Expert Systems with Applications 38 (2011) 14026–14036

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A hybrid procedure for stock price prediction by integrating self-organizing map and genetic programming

Chih-Ming Hsu *

Department of Business Administration, Minghsin University of Science and Technology, 1 Hsin-Hsing Road, Hsin-Fong, Hsinchu 304, Taiwan, ROC

ARTICLE INFO

Keywords:
Stock price prediction
Self-organizing map
Genetic programming

ABSTRACT

Stock price prediction is a very important financial topic, and is considered a challenging task and worthy of the considerable attention received from both researchers and practitioners. Stock price series have properties of high volatility, complexity, dynamics and turbulence, thus the implicit relationship between the stock price and predictors is quite dynamic. Hence, it is difficult to tackle the stock price prediction problems effectively by using only single soft computing technique. This study hybridizes a self-organizing map (SOM) neural network and genetic programming (GP) to develop an integrated procedure, namely, the SOM-GP procedure, in order to resolve problems inherent in stock price predictions. The SOM neural network is utilized to divide the sample data into several clusters, in such a manner that the objects within each cluster possess similar properties to each other, but differ from the objects in other clusters. The GP technique is applied to construct a mathematical prediction model that describes the functional relationship between technical indicators and the closing price of each cluster formed in the SOM neural network. The feasibility and effectiveness of the proposed hybrid SOM-GP prediction procedure are demonstrated through experiments aimed at predicting the finance and insurance sub-index of TAIEX (Taiwan stock exchange capitalization weighted stock index). Experimental results show that the proposed SOM-GP prediction procedure can be considered a feasible and effective tool for stock price predictions, as based on the overall prediction performance indices. Furthermore, it is found that the frequent and alternating rise and fall, as well as the range of daily closing prices during the period, significantly increase the difficulties of predicting.

1. Introduction

Stock price prediction is an important financial subject, which has received considerable attention from researchers in recent years. Stock price prediction is considered a challenging task in consideration of its high volatility, complexity, dynamics, and turbulence. In the past, many attempts have been made to predict stock prices using various methodologies, which can be broadly classified into three categories, namely, fundamental analysis, technical analysis, and traditional time series forecasting. Fundamental analysis examines the basic financial information of a corporation in order to forecast profits, supply, demand, industry strength, management abilities, and other intrinsic matters affecting the market value and growth potential of a stock (Thomsett, 1998). In fundamental analysis, investors believe that the fundamentals include a corporation's financial statements, interim reports, historical financial trends, and any forecasts concerning future growth, sales, profits, etc., should rule the processes of the selection of stocks and timing of sales (Thomsett, 1999). However, technical analysis studies the stock prices and related issues, including analysis of recent and historical price trends, cycles and factors beyond the stock price, such as dividend payments, trading volume, index trends, industry group trends and popularity, and volatility of a stock (Thomsett, 1999). Technical analysis, rather than relying solely upon historical financial information, analysts will surmise upon recent trends in stock price changes, prices and earnings relationships, the activity volume of a particular stock or industry, and other similar indicators in order to determine changes in stocks, and in the market itself (Thomsett, 1999). In addition, traditional time series forecasting techniques, such as autoregressive integrated moving average (ARIMA) (Box & Jenkins, 1970), generalized autoregressive conditional heteroskedasticity (GARCH) (Bollerslev, 1986), and multivariate regression have been applied to the prediction of stock price movements. In recent years, data mining/computational intelligence techniques have become another important approach to predict stock prices. For example, Kim and Han (2000) utilized genetic algorithms (GAs) to discretize features and determine the connection weights of artificial neural networks (ANNs), thus, predicting the stock price index. Experiments conducted on the daily Korea stock price index (KOSPI) showed that, their proposed approach outperformed the linear

* Tel.: +886 3 5593142x3581; fax: +886 3 5593142x3573.
E-mail address: cmhsu@must.edu.tw

