Expert Systems With Applications 80 (2017) 340–355

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction

Yingjun Chen, Yongtao Hao*

Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China

ARTICLE INFO

Article history:
Received 10 December 2016
Revised 27 February 2017
Accepted 28 February 2017
Available online 1 March 2017

Keywords:
Feature weighted SVM (FWSVM)
Information gain
Feature weighted K-nearest neighbor (FWKNN)
Stock market indices

ABSTRACT

This study investigates stock market indices prediction that is an interesting and important research in the areas of investment and applications, as it can get more profits and returns at lower risk rate with effective exchange strategies. To realize accurate prediction, various methods have been tried, among which the machine learning methods have drawn attention and been developed. In this paper, we propose a basic hybridized framework of the feature weighted support vector machine as well as feature weighted *K*-nearest neighbor to effectively predict stock market indices. We first establish a detailed theory of feature weighted SVM for the data classification assigning different weights for different features with respect to the classification importance. Then, to get the weights, we estimate the importance of each feature by computing the information gain. Lastly, we use feature weighted K-nearest neighbor to predict future stock market indices by computing *k* weighted nearest neighbors from the historical dataset. Experiment results on two well known Chinese stock market indices like Shanghai and Shenzhen stock exchange indices are finally presented to test the performance of our established model. With our proposed model, it can achieve a better prediction capability to Shanghai Stock Exchange Composite Index and Shenzhen Stock Exchange Component Index in the short, medium and long term respectively. The proposed algorithm can also be adapted to other stock market indices prediction.

1. Introduction

Stock market indices prediction is one of the most important and extremely challenging financial time series forecasting problems for both investors and researchers. This is mainly caused by the fact that the stock market is essentially an unstable, nonlinear and complex system of dynamic change, and is affected by many factors, such as major economic policies, government decrees, the change of political situation, investor's psychology, the future economy, and so on. However, to realize accurate forecast of stock price in the short term (1 day, 5 days ahead), medium term (10 days, 15 days ahead) and long term (20 days, 30 days ahead) is one of the most attractive and meaningful research subjects in the investment and its application fields. The benefits involved in accurate prediction have motivated researchers to develop newer and more advanced tools and methods.

With regard to the techniques used to analyze the stock markets, some are based on statistical methods, others are artificial intelligence and machine learning methods. Generally the financial time series data, being chaotic, noisy and nonlinear in nature (Guegan, 2009), does not necessarily follow a fixed pattern. Thus the statistical approaches, such as moving average, weighted moving average (Ziegel, 2002), Kalman filtering, exponential smoothing, regression analysis, autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), and autoregressive moving average with exogenous (Box, Jenkins, Reinsel, & Ljung, 2015), do not perform very well in predicting stock market indices accurately. In contrast to the statistical techniques, artificial intelligence methods can handle the random, chaotic, and nonlinear data of the stock market and have been used widely for accurate prediction of stock market indices.

A lot of artificial intelligence methods have been developed and applied to forecast stock market indices, for instance, Artificial Neural Network (ANN) (Kodogiannis & Lolis, 2002; Thirunavukarasu, 2009; Xi et al., 2014), Support Vector Machines (SVMs) (Gong, Si, Fong, & Biuk-Aghai, 2016; Hu, Zhu, & Tse, 2013; Lin, Guo, & Hu, 2013; Wen, Yang, Song, & Jia, 2010; Yu, Chen, & Zhang, 2014; Zhang & Shen, 2009), Rough Set Theory (Nair, Mohandas, & Sakthivel, 2010; Wang & Wang, 2002; Wang, 2003), Bayesian Analysis (BA) (Ives & Scandol, 2007; Miao, Wang, & Xu, 2015; Su & Peterman, 2012; Ticknor, 2013; Wang, Wang, Zhao, & Tan, 2015; Xi, Peng, Qin, Xie, & Chen, 2015), K-Nearest Neighbors (KNN) (Li, Sun, & Sun, 2009; Teixeira & De Oliveira, 2010), Particle Swarm

* Corresponding author.
E-mail addresses: cxue2006@126.com (Y.J. Chen), haoyt@vip.sina.com (Y.T. Hao).

http://dx.doi.org/10.1016/j.eswa.2017.02.044
0957-4174/

Optimization (PSO) (Fu-Yuan, 2008; Shen, Zhang, & Ma, 2009), Decision Tree (DT) (Hu, Feng, Zhang, Ngai, & Liu, 2015; Sorensen, Miller, & Ooi, 2000; Wu, Lin, & Lin, 2006), and the evolutionary learning algorithms like Genetic Algorithm (GA) (Hassan, Nath, & Kirley, 2007; Huang & Wu, 2008; Rahman, Sarker, & Essam, 2015). Often, the developed intelligence methods and technical analysis are used together with trading rules to develop an intelligent, autonomous, and adaptive decision support system. For example, Kim and Enke (2016) develop a rule change trading system (RCTS) that consists of numerous trading rules generated using rough set analysis in order to cover diverse market conditions. Podsiadlo and Rybinski (2016) investigate experimentally the feasibility of rough sets in building profitable trend prediction models for financial time series. They propose a novel time-weighted rule voting method to improve the decision process for long time series. Cervelló-Royo, Guijarro, and Michniuk (2015) propose a risk-adjusted profitable trading rule based on technical analysis and the use of the breakout and consolidation flag pattern, which defines when to buy or sell, the profit pursued in each operation, and the maximum bearable loss. Chiang, Enke, Wu, and Wang (2016) propose an adaptive intelligent stock trading decision support system that utilizes PSO and ANN to predict a stock index's future movement direction, which overcomes a major weakness of a traditional ANN approach.

In recent years, some researchers tend to hybridize other artificial intelligence techniques with ANN due to the large dimension of neurons as well as the increasing computational complexity with ANN, which though have been shown by a large number of researches to be successfully used for forecasting stock market indices. In order to select the most important and influential variables for classification, Zhong and Enke (2017) apply principal component analysis (PCA), fuzzy robust principal component analysis (FRPCA), and kernel-based principal component analysis (KPCA) to simplify and rearrange the original data structure, and then use ANNs with the transformed data sets for classification to forecast the daily direction of future market returns. Göçken, Özçalıcı, Boru, and Dosdoğru (2016) propose a hybrid model, based on a heuristic optimization methodology (HS or GA) and ANN, to improve stock market forecasting performance in terms of statistical and financial terms, which have great capability in variable selection and determining the number of neurons in hidden layer. Majhi, Panda, and Sahoo (2009) propose functional link artificial neural network model (FLANN) for prediction of stock price of Dow Jones industrial average (DJIA) and the Standard's & Poor's 500 (S&P 500) indices. FLANN is comparable to other neural network models and requires less computation during training and testing. Chakravarty and Dash (2012) develop an integrated functional link interval type-2 fuzzy neural system (FLIT2FNS) for predicting the stock market indices like DJIA, S&P 500, and Bombay stock exchange (BSE), and the results indicate that FLIT2FNS model performs better than that of FLANN and type-1 fuzzy logic system (Type-1FLS) followed by Local Linear Wavelet Neural Network (LLWNN) model irrespective of the learning algorithms used or irrespective of the periodicity of the prediction. The most recent method used for prediction of stock indices is computationally efficient functional link artificial neural network (CEFLANN). Dash, Dash, and Bisoi (2014) propose a hybrid learning framework called Self Adaptive Differential Harmony Search Based Optimized Extreme Learning Machine (SADHS-OELM) for single hidden layer feed forward neural network (SLFN). The SADHS-OELM is applied to Radial Basis Function Neural Network (RBF) and CEFLANN for prediction of closing price and volatility of five different stock indices. The performance comparison of CEFLANN and RBF with different learning schemes such as ELM, DE-OELM, DE, SADHS and two other variants of harmony search algorithm reveals that CEFLANN model trained with SADHS-OELM outperforms other learning methods and also the RBF model for both stock indices and volatility prediction.

Hybrid models based on ANN are found to be successful forecasting methods in predicting stock market indices. However, they suffer from the limitations like black box technique, over fitting, slow rate of convergence and getting trapped in local minima. To overcome these limitations, SVM developed by Vapnik and Vapnik (1998) has been employed as a popular research methodology in the area of stock market indices prediction, which employs the principle of structural risk minimization that aims to minimize the upper bound of generalization error. By applying SVM, overfitting is less likely to occur, and the optimal solution may also be global. Even though separate SVM has outstanding performance, its classification performance and classifier's generalization ability are often influenced by its dimension or the number of feature variables. Thus, researchers prefer to hybridize other techniques with SVM to develop the efficient forecasting model. Huang (2012) develops a hybrid GA-SVR methodology for effective stock selection using SVR as well as GAs. The GA is employed for the optimization of model parameters, and feature selection to acquire optimal subsets of input variables to the SVR model. It shows that the feature selection plays an important role in GA-SVR and the investment returns significantly outperform the benchmark. Wang, Liu, and Wang (2013) use a hybrid method combining DT and SVM algorithms for a stock futures prediction strategy, which can achieve an increase on the best average precision rate, best average recall rate and best average F-One rate among Bootstrap-SVM, Bootstrap-DT and BPNN methods. Nayak, Mishra, and Rath (2015) propose a hybridized framework of SVM with KNN approach for Indian stock market trend reversal analysis. The authors use SVM to predict profit or loss and the output of SVM is to predict future stock value in the horizon of 1 day, 1 week and 1 month. This framework scales relatively well to high dimensional data and controls the trade-off between classifier complexity and error, which presents better prediction capability in the Indian stock market. The results depict that this hybrid model offers significant improvement over recent developed models such as FLIT2FNS and CEFLANN by overcoming the problem of choosing too many parameters of ANN and fuzzy based model.

Most of the previous researches assume that each feature makes the same contribution to the classification, while the relative importance of each feature is not considered. However, this assumption is not always true in the real world. It is known that some features are closely relevant to the classification, some are trivial relevant, and others are irrelevant. The computing of kernel function in SVM is sensitive to trivial relevant or irrelevant features. Properly handling features can improve the robustness and accuracy of classifier's results and help to improve the quality and performance of classifier. To do this, we propose a hybrid framework of feature weighted SVM (FWSVM) and feature weighted K-nearest neighbor (FWKNN) to predict stock market indices in this paper, which assigns different weights for different features and has been compared with the recent SVM-KNN model (Nayak et al., 2015). To further analyze our proposed framework, three steps are involved. In the first part, we calculate the information gain in order to estimate the relative importance of each feature and set weight of each feature. Secondly, we utilize the weights to compute the inner product in kernel functions in SVM for predicting the direction of stock indices price movement. Lastly, we recalculate the weights and adopt them to compute Euclidean distance in KNN to predict stock price indices.

The remainder of this paper is organized as follows. Section 2 describes the proposed theory of FWSVM and FWKNN based on the information gain. Section 3 illustrates data description, feature selection and our proposed FWSVM-FWKNN framework of stock market analysis. Section 4 presents some

experiment results to validate the performance of our proposed model. Finally, concluding this work and some brief comments are presented in Section 5.

2. Research methodology

2.1. Theory of FWSVM

It is well known that classical SVM algorithm is based on the assumption that all the features of samples give the same contribution to the target value. However, this assumption is not always true in many real problems. Accordingly, we develop a new SVM algorithm model called FWSVM and derive the overall theory of FWSVM based on feature weighting in this section. In FWSVM each feature of samples is assigned a different weight value on the basis of certain principle.

