for the ith trading day by

$$I_i = \begin{cases} 1 & \text{if } sign(\hat{y}_i - 0.5) * y_i > 0, \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where $sign(a)$ is the sign function, and it equals to 1 if $a \geq 0$, -1 otherwise. $\hat{y}_i$ denotes the predicted probability of the closing price increases, and y_i represents the real direction of the closing price index changes, whose value is taken as 1 if it increases, -1 otherwise. $sign(\hat{y}_i - 0.5) * y_i > 0$ means that the model has the same trend judgment to the truth with a probability more than 0.5, which is regarded as a "hit". To assess the prediction performance of the proposed model, the *Hit ratio* [59] is evaluated using the following equation:

$$Hit\ ratio = \frac{\sum_{i=1}^{N} I_i}{N}. \tag{8}$$

The correctness of a classification can be also evaluated by computing the number of correctly recognized class examples (true positives, TP), the number of correctly recognized examples that do not belong to the class (true negatives, TN), and examples that either were incorrectly assigned to the class (false positives, FP) or that were not recognized as class examples (false negatives, FN). The metrics *Precision*, *Recall* and *F-measure* [60] are used to evaluate the prediction performance of the proposed models. These three measures are defined as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{9}$$

$$Recall = \frac{TP}{TP + FN}, \tag{10}$$

$$F\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{11}$$

where *Recall* is thus the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate. *F-measure* is an increasing function of both *Precision* and *Recall*, which achieves its maximum value when $Precision = Recall = 1$. When $Precision = Recall$, it reduces to $F\text{-}measure = Precision = Recall$.

4.4. Metrics of trading performance

Four statistical measures are chosen to evaluate the trading performances of different predicted techniques with the long–short trading strategy. They are Sharpe Ratio (SR), Maximum Drawdown (MD), Average Annual Return (PnL) and PnL/MD. Their descriptions are given in Table 3, where the SR depicts the risk-adjusted return, the MD denotes the largest cumulative loss due to a sequence of drops over the point of investment, the PnL indicates the average annual return, and PnL/MD is computed as the PnL derived by the MD and is mildly modified from the Calmar ratio. In addition, the number of Trades is also listed in the table as an indicator. In our experiments, all of the metrics will used to judge the trading performances. For SR, PnL and PnL/MD, a higher value reveals that the trading performance is better. But for MD, a lower output is better.

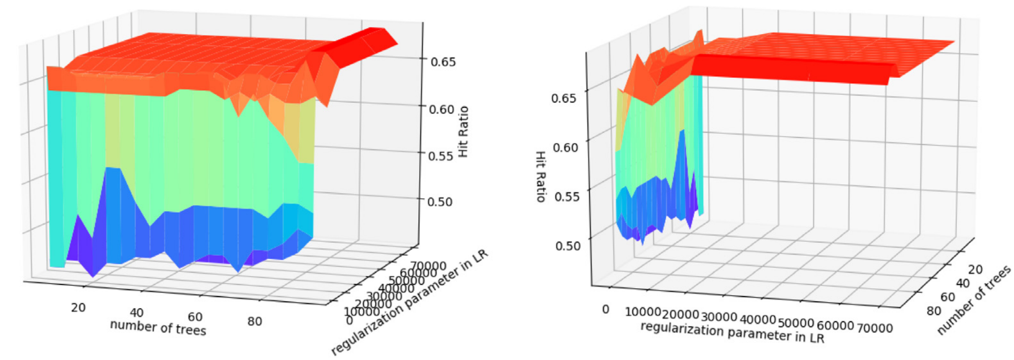

Fig. 4. *Hit ratio* as a function of number of trees and regularization parameter in LR for the NCI.

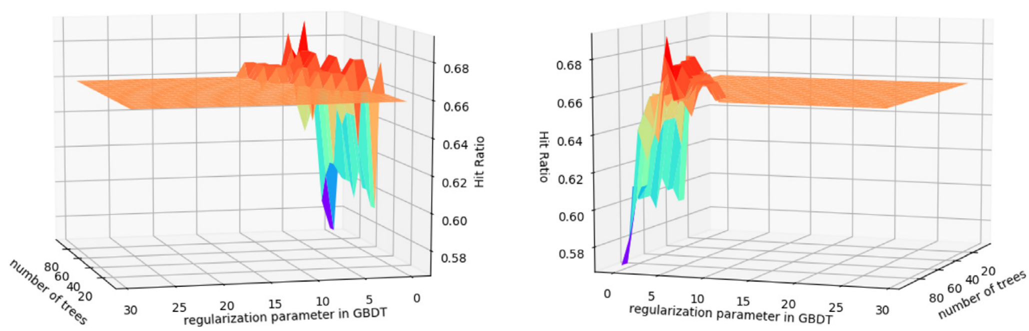

Fig. 5. *Hit ratio* as a function of number of trees and regularization parameter in GBDT for the NCI.

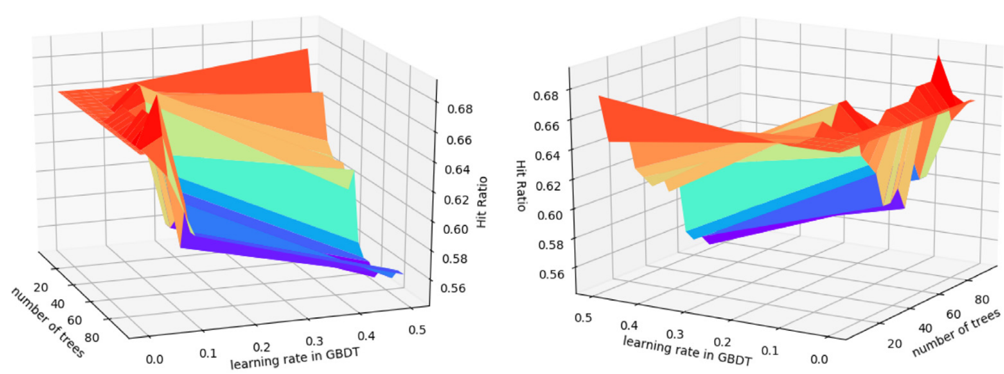

Fig. 6. *Hit ratio* as a function of number of trees and learning rate in GBDT for the NCI.