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.210

transformation functions of both a backpropagation neural network (BPLT) and a linear transformation with ANN, as trained by GA (GALT). Kim (2003) applied a support vector machine (SVM) to predict the stock price index, and the feasibility of applying SVM to financial forecasting was examined through comparisons with a backpropagation neural network (BPNN) and case-based reasoning (CBR). The experimental results of the daily Korea stock price index (KOSPI) investigation showed that, SVM provides a promising alternative for financial time series forecasting; moreover, it outperforms both BPNN and CBR approaches. Pai and Lin (2005) proposed a hybrid methodology through exploitation of the strengths of the autoregressive integrated moving average (ARIMA) and support vector machine (SVM) in order to forecast stock prices. The performance of the proposed model is evaluated by testing real data sets of ten stocks, and adequate results are obtained. Tsang et al. (2007) presented a stock buying/selling alert system using a feed-forward backpropagation neural network, called NN5. The system is tested with data from The Hong Kong and Shanghai Banking Corporation (HSBC) Holdings stock, located in Hong Kong, and achieved an overall hit rate of over 70%. Chang and Liu (2008) presented a Takagi–Sugeno–Kang (TSK) type fuzzy rule based system by applying a linear combination consequence of the significant technical index in order to predict stock prices. Their proposed approach was tested on the Taiwan Stock Exchange (TSE) and MediaTek Inc., and the experimental results outperformed other methodologies, such as a back-propagation neural network and multiple regression analysis. Ince and Trafalis (2008) assumed that the future value of a stock price depends on its financial indicators, although there is no existing parametric model able to explain the relationship coming from the technical analysis. Hence, they proposed two nonparametric data driven models, a support vector regression (SVR) and a multi-layer perceptron (MLP), for short term stock price predictions based on technical indicators. The experiments were conducted on the daily stock prices of ten companies traded on the NASDAQ, and comparison results indicated that the SVR approach outperformed the MLP networks in short term predictions, in terms of the mean square error. Huang and Tsai (2009) proposed a hybrid procedure using support vector regression (SVR), self-organizing feature map (SOFM), and filter-based feature selection in order to predict the stock market price index. Their proposed model was demonstrated through a case study of predictions of the next day's price index for Taiwan index futures (FITX), and the experiment results showed that the proposed approach can improve prediction accuracy and reduce the training time over the traditional single SVR model. Lai, Fan, Huang, and Chang (2009) proposed a decision-making system that integrates a data clustering technique, a fuzzy decision tree, and genetic algorithms in order to forecast stock price tendencies. Three particular stocks in the Taiwan Stock Exchange Corporation (TSEC) were selected to test the effectiveness of their proposed system, which yielded the best performance of an 82% average hit rate, in comparison with other approaches. Liang, Zhang, Xao, and Chen (2009) presented a nonparametric methodology based on neural networks (NNs) and support vector regression (SVR) to forecast option prices. In their study, the improved conventional option pricing methods were modified to forecast the option prices, and then, the NN and SVR were further employed to decrease the forecasting errors of the parametric methods. The proposed approach was demonstrated by experimental studies upon data taken from the Hong Kong options market, which results showed that the NN and SVR approaches can significantly shrink the average forecast errors, thus, improving forecasting accuracy. Lee (2009) developed a model based on a support vector machine (SVM) with a hybrid feature selection, namely, *F*-score and supported sequential forward search (F_SSFS), to predict the trends of stock markets. The experiments of predicting the NASDAQ index direction were used to illustrate their proposed method, and suitable results were obtained. In addition, comparisons with information gain, symmetrical uncertainty, and correlation-based feature selection methods all indicated that their proposed model could yield the highest levels of accurate and generalized performances. Yu, Chen, Wang, and Lai (2009) presented an evolving least squares support vector machine (LSSVM) learning paradigm, with a mixed kernel based on genetic algorithms (GAs), in order to predict the trends of stock markets. The GAs were used to select the input features and optimize parameters of LSSVM. The LSSVM approach was illustrated through testing the S&P 500 index, the Dow Jones Industrial Average (DJIA) index, and New York Stock Exchange (NYSE) index, and experimental results revealed that their proposed learning paradigm was more efficient than other parameter optimization methods, and outperformed all other forecasting models in terms of the hit ratio. Zhang, An, Tang, and Hong (2009) proposed a type-2 fuzzy rule based expert system that applied technical and fundamental indices as the input variables for the analysis of stock prices. Their proposed model was tested on the stock price predictions of an automotive manufactory in Asia, and successful results were obtained.

In this study, an integrated approach based on a self-organizing map (SOM) neural network and genetic programming (GP), namely, the SOM-GP procedure, is proposed for predicting stock prices. The remainder of this paper is organized as follows: In Section 2, SOM and GP are discussed. The proposed integrated approach is presented in Section 3. Section 4 evaluates the feasibility and effectiveness of the proposed approach by a case study of predicting the finance and insurance sub-index of TAIEX. Finally, Section 5 concludes the paper.

2. Self-organizing map and genetic programming

2.1. Self-organizing map

The self-organizing map (SOM) was first introduced by Kohonen (1989), as an unsupervised and competitive learning neural network able to map a high-dimensional input data space into a lower-dimensional (typically one- or two-dimensional) space. The end-product is called a feature map able to preserve the most important topological relationships of the input data. The typical SOM consists of two layers, as shown in Fig. 1, where the input

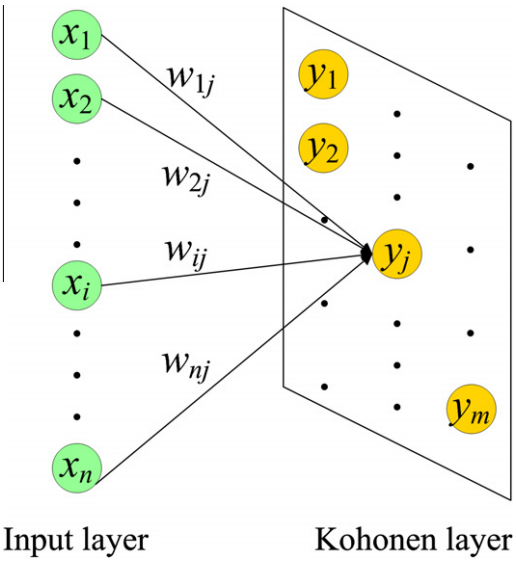


Fig. 1. Typical SOM topology consists of two layers.

