# A causal feature selection algorithm for stock prediction modeling

**Article** *in* Neurocomputing · October 2014

DOI: 10.1016/j.neucom.2014.01.057

Xiangzhou Zhang
Jinan University, Guangzhou, China
**29** PUBLICATIONS **394** CITATIONS
SEE PROFILE

Yong Hu
Fudan University
**578** PUBLICATIONS **13,117** CITATIONS
SEE PROFILE

Kang Xie
Sun Yat-Sen University
**22** PUBLICATIONS **298** CITATIONS
SEE PROFILE

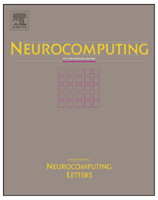
Shouyang Wang
Chinese Academy of Sciences
**698** PUBLICATIONS **12,168** CITATIONS
SEE PROFILE

Neurocomputing 142 (2014) 48–59

Contents lists available at ScienceDirect

Neurocomputing

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



# A causal feature selection algorithm for stock prediction modeling

Xiangzhou Zhang [a,1], Yong Hu [b,*,1], Kang Xie [a], Shouyang Wang [c,d], E.W.T. Ngai [e], Mei Liu [f]

[a] School of Business, Sun Yat-sen University, No. 135, Xingang Xi Road, Guangzhou 510275, PR China
[b] Business Intelligence and Knowledge Discovery, School of Management, Guangdong University of Foreign Studies School of Business, Sun Yat-sen University, Higher Education Mega Center, Guangzhou 510006, PR China
[c] Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, PR China
[d] School of Management, Graduate University of Chinese Academy of Sciences, Beijing 100049, PR China
[e] Department of Management and Marketing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, PR China
[f] Department of Computer Science, New Jersey Institute of Technology, University Heights Newark, NJ 07102, USA

ARTICLE INFO

*Article history:*
Received 16 November 2013
Received in revised form
26 January 2014
Accepted 26 January 2014
Available online 9 May 2014

*Keywords:*
Stock prediction
Data mining
Feature selection
Causal discovery
V-structure

ABSTRACT

A key issue of quantitative investment (QI) product design is how to select representative features for stock prediction. However, existing stock prediction models adopt feature selection algorithms that rely on correlation analysis. This paper is the first to apply observational data-based causal analysis to stock prediction. Causalities represent direct influences between various stock features (important for stock analysis), while correlations cannot distinguish direct influences from indirect ones. This study proposes the causal feature selection (CFS) algorithm to select more representative features for better stock prediction modeling. CFS first identifies causalities between variables and then, based on the results, generates a feature subset. Based on 13-year data from the Shanghai Stock Exchanges, comparative experiments were conducted between CFS and three well-known feature selection algorithms, namely, principal component analysis (PCA), decision trees (DT; CART), and the least absolute shrinkage and selection operator (LASSO). CFS performs best in terms of accuracy and precision in most cases when combined with each of the seven baseline models, and identifies 18 important consistent features. In conclusion, CFS has considerable potential to improve the development of QI product.

## 1. Introduction

Quantitative investment (QI) products (models/tools/systems) can provide accurate stock market prediction and help investors significantly alleviate risks of mispricing and irrational trading because of psychological factors, such as overconfidence, mental accounting, loss aversion, and so on [1,2]. One of the key issues of QI product design lies on how to select representative features for prediction. For example, Thawornwong and Enke [3] have adequately demonstrated the effectiveness of recent relevant variables (i.e., representative features) for improving stock direction prediction based on redeveloped probabilistic and feed-forward neural networks. Their work provides evidence of the importance of feature selection for QI product development.

Feature selection, a pre-processing step of data mining, can be used to filter redundant and/or irrelevant features [4]. Feature selection results in simpler model, easier interpretation, and faster induction and structural knowledge [5]. Although many studies have claimed and/or verified that feature selection is the key process in stock prediction modeling [4], identifying more representative features and improving stock prediction are challenging issues that need to be considered. Common feature selection algorithms adopted in stock prediction models include stepwise regression analysis (SRA), principle component analysis (PCA), decision tree (DT), and information gain [3,4,6]. However, all these algorithms can only reveal underlying correlations/associations and cannot determine the direct (i.e., causal) influence of stock features (inputs) on stock return (output).

This paper aims to provide further insight on the application of observational data-based causal discovery approach on feature selection. Based on Pearl's theory [7] and Hu et al.'s study [8], this study proposes the causal feature selection (CFS) algorithm, with the goal of selecting the optimal feature subset for better stock prediction performance and identifying more representative features for better stock market analysis. This study is highlighted on the following two aspects:

First, causal analysis is applied to identify direct influences between variables. Correlation does not imply causation, while causation requires additional counterfactual dependence.

[*] Corresponding author.
*E-mail addresses:* zhxzhou@mail2.sysu.edu.cn (X. Zhang), henryhu200211@163.com (Y. Hu), mnsxk@mail.sysu.edu.cn (K. Xie), swang@iss.ac.cn (S. Wang), eric.ngai@polyu.edu.hk (E.W.T. Ngai), mei.liu@njit.edu (M. Liu).
[1] Co-first authors.

http://dx.doi.org/10.1016/j.neucom.2014.01.057
0925-2312/© 2014 Elsevier B.V. All rights reserved.

Causal influences are more consistent over time, which is more attractive to stock investors.

Second, to verify the proposed algorithm more objectively, extensive experiments are conducted for multi-aspect performance comparison. These experiments involve seven baseline prediction models, three popular feature selection algorithms, and various performance measures [accuracy, precision, Sharpe ratio, Sortino ratio, information ratio, and maximum drawdown (MDD)].

To evaluate the proposed CFS, listed companies in the Shanghai Stock Exchanges of China are selected as back-testing subjects, and experimental data cover the period from 2000 to 2012. Experimental results show that CFS performs best in terms of accuracy and precision in most cases and identifies 18 representative features consistently over the entire testing period. Moreover, the constructed prediction models can obtain satisfying and stable investment returns. In conclusion, CFS has considerable potential to improve the performance of existing QI products.

The remainder of this paper is organized as follows: Section 2 reviews common filter-based feature selection algorithms and stock prediction models, and then compares related works in terms of datasets, prediction models, feature selection algorithms, and so on. Section 3 introduces the proposed feature selection algorithm based on the causal discovery algorithm. Section 4 presents the experiment setting of dataset, variables, sliding window test, and evaluation strategies, and then reports the empirical results. Section 5 provides a comprehensive conclusion.

## 2. Literature review

### 2.1. Stock prediction

Although the efficient market hypothesis [9] is against stock prediction based on past publicly available information, considerable studies suggest that some markets, especially the emerging markets, are not fully efficient, and prediction of future stock prices/returns may produce better results than random selection [10,11].

Recent studies on stock prediction can be roughly grouped into two types: (a) time series forecasting [12–16] and (b) trend prediction [4,17–20]. A time series forecasting model is trained to fit the historical return/price series of individual stock and is used to predict the future return/price. A trend prediction model is trained to obtain the relationship between various fundamental and/or technical variables and the (rise and decline) movement of stock price (i.e., positive or negative return).

Many popular data mining algorithms have been widely used in stock trend prediction models, including logistic regression (LR) [3,6,21], neural network (NN) [4,21,22], support vector machine (SVM) [23], and decision tree (DT) [21,24]. However, the Bayesian network (BN) [including its variants and naïve Bayes (NB)] has seldom been used directly for stock forecasting/prediction; only Zuo and Kita [25] used BN according to our uncomprehensive search.

Table 1 lists related works in terms of their datasets, prediction models, and feature selection algorithms. Common feature selection algorithms used in stock prediction/forecasting models include SRA, PCA, genetic algorithm (GA), information gain, and so on. Numerous related studies consider both technical and fundamental variables (including economic variables). However, the number of input features used in these studies is different. Currently, no generally agreed-upon representative features for stock prediction are available, and no “best” feature selection algorithm exists. This information motivated us to explore a novel feature selection algorithm by introducing the technique of observational data-based causal discovery to identify a more representative and compact feature set for constructing a simple prediction model with excellent performance.