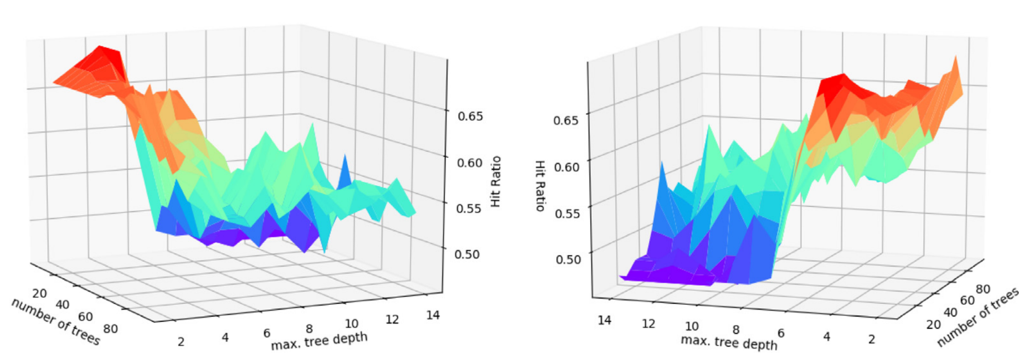

Fig. 7. *Hit ratio* as a function of number of trees and maximum tree depth for the NCI.

5. Experimental results and evaluation

To comprehensively illustrate the performances of our proposed approaches, we put forward five questions, which are: (1) Can the cascaded LR2GBDT-based model be trusted? (2) Can the best classifier be selected with good confidence (i.e. stability)? (3) Is the LR's predicting score feature relevant? (4) Can the LR2GBDT-based strategies make profits? (5) Does transaction

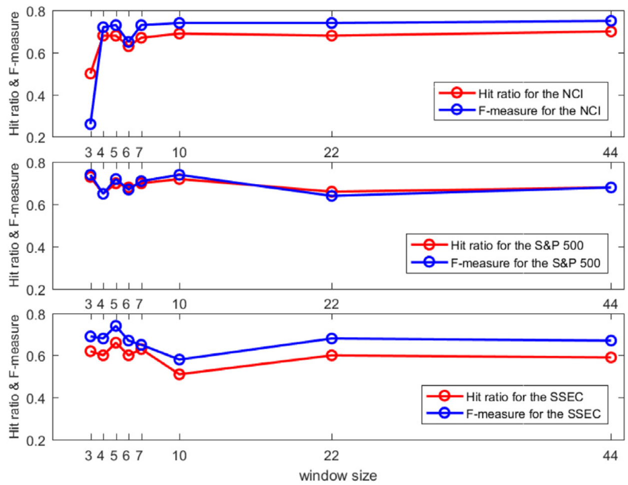

Fig. 8. *Hit ratio* and *F-measure* as functions of window size for the NCI, S&P 500 and SSEC.

Table 3
The statistical metrics of trading performance, where r_i denotes the return of year i; R_t is the cumulative return until date t over a period T; $\mu(R_t)$ and $\sigma(R_t)$ are the corresponding mean and standard deviation of the return R_t.

Statistical measure	Description
PnL	$(\prod_{i=1}^{N}(1+r_i))^{\frac{1}{N}}-1$
MD	$\max_{\tau\in(0,T)}(\max_{t\in(0,\tau)}(R_t-R_\tau))$
SR	$\frac{\mu(R_t)}{\sigma(R_t)}$
PnL/MD	$\frac{(\prod_{i=1}^{N}(1+r_i))^{\frac{1}{N}}-1}{\max_{\tau\in(0,T)}(\max_{t\in(0,\tau)}(R_t-R_\tau))}$
Number of trades	Total number of entered trades

cost remove the vested profits? And we would like to carry out experiments to address these issues one by one.

To evaluate the effectiveness of our cascaded model, we apply the LR and GBDT models as the baselines in our experiments, and compare their results with those of LR2GBDT. In addition, we also compare our proposed LR2GBDT approach against SVM,[4] NN with 2 nonlinear hidden layers and TPOT,[5] where the last three models have their respective advantages in financial prediction [31–33]. SVM has a good way to avoid over-fitting and trapping in local minimum. Compared to linear forecasting models, NN can capture complex nonlinear relationships and generalize well to be good universal approximators. TPOT can be regarded as a "Data Science Assistant" that can explore the data, discover novel features, and automatically construct machine learning pipelines by using the genetic programming with a goal of maximizing classification accuracy. At last, we would like to note that the kernel function in SVM and the hyper-parameters appeared in the models are selected by using the grid search technique, such as the learning rate in LR, GBDT and LR2GBDT, the size of trees and the depth of trees in GBDT, NN and LR2GBDT, the degree of the "poly" kernel function in SVM, the size of neurons of each hidden layer in NN, and the generations and population size in TPOT.

5.1. Can the cascaded LR2GBDT-based model be trusted?

For each dataset in Table 1, the input features in our setting contain the raw daily trading data (opening, high, low, closing prices and volume), $\log\frac{O_{t-1}}{O_{t-2}}$, $\log\frac{H_{t-1}}{H_{t-2}}$, $\log\frac{L_{t-1}}{L_{t-2}}$, $\log\frac{C_{t-1}}{C_{t-2}}$, together with the computation of the technical indicators as discussed in Table 2. All these models (i.e., LR, GBDT, SVM, NN, TPOT, LR2GBDT) are solving the classification problem with two class values "+1" and "−1", where "−1" means that the next day's closing price index is lower than the today's, "+1" otherwise. And it is not allowed to "sell" first then "buy" a stock index as an option.

Table 4 provides a summary of the results obtained for these six classifiers, using the metrics (*Hit ratio*, *Precision*, *Recall*, *F-measure*). The largest values among these models are shown in boldface.