layer is fully connected to a two-dimensional Kohonen layer, and none of the neurons are connected in the Kohonen layer. Each neuron in the Kohonen layer represents a cluster, which weight vector serves as an exemplar of the input patterns associated with only this cluster. The self-organizing process chooses a neuron whose weight vector matches the input pattern most closely (usually evaluated by the Euclidean distance) as the winner. The winner and its neighboring neurons (based on the activation zone for each neuron) would then update their weights. By following the architecture and algorithm for the SOM neural network, input data can be clustered into a certain number (i.e. the total number of neurons in the Kohonen layer) of clusters. Assuming that there are a set of continuous-valued input patterns of $\mathbf{x} = (x_1, x_2, \ldots, x_i, \ldots, x_n)$ and m clustered neurons within the feature map; the weight vector associated with neuron j in the Kohonen layer is represented by $\mathbf{w}_j = (w_{1j}, w_{2j}, \ldots, w_{ij}, \ldots, w_{nj})$; and the neighborhood function used to control the relaxation process is denoted by $h_{j'j}$ (where j' and j are the subscripts of the neurons in the Kohonen layer). The training steps include competitive and weight adjustment processes, described as follows (Fausett, 1994; Kohonen, 1995):

Step 0: Initialize weights $\mathbf{w}_j$ and neighborhood functions $h_{j'j}$; then set the radius of the topological neighborhood R, and learning rate α
Step 1: If stopping criterion is not fulfilled, repeat Steps 2–8.
Step 2: For each input vector $\mathbf{x}$, complete Steps 3–5.
Step 3: For each cluster neuron j, compute

$$D(j) = \|\mathbf{x} - \mathbf{w}_j\|.$$

Step 4: Find index c such that $D(c)$ is the minimum.
Step 5: For all neurons j, within the topological neighborhood of the radius R of neuron c:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)h_{cj}(t)[\mathbf{x} - \mathbf{w}_j(t)]$$

where, t is a discrete-time coordinate.
Step 6: Update the learning rate α and neighborhood function $h_{j'j}$.
Step 7: Reduce the radius of the topological neighborhood R at the pre-specified times.
Step 8: Test stopping criterion.

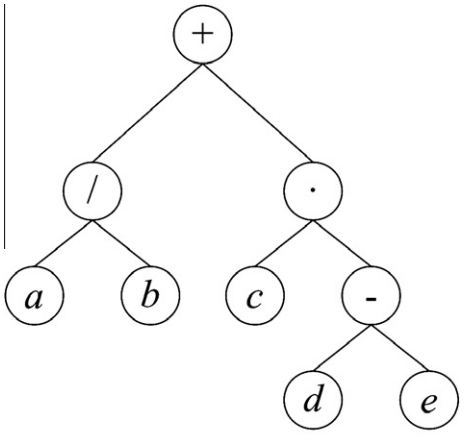
Notably, the learning rate α and radius of topological neighborhood R decrease as the clustering process progresses. The neighborhood function $h_{j'j}$ is a smoothing kernel function defined over the lattice, that is decreasing monotonically in time. There are two frequent choices for $h_{j'j}$ in literature (Kohonen, 1995). The simpler of the two refers to a neighborhood set of array points around winner c, where the neighborhood function is defined as:

$$h_{cj}(t) = \begin{cases} 1, & \text{if neuron } j \text{ lies within a radius } R \text{ of the winning neuron } c, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Another widely used smoother Gaussian neighborhood function centered at the winning neuron c is defined by:

$$h_{cj}(t) = \exp\left(-\frac{\|\mathbf{r}_c - \mathbf{r}_j\|^2}{2\sigma^2(t)}\right), \quad (2)$$

where $\mathbf{r}_c$ and $\mathbf{r}_j$ are the location vectors of neurons c and j, respectively, within the Kohonen layer; the parameter $\sigma(t)$ is a monotonically decreasing function of time that is used to define the width of the kernel. In addition, the performance of SOM is not sensitive to the exact shape of the neighborhoods, and rectangular and hexagonal neighborhoods are suggested by Kohonen (1995) for efficient implementation.



$$(a/b)+c\cdot(d-e)$$

Fig. 2. Tree-based representation of an individual in GP.

SOM has attracted substantial research interest from a wide range of applications. For example, adequate results were obtained through SOM in literature (Huang & Tsai, 2009; Lin & Wu, 2009; Szczurowska, Kuniszyk-Jozkowiak, & Smolka, 2009; Zhang et al., 2009).

2.2. Genetic programming

Genetic programming (GP), as developed by Koza (1992), is an evolutionary approach that extends genetic algorithms (GAs) (Holland, 1975) to the area of computer programs. GP can automatically create computer programs to solve a user-defined problem through iterative executions of evolutionary procedures. The evolving individuals in GP are themselves computer programs, rather than fixed-length strings consisting of numbers, alphabetic letters, or symbols. In GP, the representation of an individual can be viewed as a tree-based structure composed of terminal and function sets, as shown in Fig. 2. The terminal set defines the terminal elements available for each branch of the to-be-evolved computer program, and includes the independent variables of the problem, zero-argument functions, random constants, etc. The function set is a set of primitive functions available to each branch of the to-be-evolved computer program, e.g. addition, square root, multiplication, sine, etc. Like other evolutionary algorithms, a fitness function is defined and used to explicitly or implicitly measure the fitness (adaptability) of individuals in the population. It specifies a desired goal in the search for GP. Furthermore, in order to apply basic genetic programming, users must specify parameters and set the termination criterion. The parameters that control the generation runs of the GP include population size, maximum size of programs, crossover rate, mutation rate, etc. The termination criterion determines the time required before stopping the evolutionary procedures in GP, and may include the maximum number of generations to be run, the fitness values of the best-of-generation individuals for numerous successive generations reaching a plateau, or if a success of the run is predicated. The general steps of GP are briefly described, as follows (Ciglarič & Kidrič, 2006; Koza, Streeter, & Keane, 2008; Koza et al., 2005):