### 2.2. Feature selection

Common feature selection algorithms can be grouped into following two types: (1) filter and (2) wrapper approaches [28]. The filter approaches use the general characteristics of the training

**Table 1**
Comparison of related work.

| Work | Dataset | Prediction model | Input features | Feature selection |
|---|---|---|---|---|
| Chang et al. [26] | S&P500 index and 5 stocks in S&P500 | Case based FDT[a] | 8 Technical indices | SRA[b] |
| Tsai et al. [21] | Electronic industry of the Taiwan stock market | Classifier ensembles | 19 Financial ratios and 11 economic indicators | – |
| Tsai and Hsiao [4] | Electronic corporations in Taiwan Stock Exchange | BPNN[c] | 22 Fundamental indices and 63 macroeconomic indices | PCA,[d] CART,[e] GA[f] |
| Lai et al. [27] | 3 Taiwan Stock Exchange Corporations | $K$-means+GAFDT[g] | 7 Technical indices | SRA |
| Lee [23] | NASDAQ index | SVM[h] | 17 Financial and economic variables | F_SSFS[i] |
| Enke and Thawornwong [6] | S&P 500 index | BPNN, GRNN,[j] PNN,[k] LR[l] | 31 Financial and economic variables | Info. gain |
| Thawornwong and Enke [3] | S&P 500 index | BPNN, GRNN, PNN, LR | 31 Financial and economic variables | Info. gain |
| Lam [22] | 364 S&P companies | BPNN | 16 Financial variables and 11 macroeconomic variables | – |

[a] FDT: fuzzy decision tree.
[b] SRA: step-wise regression analysis.
[c] BPNN: back propagation neural network.
[d] PCA: principle component analysis.
[e] CART: classification and regression tree.
[f] GA: genetic algorithm.
[g] GAFDT: genetic algorithm-based fuzzy decision tree.
[h] SVM: support vector machine.
[i] F_SSFS: $F$-score and supported sequential forward search.
[j] GRNN: generalized regression neural network.
[k] PNN: probabilistic neural network.
[l] LR: linear regression.

data to select key features independently; that is, they only consider the input variables. The wrapper approaches use the prediction performance of a specified learning algorithm to evaluate and determine the optimal feature subset [5]. The latter performs better but needs additional algorithms, such as the evolutionary algorithm and GA for optimization because of large search space, which results in a large computation cost. As the issue of stock prediction involves numerous features, applying filter approaches is more appropriate. Therefore, only this kind of approach is examined in this paper. In the following section, several common filter-based feature selection algorithms are briefly reviewed.

#### 2.2.1. PCA

PCA can reduce a large set of $n$ interrelated variables to a small set of $m$ highly uncorrelated components or factors ($m < n$) while retaining as much variance of the original dataset as possible. Specifically, PCA computes eigenvalues and eigenvectors of the components, each of which is a linear combination of the original variables. The first component accounts for the largest variance, whereas the second one accounts for the largest part of the remaining variance, and so on. The level of variance for each component ranges from 0 to 1. Given a specified value, e.g., 0.9, a subset of components with cumulative variance equals to or higher than 0.9 can be selected, which can explain 90% of the variance of original dataset [4].

Usually, the result given by PCA is a set of components but not a subset of original variables, and the subset variables would be more helpful for investors. Actually, when applying PCA, a subset of original variables can be selected by following the approach provided in [4], which uses a specified factor loading threshold, e.g. 0.5, to screen out informative variables.

#### 2.2.2. DT

A DT comprises one root (node) and a number of branches, nodes, and leaves. Each feature is involved in one node, and only features that contribute to the classification appear in the tree, and others do not (controlled by a preset threshold), which is useful in selecting good features. The DT is based on the entropy theory that selects variables with the highest information gain as the discriminating and explanatory features. First, the expected information (i.e., entropy) needed to classify a given sample is computed, and then this information is re-computed after the given sample is partitioned in accordance to a selected variable $X$. The difference between these two information is called information gain of variable $X$. The classification and regression tree (CART) [29], a popular DT algorithm, is used for feature selection in this study.

#### 2.2.3. LASSO

The least absolute shrinkage and selection operator algorithm (LASSO), which was proposed by Tibshirani in [30], is a prominent penalized method for selecting individual variables [31], which is based on an assumed model as follows:

$$E(y|X=x)=\alpha+\beta' x. \tag{1}$$

The feature selection problem is formulated as finding the elements of $\beta$ that equal zero. Estimates are selected by

$$\begin{aligned}&\text{arg min } \textstyle\sum(y_i-\alpha-\beta' x_i)^2\\&\text{subject to } \textstyle\sum|\beta_j|<t,\end{aligned} \tag{2}$$

where $t>0$ is a tuning parameter, which controls the amount of shrinkage applied to the estimates. Eq. (2) is equivalent to minimizing

$$\frac{1}{2n}\sum(y_i-\alpha-\beta' x_i)^2+\lambda\sum|\beta_j|, \tag{3}$$

which is the common least squares with a penalty term determined by $\lambda$ for large coefficient estimates. $\Sigma|\beta_j|$ is a coefficient vector constraint that delivers a sparse solution vector $\beta_\lambda$; given a larger $\lambda$, more elements of $\beta_\lambda$ are zero. If $\lambda=0$, the LASSO is similar to ordinary least squares. As $\lambda$ increases, shorter vectors are generated.

## 3. Methodology

### 3.1. Causality

Causality, or causal influence, is the relationship between two events. For instance, in a first event (the cause) and a second event (the effect), the second event is a consequence of the first one. Causalities must be correlations while correlations are not necessarily causalities. Causal inference requires not only correlation but also counterfactual dependence. One of the main approaches for causal inference is observational data-based inference, which can learn causality from an observational dataset and is particularly suitable for the stock prediction research that contains large and accurate information for data mining.

### 3.2. Causal inference

Consider a simple influence network shown in Fig. 1, for example. Only features $X_2$ and $X_3$ are assumed to affect output variable $Y$ directly. Thus, associations of $Y$ with $X_2$ and $X_3$ are causal influences while associations of $Y$ with features $X_1$ and $X_4$ are correlations. However, regression or other correlation analysis methods may select all of $X_1$, $X_2$, $X_3$, and $X_4$ as final features because of their strong correlation with $Y$ in the given observational dataset, i.e., these methods cannot distinguish between these two kinds of relationships. Although correlation analysis may result in high prediction performance, sometimes it is meaningful to find out which features can directly affect the output variable. This requirement entails observational data-based causal analysis of dataset.

The development of the causal inference theory is undoubtedly contributed by Pearl [7], whose causal inference paradigm has been applied in many studies. For instance, Mani and Cooper [32] proposed a Bayesian local causal discovery algorithm, which is applied to the Linked Birth/Infant Death dataset and found six causal influences, three of which appeared plausible. As learning complete causal models is essentially impossible, Silverstein et al. [33] suggested that isolated causal influences that only involve pairs or small sets of items are easier to interpret. Similarly, the local causal influence and/or causal structure discovery algorithm

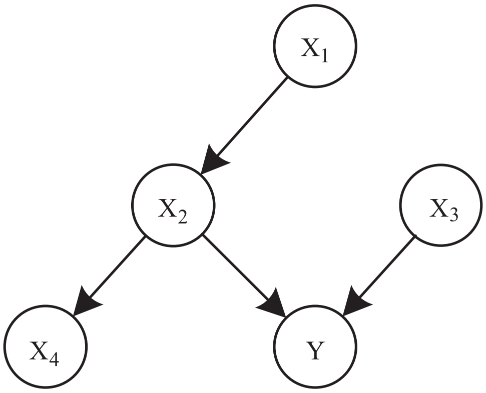


**Fig. 1.** A simple influence network.

can also be applied to stock market analysis for identifying causal influences between variables.