The first interesting observation is that the input features (i.e., the raw daily trading data and 12 technical indicators) constructed from the historical information provide significant explanatory power, as the LR, TPOT and LR2GBDT models have performances at least better than random guessing on average, reflected in the fact that all measures are larger than 50%.

From the results of "**Testing data**" in the right part of Table 4, one can see that LR2GBDT is superior to the other approaches in terms of all four metrics for the NCI and SSEC. And the TPOT outperforms the others for the S&P 500, except for the value of the *Precision* metric that is lower than that of the LR2GBDT. When we compare the results of training and testing data, Table 4 also shows that the cascaded model is useful to cope with the overfitting problems of GBDT. GBDT shows worse results on testing data than training data for the three stock price indices, with an average degradation of the *Hit ratio* larger than 20%, while the LR2GBDT's metrics deteriorate much less. Besides, the results of the training data sets by LR2GBDT are found to be better approximated than those of the LR, SVM and TPOT.

5.2. Can the best classifier be selected with good confidence?

As we know, there are several hyper-parameters in above models. In our experiments, they are selected by grid search technique. Next, we would like to analyze the reliability of our cascaded ensemble-based classifiers, i.e., LR2GBDT, by discussing the sensitivity of the *Hit ratio* metric to different parameters. The involved parameters include the number of trees, the regularization parameter in LR, the regularization parameter in GBDT, learning rate in GBDT, and the maximum tree depth in GBDT. Figs. 4–7 analyze the dependence of *Hit ratio* as a function of different pairs of parameters taken from this list of five parameters for the testing data of NCI. For each figure, we present the 3D plot from two different angles for better visibility.

These figures allow us to obtain the following insights:

(1) Figs. 4 and 5 show that *Hit ratio* is not very sensitive to the 'number of trees', when the 'regularization parameter in LR' and the 'regularization parameter in GBDT' are given and fixed. But as shown in Figs. 6 and 7, this accuracy measure could be influenced by the 'learning rate in GBDT' and the 'maximum tree depth'.
(2) *Hit ratio* is sensitive to the 'regulation parameter in LR' and the 'regulation parameter in GBDT', when the 'number of trees' is given and fixed. Since its value increases fast at first when these two parameters are small, but it stabilizes later when these two parameters increase.
(3) Observing the left part of Fig. 6, there seems to be different linear relationships between the *Hit ratio* and 'learning rate in GBDT', when the 'number of trees' is given and fixed.
(4) According to the right part of Fig. 7, an interesting finding is that the maxima of *Hit ratio* can be achieved along a parabola in red, indicating the nonlinear relationship between the 'learning rate in GBDT' and the 'number of trees'.

[4] SVM package: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.
[5] TPOT package: http://epistasislab.github.io/tpot/.

Table 4
(*Hit ratio*, *Precision*, *Recall*, *F-measure*) of six models for the training and testing data sets from NCI, S&P 500 and SSEC.

Asset	Measure	Training data						Testing data					
		LR	GBDT	SVM	NN	TPOT	LR2GBDT	LR	GBDT	SVM	NN	TPOT	LR2GBDT
NCI	*Hit ratio*	0.71	**0.92**	0.72	0.57	0.75	0.74	0.66	0.51	0.66	0.49	**0.68**	**0.68**
	Precision	0.71	**0.90**	0.70	0.57	0.72	0.72	0.68	0.63	0.67	0.58	0.63	**0.69**
	Recall	0.81	**0.96**	0.80	**0.96**	0.78	0.86	0.70	0.32	0.78	0.30	0.67	**0.79**
	F-measure	0.75	**0.93**	0.74	0.71	0.75	0.78	0.71	0.42	0.72	0.40	0.69	**0.73**
S&P 500	*Hit ratio*	0.65	**0.82**	0.63	0.55	0.65	0.65	0.69	0.56	0.69	0.48	**0.71**	0.70
	Precision	0.64	**0.80**	0.62	0.55	0.66	0.65	0.68	0.67	0.70	0.49	0.70	**0.71**
	Recall	0.77	**0.90**	0.82	0.87	0.69	0.76	0.73	0.28	0.69	0.53	**0.77**	0.75
	F-measure	0.70	**0.84**	0.71	0.68	0.68	0.70	0.71	0.40	0.69	0.47	**0.73**	0.72
SSEC	*Hit ratio*	0.66	**0.93**	0.65	0.56	0.65	0.73	0.59	0.52	0.55	0.62	0.64	**0.66**
	Precision	0.66	**0.95**	0.65	0.56	0.65	0.73	0.70	0.59	0.70	0.66	**0.71**	0.70
	Recall	0.65	**0.92**	0.66	0.56	0.65	0.71	0.56	0.69	0.49	0.77	0.68	**0.78**
	F-measure	0.66	**0.93**	0.65	0.56	0.65	0.72	0.62	0.64	0.55	0.71	0.70	**0.74**

Besides, the window size n (the number of points of daily series that is used for the technical indicators in Table 2) is a potentially important parameter that might affect the observed level of prediction accuracy. Extensive experiments are thus conducted to study the sensitivity of the *Hit ratio* and *F-measure* metrics to the window size. Fig. 8 presents the results of the window sizes of 3, 4, 5 (about one week), 6, 7, 10 (about two weeks), 22 (about one month) and 44 (about two months) trading days for the NCI (top panel), S&P 500 (middle panel) and SSEC (bottom panel). For all window sizes considered in the plot, the LR2GBDT model with window size equals to 5 yields the relatively high *Hit ratio* and *F-measure* for the NCI, S&P 500 and SSEC, although it is very close to the prediction accuracies presented for other window sizes. In addition, increasing the size of the sliding window to 10, 22, or 44 trading days does not significantly improve the prediction performance. Thus, the following sections are obtained from the window size $n = 5$ based on the evidence that the choice of 5 is appropriate and does not significantly affect the conclusions of the comparison of alternative models.