Step 1: Creating an initial population.

The first step creates an initial population (generation 0) of individual computer programs (typically random), which are composed of functions and terminals appropriate to the problem. In

general, the initial individuals are generated subject to a pre-specified maximum size, and are of different sizes and different shapes.

Step 2: Evaluating individuals.

Each program in the population is executed and measured in terms of how well it performs the task at hand (this is called the fitness value), by using a pre-defined fitness function.

Step 3: Generating the next generation.

This step first selects programs from the population using a probability based on fitness. The genetic operations, including reproduction, crossover, mutation, and architecture-altering operations are applied to the selected programs. Then, a new population (the next generation) is created by replacing the current population (the now-old generation) with the population of offspring based on a certain strategy, e.g. elitist strategy.

Step 4: Examining the termination criterion.

When the termination criterion is satisfied, the outcome is designated as the final results of the run. Typically, the single best program encountered during the entire run (i.e. the best-so-far individual) is selected as the solution for a specific problem. If the termination criterion cannot be fulfilled, execute Steps 2–4 iteratively.

Genetic programming has been a highly successful technique for solving problems in numerous fields. Various studies have obtained adequate results through GP (Bae et al., 2010; Etemadi, Rostamy, & Dehkordi, 2009; Hwang et al., 2009).

3. Proposed hybrid SOM-GP prediction procedure

In this study, a hybrid approach based on a self-organizing map (SOM) neural network and genetic programming (GP), namely, the SOM-GP procedure, is proposed to predict stock prices. The SOM-GP procedure comprises three stages. In the first stage, the essential historical stock trading data, e.g. opening price, highest price, lowest price, closing price, trade volume, etc. are first collected. Next, the required technical indicators, e.g. moving average (MA), Williams overbought/oversold index (WMS%R), psychological line (PSY), commodity channel index (CCI), etc. used for independent input variables in the stock price prediction model are calculated. The acquired technical indicators, denoted by $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, serve as the sample data in the succeeding SOM clustering steps. To avoid variables with larger numeric ranges from dominating those in smaller numeric ranges, the technical indicators are normalized into a range between -1 and 1, denoted by $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$, according to their corresponding maximum and minimum values. An SOM neural network is followed to divide the sample data $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ into an appropriate number of clusters. The purpose of clustering aims to split the sample data to form several clusters in such a manner that the objects within each cluster are similar to each other and dissimilar to the objects in other clusters. By doing so, the approximate functional model that describes an implicit mathematical relationship between the technical indicators $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, i.e. independent variables, and the closing price in the next day y, i.e. dependent variables, for the sample data in each cluster can be expectantly constructed more easily and precisely. However, it is difficult to determine the most appropriate number of clusters when clustering through SOM. This study introduces an index to measure the performance of clustering. The average distance of all possible paired normalized objects $(\mathbf{x}'_i, \mathbf{x}'_j)$, which belong to different clusters, is first calculated by:

$$D_{between\_clusters} = \sum_i \sum_j \|\mathbf{x}'_i - \mathbf{x}'_j\| / np_{between\_clusters}, \tag{3}$$

where $np_{between/clusters}$ is the total number of all possible paired objects $(\mathbf{x}'_i, \mathbf{x}'_j)$, which belong to different clusters. Similarly, the average distance of all possible paired normalized objects $(\mathbf{x}'_k, \mathbf{x}'_l)$, which are clustered into the same cluster, can be expressed by:

$$D_{within\_clusters} = \sum_k \sum_l \|\mathbf{x}'_k - \mathbf{x}'_l\| / np_{within\_clusters}, \tag{4}$$

where $np_{within/clusters}$ is the total number of all possible objects $(\mathbf{x}'_k, \mathbf{x}'_l)$, which are grouped into the same cluster. Therefore, the clustering efficiency (CE) that is used to evaluate the clustering performance can be defined as:

$$\mathrm{CE} = \frac{D_{between\_clusters}}{D_{within\_clusters}}. \tag{5}$$

By maximizing the value of CE, the optimal number of clusters in SOM clustering can then be determined.

In the second stage, the closing price in the next day y is first normalized into a range between -1 and 1 according to its maximum and minimum values, as denoted by y'. The normalized technical indicators $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$, along with the normalized closing price in the next day y' are then partitioned into training, testing, and validation data, as based on a pre-specified proportion, e.g. 4:1:1. According to the clustering results previously acquired by SOM, the GP algorithm is then applied to the training and testing sample data of each cluster and constructs several prediction models. Based on simultaneously minimizing the mean squared errors (MSEs) regarding the training and testing data, an optimal GP model is selected for each cluster to describe the functional relationship between the normalized technical indicators $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ of each day, and the normalized closing price in the next day y'.