In the process of causal discovery, following four kinds of basic local structures are distinguished: (a) $X \rightarrow Y \rightarrow Z$, (b) $X \leftarrow Y \leftarrow Z$, (c) $X \leftarrow Y \rightarrow Z$, and (d) $X \rightarrow Y \leftarrow Z$. Three of these basic structures, namely, (a)–(c), are *independence equivalent* to each other because they reveal the same conditional independence assertions related to the variables $X$, $Y$, and $Z$ [34]. For example, variable $X$ is conditionally independent of variable $Z$ given variable $Y$ (assuming that variable $X$ is not a neighbor of variable $Y$ and no dependence path between variables $X$ and $Z$ through other variables exist). On the contrary, structure (d) is not *independence equivalent* to any one of the other three. Specifically, scholars refer to $Y$ in (d) as a *collider* and refer to (d) as a *V-structure*, which implies a different assertion that variable $X$ is independent of variable $Z$ if not given variable $Y$, but conditionally dependent if given variable $Y$. The most important point is that only the *V-structure* can be discriminated from others based on observational data without any expert/subjective judgment. This is the theoretical foundation of causal discovery.

### 3.3. CFS

Based on Pearl's theory [7] and Hu et al.'s study [8], the CFS algorithm, which is shown in Fig. 2, is proposed. Specifically, CFS generates a feature subset $Vs$ from a given full feature set $V$, through the following three steps:

First, the sample with missing values are deleted, while all continuous features are discretized.

Second, the parents and children set of each feature $v_i$ in $V$, denoted by $PC(v_i)$, is identified by using mutual information criteria and $g$-square (also $g^2$) conditional independence tests. Mutual information measures the dependence between two random variables or in other words measures how much knowing one of them reduces uncertainty about the other. The mutual information of two discrete random variables $X$ and $Y$ is defined as

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left( \frac{p(x,y)}{p(x)p(y)} \right), \tag{4}$$

where $p(x,y)$ is the joint probability distribution function of $X$ and $Y$, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of $X$ and $Y$ respectively. Moreover, $I(X;Y)=0$ if and only if $X$ and $Y$ are independent, and mutual information is nonnegative (i.e. $I(X;Y) \geq 0$) and symmetric (i.e. $I(X;Y) = I(Y;X)$).

To test the independence of two categorical variables $X$ and $Y$, the $G$-square test, or $G$-test, is increasingly being used in situations where the chi-squared test is previously recommended since the 1981 edition of the popular statistics textbook by Sokol and Rohlf [35]. The $G$-square test is the likelihood-ratio or the maximum likelihood statistical significance test based on the statistic

$$G = 2\sum_i O_i \ln(O_i/E_i), \tag{5}$$

where $O_i$ is the observed frequency in a cell of the contingency table and $E_i$ is the expected frequency on the null hypothesis. If the null hypothesis is rejected, variable $X$ and variable $Y$ are not independent; otherwise, $X$ and $Y$ are independent, denoted by $X \perp Y$.

```
Function CFS($V$)
//Input: $V$: the full feature set
//Output: $Vs$: the feature subset of $V$

//Initialization:
$Vs = \emptyset$
Delete sample with missing values;
Discretize all continuous features;
//Parents and children discovery
for each $v_i \in V$ do
    Parents and children set $C(v_i) = \emptyset$ , Candidate set $C = V - \{v_i\}$;
    Rank $C$ according to mutual information with $v_i$;
    for each $c_j \in C$ do
        if $\nexists S \subset PC(v_i), c_j \perp v_i | S$ then $PC(v_i) = PC(v_i) \cup \{c_j\}$; //identify a neighbor
        $PC(v_i) = PC(v_i) - \{f | f \perp v_i\}$                    //pruning
//V-structure discovery
for each $v_i \in V$ do
    for each pair of $pc_m, pc_n \in PC(v_i), m \neq n$ do
        if $pc_m \in PC(pc_n)$ or $pc_n \in PC(pc_m)$ then continue;   //skip neighbors
        $T$ = the smaller one of $PC(pc_m) - \{v_i\}$ and $PC(pc_n) - \{v_i\}$;
        if $\nexists S \subset T, pc_m \perp pc_n | v_i, S$ then            //identify a collider
            $Vs = Vs \cup \{pc_m, pc_n, v_i\}$;                  //extract features
```

**Fig. 2.** The causal feature selection algorithm (CFS).

In the probability theory, variable $A$ and variable $B$ are conditionally independent given $C$ if and only if $P(A \cap B|C) = P(A|C)P(B|C)$, denoted by $A \perp B|C$.

Third, *V-structures* that take each feature $v_i$ in $V$ as a *collider* are identified. Each pair of neighbors (i.e., parent and/or child) of $v_i$, denoted by $pc_m$ and $pc_n$, is enumerated, together with $v_i$, which constitutes a candidate *V-structure*, $pc_m \rightarrow v_i \leftarrow pc_n$. If no subset $S$ of $PC(v_i)$, $pc_m, pc_n \notin S$ , exists, that $pc_m$ and $pc_n$ are independent given both $v_i$ and $S$, we can confirm this *V-structure*, and all $v_i$, $pc_m$, and $pc_n$ are added to $Vs$.

## 4. Experiments

### 4.1. Dataset

The data of this study are calculated from the annual financial reports of A-Shares (i.e., shares sold to domestic investors in China) of the Shanghai Stock Exchanges from 1999 to 2011. Therefore, the backtesting period is from May 1, 2000 to April 30, 2012 because annual data are only available after the end of the year (with a few months' delay). In total, there are 8347 data samples (i.e., case companies) composed of 4116 and 4231 samples for positive and negative excess return (see Section 4.1.2), respectively.

#### 4.1.1. Input features

As mentioned previously, no generally agreed-upon representative features or "best" feature selection algorithm for stock prediction currently exists. This situation motivated us to collect as many relevant stock prediction features as possible from previous studies such as [4,21,22,36]. Considering the limited available data, 50 features were selected as listed in Table 2. Definitions of these features are given in Appendix A. In literature, these features were used for comprehensive stock evaluation from various perspectives, including valuation, profitability, growth, leverage, liquidity, operation, momentum, size, cash flow, bonus, volatility, and volume.

To reduce the influence of missing data on the performance comparison between different baseline models, we eliminate cases with missing data, i.e., only cases with all the required features are included. Moreover, continuous features were separately discretized into 0/1 binary ones in a simple way, namely, the 80/20 rule. First, each feature was ranked in descending order. Second, cases were classified into 1 and 0 by choosing the top 20% into 1 and the remaining 80% into 0 categories.

#### 4.1.2. Output variable

As an output variable, stock trend is defined differently in literature (including but not limited to) (1) as a 0/1 binary variable that indicates whether the ending stock price is higher than or equal to the previous price [4], (2) as a $-1/0/1$ tri-value variable by setting a threshold $r$ compared with when the stock price is regarded as uptrend (or downtrend) if the rate of return is higher (or lower) than $r$ (or $-r$) otherwise as a steady state [26,27], and (3) as a 0/1 binary variable that indicates whether the stock price beats the market (benchmark) or not [3,6]. As the third one can, to some extent, reduce the influence of different macromarket conditions (e.g., economic environment), this definition was adopted in this paper. Therefore, the output variable contains two class labels of "1" and "0," where "1" means that the excess return is positive, i.e., the stock return outperforms that of the buy-and-hold strategy on the benchmark (Shanghai Composite Index in this paper), and "0" represents that the excess return is negative, i.e., the stock return is lower than that of the benchmark. Using excess return can reduce the market effects of different years on individual stocks, that is, a stock that outperforms the market can be regarded as a good stock.

The Chinese stock market stipulates that the deadline of the annual financial report publication by the listed companies is April 30 of every year. Therefore, our study computes the annual stock return over the period from May 1 of year $t$ to April 30th of year $t+1$.