5.3. Is the LR's predicting score feature relevant?

It seems that the input features of LR2GBDT are the raw daily trading data, $\log\frac{O_{t-1}}{O_{t-2}}$, $\log\frac{H_{t-1}}{H_{t-2}}$, $\log\frac{L_{t-1}}{L_{t-2}}$, $\log\frac{C_{t-1}}{C_{t-2}}$ and the 12 technical indicators listed in Table 2. However, after taking a closer look at Figs. 2 and 3, we can find that the LR2GBDT has the same setting of hyper-parameters as the GBDT model except an additional feature derived from the LR model. Thus, the difference between the GBDT and LR2GBDT models can be attributed to the derived feature via the LR model. Next, we verify its importance and rank the input features. Based on the average gain of each feature when it was used in a tree [61], Fig. 9 thus presents the most important features of the model with the highest *Hit ratio* in Table 4 for the NCI, S&P 500 and SSEC, respectively.

By building the decreasing feature important lists, these figures show that the LR's predicting score is the most important feature to predict the changes of all these three stock price indices in our model. Besides, comparing the results derived from the GBDT model and LR2GBDT model in Table 4, we can also see the improvement of the predicted values derived from the LR feature.

5.4. Can the LR2GBDT-based strategies make profits?

To evaluate the performance of the long–short trading strategy for the LR2GBDT approach, we compare the results against those obtained from the LR and GBDT models with the same trading strategy. Furthermore, we also compare the long–short strategy to the other strategies that are not rely on the predicted models, such as the benchmark 'buy-and-hold strategy' and 'ex post trading strategy', where the former represents buying one unit at the beginning and holding the position until the end, the latter is realized by assuming one can truly foresee the price's rise and fall from today to tomorrow. Hence, the profit of the 'buy-and-hold strategy' is entirely determined by the initial price of entry into the market and the final price of exiting from the market. And the 'ex post trading strategy' cannot be happened in reality, it nevertheless provides an interesting profit ceiling.

The results of the metrics given in Table 3 of the LR2GBDT, LR and GBDT predicted models with the long–short strategy, together with those of the 'ex post trading strategy' and the benchmark 'buy-and-hold strategy' are shown in Table 5. From the results, we have that

(1) As we expected, the 'ex post trading strategy' would obtain a significant performance from the PnL metric, which is ranged from 90% to 418%, and from the MD metric, which equals 0 since foresight is assumed.
(2) For the benchmark 'buy-and-hold strategy', it delivers the PnL metric ranged from 20% to 156%, and the MD metric is taken as 5%.
(3) For the LR and LR2GBDT models, their PnL metrics are close to one-half those of the 'ex post trading strategy', and their MD values are taken in the range of 1%–5%, the SR values are also impressive.
(4) We also see that the GBDT alone performs much worse than LR and LR2GBDT, while LR alone is already impressive. The cascading of LR onto GBDT in the form of the LR2GBDT model provides a clear improvement over the LR approach, especially in the SSEC market.

These are very promising results that are useful to demonstrate genuine forecasting abilities of the cascaded model. The next section quantifies if these statistically significant results are also economically significant in the presence of transaction cost.

5.5. Does transaction cost remove the vested profits?

In practice, a transaction cost is charged for each trade, where the trade covers buying or selling the stock index. The transaction cost is primarily composed of the stamp duty, transfer and commission fees. In absence of transaction cost, it is easy to create the illusion of good earnings. Hence, the trading strategy performing well with not taking transaction cost into consideration do not lead economically relevant. In this part, we take the transaction cost as 0.3% for simplicity, and we explore the arbitrage performances of the models with the long–short trading strategy in consideration of the transaction cost. The results are also compared with those obtained from the 'buy-and-hold strategy' and 'ex post trading strategy'.

The results are shown in Table 6. We clearly see that the transaction cost dramatically weakens the trading performance, where