In the third stage, the normalized technical indicators $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ of each day in the validation data are first grouped into a cluster, denoted as cluster c, by inputting $\mathbf{x}'$ into the SOM neural model constructed in Stage 1. Then, the normalized technical indicators $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ of each day in the validation data are fed into the well-trained GP model corresponding to cluster c, as obtained in Stage 2, in order to acquire the predicted normalized closing price in the next day, denoted by $\hat{y}'$. Therefore, the predicted closing price of the next day $\hat{y}$, given the technical indicators of $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ for a certain day then can be obtained through de-normalizing $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ and $\hat{y}'$. Finally, the effectiveness of the proposed SOM-GP prediction procedure is evaluated by using statistical metrics of the root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), which are defined as follows:

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{no} \frac{(y_i - \hat{y}_i)^2}{no}}, \tag{6}$$

$$\mathrm{MAE} = \sum_{i=1}^{no} \frac{|y_i - \hat{y}_i|}{no}, \tag{7}$$

$$\mathrm{MAPE} = \sum_{i=1}^{no} \frac{|y_i - \hat{y}_i|/y_i}{no} \times 100\%, \tag{8}$$

where no is the total number of objects in the validation data.

The proposed hybrid SOM-GP prediction procedure in this study is conceptually illustrated in Fig. 3 and re-stated summarily, as follows:

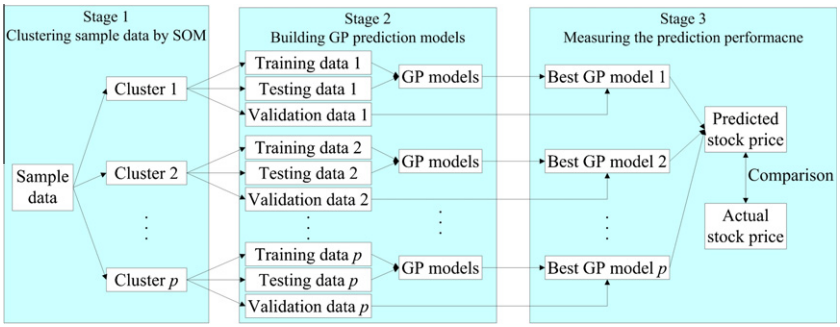


Fig. 3. Proposed hybrid SOM-GP prediction procedure.

Stage 1: Clustering sample data by SOM.

1. Collect the essential historical stock trading data and calculate the required technical indicators.
2. Normalize the technical indicators $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ into the range of $(-1, 1)$, represented by $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$.
3. Apply an SOM neural network to divide the sample data $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ into several clusters, based on the clustering efficiency (CE) in Eq. (5).

Stage 2: Building GP prediction models.

1. Normalize the closing price of the next day y into the range between -1 and 1, as denoted by y'.
2. Partition the sample data $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ and y' into the training, testing, and validation data, as based on a pre-specified proportion.
3. Apply the GP algorithm to construct several prediction models from the training and testing sample data in each cluster generated by SOM in Stage 1.
4. Select an optimal GP model for each cluster to approximate the implicit functional relationship between the normalized technical indicators $(x'_1, x'_2, \ldots, x'_n)$ and the closing price of the next day y', as based on minimizing training and testing MSEs.

Stage 3: Measuring the prediction performance.

1. Input the normalized technical indicators $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ of each day in the validation data into the SOM neural model, as constructed in Stage 1, and group $\mathbf{x}'$ into cluster c.
2. Feed the normalized technical indicators $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ of each day in the validation data into the well-trained GP model corresponding to cluster c, as obtained in Stage 2, in order to acquire the predicted normalized closing price of the next day $\hat{y}'$.
3. De-normalize $(x'_1, x'_2, \ldots, x'_n)$ and $\hat{y}'$ in order to acquire the predicted closing price of the next day $\hat{y}'$, when given the technical indicators $(x_1, x_2, \ldots, x_n)$ of a certain day.
4. Evaluate the effectiveness of the proposed SOM-GP prediction procedure, via RMSE, MAE, and MAPE in Eqs. (6)–(8).

4. Experiments

4.1. Experimental data

To demonstrate the feasibility and effectiveness of the proposed hybrid SOM-GP prediction procedure, experiments on predicting the finance and insurance sub-index of TAIEX (Taiwan stock exchange capitalization weighted stock index), called TAIEX-FISI in this study, are conducted. There are two major reasons for selecting the TAIEX-FISI as the research target. First, it is difficult to predict the price of an individual stock because the stock market news and contrived manipulations, which usually cannot be anticipated, have great impact on the individual stock price. Hence, the weighted stock index in a sector, but not an individual stock price, is selected for exploration in this study. Second, the investment market of exchange traded funds (ETFs), futures, and options in Taiwan have flourished and grown in recent years. For example, Fubon Taiwan Finance ETF, Finance Sector Index Futures (TF), and Finance Sector Index Options (TFO), whose underlying index is TAIEX-FISI, achieved trade volumes of 47,018,000 (shares), 1,285,074 (lots), and 927,888 (lots), respectively, in 2008 (http://www.twse.com.tw; http://www.taifex.com.tw). Furthermore, the trade volumes of TF and TFO grew 41.3% and 22.6%, respectively, from 2005 to 2008. The finance and insurance sub-index of TAIEX is therefore selected as the object of experimentation in order to examine the practicability and performance of the proposed hybrid SOM-GP prediction procedure. In this study, the TAIEX-FISI experimental data are collected from the Taiwan Stock Exchange Corporation (TWSE) over a period of approximately thirteen years, from January 4, 1996 to September 18, 2009. A total of 3,540 pairs of daily trading data, including opening price, highest price, lowest price, closing price, and trade volume are the initial sample data.