**Table 2**
The fundamental and momentum variables.

| Category | (#) | Features | Category | (#) | Features |
|---|---|---|---|---|---|
| Valuation factors | (1) | *E/P* | | (26) | Inventory turnover |
| | (2) | *B/P* | | (27) | Total assets turnover |
| | (3) | *S/P* | | (28) | Current assets turnover |
| Profitability factors | (4) | ROE | | (29) | Long-term liabilities to operating capital |
| | (5) | ROA | Momentum factors | (30) | Buy–hold return 1-month % |
| | (6) | Gross profit margin | | (31) | Buy–hold return 3-month % |
| | (7) | Net profit margin on sales | | (32) | Buy–hold return 6-month % |
| | (8) | Operating expense ratio | | (33) | Buy–hold return 12-month % |
| | (9) | Financial expense ratio | Size factors | (34) | Circulation market value |
| | (10) | Earnings before interest and tax to operating income | | (35) | Total market value |
| Growth factors | (11) | Operating profit % | | (36) | Circulation market value to total market value |
| | (12) | Gross profit % | | (37) | ln(total market value) |
| | (13) | Net income % | | (38) | ln(circulation market value) |
| | (14) | Net asset value per share % | | (39) | ln(total assets) |
| | (15) | Total assets % | Cash flow factors | (40) | Operating net cash flow |
| | (16) | Leverage % | Bonus factors | (41) | Dividend payout ratio |
| Leverage factors | (17) | Equity to debt ratio | Volatility factors | (42) | Amplitude 6-month % |
| | (18) | Asset liability ratio | | (43) | Amplitude 12-month % |
| | (19) | Long-term liabilities ratio | | (44) | SD of daily return rate 3-month |
| | (20) | Fixed assets ratio | | (45) | SD of daily return rate 6-month |
| | (21) | Current liabilities rate | | (46) | SD of daily return rate 12-month |
| Liquidity factors | (22) | Current ratio | Volume factors | (47) | Turnover rate 1-month % |
| | (23) | Quick ratio | | (48) | Turnover rate 3-month % |
| Operation factors | (24) | Cash to assets | | (49) | Turnover to total market turnover 1-month % |
| | (25) | Fixed assets turnover | | (50) | Turnover to total market turnover 3-month % |

### 4.2. Parameter setting

#### 4.2.1. PCA

When applying PCA, the factor loading threshold is set to 0.5, i.e., only variables with factor loading equal to or greater than 0.5 are retained [4]. To ensure factor interpretability, the varimax factor rotation method was adopted to reduce variables that have high loading on a factor. (After variables with factor loading lower than 0.5 are deleted, the processed dataset can still explain more than 80% of the total variance of the original dataset in the subsequent experiments.)

#### 4.2.2. CART

When applying the CART algorithm, the minimum support and the score method are considered to create the tree branches and then prune the initial DT, which results in a set of explanatory variables and split points with the highest impurity reduction. The minimum support was set to be 10% of the training set size. (The corresponding sample size was at least 187 over different training sets in the subsequent experiments, compared with 100 in [4].)

#### 4.2.3. LASSO

The parameter $\lambda$ in formula (3) is used to control the penalty strength. If $\lambda$ is set too large, the final solution vector $\beta_\lambda$ tends to become zero, which results in underfitting; however, if $\lambda$ is too small, the model will be overfitting. Generally, 10-fold cross-validation is used to determine a proper $\lambda$, which minimizes the mean squared error (MSE) of LASSO estimates [30].

#### 4.2.4. CFS

The most important parameter of CFS is the significance level, which affects the conditional independence test in the underlying *collider/V-structure* discovery procedure. Throughout the experiments, a significance level of 0.01 (1%) was used [8].

#### 4.2.5. Baseline prediction models

The present study employed the Waikato Environment for Knowledge Analysis (Weka), which is freely available on www.cs.waikato.ac.nz/ml/weka/. Weka provides a comprehensive suite of Java libraries that implement many state-of-the-art data mining algorithms. Baseline prediction models chosen for subsequent comparative experiments include LR, NB, BN, NN, SVM, DT, and Random Forest (RF). Correspondingly, Weka classifiers *Logistic*, *NaiveBayes*, *BayesNet*, *MultilayerPerceptron*, *LibSVM* (with linear kernel), *J48* (C4.5), and *RandomForest* were adopted with their default parameters.

### 4.3. Evaluation strategy

#### 4.3.1. Sliding window test

This study employs the sliding window method to divide the sample into different groups of training and testing sets. The sliding window method has been widely used in stock market prediction [4]. It is intuitive that investors always make predictions based on the recent data. However, there is no consistent view on the optimal proportion of the training set and testing set size. Therefore, several proportion settings including 1:1, 2:1, 3:1, 4:1, 5:1, 2:2, 3:2, 4:2, etc., were evaluated, and the optimal one turned out to be 4:1. For example, the training set of the first group is over the period from 2000 to 2003 while the testing set is based on the succeeding year 2004. Similarly, the last group contains a training set of data from 2005 to 2009 and a testing set based on 2012. The specific partitioning strategy is shown in Fig. 3.

The training and testing procedure of each group is as follows:

(1) feature selection is performed on the original training set to obtain the optimal feature subset $S$;
(2) the actual training set is extracted from the original one according to $S$, and then a prediction model is trained;
(3) the testing set is preprocessed in accordance with $S$ before advancing to the prediction model, and then the prediction performance is measured by comparing the expected outputs with the actual ones.

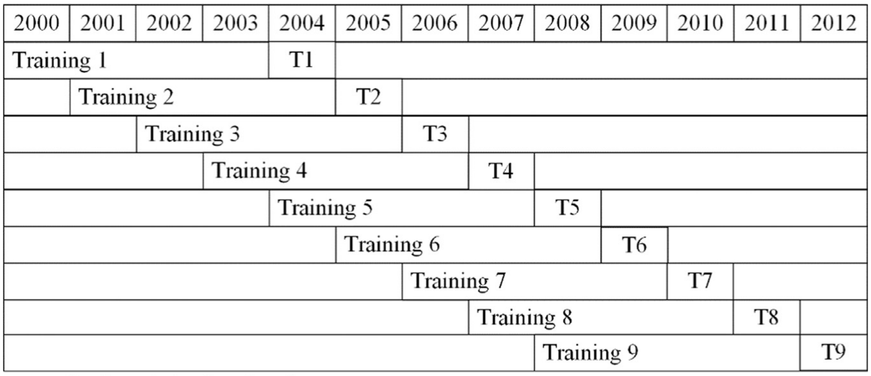
#### 4.3.2. Prediction performance evaluation

From the perspective of prediction capability, this paper compares the classification performances of the stock prediction models when combined with different feature selection algorithms. Specifically, accuracy and precision measures are examined, which can be calculated from a confusion matrix, as shown in Table 3.

**Table 3**
Confusion matrix.

| Actual/predicted | Positive | Negative |
|---|---|---|
| Positive | *a* | *b* |
| Negative | *c* | *d* |

Precision $= (a/(a+c))$ .
Accuracy $= ((a+d)/(a+b+c+d))$.



| 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Training 1 | | | | T1 | | | | | | | | |
| | Training 2 | | | | T2 | | | | | | | |
| | | Training 3 | | | | T3 | | | | | | |
| | | | Training 4 | | | | T4 | | | | | |
| | | | | Training 5 | | | | T5 | | | | |
| | | | | | Training 6 | | | | T6 | | | |
| | | | | | | Training 7 | | | | T7 | | |
| | | | | | | | Training 8 | | | | T8 | |
| | | | | | | | | Training 9 | | | | T9 |

Note: "Tn" stands for "Testing n" (n=1...9)

**Fig. 3.** Sliding window by one year based testing data.

This study considers that prediction precision is more important for common Chinese individual investors because of the short-selling restriction (on most A-shares stocks). Under this restriction, investors are preferred to buy/hold up-trend stocks, which is defined as positive sample (i.e. “good” stocks) in this paper. Therefore, precision is an intuitional measure for this task, which indicates the success rate of identifying “good” stocks that outperform the benchmark. Consider an extreme case for example, where $a=b=0$, $c=50$, and $d=150$, which results in a precision of 0% and an accuracy of 75%. Considering no short-selling, investors should only buy stocks that predicted as positive, i.e., totally $a+c=50$ stocks. However, all these stocks are of negative returns ($a=0$), which obviously indicates a bad investment.

### 4.3.3. Investment performance evaluation

From the perspective of stock investment, this paper compares the profitability of the stock prediction models when combined with different feature selection algorithms. Specifically, profitability is measured based on a simple trading strategy.

In each testing stage $t$ $(t=1,2,\ldots,T)$ of the sliding window test, the stocks predicted to have positive excess return are bought. Denote these stocks as $s_1,s_2,\ldots,s_{N_t}$, where $N_t$ is the number of bought stocks, and the actual returns or excess returns (i.e., returns that surpass the benchmark returns) of these stocks as $r_1,r_2,\ldots,r_{N_t}$. Then, the average of these return, $R_t=(\Sigma_{i=1}^{N_t} r_i)/N_t$, is used as the trading reward (i.e., investment profit) of testing stage $t$. Finally, the accumulated return over all testing stages can be calculated by $AR=\Pi_{t=1}^{T}(1+R_t)$, i.e., the reward of testing stage $t$ is also invested in the testing stage $t+1$.