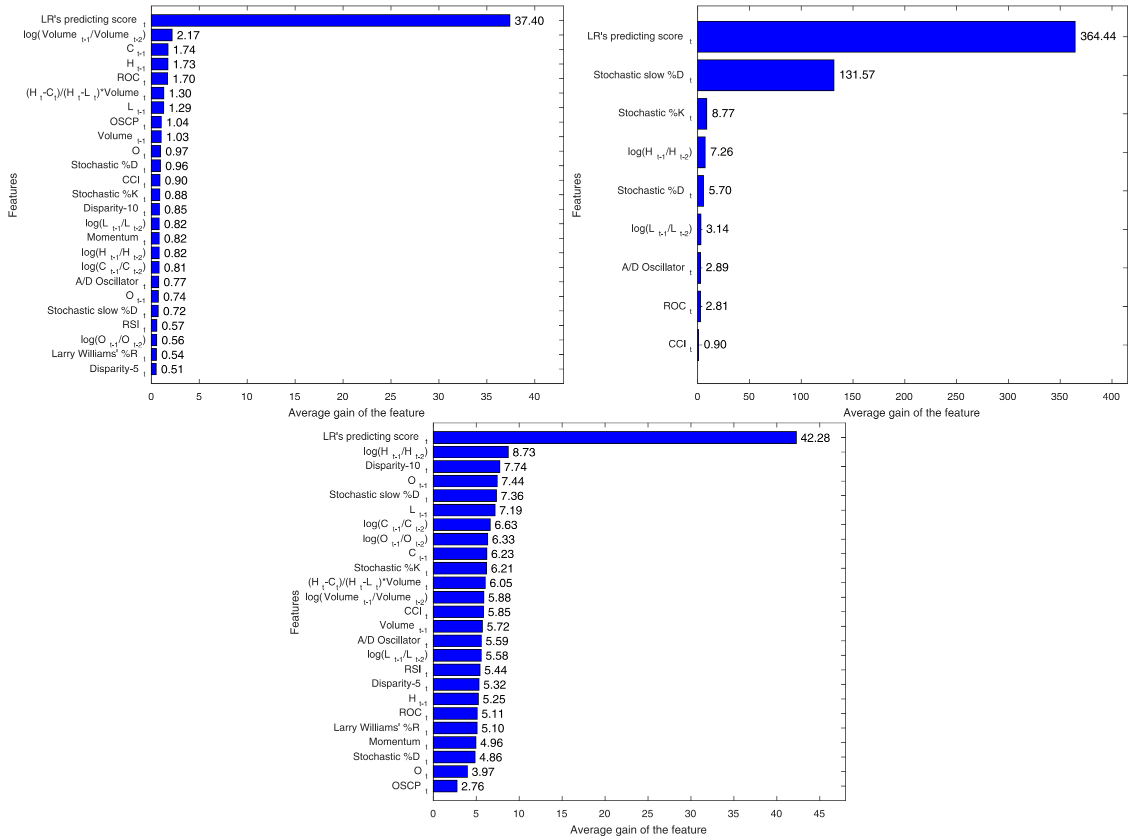

Fig. 9. The relative importance of features based on how many times each feature is split by the GBDT model for the different stock prices. Top left: for the NCI. Top right: for the S&P 500. Bottom: for the SSEC.

Table 5
(PnL, MD, SR, PnL/MD, Number of trades) of the long–short strategy using the LR2GBDT, LR and GBDT models, together with those of the 'buy-and-hold strategy' denoted **Benchmark** and of the 'ex post trading strategy' denoted **Ex post** for the testing sets of three financial markets (i.e., NCI, S&P 500 and SSEC).

Asset	Measure	Ex post	Benchmark	LR	GBDT	LR2GBDT
NCI	PnL	1.25	0.27	**0.52**	0.24	0.47
	MD	0.00	0.05	0.04	**0.02**	0.03
	SR	11.33	2.22	4.93	3.44	**5.48**
	PnL/MD	$+\infty$	5.40	13.00	12.00	**15.67**
	Number of trades	33	1	26	17	27
S&P 500	PnL	0.90	0.20	0.48	0.07	**0.49**
	MD	0.00	0.05	**0.01**	**0.01**	**0.01**
	SR	10.27	2.00	6.21	1.57	**6.39**
	PnL/MD	$+\infty$	4.00	48.00	7.00	**49.00**
	Number of trades	36	1	34	8	34
SSEC	PnL	4.18	1.56	1.21	0.68	**1.57**
	MD	0.00	0.05	**0.05**	0.06	**0.05**
	SR	10.80	4.61	4.61	3.11	**5.21**
	PnL/MD	$+\infty$	31.20	24.20	11.33	**31.40**
	Number of trades	26	1	27	24	26

all of the metric values decrease significantly. The exposure of returns to common sources of systematic risk is evaluated for the strategy. As such, some structure is detected, but the effects are not large enough to construct a profitable strategy in the light of transaction cost. However, compared with the baseline LR and GBDT models, the LR2GBDT still gives better results with PnL between 8% and 121% and MD about 5% or 6%. In the presence of transaction cost and the introduction of the adaptive adjustment of the buy and sell thresholds, the trading performance of GBDT is now much better in comparison with LR model.

Furthermore, Figs. 10–12 are plotted the evolution of the trading performance as the time series during the testing phase based on the predicted values of the LR2GBDT, whose performance metrics have been presented in Table 5 for no transaction cost and in Table 6 with transaction cost, for the NCI, the S&P 500 and the SSEC, and the results from the buy-and-hold strategy. The normalized compounded profits of the LR2GBDT-based long–short strategy without transaction cost (red line) and with transaction cost (blue line), the buy-and-hold strategy (black line) are presented in the top panels. The other panels list the relevant signs of entry into and exiting from the market of the LR2GBDT-based long–short strategy with taking the transaction cost into account (middle panel) or not (bottom panel).

To investigate the influence of the buy threshold and sell threshold parameters on the results, Fig. 13 presents two 3D plots

Table 6
Same as Table 5 with taking transaction cost into consideration.

Asset	Measure	Ex post	Benchmark	LR	GBDT	LR2GBDT
NCI	PnL	0.46	0.27	0.06	0.13	**0.22**
	MD	0.01	0.05	0.08	**0.06**	**0.06**
	SR	5.01	2.16	0.65	1.14	**2.34**
	PnL/MD	46.00	5.40	0.75	2.17	**3.67**
	Number of trades	36	1	25	7	11
S&P 500	PnL	0.20	0.20	−0.04	0.03	**0.08**
	MD	0.03	0.05	0.06	0.06	0.06
	SR	2.60	1.93	−0.52	0.37	**0.85**
	PnL/MD	6.67	4.00	−0.67	0.50	**1.33**
	Number of trades	33	1	34	6	10
SSEC	PnL	2.65	1.55	0.56	1.06	**1.21**
	MD	0.02	0.05	0.06	**0.05**	**0.05**
	SR	8.17	4.58	2.62	3.65	**4.05**
	PnL/MD	265.00	31.00	9.33	21.20	**24.20**
	Number of trades	26	1	27	8	14

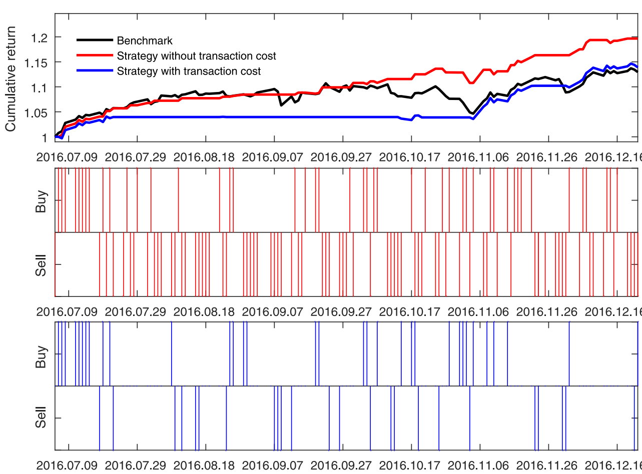

Fig. 10. Top panel: Time evolution of the trading performance in the testing period of the NCI based on the predictions of the LR2GBDT model, for the benchmark strategy (black line), for the strategy without transaction cost (red line) and strategy with transaction cost (blue line), in comparison with the buy-and-hold strategy. Middle panel: buy and sell signals of the LR2GBDT-based trading strategy without transaction cost. Bottom panel: buy and sell signals of the LR2GBDT-based trading strategy with transaction cost.