4.2. Clustering sample data

First, 16 technical indicators are selected as the independent input variables for predicting stock price, according to previous studies of Kim and Han (2000), Kim and Lee (2004), Tsang et al. (2007), Chang and Liu (2008), Ince and Trafalis (2008), Huang and Tsai (2009) and Lai et al. (2009). These technical indicators, which formulas are presented in Appendix A, include 10-day moving average (MA_10), 20-day bias (BIAS_20), moving average convergence/divergence (MACD), 9-day stochastic indicator K (K_9), 9-day stochastic indicator D (D_9), 9-day Williams overbought/oversold index (WMS%R_9), 14-day plus directional indicator (+DI_14), 14-day minus directional indicator (−DI_14), 10-day momentum (MTM_10), 10-day rate of change (ROC_10), 5-day relative strength index (RSI_5), 24-day commodity channel index (CCI_24), 26-day buying/selling momentum indicator (AR_26), 26-day buying/selling willingness indicator (BR_26), 26-day volume ratio (VR_26), and 13-day psychological line (PSY_13). According to the formulas in Appendix A and the initial 3540 pairs of daily trading data, the 16 technical indicators are calculated. Notably, the technical indicators in the first few days from January 4, 1996, are not available due to the definitions of technical indicators. For example, the 10-day moving average (MA_10) can be obtained only for the period after the 10th trading day from January 4, 1996. By considering the above limitation and validity of the technical indicators, a total of 3497 pairs of technical indicator data from March 1, 1996 to September 17, 2009 are used as the sample data for clustering. Next, the above 16 technical indicators are normalized into a range between -1 and 1, according to their

Table 1
Clustering results of SOM.

Number of neurons in the Kohonen layer	3	4	5	6	7	8	9	10
$D_{\text{between-clusters}}$	0.0192	0.0193	0.0194	0.0193	0.0192	0.0190	0.0190	0.0188
$D_{\text{within-clusters}}$	0.0165	0.0155	0.0141	0.0137	0.0132	0.0136	0.0132	0.0137
Clustering efficiency (CE)	1.1636	1.2452	1.3759	1.4088	1.4545	1.3971	1.4394	1.3723

Table 2
Total numbers of sample data in the seven clusters formed by SOM.

Cluster	1	2	3	4	5	6	7	Total
Total number of sample data	397	669	443	358	534	292	804	3497

corresponding maximum and minimum values. The SOM neural network is further designed using NeuralWorks Professional II/Plus (http://www.neuralware.com) to classify the normalized technical indicators into clusters. The SOM consists of sixteen neurons in an input layer, and several neurons arranged in a one-dimensional Kohonen layer. The initial weight vector of each neuron in the Kohonen layer is randomly set, and the total number of learning iterations for the Kohonen layer is set as 104,910 (30 times the total number of the sample data, i.e. 30×3497). The learning rate is initially set as 0.06, and is reduced by half at 52,454 and 78,681 learning iterations. The simple neighborhood function, as shown in Eq. (1), is applied to control the relaxation process when updating weights. In addition, the size of the topological neighborhood is initially set as 7, and is reduced to 5 and 3 at 52,454 and 78,681 learning iterations, respectively. Experiments are conducted by setting the total number of neurons in the one-dimensional Kohonen layer between 3 and 10, and the results are summarized in Table 1. Based on maximized clustering efficiency (CE), an SOM neural model with seven neurons in the Kohonen layer is selected to group the normalized technical indicators into seven clusters, as summarized in Table 2.

4.3. Building GP prediction models

The closing prices from March 2, 1996 to September 18, 2009 are first normalized into a range between −1 and 1. The normalized technical indicators from March 1, 1996 to September 17, 2009, along with the normalized closing price of the next day, i.e. from March 2, 1996 to September 18, 2009, are then partitioned into training, testing, and validation sample data groups, based on the proportion of 4:1:1, as shown in Table 3. Notably, the sample data, which consist of normalized technical indicators along with the normalized closing price of the next day, are split into 10 subsets, which is achieved by slicing periods of time to ensure that the training, testing, and validation sample data cover the entire period of research. In this manner, it is believed that the GP models, which are constructed later, can more accurately estimate the relationship between the normalized technical indicators and the closing price of the next day. Table 4 summarizes the distribution of training, testing, and validation sample data in the seven clusters previously formed by SOM. Next, a genetic programming (GP) technique is applied to the training and testing sample data of each cluster in order to establish the estimated mathematical function between the independent input variables (the normalized technical indicators) and the dependent output variable (the normalized closing price in the next day). Here, the GP system Discipulus 4.0 (http://www.rmltech.com), with its default parameter settings, is employed, while the fitness of an individual (program) is evaluated through mean squared error (MSE). For each dataset, a GP algorithm is implemented for 5 runs, and Table 5 summarizes the results. Based on the training and testing MSEs, the 4th, 3rd, 3rd, 2nd, 5th, 2nd, and 4th models from clusters 1 through 7 of Table 5, described as GP_MODEL$_1$ through GP_MODEL$_7$, are selected to predict the normalized closing price of the next day, when given the normalized technical indicators of a certain day that belong to clusters 1 through 7, as formed by SOM, respectively.