Moreover, several risk-return measures are adopted in this study, including Sharpe ratio, Sortino ratio, information ratio, and MDD. The Sharpe ratio is proposed in [37], which measures the average excess return on a risk-free return per unit of standard deviation in an investment asset or a trading strategy. A higher Sharpe ratio indicates higher return and lower volatility. Similarly, information ratio is defined as the average excess return on a benchmark per unit of standard deviation [38]. However, the Sharpe and information ratios implicitly assume that investors are indifferent to upside and downside risk, which is sometimes impractical. As an extension of the Sharpe ratio, the Sortino ratio is proposed in [39], which argues that any returns below the minimal acceptable return will produce unfavorable outcomes and any returns greater will produce good ones. Considering that risk is associated only with bad outcomes, the Sortino ratio estimates the average excess return relative to the downside deviation (by using the lower partial moment approach). Finally, the MDD measures the maximum drop from a market peak to bottom during a specified period, which measures how sustained one's losses can be.

**Table 4**
Prediction precision of different feature selection algorithms.

| | NoFS (%) | CFS (%) | PCA (%) | CART (%) | Lasso (%) |
|---|---|---|---|---|---|
| LR | 52.82 | **54.40** (+1.58) | *49.90* (−2.92) | *52.63* (−0.19) | *52.22* (−0.60) |
| NB | 53.82 | **54.00** (+0.18) | *52.38* (−1.44) | *53.36* (−0.46) | *53.37* (−0.46) |
| BN | 53.40 | **53.98** (+0.58) | *52.38* (−1.02) | *53.37* (−0.03) | *53.35* (−0.05) |
| NN | 53.33 | **54.61** (+1.28) | *52.08* (−1.25) | *51.04* (−2.29) | *53.06* (−0.27) |
| SVM | 52.74 | **53.24** (+0.50) | *51.61* (−1.13) | *51.85* (−0.89) | *52.46* (−0.27) |
| J48 | 52.69 | **53.23** (+0.54) | *49.03* (−3.66) | *51.20* (−1.49) | *51.12* (−1.57) |
| RF | 51.21 | **53.26** (+2.05) | 52.04 (+0.83) | 51.25 (+0.04) | *52.08* (+0.87) |

*Note*: “NoFS” indicates no feature selection. Best results are highlighted in bold. Results that underperform the baseline models are italicized. The value in parentheses indicates the performance difference with the corresponding baseline model.

### 4.4. Result

#### 4.4.1. Prediction performance comparison

As the sliding window test is adopted in this study, the prediction model is trained and tested several times. As a result, several sets of prediction measures – one for each year – of the prediction model are observed. For convenience, only the averages of nine testing sets (refer to Fig. 3) were calculated for comparison over different feature selection algorithms.

Table 4 shows the prediction precisions of the four feature selection algorithms compared with the results when no feature selection algorithm is applied. When all input features are directly used to train the prediction model (i.e., feature selection is not performed, indicated by NoFS in Table 4), the prediction precision of NB model is 53.82%, which is the best among seven different baseline models, followed by NN (53.33%), LR (52.82%), and so on. When feature selection algorithms are applied, CFS outperforms other feature selection algorithms over all modeling algorithms (highlighted in bold in Table 4). Furthermore, CFS always outperforms NoFS, while the performances of PCA, CART, and LASSO are unsatisfactory (i.e., degrade precision in most cases). For instance, PCA underperforms NoFS by 1.51% on average and improves precision by merely 0.83% in the case of RF+PCA.

Table 5 shows the comparison result of prediction accuracy, from which an analysis similar to Table 4 can be conducted. Among the baseline models, BN model performs the best (53.40%), followed by NB (52.36%), NN (51.82%), and so on. Regarding feature selection algorithms, CFS outperforms PCA, CART, and LASSO over different modeling algorithms (highlighted in bold in Table 5) except two combinations, NN+LASSO and SVM+LASSO. Moreover, CFS always outperforms NoFS, while the other three feature selection algorithms underperform NoFS in some cases. For example, NB+PCA (50.47%) is nearly 2% lower than NB+NoFS (52.36%).

In conclusion, the above results imply that CFS is more suitable and stable compared with the other algorithms in the experiment. Moreover, as mentioned, precision is more essential than accuracy. From this point of view, CFS outperforms the other algorithms.

**Table 5**
Prediction accuracy of different feature selection algorithms.

| | NoFS (%) | CFS (%) | PCA (%) | CART (%) | Lasso (%) |
|---|---|---|---|---|---|
| LR | 51.58 | **51.81** (+0.23) | *50.40* (−1.18) | *51.22* (−0.36) | *51.30* (−0.28) |
| NB | 52.36 | **52.61** (+0.25) | *50.47* (−1.89) | *51.70* (−0.66) | *51.67* (−0.69) |
| BN | 51.69 | **52.60**(+0.91) | *50.48* (−1.21) | *51.71* (−0.02) | *51.36* (−0.33) |
| NN | 51.82 | 54.57 (+2.75) | 52.89 (+1.07) | 51.91 (+0.09) | **54.70** (+2.87) |
| SVM | 51.24 | 51.82 (+0.58) | *50.37* (−0.87) | *50.36* (−0.88) | **52.33** (+1.09) |
| J48 | 51.49 | **52.38** (+0.89) | *51.34* (−0.15) | *50.95* (−0.54) | *50.49* (−1.00) |
| RF | 50.07 | **51.51** (+1.44) | 51.29 (1.22) | 50.84 (+0.77) | 51.11 (+1.04) |

*Note:* “NoFS” indicates no feature selection. Best results are highlighted in bold. Results that underperform the baseline models are italicized. The value in parentheses indicates the performance difference with the corresponding baseline model.

**Table 6**
Number of selected features of different feature selection algorithms.

| Year | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | Avg. | Std. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CFS | 30 | 27 | 32 | 31 | 35 | 36 | 34 | 40 | 27 | 32 | 4.28 |
| PCA | 26 | 24 | 18 | 19 | 18 | 19 | 19 | 24 | 24 | 21 | 3.19 |
| CART | 25 | 16 | 7 | 29 | 26 | 31 | 19 | 27 | 26 | 23 | 7.57 |
| Lasso | 32 | 19 | 17 | 7 | 6 | 16 | 21 | 27 | 19 | 18 | 8.35 |

### 4.4.2. *Feature subset comparison*

Table 6 and Fig. 4 compare these feature selection algorithms in terms of the number of features selected in each period. CFS selects more features on average (32/50 or 64%), whereas it performs best in stock prediction (as shown in the previous section). However, although PCA, CART, and LASSO select fewer features on average (about 20/50 or 40%), the models based on these feature selection algorithms perform worse than the baseline models in some cases (i.e., LR+NoFS, NB+NoFS, BN+NoFS, and J48+NoFS). Therefore, one could make a tradeoff between model complexity and (back-testing) prediction performance when considering either CFS or the other feature selection algorithms. Another interesting result is that PCA has the least variance (Std.=3.19) among the four algorithms, followed by CFS (Std.=4.28), while CART and LASSO have much larger variance.

Because of space limitations, this paper does not present the complete lists of the selected features by each algorithm over each period (totally 4 × 9 or 36 feature subset tables). Upon further analysis of the complete lists, it is found that the number of features that are selected at least eight times (in nine training sets) by CFS is 23, compared with 14, 3, and 1 selected by PCA, CART, and LASSO, respectively. This analysis implies that CART and LASSO are very sensitive to noise in the experiment dataset, which makes them unsuitable for finding consistent and convincing features for further analysis. Table 7 lists 18 features selected by CFS, which are reserved over the entire testing period. (That is, these 18 features were selected in every training set and appeared in all 9 feature subset tables generated by CFS.) This finding indicates that most valuation, probability, leverage, liquidity, operation, and cash flow factors have a consistent effect on the Chinese stock market. As for momentum factors, "Buy–hold return 1-month %" has better explanatory power than the other three, which is quite reasonable because this factor can capture the most-recent influence (before and after the publication) of the annual financial report on stock price before our model is employed to select stocks (the first trading day of May).