Let $T_{train} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ be the training dataset, where $x_i = (x_i^1, x_i^2, \ldots, x_i^n) \in X = R^n$ is ith input feature vector, n is the number of features, and $y_i \in Y = \{+1, -1\}, i = 1, 2, \ldots N$ is the class label. Consider the problem of separating the training dataset T_{train} with a hyperplane $\omega \cdot \phi(xP) + b = 0$ into two separate classes, where P is a $n \times n$ matrix called feature weighted matrix. FWSVM classifier satisfies the following conditions:

$$y_i(\omega \cdot \phi(xP) + b) \geq 1 - \zeta_i, (\zeta_i \geq 0, i = 1, 2, \ldots, N) \tag{1}$$

where ϕ: $R^n \rightarrow R^m$ is the feature mapping function, which maps the input space to a usually high dimensional feature space where the data points become linearly separable, ζ_i is the slack variable.

In this way, when an error occurs, the corresponding ζ_i is added and the upper bound on the number of training errors is $\sum_{i=1}^{N} \zeta_i$. The objective optimization problem is as follows:

$$\min_{\omega,b,\zeta} : J(\omega, \zeta) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N} \zeta_i \tag{2}$$

$$\text{subject to: } \begin{cases} y_i(\omega \cdot \phi(xP) + b) \geq 1 - \zeta_i, i = 1, 2, \ldots, N \\ \zeta_i \geq 0, i = 1, 2, \ldots, N \end{cases}$$

where C is a positive constant parameter used to control the trade-off between the margin and the classification error. Eq. (2) has two meanings: (a) make $\frac{1}{2}||\omega||^2$ as small as possible in order to get the smallest margin; (b) make the mistake of classifier as fewer as possible. Thus, the corresponding Lagrangian function is Eq. (3).

$$L_p(\omega, b, \zeta, \alpha, \mu) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N} \zeta_i - \sum_{i=1}^{N} \alpha_i[y_i(\omega \cdot \phi(x_iP) + b) - 1 + \zeta_i] - \sum_{i=1}^{N} \mu_i\zeta_i \tag{3}$$

where α_i, μ_i are the non-negative Lagrange multipliers. By introducing the Lagrangian multipliers, the original optimization problem can be transformed into its dual problem. In order to represent Eq. (3) as a dual problem, the Karush–Kuhn–Tucker (KKT) conditions (Fletcher, 2013) for the primal problem need to be calculated.

$$\frac{\partial L_p(\omega, b, \zeta, \alpha, \mu)}{\partial \omega} = \omega - \sum_{i=1}^{N} \alpha_i y_i \phi(x_iP) = 0 \tag{4}$$

$$\frac{\partial L_p(\omega, b, \zeta, \alpha, \mu)}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{5}$$

$$\frac{\partial L_p(\omega, b, \zeta, \alpha, \mu)}{\partial \zeta_i} = C - \alpha_i - \mu_i = 0 \tag{6}$$

$$\alpha_i[y_i(\omega \cdot \phi(x_iP) + b) - 1 + \zeta_i] = 0 \tag{7}$$

Table 1
FWKNN algorithm for prediction.

Step 1. Determine the number of nearest neighbors of K.
Step 2. According to Eq. (27), calculate the weight of feature matrix P in Eq. (28).
Step 3. According to Eq. (30), calculate the weighted Euclidean distance. Sort the distance in ascending order and pick the first K distance samples $x_1, x_2, \ldots, x_k$ as the neighbors.
Step 4. Average $V_1, V_2, \ldots, V_k$ of these neighbors and set it as the predicted value of V.

$$\mu_i\zeta_i = 0 \tag{8}$$

$$y_i(\omega \cdot \phi(x_iP) + b) \geq 1 - \zeta_i \tag{9}$$

$$\zeta_i \geq 0 \tag{10}$$

$$\alpha_i \geq 0 \tag{11}$$

$$\mu_i \geq 0 \tag{12}$$

Hence, $\omega = \sum_{i=1}^{N} \alpha_i y_i \phi(x_iP)$. According to the KKT complementarity conditions, we can simply take any training data (x_j, y_j) which satisfies $0 < a_j < C$ to compute b as follows.

$$b = y_j - \omega \cdot \phi(x_jP) \tag{13}$$

where the value of b is not unique. It is reasonable to take the mean value of all b resulting from Eq. (13). Hence,

$$b^* = \frac{1}{N_S} \sum_{0<\alpha_j<C} [y_j - \sum_{i=1}^{N} \alpha_i y_i (\phi(x_iP) \cdot \phi(x_jP))] \tag{14}$$

where N_S is the number of the support vectors.

For a new data sample x, the classification decision function is given by Eq. (15).

$$f(x) = sign(\omega \cdot \phi(xP) + b^*) \tag{15}$$

Substituting ω and b^* in Eq. (14) into Eq. (15), we get the classification decision function

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i (\phi(x_iP) \cdot \phi(xP)) + \frac{1}{N_S} \sum_{0<\alpha_j<C} [y_j - \sum_{i=1}^{N} \alpha_i y_i (\phi(x_i \cdot P)\phi(x_jP))]) \tag{16}$$

In order to express FWSVM, The feature weighted kernel function $K_p(x_i, x_j)$ is defined as follows:

$$K_p(x_i, x_j) = \phi(x_iP) \cdot \phi(x_jP) = K(x_iP, x_jP) \tag{17}$$

where P is feature weighted matrix. The different selection of P leads to different weighted cases.

Case 1: When P is an $N \times N$ identity matrix, this is weighted-free case.

$$P = \begin{Bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{Bmatrix} \tag{18}$$

Case 2: When P is an $N \times N$ diagonal matrix, there is one weight value per feature and no interaction between features. f_{ii} represents the weight of ith feature of the sample and is not always

Table 2
Used technical indicators with formula.

Technical indicator	Formulae
1.Moving Average (MA)	$MA(N) = \frac{1}{N}\sum_{i=1}^{N} s_{i,close}$
2.Exponential Moving Average (EMA)	$EMA(N) = \begin{cases} s_{1,close} & if\ N = 1 \\ \frac{2}{N+1} * s_{N,close} + \frac{N-1}{N+1} * EMA(N-1) & if\ N > 1 \end{cases}$
3.Moving Average Convergence/ Divergence (MACD)	$DIF(i) = EMA(N_{fast}) - EMA(N_{slow})$ $DEA(i) = \alpha * DEA(i-1) + (1-\alpha) * DIF(i)$ $MACD(i) = 2 * (DIF(i) - DEA(i))$
4.Volume Ratio (VR)	$AVS(N) = \sum_{i=1}^{N-1} I(s_{i,close} < s_{i+1,close}) * vol_i$ $BVS(N) = \sum_{i=1}^{N-1} I(s_{i,close} > s_{i+1,close}) * vol_i$ $CVS(N) = \sum_{i=1}^{N-1} I(s_{i,close} = s_{i+1,close}) * vol_i$ $VR(N) = (AVS(N) + 1/2CVS(N))/(BVS(N) + 1/2CVS(N))$ $I(x)$ is indicative function
5.Relative Strength Index (RSI)	$RSI(N) = 100 - \frac{100}{1+EMA(N)_{up}/EMA(N)_{down}}$ $EMA(N)_{up}$ is upward changes, $EMA(N)_{down}$ is downward changes for N
6.On Balance Volume (OBV)	$OBV(N) = \begin{cases} 0 & if\ N = 1 \\ OBV(N-1) + sgn(s_{N,close} - s_{N-1,close}) * vol_N & if\ N > 1 \end{cases}$ $sgn(x)$ is signum function
7.Momentum Index (MTM)	$MTM(i,N) = s_{i,close} - s_{i-N,close}$
8.AR	$AR(N) = \sum_{i=1}^{N}(s_{i,hight} - s_{i,open})/\sum_{i=1}^{N}(s_{i,open} - s_{i,low}) * 100$
9.BR	$BR(N) = \sum_{i=1}^{N}(s_{i,hight} - s_{i-1,close})/\sum_{i=1}^{N}(s_{i-1,close} - s_{i,low}) * 100$

Table 3
Description of data samples and data range.

Time horizon	SSE composite index/SZSE COMP SUB IND				
	Total samples	Data range	Training data	Testing data	Step window
1 day	1500	31/Oct/2008 to 31/Dec/2014	999	500	1
5 days	1500	31/Oct/2008 to 31/Dec/2014	995	500	5
10 days	1500	31/Oct/2008 to 31/Dec/2014	990	500	10
15 days	1500	31/Oct/2008 to 31/Dec/2014	985	500	15
20 days	1500	31/Oct/2008 to 31/Dec/2014	980	500	20
30 days	1500	31/Oct/2008 to 31/Dec/2014	970	500	30

the same, where $i = 1, 2, \ldots, N$.

$$P = \begin{Bmatrix} f_{11} & 0 & \cdots & 0 \\ 0 & f_{22} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & f_{NN} \end{Bmatrix} \tag{19}$$

Case 3: When P is $N \times N$ square matrix, it is full weighted case.

$$P = \begin{Bmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & f_{22} & \cdots & f_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N1} & \cdots & 0 & f_{NN} \end{Bmatrix} \tag{20}$$

Therefore, the non-linear classification decision function is

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y K_p(x_i, x) + \frac{1}{N_S} \sum_{0<\alpha_j<C} [y_j - \sum_{i=1}^{N} \alpha_i y_i K_p(x_i, x_j)]) \tag{21}$$

The three most commonly used feature weighted kernels can be expressed as follows.

(a) Linear kernel :

$$K_p(x_i, x_j) = (x_i P) \cdot (x_j P) = x_i P P^T x_j^T \tag{22}$$

(b)Polynomial kernel :

$$K_p(x_i, x_j) = [\gamma (x_i P) \cdot (x_j P) + r]^d = (\gamma x_i P P^T x_j^T + r)^d, \gamma > 0 \tag{23}$$

(c)Gaussian kernel :

$$K_p(x_i, x_j) = \exp(-\gamma \frac{||x_i P - x_j P||^2}{2\delta^2}) = \exp(-\gamma \frac{(x_i - x_j) P P^T (x_i - x_j)^T}{2\delta^2}), \gamma > 0 \tag{24}$$

2.2. FWSVM based on the information gain

This section briefly describes the concept of basic information gain for FWSVM. The similar procedure can be found in Sheng-feng and Hou-kuan (2009) and Wang, Guan, Wang, and Wang (2006). The concept of feature weighting is to assign the weight values to different features based on some certain principle and thus the key question is how to measure the correlation of feature. In this paper, the information gain is used to measure the importance of each feature (Dai & Xu, 2013).

The training dataset as mentioned above is $T_{train} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, $x_i = (x_i^1, x_i^2, \ldots, x_i^n) \in X = R^n$, $y_i \in \{C_{-1}, C_{+1}\} = \{-1, +1\}$. Let $C_{\{i,D\}}$ denote the subset of D belonging to class C_i, where $i = -1, +1$. Thus, the following relationships are obtained: $C_{\{-1,D\}} \cup C_{\{+1,D\}} = D$ and $C_{\{-1,D\}} \cap C_{\{+1,D\}} = \emptyset$. Also, let $|D|$ denote the size of sample set D and $|C_{\{i,D\}}|$ denote the size of sample set $C_{\{i,D\}}$. The probability of a sample belonging to class C_i can be approximately calculated by $|C_{\{i,D\}}|/|D|$. So the expected information to classify dataset can be expressed by the following formula.

$$Info(D) = -\sum_{i \in \{-1,+1\}} \frac{|C_{\{i,D\}}|}{|D|} \log(\frac{|C_{\{i,D\}}|}{|D|}) \tag{25}$$

Suppose the dataset T_{train} is split on the feature $A_{feature}$ into some subsets indicated by $D_1, D_2, \ldots, D_v$. Then, based on this partition of T_{train}, the expected information is as follows.