Table 3
Dividing TAIEX-FISI sample data into 10 subsets.

Dataset	Training period	Testing period	Validation period	Dataset size
1	1996/03/01–1996/12/24	1996/12/26–1997/03/15	1997/03/17–1997/05/29	360
2	1997/05/30–1998/04/09	1998/04/10–1998/06/26	1998/06/29–1998/09/11	360
3	1998/09/14–1999/08/06	1999/08/07–1999/10/27	1999/10/28–2000/01/18	360
4	2000/01/19–2000/12/05	2000/12/06–2001/03/08	2001/03/09–2001/06/04	360
5	2001/06/05–2002/05/30	2002/05/31–2002/08/22	2002/08/23–2002/11/18	360
6	2002/11/19–2003/11/05	2003/11/06–2004/02/06	2004/02/09–2004/04/30	360
7	2004/05/03–2005/04/20	2005/04/21–2005/07/14	2005/07/15–2005/10/12	360
8	2005/10/13–2006/09/27	2006/09/28–2006/12/22	2006/12/25–2007/03/28	360
9	2007/03/29–2008/03/18	2008/03/19–2008/06/12	2008/06/13–2008/09/05	360
10	2008/09/08–2009/05/15	2009/05/18–2009/07/17	2009/07/20–2009/09/17	257
Total number of sample data	2,330	584	583	3497

Table 4
The distribution of training, testing, and validation sample data in the seven clusters formed by SOM.

Cluster	1	2	3	4	5	6	7	Total
Total number of training data	282	462	300	258	366	194	468	2330
Total number of testing data	71	115	75	65	92	49	117	584
Total number of validation data	44	92	68	35	76	49	219	583
Total number of sample data	397	669	443	358	534	292	804	3497

4.4. Measuring the prediction performance

First, the normalized technical indicators of each day lying within the validation period, as shown in Table 3, are fed into the SOM neural model, as constructed in Section 4.2 and clustered as cluster c. The normalized technical indicators then act as independent input variables of the well-trained GP model corresponding to cluster c, i.e. GP_MODEL$_c$, as obtained in Section 4.3, in order to acquire the predicted normalized closing price of the next day.

Table 5
Implementation results of the genetic programming algorithm.

Cluster	Model No.	Training MSE	Testing MSE	Training R^2	Testing R^2
1	1	0.000938	0.000710	0.99195	0.99346
	2	0.000822	0.000884	0.99282	0.99158
	3	0.000884	0.000789	0.99222	0.99265
	4	**0.000791**	**0.000770**	**0.99318**	**0.99272**
	5	0.000702	0.000913	0.99387	0.99144
2	1	0.000519	0.000658	0.99138	0.99141
	2	0.000480	0.000660	0.99382	0.99124
	3	**0.000499**	**0.000641**	**0.99346**	**0.99152**
	4	0.000516	0.000691	0.99318	0.99076
	5	0.000549	0.000615	0.99291	0.99182
3	1	0.000552	0.000329	0.99338	0.99667
	2	0.000525	0.000398	0.99368	0.99579
	3	**0.000544**	**0.000318**	**0.99356**	**0.99661**
	4	0.000540	0.000340	0.99349	0.99653
	5	0.000530	0.000421	0.99372	0.99567
4	1	0.000876	0.000980	0.98008	0.98246
	2	**0.000833**	**0.000921**	**0.98191**	**0.98358**
	3	0.000738	0.001092	0.98399	0.98105
	4	0.000836	0.000962	0.98198	0.98273
	5	0.000706	0.001144	0.98440	0.97988
5	1	0.000466	0.000463	0.99340	0.99271
	2	0.000443	0.000528	0.99351	0.99157
	3	0.000425	0.000519	0.99372	0.99199
	4	0.000504	0.000463	0.99278	0.99252
	5	**0.000411**	**0.000482**	**0.99394**	**0.99254**
6	1	0.001290	0.000478	0.97780	0.99026
	2	**0.001077**	**0.000464**	**0.98149**	**0.99096**
	3	0.001082	0.000577	0.98094	0.98856
	4	0.001166	0.000555	0.97954	0.98866
	5	0.001177	0.000543	0.97948	0.98919
7	1	0.000713	0.000492	0.98847	0.99275
	2	0.000711	0.000486	0.98858	0.99261
	3	0.000717	0.000487	0.98844	0.99275
	4	**0.000703**	**0.000428**	**0.98860**	**0.99356**
	5	0.000706	0.000473	0.98882	0.99312