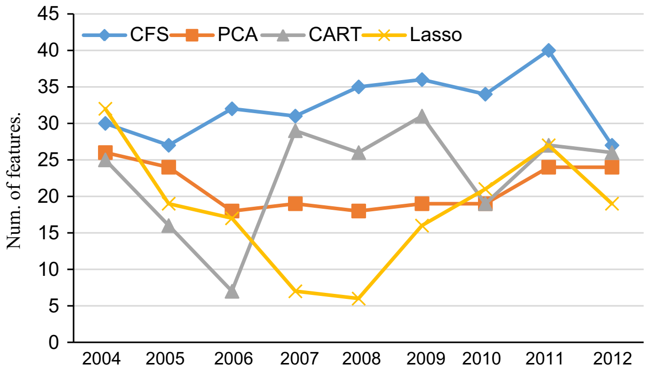


**Fig. 4.** Number of selected features of different feature selection algorithms

### 4.4.3. *Investment profit comparison*

Tables 8 and 9 show the profitability of different models in terms of absolute return and excess return, respectively. As a benchmark, the buy-and-hold return of the Shanghai Composite Index during the entire testing period is 136.49%. This result is somewhat inconsistent with Tables 4 and 5, as better prediction performance cannot ensure higher investment return. Evidently, "good" stocks have (excess) returns that differ from each other, while our precision and accuracy-based evaluation criteria cannot capture this characteristic. Another important factor that affects investment return is capital allocation among selected stocks, which is beyond the scope of this study.

Tables 8 and 9 show that the baseline models LR, NB, and NN perform well, as none of the four feature selection algorithms can result in higher return. However, CFS performs best when

**Table 8**
Absolute return of different feature selection algorithms

| | NoFS (%) | CFS (%) | PCA (%) | CART (%) | Lasso (%) |
|---|---|---|---|---|---|
| LR | **295.81** | 283.08 | 209.19 | 270.83 | 259.61 |
| NB | **287.27** | 265.87 | 229.48 | 276.10 | 284.76 |
| BN | 281.51 | 265.64 | 229.73 | 276.58 | **285.23** |
| NN | **267.36** | 264.81 | 223.21 | 203.43 | 251.80 |
| SVM | 249.04 | **252.98** | 213.18 | 225.58 | 238.44 |
| J48 | 228.76 | **230.66** | 192.93 | 219.41 | 219.80 |
| RF | 230.02 | **257.29** | 222.42 | 218.71 | 233.41 |

*Note*: best results are highlighted in bold. "NoFS" indicates no feature selection.

**Table 9**
Excess return of different feature selection algorithms.

| | NoFS (%) | CFS (%) | PCA (%) | CART (%) | Lasso (%) |
|---|---|---|---|---|---|
| LR | **262.36** | 249.29 | 111.98 | 225.36 | 205.55 |
| NB | **246.24** | 220.61 | 170.10 | 230.74 | 242.29 |
| BN | 239.47 | 220.35 | 170.26 | 231.01 | **243.07** |
| NN | **274.45** | 252.93 | 147.66 | 188.67 | 195.59 |
| SVM | 252.62 | **260.43** | 188.22 | 204.70 | 216.65 |
| J48 | **209.50** | 202.70 | 142.48 | 196.30 | 188.53 |
| RF | 213.83 | **261.12** | 202.25 | 195.79 | 219.55 |

*Note*: best results are highlighted in bold. "NoFS" indicates no feature selection.

**Table 7**
The features selected and reserved by CFS over all testing sets.

| Category | (#) Features | Category | (#) Features |
|---|---|---|---|
| Valuation (2/3) | (2) *B*/*P*<br>(3) *S*/*P* | Liquidity (2/2) | (22) Current ratio<br>(23) Quick ratio |
| Profitability (4/7) | (4) ROE<br>(5) ROA<br>(7) Net profit margin on sales<br>(10) Earnings before interest and tax to operating income | Operation (4/6) | (26) Inventory turnover<br>(27) Total assets turnover<br>(28) Current assets turnover<br>(29) Long-term liabilities to operating capital |
| Leverage (4/5) | (18) Asset liability ratio<br>(19) Long-term liabilities ratio<br>(20) Fixed assets ratio<br>(21) Current liabilities rate | Momentum (1/4)<br><br><br>Cash flow (1/1) | (30) Buy–hold return 1-month %<br><br><br>(40) Operating net cash flow |

*Note*: (*m*/*n*) represents "*m* features are selected out of total *n* features in the corresponding category".

**Table 10**
NN model's absolute return based on different feature selection algorithms.

| | Ret. (%) | ARet. (%) | Sharpe | Sortino | Info | MDD (%) |
|---|---|---|---|---|---|---|
| NoFS | 267.36 | 15.56 | 0.35 | 1.82 | 0.71 | 31.77 |
| CFS | 264.81 | 15.47 | 0.35 | 1.86 | 0.78 | 27.20 |
| PCA | 223.21 | 13.92 | 0.35 | 1.60 | 1.16 | 27.17 |
| CART | 203.43 | 13.13 | 0.32 | 1.51 | 0.72 | 32.79 |
| Lasso | 202.85 | 13.10 | 0.33 | 1.43 | 0.83 | 32.87 |

*Note*: “Ret.” represents the return ratio, “ARet.” represents the annualized return, “Sharpe” represents the Sharpe ratio, “Sortino” represents the Sortino ratio, “MDD” represents the maximum drawdown ratio, and “Info” represents information ratio. “NoFS” indicates no feature selection.

**Table 11**
NN model's excess return based on different feature selection algorithms.

| | Ret. (%) | ARet. (%) | Sharpe | Sortino | MDD (%) |
|---|---|---|---|---|---|
| NoFS | 274.45 | 15.80 | 0.59 | 4.54 | 4.42 |
| CFS | 252.92 | 15.04 | 0.63 | 5.81 | 3.41 |
| PCA | 147.66 | 10.60 | 0.84 | 3.69 | 3.26 |
| CART | 188.67 | 12.50 | 0.57 | 3.01 | 5.43 |
| Lasso | 165.81 | 11.47 | 0.63 | 2.61 | 5.52 |

*Note*: “Ret.” represents the return ratio, “ARet.” represents the annualized return, “Sharpe” represents the Sharpe ratio, “Sortino” represents the Sortino ratio, and “MDD” represents the maximum drawdown ratio. “NoFS” indicates no feature selection.

combined with SVM, J48, or RF in the case of absolute return and with SVM or RF in the case of excess return.

For further analysis, the return details of NN model over different feature selection algorithms are specifically shown in Tables 10 and 11 because of its high excess returns. Although NN+CFS produces lower return than NN+NoFS, the former has a higher Sortino ratio (1.86 vs. 1.82 in the case of absolute return, and 5.81 vs. 4.54 in the case of excess return), which implies that NN+CFS has more stable and consistent positive return. Moreover, NN+CFS achieved the highest Sortino ratio and lowest max–drawdown ratio among five models except the NN+PCA; however, NN+PCA has a poor return.

## 5. Conclusion

In the development of QI products, it is important to select informative features for stock prediction modeling. The feature selection algorithm can contribute to identifying the most representative features and improving the prediction performance. Conventional feature selection algorithms, such as PCA, CART, and LASSO, are mainly based on the characteristics of data correlation. Although these algorithms can, to some extent, improve prediction performance, they are not designed to discover the underlying causalities between variables from a given dataset. Therefore, based on Pearl's theory [7] and Hu et al.'s study [8], this paper proposes CFS to improve the performance of the stock prediction model by identifying more representative features. Features selected by CFS are reasonable and deserve further studies for investment decisions.

The effectiveness of CFS is demonstrated in comparative experiments with three popular feature selection algorithms, namely, PCA, CART, and LASSO. When combined with each of the seven baseline models (i.e., LR, NB, BN, NN, SVM, J48, and RF), CFS can always improve both prediction accuracy and precision. It also outperforms PCA, CART, and LASSO in most cases. In addition, although CFS generates fewer compact feature subsets, CFS obtained stable prediction accuracy and precision over different baseline models. This robustness of CFS may be due to the fact that CFS is less likely to miss any important features. In terms of investment profitability, CFS models also perform well. (Note that, as mentioned, good prediction precision/accuracy does not guarantee a high investment return.) Moreover, CFS identifies and reserves 18 representative features over the entire back-testing period. These features can be used not only for practical stock investment decisions but also for future study as the “standard” features to develop new prediction models for comparison.