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} Info(D_j) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \sum_{i \in \{-1,+1\}} \times \frac{|C_{\{i,D_j\}}|}{|D_j|} \log(\frac{|C_{\{i,D_j\}}|}{|D_j|}) \tag{26}$$

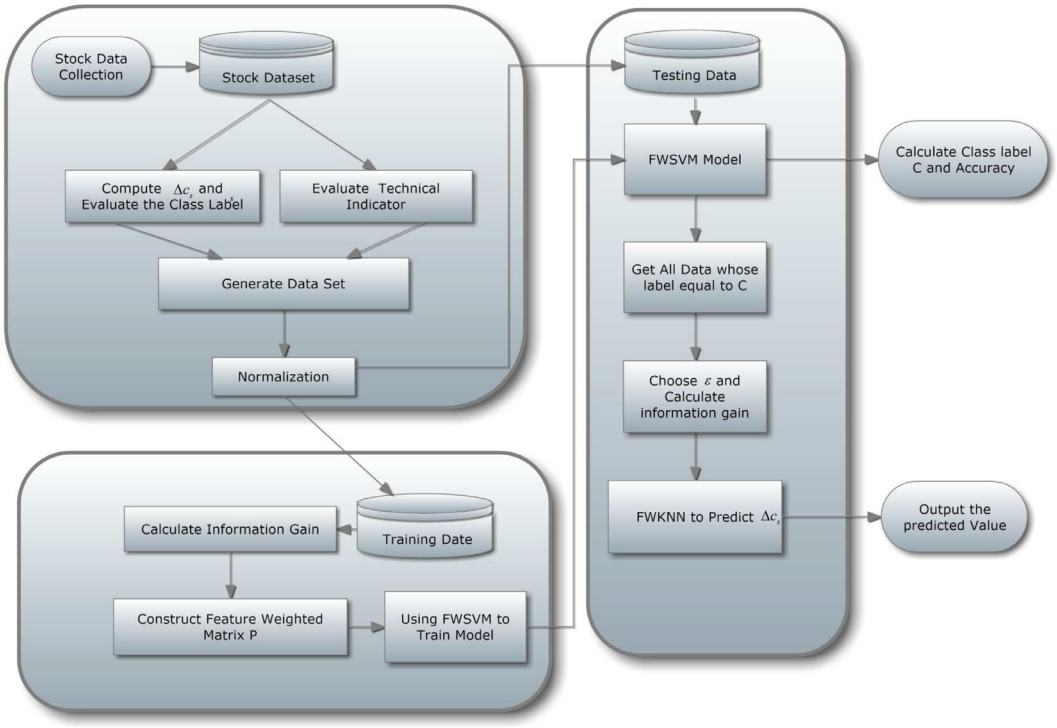


Fig. 1. Schematic layout of FWSVM combined with FWKNN stock analysis algorithm.

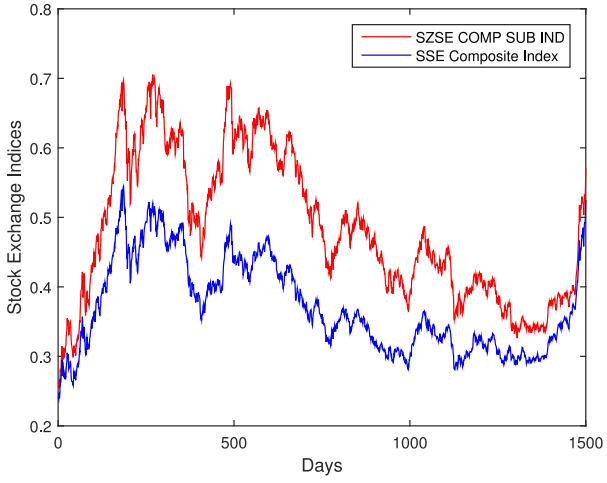


Fig. 2. Two normalized stock indices.

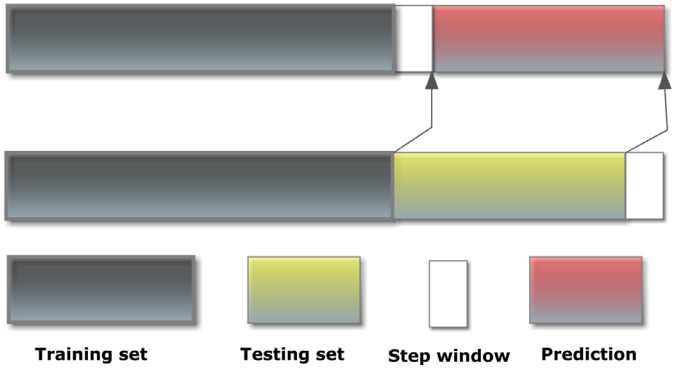


Fig. 3. Training-testing sets generation and prediction.

where $|D_j|$ is the size of sample set D_j, $|C_{\{i,D_j\}}|$ is the number of samples belonging to the class C_i in set D_j and $i = -1, +1$, $j = 1, 2, \ldots, v$. Thus, the information gain with respect to each feature is defined as the square root of subtraction between the original information $Info(D)$ and the new information $Info_A(D)$.

$$InfoGain(A) = \sqrt[2]{Info(D) - Info_A(D)} \tag{27}$$

If this information gain is bigger, the corresponding feature is more distinguished and has greater contribution to classification. Consequently, the information gain could be applied to measure the importance of a feature to classification. The feature weighted matrix based on the information gain is as follows.

$$P = \begin{Bmatrix} InfoGain(f_1) & 0 & \cdots & 0 \\ 0 & InfoGain(f_2) & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & InfoGain(f_n) \end{Bmatrix} \tag{28}$$

where $InfoGain(f_i)$ describes the weight of each feature and $i = 1, 2, \ldots, n$.

2.3. Theory of FWKNN

KNN is a nonparametric learning algorithm and sensitive to the distance function because of the inherent sensitivity to the irrelevant features. Therefore, we proceed to derive FWKNN forecasting model based on the feature weighted matrix in this section. Let the training dataset be $T_{train} = \{(x_1, V_1), (x_2 V_2), \ldots, (x_N, V_N)\}$, where $x_i = (x_i^1, x_i^2, \ldots, x_i^n) \in X = R^n$ is ith input feature vector, n is the number of features, $V_i \in R, i = 1, 2, \ldots, N$ is the output value. We construct the class label y_i by the following equation.

$$y_i = \begin{cases} +1 \ if \ V_i \geq \epsilon \\ -1 \ if \ V_i < \epsilon \end{cases} \tag{29}$$

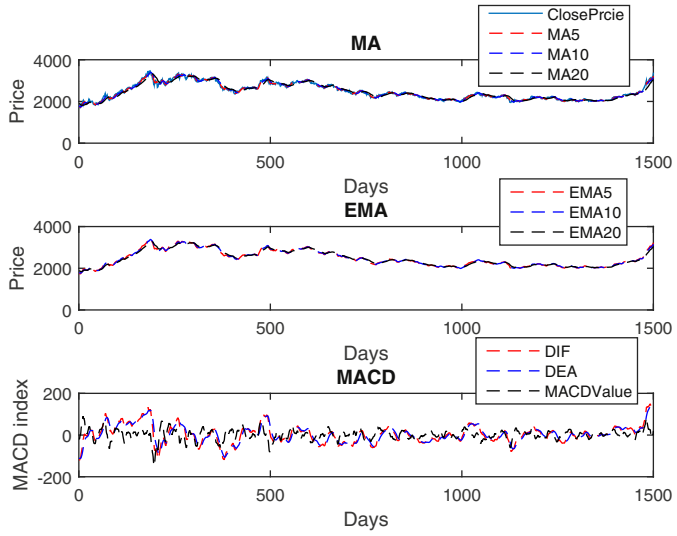


Fig. 4. MA, EMA and MACD technical indicator of SSE composite index.

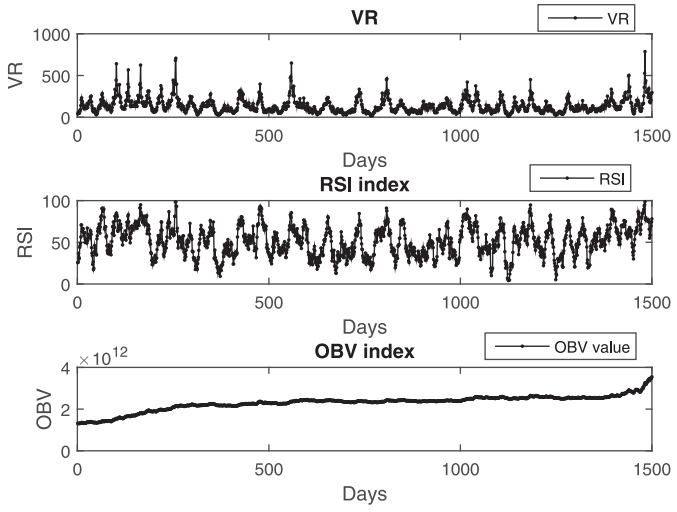


Fig. 5. VR, RSI and OBV technical indicator of SSE composite index.

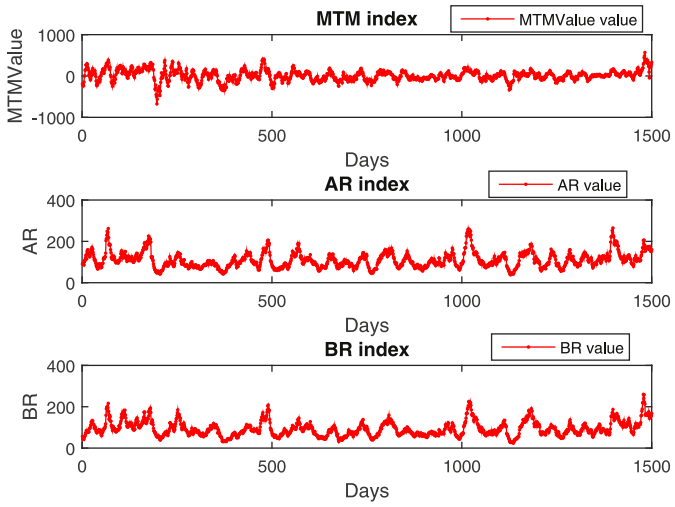


Fig. 6. MTM, AR and BR technical indicator of SSE composite index.

where ϵ is the threshold value chosen depending upon the realistic demands. FWKNN makes a decision by identifying the k weighted nearest neighbors, which share the highest similarity with the testing data. Thus, for FWKNN, it requires a matrix that measures the distance between the testing data and cases from the training samples. To measure the similarity, one of the most popular choices is Euclidean distance. FWKNN uses the modified distance by using feature weighted matrix P to expand the standard Euclidean distance. The modified distance formula between two points x_i and x_j is given as follows.

$$d^w(x_i, x_j) = \sqrt[2]{\sum_{k=1}^{n} P_{k,k} \cdot (x_{i,k} - x_{j,k})^2} \tag{30}$$

where $d^w(x_i, x_j)$ is weighted Euclidean distance between sample x_i and x_j, $x_{i,\,k}$ is the kth feature value of x_i and $x_{j,\,k}$ is the kth feature value of x_j. $P_{k,k} \geq 0, (i = 1, 2, \ldots, n)$ is weighted feature value, which indicates the importance of each feature. From the formula, we can see that feature-weighted transformation changes the positional relation of the sample points. The detailed algorithm is depicted in Table 1.