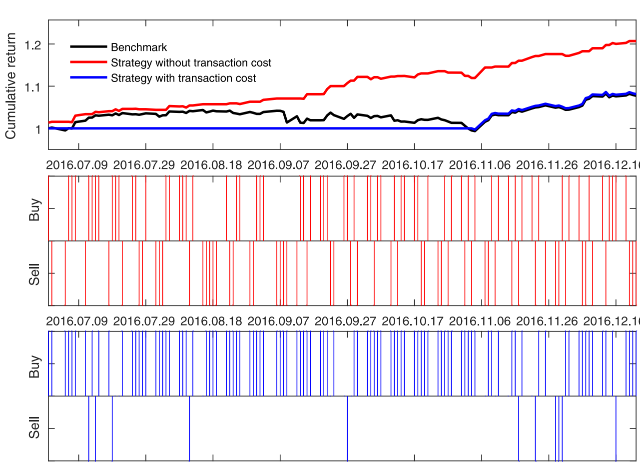

Fig. 11. Same as Fig. 10 for the S&P 500.

with two different views of the Sharpe Ratio as a function of the two thresholds for the Nasdaq Composite Index data. This figure

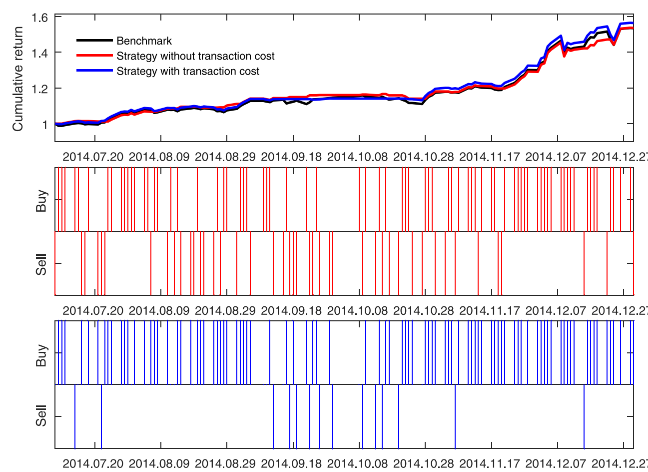

Fig. 12. Same as Fig. 10 for the SSEC.

shows that the maximum of the Sharpe Ratio is reached when the buy threshold equals to 0.64 and the sell threshold is 0.22.

6. Concluding remarks

Financial markets are particularly efficient when it comes to processing information; such information is typically embedded in the price that is then interpreted by market participants. We have aimed at forecasting the direction of the daily changes of stock indices of two different stock markets, which are an emerging market (SSEC) and a mature stock market (NCI and S&P 500). For this, we have cascaded the LR model onto the GBDT model to develop a learning architecture LR2GBDT, and used raw price data and twelve technical indicators fed as the input features. Comparing the results of the LR, GBDT, SVM, NN and TPOT models, we have demonstrated the genuine forecasting ability of the cascaded model. The results have also suggested that the feasibility of profitable model-based trading are influenced by the maturity of the market, the learning model employed, the cost of transaction and buy–sell thresholds strategy.

To the best of our knowledge, this study is the first in deploying the cascaded ensemble learning architecture to forecast and trade stock indices. We have also shown the positive contribution of the LR's predicting score feature on the performance of cascaded ensemble-based learning classifiers for the task of predicting the direction of price indices changes. For the three stock indices studied here, the LR2GBDT model has in general the highest accuracy (measured by *Hit ratio*) and profit

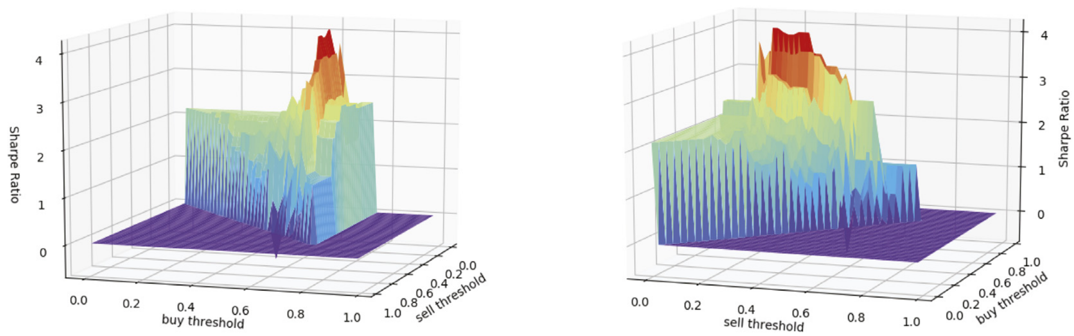

Fig. 13. Sharpe Ratio as a function of the buy threshold and sell threshold for the NCI.

(measured by Sharpe Ratio) and the lowest loss (measured by Maximum Drawdown), than the baseline LR model and GBDT model. When transaction cost and buy–sell thresholds are taken into account, the best trading strategy derived from LR2GBDT model still reaches the highest Sharpe Ratio than that of two baseline models. Although the transaction cost decrease profits, the LR2GBDT-based trading strategy is still profitable. Overall, our results demonstrate both statistically as well as economically significant predictive power of the cascaded LR2GBDT model.

It is worth noting that the cascading method adopted offers the prospect of studying the informational efficiency of stock markets as well as provides new insights concerning the supervised classification problem. Although there is no guarantee that the cascaded model will find all patterns in any stock indices, it still stands a better chance to do so, as the implementation avoids the subjective judgment and thus directly reduces the risk. We also surmise that increasing public availability of powerful machine learning methods and decreasing entry barriers in terms of technological investment has led to profits being arbitraged away in recent years.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.105747.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (71801057; 11901113; 11701106); the Humanities and Social Sciences Research Youth Foundation of the Ministry of Education in China (17YJCZH248); the Natural Science Foundation of Guangdong Province, China (2018A030313968; 2018A030313996); the Guangdong Youth Innovation Talent Project, China (natural sciences, 2017KQNCX083) and the Guangzhou Science and Technology Plan Project, China (201904010225).