Limitations of this study mainly lie on the uncomprehensive comparative experiments in following two aspects: (1) not considering other popular feature selection methods, such as SRA [26], information gain [6] and GA [4], and (2) not considering data from other popular stock markets, such as US [22] and Taiwan [21]. However, from the practical standpoint, it is difficult to conduct such a comprehensive study. Future research directions of CFS include but are not limited to (1) identifying smaller feature subset without sacrificing prediction performance, and (2) reducing computational complexity.

## Acknowledgments

This research was partly supported by the National Natural Science Foundation of China (71271061 and 70801020), the Science and Technology Planning Project of Guangdong Province, China (2010B010600034 and 2012B091100192), and the Ministry of Education Innovation Team Development Plan, Guangdong Natural Science Foundation Research Team, and Business Intelligence Key Team of Guangdong University of Foreign Studies (S2013030015737, IRT1224, and TD1202).

## Appendix A. Feature definitions

| (#) | Feature | Definition |
|---|---|---|
| (1) | *E*/*P* | Earnings-to-price ratio=earnings per share/price per share |
| (2) | *B*/*P* | Book-to-price ratio=book value/price per share |
| (3) | *S*/*P* | Sales-to-price ratio=sales/price per share |
| (4) | ROE | Return on equity=net profit after tax before extraordinary items/shareholders' equity |
| (5) | ROA | Return on assets=net profit after tax before extraordinary items/total assets |
| (6) | Gross profit margin | (Net sales – cost of goods sold)/net sales |
| (7) | Net profit margin on sales | Net income after tax/net sales |
| (8) | Operating expense ratio | Operating expense/operating income |
| (9) | Financial expense ratio | Financial expense/operating income |
| (10) | Earnings before interest and tax to operating income | Earnings before interest and tax/operating income |

| | | |
|---|---|---|
| (11) | Operating profit % | Operating profit growth rate=(operating profit at the current year − operating profit at the previous year)/operating profit at the previous year |
| (12) | Gross profit % | Gross profit growth rate=(gross profit at the current year − gross profit at the previous year)/gross profit at the previous year |
| (13) | Net income % | Net income growth rate=(net income after tax at the current year − net income after tax at the previous year)/net income after tax at the previous year |
| (14) | Net asset value per share % | Net asset value per share growth rate=(net asset value per share at the current year − net asset value per share at the previous year)/net asset value per share at the previous year |
| (15) | Total assets % | Total assets growth rate=(total assets at the current year − total assets at the previous year)/total assets at the previous year |
| (16) | Leverage % | Debt to equity growth rate=(debt to equity at the current year − debt to equity at the previous year)/debt to equity at the previous year |
| (17) | Equity to debt ratio | Shareholders' equity/total liabilities |
| (18) | Asset liability ratio | Total assets/total liabilities |
| (19) | Long-term liabilities ratio | Long-term liabilities/total liabilities |
| (20) | Fixed assets ratio | Net fixed assets/total assets |
| (21) | Current liabilities rate | Current liabilities/total liabilities |
| (22) | Current ratio | Current assets/current liabilities |
| (23) | Quick ratio | (Current assets − inventory)/current liabilities |
| (24) | Cash to assets | Operating net cash flow/total assets |
| (25) | Fixed assets turnover | Operating income/average fixed assets, where average fixed assets=(fixed assets closing balance + fixed assets opening balance)/2 |
| (26) | Inventory turnover | Operating cost/average inventory, where average inventory=(inventory closing balance + inventory opening balance)/2 |
| (27) | Total assets turnover | Operating income/total average assets, where total average assets=(total assets closing balance + total assets opening balance)/2 |
| (28) | Current assets turnover | Operating income/average current assets, where average current assets=(current assets closing balance + current assets opening balance)/2 |
| (29) | Long-term liabilities to operating capital | Long-term liabilities/operating capital |
| (30) | Buy–hold return 1-month % | (Closing price of the last trading day − closing price of the first trading day)/closing price of the first trading day × 100% (in the recent month) |
| (31) | Buy–hold return 3-month % | (Closing price of the last trading day − closing price of the first trading day)/closing price of the first trading day × 100% (in the recent 3 months) |
| (32) | Buy–hold return 6-month % | (Closing price of the last trading day − closing price of the first trading day)/closing price of the first trading day × 100% (in the recent 6 months) |
| (33) | Buy–hold return 12-month % | (Closing price of the last trading day − closing price of the first trading day)/closing price of the first trading day × 100% (in the recent 12 months) |
| (34) | Circulation market value | Price per share × number of shares outstanding |
| (35) | Total market value | Price per share × capital size |
| (36) | Circulation market value to total market value | Circulation market value/total market value |
| (37) | ln(total market value) | The natural logarithm of total market value |
| (38) | ln(circulation market value) | The natural logarithm of circulation market value |
| (39) | ln(total assets) | The natural logarithm of total assets |
| (40) | Operating net cash flow | Operating cash inflows − operating cash outflows |
| (41) | Dividend payout ratio | Dividend per share/earnings per share |
| (42) | Amplitude 6-month % | (The highest price of the current 6 months − the lowest price of the current 6 months)/the closing price of the previous 6 months |
| (43) | Amplitude 12-month % | (The highest price of the current 12 months − the lowest price of the current 12 months)/the closing price of the previous 12 months |
| (44) | SD of daily return rate 3-month | The deviation amplitude of the daily return rate relative to average daily return rate in 3 months |
| (45) | SD of daily return rate 6-month | The deviation amplitude of the daily return rate relative to average daily return rate in 6 months |
| (46) | SD of daily return rate 12-month | The deviation amplitude of the daily return rate relative to average daily return rate in 12 months |
| (47) | Turnover rate 1-month % | Trading volume/number of shares outstanding (in 1 month) |
| (48) | Turnover rate 3-month % | Trading volume/number of shares outstanding (in 3 months) |
| (49) | Turnover to total market turnover 1-month % | Turnover/total market turnover (in 1 month) |
| (50) | Turnover to total market turnover 3-month % | Turnover/total market turnover (in 3 months) |

## References

[1] N. Barberis, M. Huang, Mental accounting, loss aversion, and individual stock returns, J. Financ. 56 (2001) 1247–1292.

[2] T. Odean, Do investors trade too much? Am. Econ. Rev. 89 (1999) 1279–1298.

[3] S. Thawornwong, D. Enke, The adaptive selection of financial and economic variables for use with artificial neural networks, Neurocomputing 56 (2004) 205–232.

[4] C. Tsai, Y. Hsiao, Combining multiple feature selection methods for stock prediction: union, intersection, and multi-intersection approaches, Decis. Support Syst. 50 (2010) 258–269.

[5] Y. Chen, C. Cheng, Evaluating industry performance using extracted RGR rules based on feature selection and rough sets classifier, Expert Syst. Appl. 36 (2009) 9448–9456.

[6] D. Enke, S. Thawornwong, The use of data mining and neural networks for forecasting stock market returns, Expert Syst. Appl. 29 (2005) 927–940.

[7] J. Pearl, Causality: Models, Reasoning, and Inference, Cambridge University Press, Los Angeles, USA, 2000.

[8] Y. Hu, X. Zhang, E.W.T. Ngai, R. Cai, M. Liu, Software project risk analysis using Bayesian networks with causality constraints, Decis. Support Syst. 56 (2013) 439–449.

[9] E.F. Fama, Efficient capital markets: II, J. Financ. 46 (1991) 1575–1617.

[10] A.W. Lo, A.C. MacKinlay, Stock market prices do not follow random walks: evidence from a simple specification test, Rev. Financ. Stud. 1 (1988) 41–66.

[11] Q. Cao, M.E. Parry, K.B. Leggio, The three-factor model and artificial neural networks: predicting stock price movement in China, Ann. Oper. Res. 185 (2011) 25–44.