3. Research design

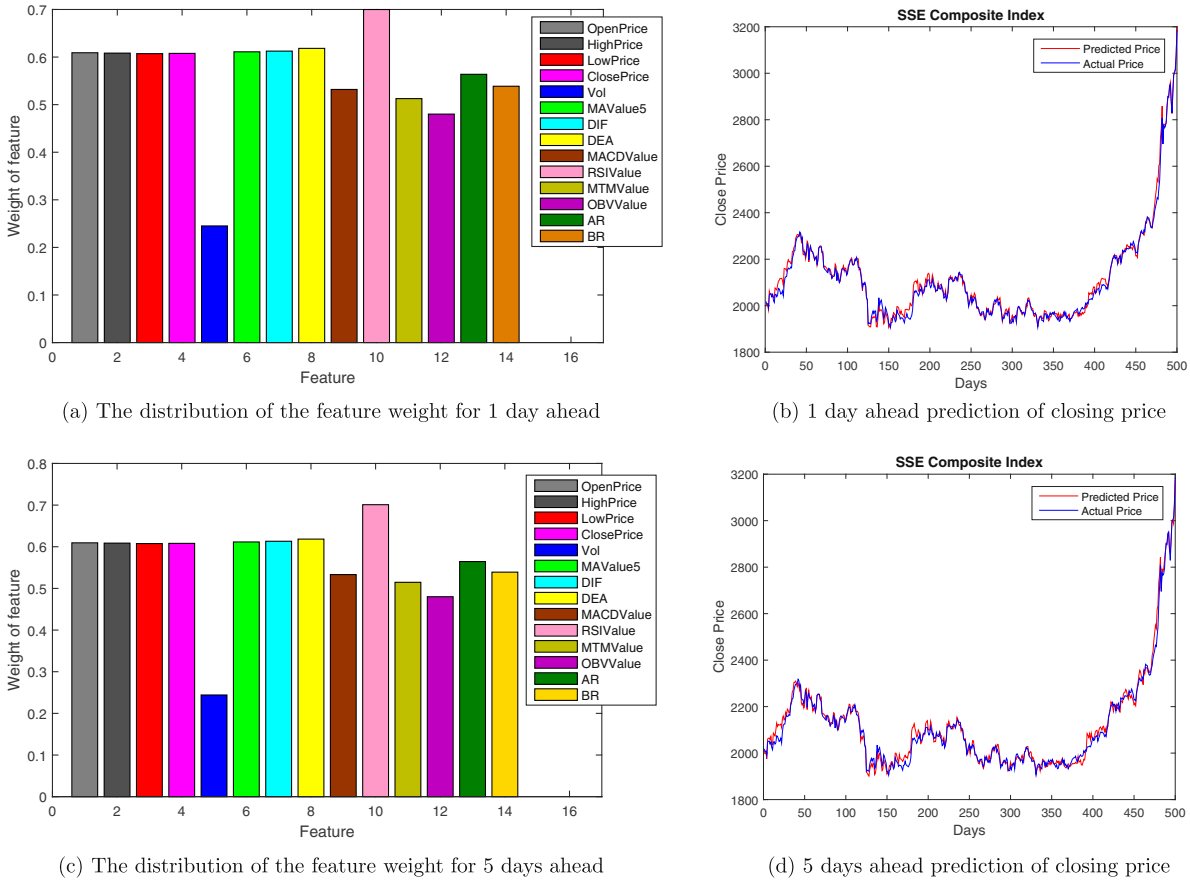
3.1. Data description

The datasets utilized for this study involve Shanghai Stock Exchange Composite Index (SSE Composite Index) and Shenzhen Stock Exchange Component Index (SZSE COMP SUB IND) for analysis of Chinese stock market trend throughout this paper for experimental purpose.

SSE Composite Index is a capitalization-weighted index, developed on December 19, 1990 with a base value of 100. It tracks the daily price performance of all A-shares and B-shares listed on Shanghai Stock Exchange, which is the world's fastest-growing and emerging securities market. Index trade volume on Q is scaled down by a factor of 1000.

SZSE COMP SUB IND is also a Capitalization Weighted Index. The constituents consist of the 500 top companies that issue A-shares on Shenzen Stock Exchange. The index was developed with a base value of 1000 as of July 20, 1994. Index trade volume on Q is scaled down by a factor of 1000. On May 20, 2015, number of members changed from 40 to 500.

SSE Composite Index and SZSE COMP SUB IND play an important role on the development of the whole national economy, which develop in the macroeconomic environment, and serve the development of national economy at the same time. The trend of the stock market and macroeconomic operation should be consistent and thus the change of the stock market reflects the economic cycle. An economic cycle, also referred to as the stock market cycle, has four stages: recession, crisis, recovery and prosperity. Generally speaking, the decline of stock market index will be accompanied by the recession of economic and vice versa. Consequently, it is very important to predict the stock market trend. In the stock market, there exists bullish and bearish market. In bullish market, more people buying stock leads to the growing of the stock prices. Bullish market generally refers to a continuously rising market where more buyers than sellers, especially with the stock indices increase by more than 20% lasting for more than 2 weeks. On the contrary, bearish market generally refers to a continuously falling and flagging market, where the stock price is downward with stock indices decrease by more than 20% lasting for more than 2 weeks. In order to know whether the stock market will be bullish or bearish, we need to be closely concerning about not only stock price but also the trading volume of the stock. In this paper, we keep a closer view on SSE Composite Index and SZSE COMP



(a) The distribution of the feature weight for 1 day ahead

(b) 1 day ahead prediction of closing price

(c) The distribution of the feature weight for 5 days ahead

(d) 5 days ahead prediction of closing price

Fig. 7. Short term prediction (SSE composite index).

SUB IND, so that we can predict the future closing price. The data collected for the stock indices consists of the daily opening price, the closing price, the lowest price, the highest price and the total trading volume.

Let S be the stock dataset, it is described as follows:

$$S = \begin{bmatrix} s_{1,open} & s_{1,low} & s_{1,heigh} & s_{1,close} & vol_1 \\ s_{2,open} & s_{2,low} & s_{2,heigh} & s_{2,close} & vol_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{n,open} & s_{n,low} & s_{n,heigh} & s_{n,close} & vol_n \end{bmatrix} \tag{31}$$

where $s_{i,open}(i = 1, 2, 3, \ldots, n)$ represents the opening price of stock for ith day, $s_{i,low}(i = 1, 2, 3, \ldots, n)$ represents the lowest price of stock for ith day, $s_{i,heigh}(i = 1, 2, 3, \ldots, n)$ represents the highest price of stock for ith day, $s_{i,close}(i = 1, 2, 3, \ldots, n)$ represents the closing price of stock for ith day, and $vol_i(i = 1, 2, 3, \ldots, n)$ represents the trading volume for ith day. A major issue related to any stock market dataset is that it does not contain class label. Hence, we introduce a new attribute in order to identify class label, which is the change of the closing price and formulated as follows.

$$\Delta c_s^i = s_{i,close} - s_{i+m,close}, \forall i = 1, 2, 3, \ldots, n - m \tag{32}$$

where m is the step size of time horizon, Δc_s^i is the difference of closing values between ith day and $(i + m)$th day, it can indicate returns. If $\Delta c_s^i \leq 0$, it represents profit otherwise loss. Hence, Δc_s^i can be used as the class label as well as the direction of stock indices price movement. If Δc_s^i is larger than zero, we set $y_i = -1$ as the class label. If Δc_s^i is less than or equal to zero, we set $y_i = +1$ as the class label.

3.2. Feature selection

Historical opening price, closing price, lowest price, highest price and total volume on stock market indices are usually used as the input features. Recently, some researches show that technical indicators are good predictors (Hsu, Lessmann, Sung, Ma, & Johnson, 2016; Neely, Rapach, Tu, & Zhou, 2014; Żbikowski, 2015). Consequently, under different circumstances many important technical indicators defined in Table 2 will be taken into consideration along with the daily prices and trading volume of the specific stocks. These technical indicators are of any class of metrics whose values are derived by applying a formula to the opening price, the lowest price, the highest price and trading volume data. A brief explanation of each indicator is provided here.

(i) Moving Average (MA)

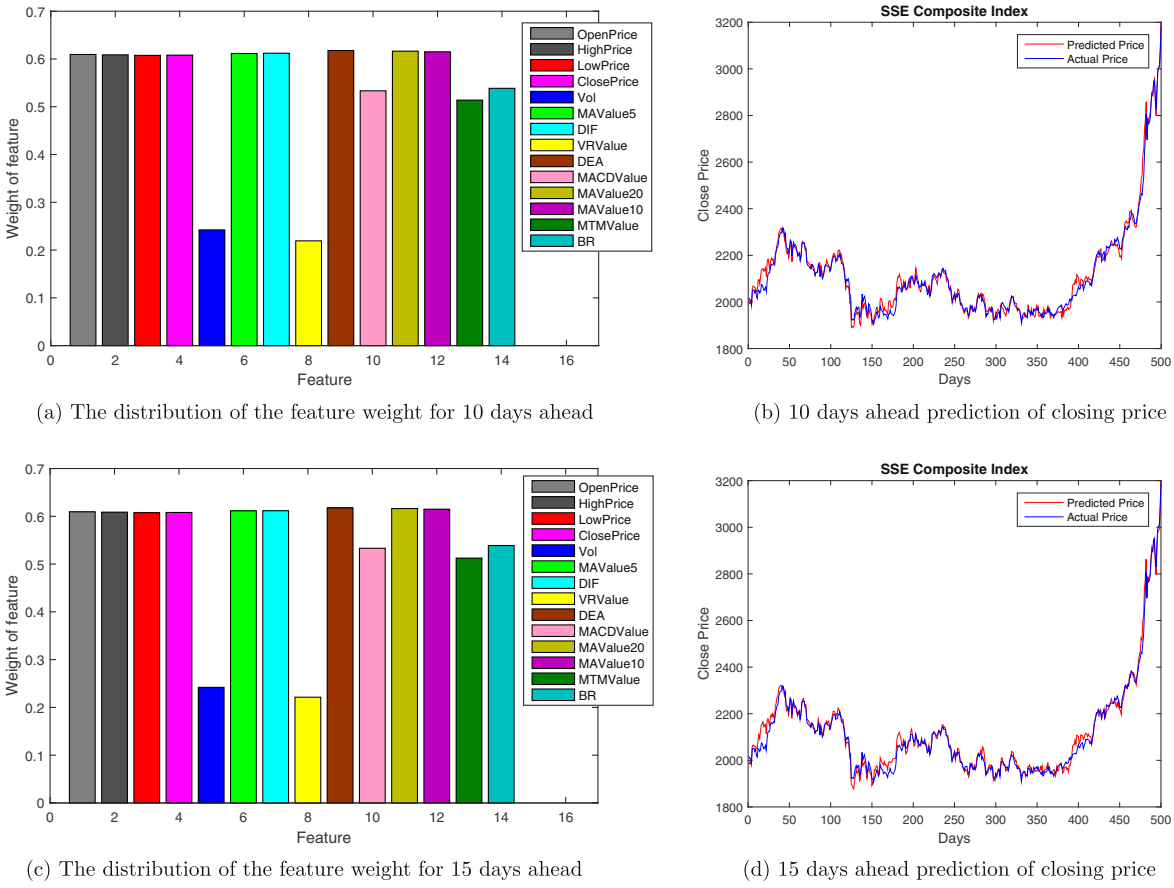
It is the average of the values of the neighbors by taking a window of the specified period, which is used to smooth the datas.

(ii) Exponential Moving Average (EMA)

It is also known as an average of the values in the specified period, except that more weight is given to the latest data and thus it is more close to the actual values.

(iii) Moving Average Convergence / Divergence (MACD)

MACD is a trend-following momentum indicator that shows the relationship between two moving averages of prices. It is supposed to reveal changes in the strength, direction, momentum, and duration of a trend in a stock's price. It turns two trend-following indicators, moving averages, into a momentum oscillator by subtracting the longer EMA from the shorter EMA.



(a) The distribution of the feature weight for 10 days ahead

(b) 10 days ahead prediction of closing price

(c) The distribution of the feature weight for 15 days ahead

(d) 15 days ahead prediction of closing price

Fig. 8. Medium term prediction (SSE composite index).