References

[1] E.F. Fama, Efficient capital markets: A review of theory and empirical work, J. Financ. 25 (2) (1970) 383–417.
[2] E.F. Fama, Efficient capital markets: II, J. Financ. 46 (5) (1991) 1575–1617.
[3] D. Sornette, Physics and financial economics (1776–2014): Puzzles, Ising and agent-based models, Rep. Progr. Phys. 77 (6) (2014) 062001.
[4] S.J. Grossman, J.E. Stiglitz, On the impossibility of informationally efficient markets, Amer. Econ. Rev. 70 (3) (1980) 393–408.
[5] W.F.M.D. Bondt, R. Thaler, Does the stock market over-react? J. Financ. 40 (3) (1985) 793–805.
[6] K. Daniel, D. Hirshleifer, A. Subrahmanyam, Investor psychology and security market under- and overreactions, J. Financ. 53 (6) (1998) 1839–1885.
[7] M. Cipriani, A. Guarino, Herd behavior in a laboratory financial market, Amer. Econ. Rev. 95 (5) (2005) 1427–1443.
[8] A.W. Lo, A.C. MacKinlay, Stock market prices do not follow random walks: Evidence from a simple specification test, Rev. Financ. Stud. 1 (1) (1988) 41–66.
[9] N. Jegadeesh, Evidence of predictable behavior of security returns, J. Financ. 45 (3) (1990) 881–898.
[10] N. Jegadeesh, S. Titman, Returns to buying winners and selling losers: Implications for stock market efficiency, J. Financ. 48 (1) (1993) 65–91.
[11] D. Sornette, S. von der Becke, Crashes and high frequency trading: An evaluation of risks posed by high-speed algorithmic trading, in: The Future of Computer Trading in Financial Markets, Foresight Driver Review, Government Office for Science, London, DR 7, 2011, pp. 1–26.
[12] V. Filimonov, D. Sornette, Quantifying reflexivity in financial markets: Towards a prediction of flash crashes, Phys. Rev. E 85 (5) (2012) 056108.
[13] V. Filimonov, D. Bicchetti, N. Maystre, D. Sornette, Quantification of the high level of endogeneity and of structural regime shifts in commodity markets, J. Int. Money Financ. 42 (2014) 174–192.
[14] J.P. Bouchaud, D. Challet, Why have asset price properties changed so little in 200 years, in: Abergel F., et al. (Eds.), Econophysics and Sociophysics: Recent Progress and Future Directions, New Economic Windows, Springer, 2017, pp. 3–17.
[15] W. Leigh, R. Purvis, J.M. Ragusa, Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: A case studying decision support, Decis. Support Syst. 32 (2002) 361–377.
[16] I.H. Laradji, M. Alshayeb, L. Ghouti, Software defect prediction using ensemble learning on selected features, Inf. Softw. Tech. 58 (2015) 388–402.
[17] E.F. Fama, The behavior of stock-market prices, J. Bus. 38 (1) (1965) 34–105.
[18] P.A. Samuelson, Proof that properly anticipated prices fluctuate randomly, Ind. Manage. Rev. 6 (2) (1965) 41–49.
[19] E.F. Fama, M.E. Blume, Filter rules and stock-market trading, J. Bus. 39 (1) (1966) 226–241.
[20] M.C. Jensen, G.A. Benington, Random walks and technical theories: Some additional evidence, J. Financ. 25 (2) (1970) 469–482.
[21] A. Timmermann, W.J. Granger, Efficient market hypothesis and forecasting, Int. J. Forecast. 20 (1) (2004) 15–27.
[22] C.H. Park, S.H. Irwin, What do we know about the profitability of technical analysis? J. Econ. Surv. 21 (4) (2007) 786–826.
[23] J.L. Wu, L.C. Yu, P.C. Chang, An intelligent stock trading system using comprehensive features, Appl. Soft Comput. 23 (2014) 39–50.
[24] A.W. Lo, A.C. MacKinlay, Stock market prices do not follow random walks: Evidence from a simple specification test, Rev. Financ. Stud. 1 (1) (1988) 41–66.
[25] A.W. Lo, A.C. MacKinlay, A Non-Random Walk Down Wall Street, Princeton University Press, New Jersey, USA, 2002.
[26] P.H. Hsu, C.M. Kuan, Reexamining the profitability of technical analysis with data snooping checks, J. Financ. Econom. 3 (4) (2005) 606–628.
[27] P.H. Hsu, Y.C. Hsu, C.M. Kuan, Testing the predictive ability of technical analysis using a new stepwise test without data snooping bias, J. Empir. Financ. 17 (3) (2010) 471–484.
[28] P.C. Chang, T.W. Liao, J.J. Lin, C.Y. Fan, A dynamic threshold decision system for stock trading signal detection, Appl. Soft Comput. 11 (5) (2011) 3998–4010.
[29] Q. Sun, C.C. Lim, P. Shi, F. Liu, Moving horizon estimation for Markov jump systems, Inform. Sci. 367 (2016) 143–158.
[30] H. Jiang, H. Zhang, Y. Luo, J. Wang, Optimal tracking control for completely unknown nonlinear discrete-time Markov jump systems using data-based reinforcement learning method, Neurocomputing 194 (2016) 176–182.