[12] A. Kanas, Non-linear forecasts of stock returns, J. Forecast. 22 (2003) 299–315.

[13] Q. Cao, K.B. Leggio, M.J. Schniederjans, A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market, Comput. Oper. Res. 32 (2005) 2499–2512.

[14] C. Huang, C. Tsai, A hybrid SOFM–SVR with a filter-based feature selection for stock market forecasting, Expert Syst. Appl. 36 (2009) 1529–1539.

[15] C. Huang, A hybrid stock selection model using genetic algorithms and support vector regression, Appl. Soft Comput. 12 (2012) 807–818.

[16] M.R. Hassan, K. Ramamohanarao, J. Kamruzzaman, M. Rahman, M. Maruf Hossain, A HMM-based adaptive fuzzy inference system for stock market forecasting, Neurocomputing 104 (2013) 10–25.

[17] Y. Kishikawa, S. Tokinaga, Prediction of stock trends by using the wavelet transform and the multi-stage fuzzy inference system optimized by the GA, IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E83A (2000) 357–366.

[18] C. Huang, D. Yang, Y. Chuang, Application of wrapper approach and composite classifier to the stock trend prediction, Expert Syst. Appl. 34 (2008) 2870–2878.

[19] L. Yu, H. Chen, S. Wang, K.K. Lai, Evolving least squares support vector machines for stock market trend mining, IEEE Trans. Evol. Comput. 13 (2009) 87–102.

[20] C. Cheng, L. Wei, Y. Chen, Fusion ANFIS models based on multi-stock volatility causality for TAIEX forecasting, Neurocomputing 72 (2009) 3462–3468.

[21] C. Tsai, Y. Lin, D.C. Yen, Y. Chen, Predicting stock returns by classifier ensembles, Appl. Soft Comput. 11 (2011) 2452–2459.

[22] M. Lam, Neural network techniques for financial performance prediction: integrating fundamental and technical analysis, Decis. Support Syst. 37 (2004) 567–581.

[23] M. Lee, Using support vector machine with a hybrid feature selection method to the stock trend prediction, Expert Syst. Appl. 36 (2009) 10896–10904.

[24] N. Ren, M. Zargham, S. Rahimi, A decision tree-based classification approach to rule extraction for security analysis, Int. J. Inf. Technol. Decis. Mak. 5 (2006) 227–240.

[25] Y. Zuo, E. Kita, Stock price forecast using Bayesian network, Expert Syst. Appl. 39 (2012) 6729–6737.

[26] P. Chang, C. Fan, J. Lin, Trend discovery in financial time series data using a case based fuzzy decision tree, Expert Syst. Appl. 38 (2011) 6070–6080.

[27] R.K. Lai, C. Fan, W. Huang, P. Chang, Evolving and clustering fuzzy decision tree for financial time series data forecasting, Expert Syst. Appl. 36 (2009) 3761–3773.

[28] M.A. Hall, G. Holmes, Benchmarking attribute selection techniques for discrete class data mining, IEEE Trans. Knowl. Data Eng. 15 (2003) 1437–1447.

[29] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, Chapman & Hall/CRC, Florida, 1984.

[30] R. Tibshirani, Regression shrinkage and selection via the LASSO, J. R. Stat. Soc., Ser. B 58 (1996) 267–288.

[31] Y. Ming, L. Yi, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. Ser. B: Stat. Methodol. 68 (2005) 49–67.

[32] S. Mani, G.F. Cooper, Causal discovery using a Bayesian local causal discovery algorithm, in: M. Fieschi, E. Coiera, Y.J. Li (Eds.), Medinfo, IOS Press, Amsterdam, Netherlands, 2004, pp. 731–735.

[33] C. Silverstein, S. Brin, R. Motwani, J. Ullman, Scalable techniques for mining causal structures, Data Min. Knowl. Discov. 4 (2000) 163–192.

[34] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Francisco, California, USA, 1991.

[35] R.R. Sokal, F.J. Rohlf, Biometry: The Principles and Practice of Statistics in Biological Research, WH Freemand and Company, New York, 1995.

[36] M.Y. Kiang, R.T. Chi, K.Y. Tam, DKAS: a distributed knowledge acquisition system in a DSS, J. Manag. Inf. Syst. 9 (1993) 59–82.

[37] W.F. Sharpe, The Sharpe ratio, J. Portf. Manag. 21 (1994) 49–58.

[38] T.H. Goodwin, The information ratio, Financ. Anal. J. 54 (1998) 34–43.

[39] F.A. Sortino, L.N. Price, Performance measurement in a downside risk framework, J. Invest. 3 (1994) 59–64.

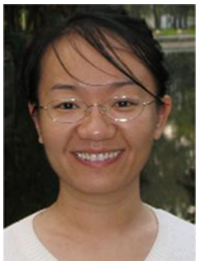
**Xiangzhou Zhang** is a Ph.D. student in the Sun Yat-sen University and working as an assistant researcher in the Institute of Business Intelligence and Knowledge Discovery at the Guangdong University of Foreign Studies and the Sun Yat-Sen University. He has received his B.Sc. in Computer Science from the Sun Yat-Sen University, and M.Phil. in the Guangdong University of Foreign Studies. His research interests are quantitative investment, design science, software project risk management and business intelligence. He has published papers in a number of international journals and conferences including Decision Support Systems, Journal of Software, FSKD'09, EIDWT2013 and WHICEB2013.

**Prof. Yong Hu** is a professor and director of the Institute of Business Intelligence and Knowledge Discovery at the Guangdong University of Foreign Studies (GDUFS) and the Sun Yat-Sen University (SYSU). He received his B.Sc. in Computer Science, M.Phil. and Ph. D. in Management Information Systems from SYSU. His research interests include business intelligence, quantitative investment, medical informatics, software project risk management, spam filtering and decision support systems. He has published more than 50 papers in JAMIA, DSS, JASIST, IJPR, ESWA, IST, IEEE ICDM, etc. His research is supported by the National Science Foundation of China, the Science and Technology Planning Project of Guangdong Province, and the Key Team of Business Intelligence from GDUFS.

**Prof. Kang Xie** is currently a professor of management science, the School of Business, Sun Yat-sen University. He is Standing Associate Director-General of China Information Economics Society, Standing Director of China Association for Information Systems, and Advanced Consultant of Ministry of Commence on Electronic Commence. He received his Ph.D. in management from the Renmin University of China. He has published more than 11 books and numerous papers in a number of international journals and conferences. His research interests are in the areas of management science, Information Economics, E-commerce Economics, and Enterprise informatization.

**Prof. Shouyang Wang** is a professor and vice president of the Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS). His research includes Investment Analysis and Risk Management, Integrated Logistics, Game Theory, and Multilevel Programming. He has published 18 books and over 120 journal papers in leading journals including Journal of Optimization Theory and Applications, Computers & Operations Research, European Journal of Operational Research, and Annals of Operations Research. He is the editor-in-chief or a co-editor of 12 journals, including Journal of Management Systems, Information Technology and Decision Making, and Journal of Systems Science and Complexity.

**Prof. E.W.T. Ngai** is a professor in the Department of Management and Marketing at The Hong Kong Polytechnic University. His research interests include E-commerce, Supply Chain Management, Decision Support Systems and RFID Technology and Applications. He has published papers in a number of international journals including MIS Quarterly, Journal of Operations Management, Decision Support Systems, and IEEE Transactions on Systems, Man and Cybernetics, Information & Management. He is an Associate Editor of European Journal of Information Systems and serves on editorial board of six international journals. He has attained an *h*-index of 13, and received 510 citations, ISI Web of Science.



**Prof. Mei Liu** is an Assistant Professor in the Department of Computer Science at New Jersey Institute of Technology. She received her Ph.D. degree in computer science from the University of Kansas, Lawrence, USA, and completed her postdoctoral training as an NIH-NLM research fellow in the Department of Biomedical Informatics at Vanderbilt University, Nashville, USA. Her research interest includes data mining, text mining, decision support systems, quantitative investment, and medical informatics. She has published a number of papers in Bioinformatics, JAMIA, DSS, ESWA, EURASIP Journal on Applied Signal Processing, BMC Bioinformatics, PLoS ONE, IEEE ICDM, etc.