(iv) Volume Ratio (VR)

VR is used to identify price ranges and breakouts. It uses a true price range to determine a stock's true trading range and is able to identify situations where the price has moved out of this true range.

$$S = \begin{bmatrix} s_{1,open} & s_{1,low} & s_{1,heigh} & s_{1,close} & v_1 & TI_{1,1} & TI_{1,2} & TI_{1,3} & \cdots & TI_{1,m} & \Delta c_s^1 & y_1 \\ s_{2,open} & s_{2,low} & s_{2,heigh} & s_{2,close} & v_2 & TI_{2,1} & TI_{2,2} & TI_{2,3} & \cdots & TI_{2,m} & \Delta c_s^2 & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{n,open} & s_{n,low} & s_{n,heigh} & s_{n,close} & v_n & TI_{n,1} & TI_{n,2} & TI_{n,3} & \cdots & TI_{n,m} & \Delta c_s^n & y_n \end{bmatrix} \tag{33}$$

(v) Relative Strength Index (RSI)

RSI is a momentum indicator, that compares the magnitude of recent gains and losses over a specified time period to measure speed and change of price movements of a security. It is primarily used to attempt to identify overbought or oversold conditions in the trading of an asset.

(vi) On Balance Volume (OBV)

OBV is a momentum indicator that uses volume flow to predict changes in stock price.

(vii) Momentum Index (MTM)

MTM is the rate of acceleration of a security's price or volume and refers to the force or speed of movement. it is usually defined as a rate.

(viii) AR

AR is a technical indicator that shows the degree of exchange activity in the stock market by studying the level of the opening price during a period of exchange time. It lays emphasis on the stock opening price.

(ix) BR

BR is a technical indicator that shows the degree of exchange willingness in the stock market by studying the level of the closing price during a period of exchange time. It lays emphasis on the stock closing price.

After applying technical indicators discussed in Table 2, the dataset S is reformulated as follows.

The dataset S is scaled between −1 and +1 by using the standard formula described as follows.

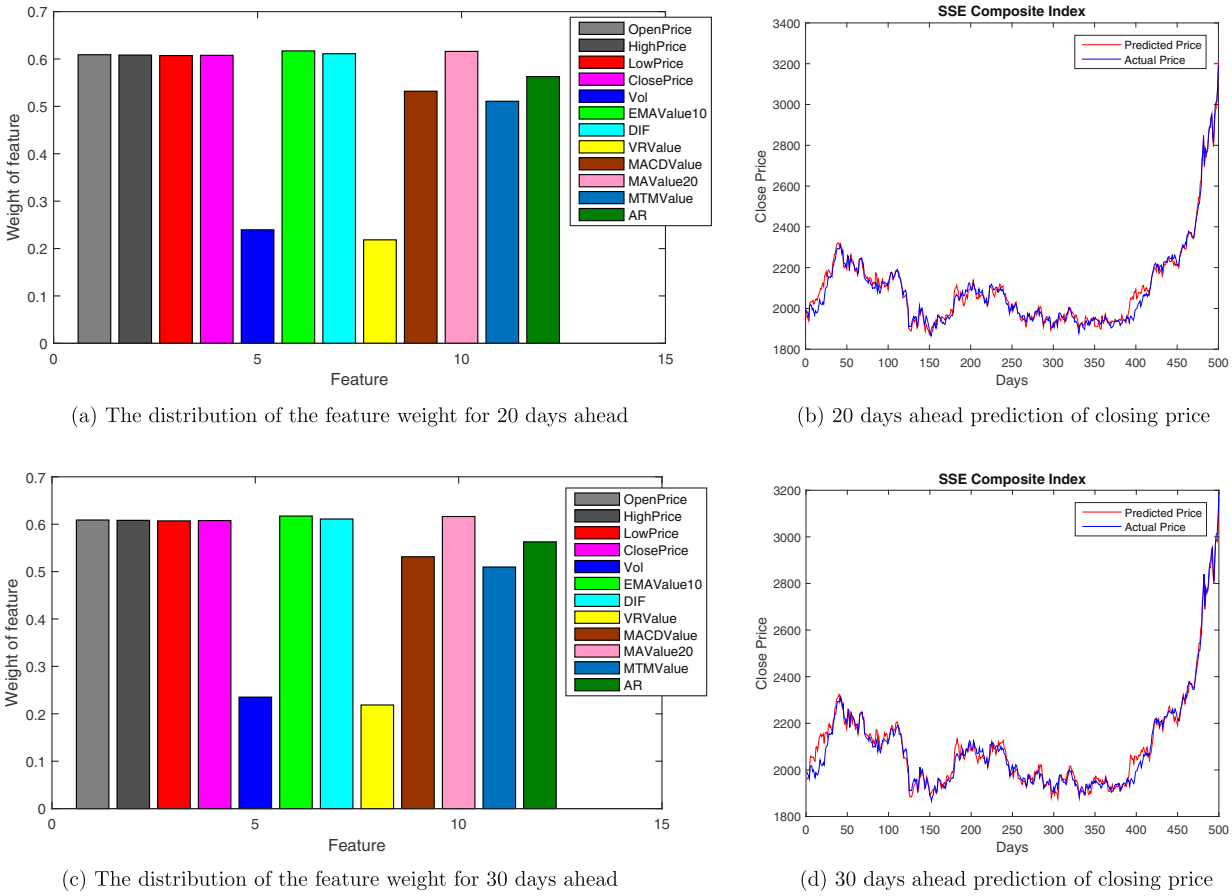
$$\widehat{s}_{i,j} = sign(s_{i,j}) \frac{abs(s_{i,j}) - \min abs(s_j)}{\max abs(s_j) - \min abs(s_j)} \tag{34}$$

$$sign(s_{i,j}) = \begin{cases} +1 & if\ s_{i,j} \geq 0 \\ -1 & if\ s_{i,j} < 0 \end{cases}$$

where $s_{i,j}$ is the current day value and $\widehat{s}_{i,j} \in [-1, 1]$ is scaled value, $\min abs(s_j) = \min\{abs(s_{i,j}), i = 1, 2, \ldots, n\}$ is the minimum absolute value of jth feature of S, $\max abs(s_j) = \max\{abs(s_j), i = 1, 2, \ldots, n\}$ is the maximum absolute value of jth feature of S.

To evaluate the performance of our proposed methods, Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) are adopted and defined in Eqs. (35) and (36) as follows.

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{s_{i.close} - \widehat{s}_{i.close}}{s_{i.close}} \right| \times 100 \tag{35}$$



(a) The distribution of the feature weight for 20 days ahead

(b) 20 days ahead prediction of closing price

(c) The distribution of the feature weight for 30 days ahead

(d) 30 days ahead prediction of closing price

Fig. 9. Long term prediction (SSE composite index).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_{i.close} - \widehat{s}_{i.close})^2} \tag{36}$$

where $s_{i.close}$ is the actual price on ith day, $\widehat{s}_{i.close}$ is the predicted price on ith day, and N is the total number of testing data.

3.3. Schematic layout of proposed FWSVM-FWKNN stock analysis algorithm

This section outlines the theoretical framework of the proposed hybridized stock market analysis algorithm. As illustrated in Fig. 1, the framework consists of three basic phases: the data collection and preprocessing phase, the training phase, and the testing phase. In the data collection and preprocessing phase, the opening price, the lowest price, the highest price, the closing price and trading volume data are firstly collected and stored into stock database indicated as Eq. (31). Later we compute Δc_s based on the change of the closing price according to Eq. (32) to generate the class label y_i, and evaluate the technical indicators $TI_{i,\,m}$ with the opening price, the lowest price, the highest price, the closing price and trading volume data according to Table 2 depending upon the requirement. The initial dataset is generated by adding $TI_{i,\,m}$ and y_i to the old dataset according to Eq. (33) and the data points are normalized and mapped within range $[-1, 1]$ using Eq. (34). The dataset is then divided into the training dataset S_{train} and the testing dataset S_{test}. In the training phase, the information gain is calculated according to Eq. (27) and the feature weighted matrix P is constructed. FWSVM is then trained on the training data and the trained model is exported. In the testing phase, the testing data

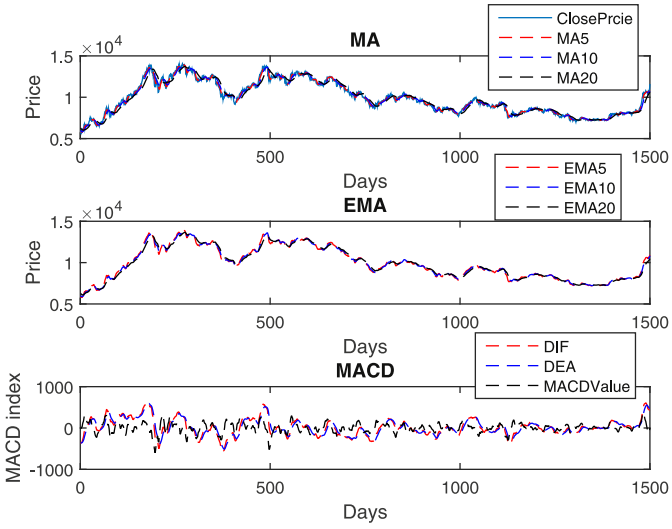


Fig. 10. MA, EMA and MACD technical indicator of SZSE COMP SUB IND.

is classified using the trained FWSVM model and the output of the model is either $+1$ or -1, which is indicated as C. Then this value of C is compared with class label to compute the accuracy of the model. Again from previous data, find data whose class label equals to C and set median of those data as ϵ in order to recompute the information gain. Afterwards, FWKNN algorithm is used to find k nearest neighbors of the testing data and evaluate the mean of those neighbors to predict Δc_s. Result is then added with

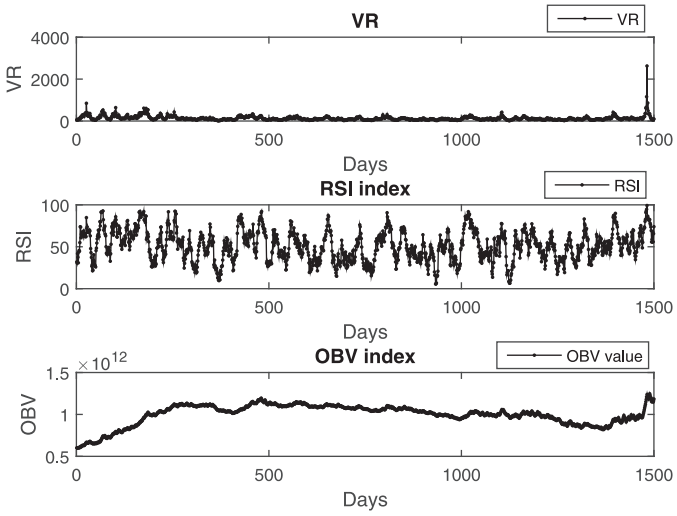


Fig. 11. VR, RSI and OBV technical indicator of SZSE COMP SUB IND.

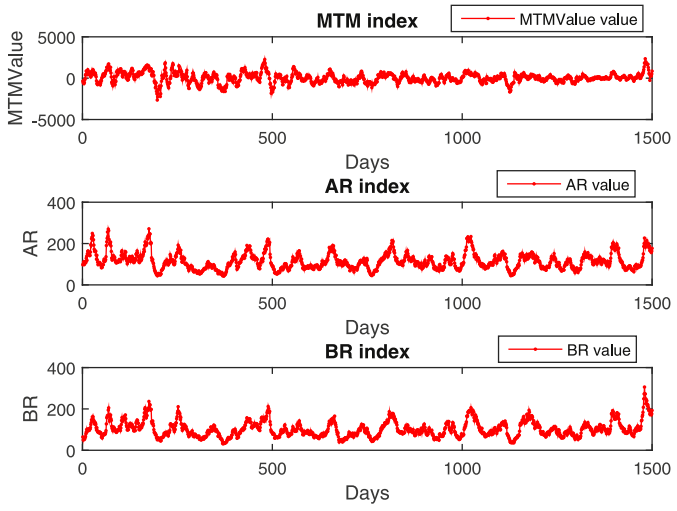


Fig. 12. MTM, AR and BR technical indicator of SZSE COMP SUB IND.