[31] M.L. de Prado, Advances in Financial Machine Learning, John Wiley & Sons, Inc., 2018.
[32] L.K. Luo, X. Chen, Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction, Appl. Soft Comput. 13 (2) (2013) 806–816.
[33] R.S. Olson, R.J. Urbanowicz, P.C. Andrews, N.A. Lavender, L.C. Kidd, J.H. Moore, Automating biomedical data science through tree-based pipeline optimization, in: European Conference on the Applications of Evolutionary Computation, Springer, Cham, 2016, pp. 123–137.
[34] D. Senol, M. Oezturan, Stock price direction prediction using artificial neural network approach: The case of Turkey, J. Artif. Intell. 3 (4) (2010) 261–268.
[35] C.L. Dunis, R. Rosillo, D. de la Fuente, R. Pino, Forecasting IBEX-35 moves using support vector machines, Neural Comput. Appl. 23 (1) (2013) 229–236.
[36] C.J. Lu, Integrating independent component analysis-based denoising scheme with neural network for stock price prediction, Expert Syst. Appl. 37 (10) (2010) 7056–7064.
[37] Y. Kara, M.A. Boyacioglu, Ö.K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange, Expert Syst. Appl. 38 (5) (2011) 5311–5319.
[38] F. Zhou, H.M. Zhou, Z.H. Yang, L.H. Yang, EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction, Expert Syst. Appl. 115 (2019) 136–151.
[39] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1) (2010) 1–39.
[40] J.Q. Zhao, L.C. Jiao, S.X. Xia, V.B. Fernandes, I. Yevseyeva, Y. Zhou, M.T.M. Emmerich, Multiobjective sparse ensemble learning by means of evolutionary algorithms, Decis. Support Syst. 111 (2018) 86–100.
[41] M. Kaucic, Investment using evolutionary learning methods and technical rules, European J. Oper. Res. 207 (3) (2010) 1717–1727.
[42] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, J.Q. Candela, Practical lessons from predicting clicks on Ads at Facebook, in: 8th International Workshop on Data Mining for Online Advertising, 2014, pp. 1–9.
[43] X. Ling, W. Deng, C. Gu, H. Zhou, C. Li, F. Sun, Model ensemble for click prediction in Bing search Ads, in: Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 689–698.
[44] D. Pradeepkumar, V. Ravi, Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network, Appl. Soft Comput. 58 (2017) 35–52.
[45] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA 113 (15) (2016) 3932–3937.
[46] N. Terui, H.K.V. Dijk, Combined forecasts from linear and nonlinear time series models, Int. J. Forecast. 18 (3) (2002) 421–438.
[47] L.Y. Wei, A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting, Appl. Soft Comput. 42 (2016) 368–376.
[48] H.F. Yu, F.L. Huang, C.J. Lin, Dual coordinate descent methods for logistic regression and maximum entropy models, Mach. Learn. 85 (1) (2011) 41–75.
[49] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 229–236.
[50] W.X. Zhou, G.H. Mu, W. Chen, D. Sornette, Investment strategies used as spectroscopy of financial markets reveal new stylized facts, PLoS One 6 (9) (2011) e24391.
[51] K.J. Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (1) (2003) 307–319.
[52] K.J. Kim, I. Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Syst. Appl. 19 (2) (2000) 125–132.
[53] M. Thenmozhi, G.S. Chand, Forecasting stock returns based on information transmission across global markets using support vector machines, Neural Comput. Appl. 27 (4) (2016) 805–824.
[54] S.B. Achelis, Technical Analysis from A to Z: Covers Every Trading Tool from the Absolute Breadth Index to the Zig Zag, McGraw-Hill, Chicago, 2001.
[55] E. Gifford, Investor's Guide to Technical Analysis: Prediction Price Action in the Markets, Pitman Publishing, London, 1995.
[56] J. Chang, Y. Jung, K. Yeon, J. Jun, D. Shin, H. Kim, Technical Indicators and Analysis Methods, Jinritamgu Publishing, Seoul, 1996.
[57] J.J. Murphy, Technical Analysis of the Futures Markets: A Comprehensive Guide to Trading Methods and Applications, Prentice Hall Press, New York, USA, 1986.
[58] J. Choi, Technical Indicators, Jinritamgu Publishing, Seoul, 1995.
[59] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inform. Process. Manag. 45 (4) (2009) 427–437.
[60] J. Patel, S. Shah, P. Thakkar, K. Kotecha, Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Syst. Appl. 42 (1) (2015) 259–268.
[61] Y. Liu, Y. Gu, J.C. Nguyen, H. Li, J. Zhang, Y. Gao, Y. Huang, Symptom severity classification with gradient tree boosting, J. Biomed. Inform. 75 (2017) S105–S111.

Feng Zhou received the B.S. degree in Information Computing Science from Minnan Normal University, China, in 2010 and Ph.D. degree in Computational Mathematics from Sun Yat-sen University, China, in 2015. As a visiting student, he studied in the School of Mathematics, Georgia Institute of Technology from Sep. 2013 to Sep. 2014. Now he is an Assistant Professor at the School of Information, Guangdong University of Finance and Economics. His research interests include signal analysis, machine learning and ensemble learning.

Qun Zhang is Yushan Young Scholar and Associate Professor at School of Finance, Guangdong University of Foreign Studies. She was visiting researcher of Entrepreneurial Risks in the Department of Management, Technology and Economics at ETH Zurich from September 2014 to March 2016, supported by Guangzhou Elites Project of Guangzhou Municipal Government. She received her doctoral degree in management science and engineering at South China University of Technology (SCUT), China, in 2016, within a Successive Postgraduate and Doctoral Program after two years postgraduate study in School of Science at SCUT. Her research interests are primarily in the complex financial system, machine learning, decision making and risk management.

Didier Sornette is Professor on the Chair of Entrepreneurial Risks at the Swiss Federal Institute of Technology Zurich (ETH Zurich) since March 2006. He is also a professor of the Swiss Finance Institute, and a professor associated with both the department of Physics and the department of Earth Sciences at ETH Zurich. He is a founding member of the Risk Center at ETH Zurich since June 2011. His research interests include the prediction and control of crises and extreme events in complex systems, with applications to financial bubbles and crashes, earthquake physics and geophysics, the dynamics of success towards the diagnostic of systemic instabilities.

Liu Jiang received the B.S. degree in the Department of life and science and majored in Physical Geography, University of Exeter, in 2017. Now she is a master 's student in the Surrey Business School, University of Surrey. Her research interests include data analysis and risk management.