Table 4
Description of parameters used for FWSVM and FWKNN on SSE Composite Index.

Parameter	Short term		Medium term		Long term	
	Time horizon		Time horizon		Time horizon	
	1 day	5 days	10 days	15 days	20 days	30 days
C	38	41	30	36	45	32
γ	0.5	0.6	0.4	0.8	0.7	0.4
K	10	10	10	10	10	10

the current original non-normalized closing value as the predicted value in advance.

4. Experiment results

4.1. Analysis of datasets and input selection

The initial experimental data that is the collection of the daily opening price, closing price, lowest price, highest price and the total trading volume, is obtained from two significant Chinese stock markets, i.e. the SSE Composite Index and SZSE COMP SUB IND. The proposed FWSVM-FWKNN forecasting model is used to predict the SSE Composite Index and SZSE COMP SUB IND for 1 day, 5 days, 10 days, 15 days, 20 days and 30 days ahead. The collected datasets and their dimensions are depicted in Table 3, where 1500 samples are from SSE Composite Index and SZSE COMP SUB IND from Oct 31st 2008 to Dec 31st 2014. The reason why we chose this period is that the Chinese stock market experienced a stable and healthy growth because of the stable economic condition during the period. The generating and forecasting process of the datasets used in our experiment are depicted in Fig. 3. To further illustrate the process, we let the size of the initial dataset be l_{all}, step window of time horizon be l_{step}, the size of each training set and testing set be l_{train} and l_{test} respectively. In this way, the size of new dataset will be $l_{all} - l_{step}$, where the input features are features in current time and the actual target values are values l_{step} step later from current time. In our experiment, the last 500 (l_{test}) samples are used for testing and the remaining samples are used for training. In the proposed models, inputs are normalized according to Eq. (34) and the normalized closing values of these datasets are shown in Fig. 2.

Table 5
Description of parameters used for FWSVM and FWKNN on SZSE COMP SUB.

Parameter	Short term		Medium term		Long term	
	Time horizon		Time horizon		Time horizon	
	1 day	5 days	10 days	15 days	20 days	30 days
C	29	35	38	39	48	52
γ	0.2	0.5	0.6	0.6	0.3	0.8
K	10	10	10	10	10	10

Table 6
The comparison results of SSE composite index between FWSVM and SVM in percentage (%) for predicting Profit or Loss.

Method	Short term		Medium term		Long term	
	Time horizon		Time horizon		Time horizon	
	1 day	5 days	10 days	15 days	20 days	30 days
SVM	96.4	94.6	97.8	97.2	97.0	98.0
FWSVM	98.2	96.2	98.8	97.8	97.4	98.4

Table 7
The comparison results of SSE composite index between FWSVM and SVM on AUC scores.

Method	Short term		Medium term		Long term	
	Time horizon		Time horizon		Time horizon	
	1 day	5 days	10 days	15 days	20 days	30 days
SVM	0.9977	0.9830	0.9981	0.9984	0.9976	0.9993
FWSVM	0.9990	0.9966	0.9999	0.9987	0.9976	0.9996

For forecasting the SSE Composite Index and SZSE COMP SUB IND, in addition to the opening price, closing price, lowest price, highest price, the total trading volume, nine appropriate technical indicators are considered as initial feature pool since technical indicators are effective tools to characterize the real market situation in financial time series prediction (Leigh, Hightower, & Modani, 2005; Yeh, Lien, & Tsai, 2011) and can be more informative than using pure prices (Nikfarjam, Emadzadeh, & Muthaiyah, 2010). Based on the review of domain experts and literatures, the selected nine technical indicators are MA, EMA, MACD, VR, RSI, OBV, MTM, AR and BR. These technical indicators on SSE Composite Index are depicted in Figs. 4, 5 and 6 and that on SZSE COMP SUB IND are shown in Figs. 10, 11 and 12. Figs. 4 and 10 show the MA index and EMA index for 5 days, 10 days and 20 days, which are used to smooth the value. Figs. 4 and 10 also show the DIF, DEA and MACD index, which implicit that the overall market will be in rising or reducing trend. When the DIF and MACD are greater than zero (i.e., they are expressed above the zero line in

Table 8
Performance comparison of SSE composite index between FWSVM-FWKNN and SVM-KNN in MAPE and RMSE.

	Method	Short term		Medium term		Long term	
		Time horizon		Time horizon		Time horizon	
		1 day	5 days	10 days	15 days	20 days	30 days
MAPE	SVM-KNN	0.653	0.83	0.86	0.86	0.93	1.06
	FWSVM-FWKNN	0.646	0.80	0.86	0.85	0.93	1.03
RMSE	SVM-KNN	0.0147	0.0198	0.0225	0.0226	0.0226	0.0239
	FWSVM-FWKNN	0.0143	0.0197	0.0222	0.0224	0.0206	0.0230

the figure) and move up, the stock prices tend to grow. On the contrary, when the DIF and the MACD is less than zero (i.e., they are expressed below the zero line in the figure) and move down, it will be in bearish market. Figs. 5 and 11 show the VR, RSI, and OBV index. VR index measures the heat of the stock market from the perspective of trading volume, RSI index predicts the price change direction in future in accordance with the price changes in a specific period, and OBV index predicts the stock price trend through the statistical volume change. Figs. 6 and 12 show MTM index, AR index, and BR index. MTM index denotes the process of acceleration in the stock price by the kinetics principle, of which the theory basis is the relevance between supply-demand relationship and the stock price. AR index and BR index are the technical indicators based on the analysis of the historical stock prices. AR index attaches importance to the opening price, and BR to the closing price.

4.2. Kernel function selection and parameter selection

When applying SVM to financial forecasting, the important thing that needs to be considered is the choosing of kernel function. Since the dynamics of financial time series are strongly non-linear, it is intuitively believed that using non-linear kernel functions could achieve better performance than the linear kernel. Many researchers have discussed the choosing of kernel functions (Cao & Tay, 2003) in financial time series forecasting. In this paper, we use the Gaussian kernel function, because Gaussian kernels tend to give good performance under general smoothness assumptions.

When the kernel function is selected, two important parameters (C, γ) need to be fixed. Parameter C is the cost of C-SVM and parameter γ is the value of gamma in kernel function. The value of C and γ can obviously affect the performance of SVM. In order to find an optimal value of these parameters, the grid search based on 5-fold cross validation is used to select for traditional SVM according to the method in paper (Cao & Tay, 2003), and we choose the same parameters for FWSVM to test for possible improvement. In the experiments, the Matlab 2014B and libsvm-3.20 (Chang & Lin, 2011) are used. Table 4 depicts parameters used in FWSVM and FWKNN on SSE Composite Index and Table 5 depicts parameters used in FWSVM and FWKNN on SZSE COMP SUB.

4.3. Results and discussion

4.3.1. Experiments on SSE composite index

In this section, to fairly evaluate the performance of the proposed model, we carry out some comparative experiments between FWSVM-FWKNN and SVM-KNN, running on the same training set and testing set of SSE Composite Index. In the experiments, libsvm-3.20 is adopted to solve FWSVM and SVM model. Table 6 lists the comparison results of SSE Composite Index between FWSVM and SVM on the same testing set for predicting two class label profit or loss in the short, medium and long term,

Table 9
The comparison results of SZSE COMP SUB IND between FWSVM and SVM in percentage (%) for predicting Profit or Loss.

Method	Short term		Medium term		Long term	
	Time horizon		Time horizon		Time horizon	
	1 day	5 days	10 days	15 days	20 days	30 days
SVM	92.6	93.6	95.0	94.0	94.4	97.4
FWSVM	96.0	96.8	97.6	95.6	95.8	98.4

Table 10
The comparison results of SZSE COMP SUB IND between FWSVM and SVM on AUC scores.

Method	Short term		Medium term		Long term	
	Time horizon		Time horizon		Time horizon	
	1 day	5 days	10 days	15 days	20 days	30 days
SVM	0.9928	0.9950	0.9945	0.9987	0.9810	0.9985
FWSVM	0.9973	0.9990	0.9981	0.9988	0.9955	0.9996

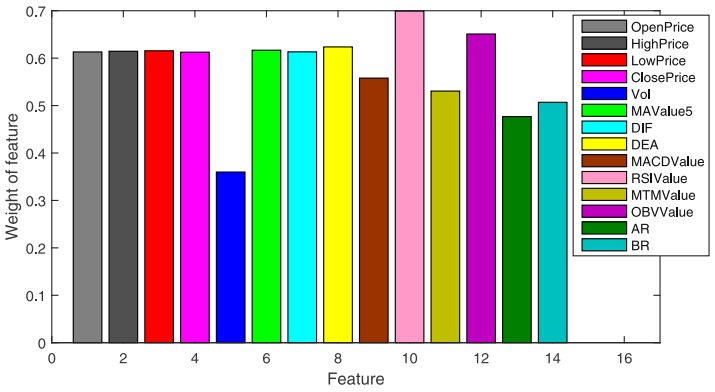
respectively. Table 7 presents the comparison results of SSE Composite Index between FWSVM and SVM on AUC scores. Figs. 7, 8 and 9 show the distribution of the feature weight and the comparison of target prices vs. the predicted stock closing prices of SSE Composite Index in the short, medium and long term, respectively. Table 8 shows the comparison results of MAPE and RMSE between FWSVM-FWKNN and SVM-KNN on SSE Composite Index in the short, medium and long term during the testing.

From Table 6, we can see that FWSVM gets better prediction accuracy than SVM in the short, medium and long term, respectively. To be specific, FWSVM outperforms SVM with 1.8% for 1 day ahead prediction, 1.6% for 5 days ahead prediction, 1.0% for 10 days ahead prediction, 0.6% for 15 days ahead prediction, 0.4% for 20 days ahead prediction, 0.4% for 30 days ahead prediction. The results indicate that FWSVM approach is useful for predicting profit or loss. Above all, FWSVM outperforms the most for 1 day ahead prediction comparing to SVM .

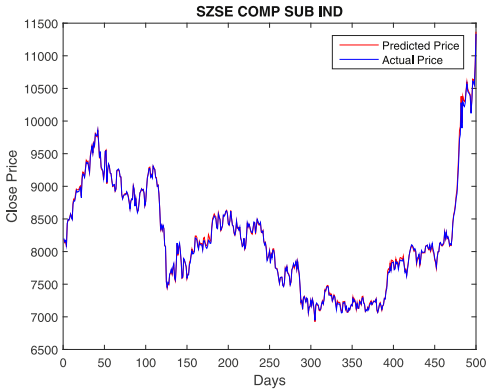
The program used to calculate the AUC scores is available in the web site https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/. The results shown in Table 7 demonstrate that both of these two methods, the classical SVM and FWSVM, are good predictor on SSE Composite Index in the short, medium and long term during testing, but the performance of later is better than the former.

Figs. 7(b), (d), 8(b), (d), 9(b) and (d) reflect that the stock closing prices predicted by FWSVM-FWKNN are almost the same as the actual prices. That means that our proposed model can forecast the stock closing prices well in the short, medium and long term respectively.

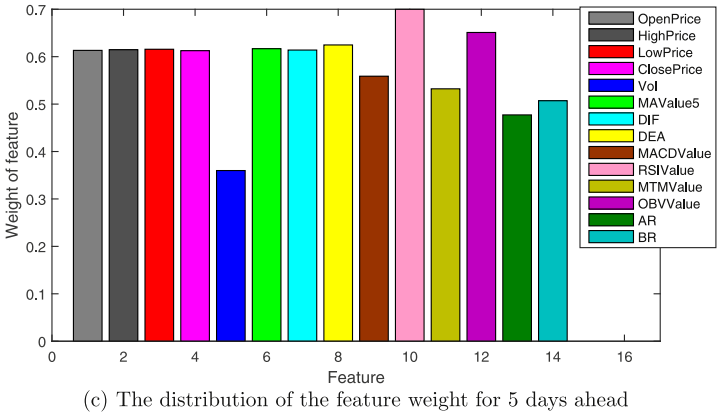
From Table 8, RMSE of FWSVM-FWKNN is smaller than SVM-KNN on SSE Composite Index in the short, medium and long term respectively, which illustrates FWSVM-FWKNN has better performance than SVM-KNN for these forecasting periods. In cases of 10 days and 20 days ahead prediction, MAPE of FWSVM-FWKNN is



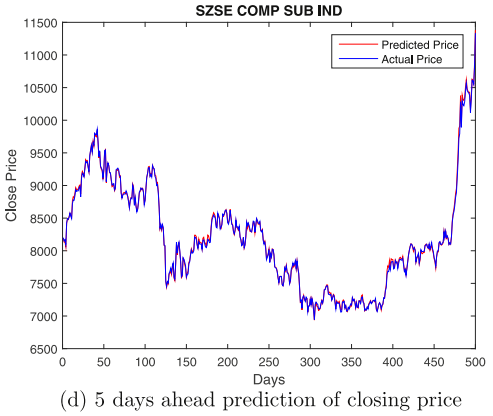
(a) The distribution of the feature weight for 1 day ahead



(b) 1 day ahead prediction of closing price

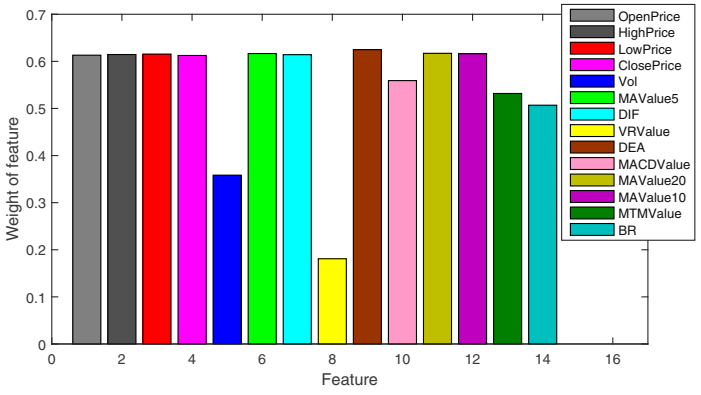


(c) The distribution of the feature weight for 5 days ahead

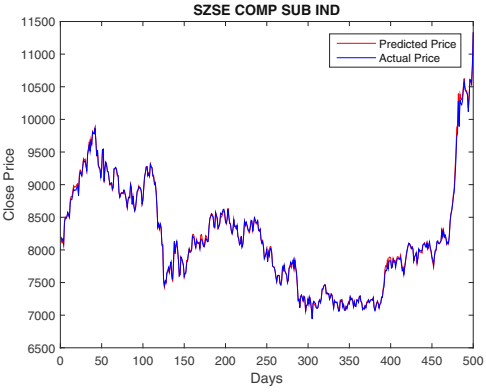


(d) 5 days ahead prediction of closing price

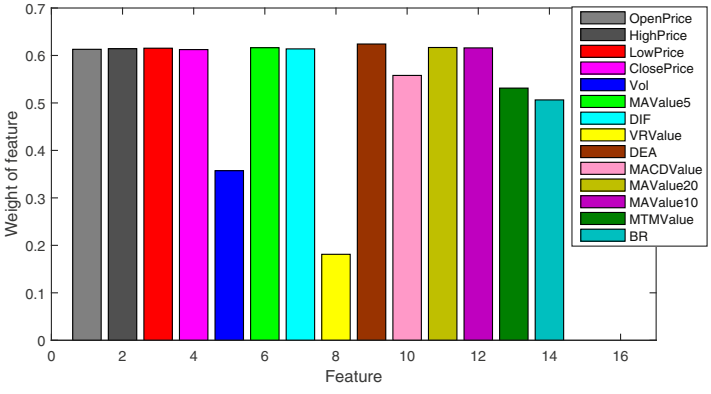
Fig. 13. Short term prediction (SZSE COMP SUB IND).



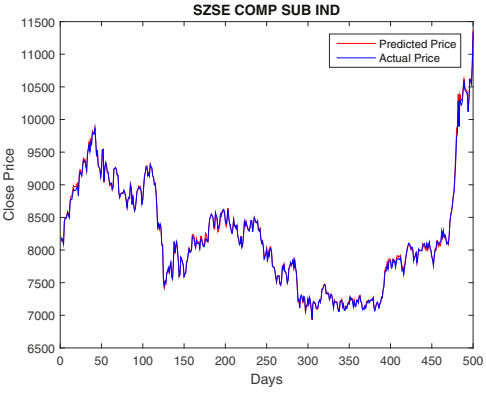
(a) The distribution of the feature weight for 10 days ahead



(b) 10 days ahead prediction of closing price

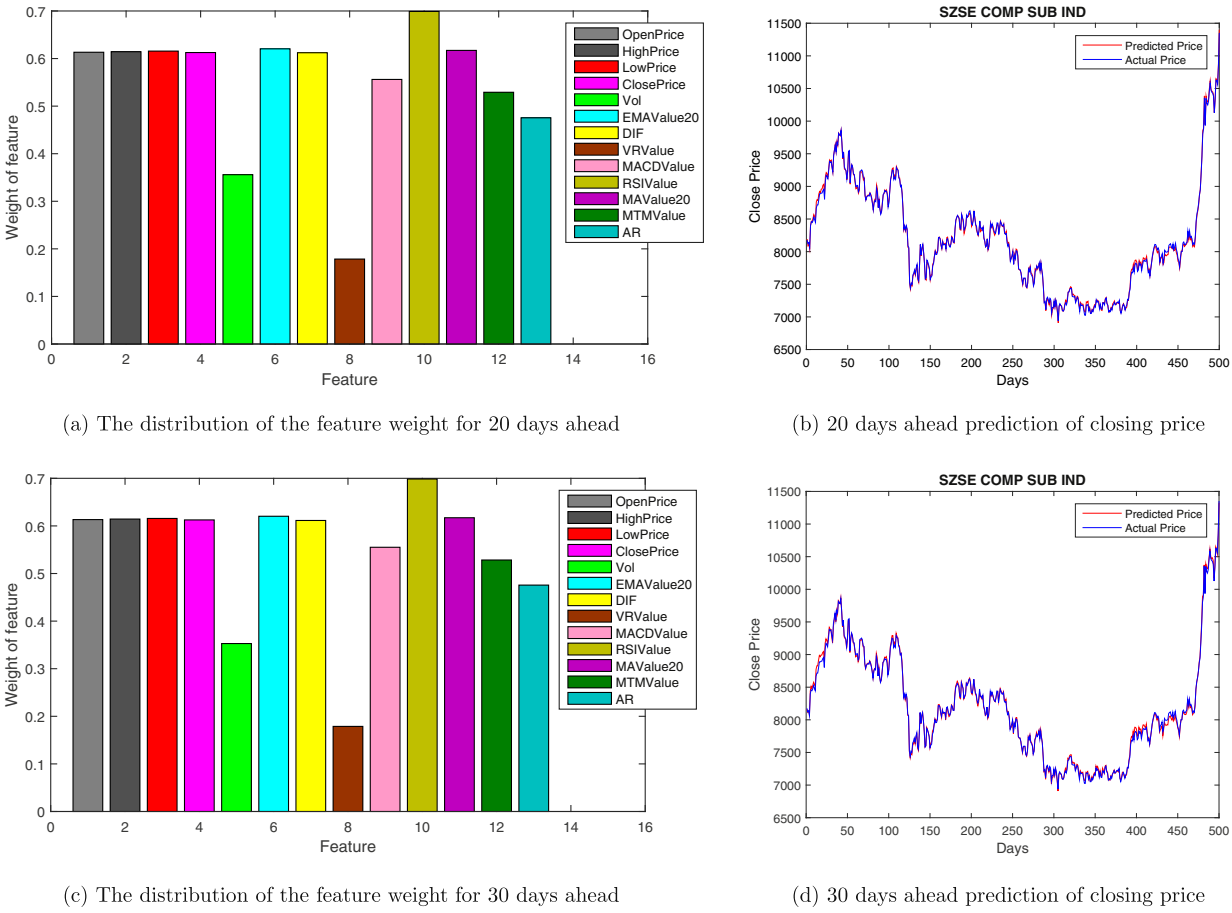


(c) The distribution of the feature weight for 15 days ahead



(d) 15 days ahead prediction of closing price

Fig. 14. Medium term prediction (SZSE COMP SUB IND).



(a) The distribution of the feature weight for 20 days ahead

(b) 20 days ahead prediction of closing price

(c) The distribution of the feature weight for 30 days ahead

(d) 30 days ahead prediction of closing price

Fig. 15. Long term prediction (SZSE COMP SUB IND).

Table 11
Performance comparison of SZSE COMP SUB IND between FWSVM-FWKNN and SVM-KNN in MAPE and RMSE.

	Method	Short term		Medium term		Long term	
		Time horizon		Time horizon		Time horizon	
		1 day	5 days	10 days	15 days	20 days	30 days
MAPE	SVM-KNN	0.20	0.22	0.22	0.24	0.27 0.26	0.31 0.30
	FWSVM-FWKNN	0.18	0.22	0.21	0.23		
RMSE	SVM-KNN	0.0051	0.0056	0.0061	0.0062	0.0072	0.0105
	FWSVM-FWKNN	0.0050	0.0056	0.0061	0.0060	0.0070	0.0105

the same as SVM-KNN and RMSE is smaller than SVM-KNN on SSE Composite Index, which indicates that FWSVM-FWKNN and SVM-KNN can learn the same valuable knowledge hidden in the history data for 10 days and 20 days ahead. RMSE of FWSVM-FWKNN for 20 days ahead is 0.0018 smaller than that for 15 days ahead. It is in a very small range, so this change is reasonable.

4.3.2. Experiments on SZSE COMP SUB IND

In this section, to make the performance of the proposed model more persuasive, we change SSE Composite Index to SZSE COMP SUB IND and repeat the comparative experiments between FWSVM-FWKNN and SVM-KNN again. Table 9 lists the comparison results between FWSVM and SVM on the same testing set of SZSE COMP SUB IND for predicting two class label profit or loss in the short, medium and long term respectively. Table 10 presents the comparison results of SZSE COMP SUB IND between FWSVM and SVM on AUC scores. Figs. 13, 14 and 15 show the weight values assigned to each feature and the comparison of target prices vs. the predicted stock closing prices of SZSE COMP SUB IND in the short, medium and long term respectively. Table 11 shows the comparison results of MAPE and RMSE between FWSVM-FWKNN and SVM-KNN on SZSE COMP SUB IND in the short, medium and long term during the testing.

From Table 9, we can see that the accuracy achieved by FWSVM is better than that achieved by SVM for predicting two class label (profit or loss) in the short, medium and long term respectively. FWSVM outperforms SVM with 3.4% for 1 day ahead prediction, 3.2% for 5 days, 2.6% for 10 days, 1.6% for 15 days, 1.4% for 20 days and 1.0% for 30 days. Through this comparison, FWSVM has an obvious improvement for 1 day ahead prediction. The results are similar to the former experiment in Section 4.3.1: the proposed FWSVM shows a better performance than the classical SVM.

It can be seen from the Table 10 that both FWSVM and SVM are good predictor on SZSE COMP SUB IND in the short, medium and long term respectively and FWSVM is better than SVM.

From Figs. 13(b), (d), 14(b), (d), 15(b), (d), it is hard to distinguish the predicted stock closing prices curve utilizing FWSVM-FWKNN from the actual curve.

From Table 11, we can see that FWSVM-FWKNN has the smallest MAPE and RMSE on SZSE COMP SUB IND for 1 day ahead among 1 day, 5 days, 10 days, 15 days, 20 days and 30 days ahead. Therefore, FWSVM-FWKNN have the best performance for 1 day ahead. MAPE of FWSVM-FWKNN is smaller than that achieved by SVM-KNN in the medium and long term respectively, which means that FWSVM-FWKNN has better performance than SVM-KNN in the medium and long term respectively.

5. Conclusion

In this paper, we propose a hybridized framework composed of FWSVM and FWKNN based on the information gain, and apply this new framework to forecast two Chinese stock market indices such as, SSE Composite Index and SZSE COMP SUB IND in the short, medium and long term respectively. An important characteristic of this method is that the relative importance of each feature is taken into account in the classification in SVM and prediction in KNN, without being dominated by trivial relevant or irrelevant features. This method has been compared with SVM-KNN. The experiment results clearly show that FWSVM-FWKNN stock analysis algorithm where the classification by FWSVM and the prediction by FWKNN, is robust, presenting significant improvement and good prediction capability for Chinese stock market indices over other compared model. However, it is not the only method to structure FWSVM and FWKNN by giving different weight values to the features according to the information gain. For future work, other correlation weighted methods is to be considered.

Acknowledgments

This work is partly supported by National Natural Science Foundation of China (E050604/51075306), The National Science & Technology Pillar Program during the Twelve Five-Year Plan Period (2015BAF10B01).

References

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control*. John Wiley & Sons.

Cao, L.-J., & Tay, F. E. H. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks, 14*, 1506–1518.

Cervelló-Royo, R., Guijarro, F., & Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the djia index with intraday data. *Expert Systems with Applications, 42*, 5963–5975.

Chakravarty, S., & Dash, P. (2012). A pso based integrated functional link net and interval type-2 fuzzy logic system for predicting stock market indices. *Applied Soft Computing, 12*, 931–941.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*, 27.

Chiang, W.-C., Enke, D., Wu, T., & Wang, R. (2016). An adaptive stock index trading decision support system. *Expert Systems with Applications, 59*, 195–207.

Dai, J., & Xu, Q. (2013). Attribute selection based on information gain ratio in fuzzy rough set theory with application to tumor classification. *Applied Soft Computing, 13*, 211–221.

Dash, R., Dash, P. K., & Bisoi, R. (2014). A self adaptive differential harmony search based optimized extreme learning machine for financial time series prediction. *Swarm and Evolutionary Computation, 19*, 25–42.

Fletcher, R. (2013). *Practical methods of optimization*. John Wiley & Sons.

Fu-Yuan, H. (2008). Forecasting stock price using a genetic fuzzy neural network. IEEE. Computer Science and Information Technology, 2008. ICCSIT'08. International Conference on, 549–552.

Göçken, M., Özçalıcı, M., Boru, A., & Dosdoğru, A. T. (2016). Integrating metaheuristics and artificial neural networks for improved stock price prediction. *Expert Systems with Applications, 44*, 320–331.

Gong, X., Si, Y.-W., Fong, S., & Biuk-Aghai, R. P. (2016). Financial time series pattern matching with extended ucr suite and support vector machine. *Expert Systems with Applications, 55*, 284–296.

Guegan, D. (2009). Chaos in economics and finance. *Annual Reviews in Control, 33*, 89–93.

Hassan, M. R., Nath, B., & Kirley, M. (2007). A fusion model of hmm, ann and ga for stock market forecasting. *Expert systems with Applications, 33*, 171–180.

Hsu, M.-W., Lessmann, S., Sung, M.-C., Ma, T., & Johnson, J. E. (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems with Applications, 61*, 215–234.

Hu, Y., Feng, B., Zhang, X., Ngai, E., & Liu, M. (2015). Stock trading rule discovery with an evolutionary trend following model. *Expert Systems with Applications, 42*, 212–222.

Hu, Z., Zhu, J., & Tse, K. (2013). Stocks market prediction using support vector machine. IEEE. 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering, 115–118, 2.

Huang, C.-F. (2012). A hybrid stock selection model using genetic algorithms and support vector regression. *Applied Soft Computing, 12*, 807–818.

Huang, S.-C., & Wu, T.-K. (2008). Integrating ga-based time-scale feature extractions with svms for stock index forecasting. *Expert Systems with Applications, 35*, 2080–2088.

Ives, M. C., & Scandol, J. P. (2007). A bayesian analysis of nsw eastern king prawn stocks (melicertus plebejus) using multiple model structures. *Fisheries Research, 84*, 314–327.

Kim, Y., & Enke, D. (2016). Developing a rule change trading system for the futures market using rough set analysis. *Expert Systems with Applications, 59*, 165–173.

Kodogiannis, V., & Lolis, A. (2002). Forecasting financial time series using neural network and fuzzy system-based techniques. *Neural Computing & Applications, 11*, 90–102.

Leigh, W., Hightower, R., & Modani, N. (2005). Forecasting the new york stock exchange composite index with past price and interest rate on condition of volume spike. *Expert Systems with Applications, 28*, 1–8.

Li, H., Sun, J., & Sun, B.-L. (2009). Financial distress prediction based on or-cbr in the principle of k-nearest neighbors. *Expert Systems with Applications, 36*, 643–659.

Lin, Y., Guo, H., & Hu, J. (2013). An svm-based approach for stock market trend prediction. IEEE. Neural Networks (IJCNN), The 2013 International Joint Conference on, 1–7.

Majhi, R., Panda, G., & Sahoo, G. (2009). Development and performance evaluation of flann based model for forecasting of stock markets. *Expert Systems with Applications, 36*, 6800–6808.

Miao, J., Wang, P., & Xu, Z. (2015). A bayesian dynamic stochastic general equilibrium model of stock market bubbles and business cycles. *Quantitative Economics, 6*, 599–635.

Nair, B. B., Mohandas, V., & Sakthivel, N. (2010). A decision treeâ;;rough set hybrid system for stock market trend prediction. *International Journal of Computer Applications, 6*, 1–6.

Nayak, R. K., Mishra, D., & Rath, A. K. (2015). A Naïve svm-knn based stock market trend reversal analysis for indian benchmark indices. *Applied Soft Computing, 35*, 670–680.

Neely, C. J., Rapach, D. E., Tu, J., & Zhou, G. (2014). Forecasting the equity risk premium: The role of technical indicators. *Management Science, 60*, 1772–1791.

Nikfarjam, A., Emadzadeh, E., & Muthaiyah, S. (2010). Text mining approaches for stock market prediction. IEEE. Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on, 256–260, 4.

Podsiadlo, M., & Rybinski, H. (2016). Financial time series forecasting using rough sets with time-weighted rule voting. *Expert Systems with Applications, 66*, 219–233.

Rahman, H. F., Sarker, R., & Essam, D. (2015). A genetic algorithm for permutation flow shop scheduling under make to stock production system. *Computers & Industrial Engineering, 90*, 12–24.

Shen, W., Zhang, Y., & Ma, X. (2009). Stock return forecast with ls-svm and particle swarm optimization. IEEE. Business Intelligence and Financial Engineering, 2009. BIFE'09. International Conference on, 143–147.

Sheng-feng, W. T.-h. T., & Hou-kuan, H. (2009). Feature weighted support vector machine [j]. *Journal of Electronics & Information Technology, 3*, 003.

Sorensen, E. H., Miller, K. L., & Ooi, C. K. (2000). The decision tree approach to stock selection. *The Journal of Portfolio Management, 27*, 42–52.

Su, Z., & Peterman, R. M. (2012). Performance of a bayesian state-space model of semelparous species for stock-recruitment data subject to measurement error. *Ecological Modelling, 224*, 76–89.

Teixeira, L. A., & De Oliveira, A. L. I. (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert Systems with Applications, 37*, 6885–6890.

Thirunavukarasu, P. (2009). Estimation of return on investment in share market through ann. *Global Journal of Finance and Management, 1*, 113–122.

Ticknor, J. L. (2013). A bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications, 40*, 5501–5506.

Vapnik, V. N., & Vapnik, V. (1998). *Statistical learning theory*: 1. Wiley New York.

Wang, D., Liu, X., & Wang, M. (2013). A dt-svm strategy for stock futures prediction with big data. IEEE. Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on, 1005–1012.

Wang, L., Wang, Z., Zhao, S., & Tan, S. (2015). Stock market trend prediction using dynamical bayesian factor graph. *Expert Systems with Applications, 42*, 6267–6275.

Wang, L.-J., Guan, S.-Y., Wang, X.-L., & Wang, X.-Z. (2006). Fuzzy c mean algorithm based on feature weights. *Chinese Journal of Computers, 29*, 1797.

Wang, X.-Y., & Wang, Z.-O. (2002). Stock market time series data mining based on regularized neural network and rough set. IEEE. Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on, 315–318, 1.

Wang, Y.-F. (2003). Mining stock price using fuzzy rough set system. *Expert Systems with Applications, 24*, 13–23.

Wen, Q., Yang, Z., Song, Y., & Jia, P. (2010). Automatic stock decision support system based on box theory and svm algorithm. *Expert Systems with Applications, 37*, 1015–1022.

Wu, M.-C., Lin, S.-Y., & Lin, C.-H. (2006). An effective application of decision tree to stock trading. *Expert Systems with Applications, 31*, 270–274.

Xi, L., Muzhou, H., Lee, M. H., Li, J., Wei, D., Hai, H., & Wu, Y. (2014). A new constructive neural network method for noise processing and its application on stock market prediction. *Applied Soft Computing, 15*, 57–66.

Xi, Y., Peng, H., Qin, Y., Xie, W., & Chen, X. (2015). Bayesian analysis of heavy-tailed market microstructure model and its application in stock markets. *Mathematics and Computers in Simulation, 117*, 141–153.

Yeh, I.-C., Lien, C.-h., & Tsai, Y.-C. (2011). Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications, 38*, 794–803.

Yu, H., Chen, R., & Zhang, G. (2014). A svm stock selection model within pca. *Procedia Computer Science, 31*, 406–412.

Żbikowski, K. (2015). Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. *Expert Systems with Applications, 42*, 1797–1805.

Zhang, Y., & Shen, W. (2009). Stock yield forecast based on ls-svm in bayesian inference. IEEE. Future Computer and Communication, 2009. FCC'09. International Conference on, 8–11.

Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications, 67*, 126–139.

Ziegel, E. R. (2002). Analysis of financial time series. *Technometrics, 44*. 408–408.

Yingjun Chen received his BE degree from Hefei University of Technology, China in 2009, and ME degree from Hefei University of Technology, China in 2012. He is currently a PhD candidate of Tongji University, China. His research interests include machine learning, data mining, reinforcement learning and financial time series.

Yongtao Hao received PhD degree in mechanical engineering from Shanghai Jiaotong University in 2000. He is an professor in CAD research center of Tongji University, and Senior Member of Council, Chinese Mechanical Engineering Society (CMES). His research interests include machine learning, data mining, intelligent design method based on CAX integrated knowledge, and virtual reality.