By de-normalizing the predicted normalized closing price, the predicted closing price of the next day can be obtained. Fig. 4 illustrates the predicted and actual values of the closing prices during the period from March 17, 1997 to May 29, 1997. To evaluate the overall performance of the proposed SOM-GP prediction procedure, the statistics including root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are used to assess prediction errors, as shown in Table 6. This table also lists the maximum and minimum absolute percentage errors, as denoted by APE_{max} and APE_{min}, respectively. According to Table 6, the overall RMSE and MAE are 19.44 and 14.20, respectively. Furthermore, the overall MAPE (1.44×10^{-2}) indicates that the absolute percentage of the difference between the actual and predicted closing prices is only 1.44%, on average. The minimum absolute percentage error attains the excellent level of 0.00156%, and the maximum absolute percentage of the differences between the actual and predicted closing prices is mere 7.32%. Consequently, the proposed SOM-GP prediction procedure is considered an effective method for predicting the TAIEX-FISI for the next day by using 16 technical indicators. In addition, the proposed SOM-GP prediction procedure yields the maximum RMSE (27.37) and MAE (22.04) for the sample data of the 1st validation period, while providing the minimum RMSE (10.25) and MAE (7.29) for the sample data of the 7th validation period. By comparing Fig. 5, which draws the predicted and actual values of the closing prices of the 7th validation period, with Fig. 4, it is found that the TAIEX-FISI closing prices within the 1st validation period fluctuated more frequently than in the 7th validation period. It is believed that such frequent fluctuation renders a prediction more difficult to obtain, and thus, results in larger prediction errors. Further observation of the distribution of the predicted and actual closing prices in the 9th period, as shown in Fig 6, reveals that the TAIEX-FISI closing prices, as compared with Fig. 5, progress much more immoderately, and the difference between the maximum and minimum closing prices is much larger than that in the 7th validation period. This study considers this the reason for the proposed SOM-GP prediction procedure to produce the maximum MAPE (2.05×10^{-2}) for the sample data in the 9th validation period, whereas it provides the minimum MAPE (7.72×10^{-3}) for the sample data in the 7th validation period. In conclusion, this study believes that the frequent, alternating rise and fall, as well as the range of the daily closing price during a period, will significantly increase prediction difficulty, as based on the above analysis.

5. Conclusions

With the inherent high volatility, complexity, dynamics, and turbulence of stock prices, the prediction of a stock price is a

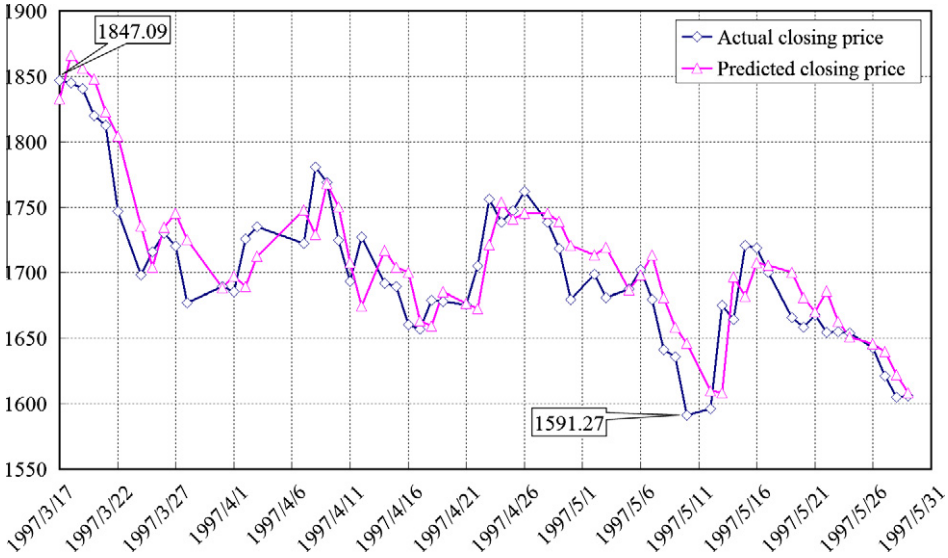


Fig. 4. Predicted and actual values of the closing prices during the period from March 17, 1997 to May 29, 1997.

Table 6
Prediction performance of the proposed SOM-GP prediction procedure.

Period No.	Validation period	RMSE	MAE	MAPE	APE_{max}	APE_{min}
1	1997/03/17–1997/05/29	27.37	22.04	1.30×10^{-2}	3.99×10^{-2}	2.79×10^{-4}
2	1998/06/29–1998/09/11	24.70	17.99	1.52×10^{-2}	6.62×10^{-2}	2.76×10^{-5}
3	1999/10/28–2000/01/18	18.41	14.62	1.50×10^{-2}	5.28×10^{-2}	8.72×10^{-5}
4	2001/03/09–2001/06/04	13.73	10.79	1.50×10^{-2}	6.47×10^{-2}	1.56×10^{-5}
5	2002/08/23–2002/11/18	13.48	9.86	1.51×10^{-2}	7.24×10^{-2}	4.57×10^{-5}
6	2004/02/09–2004/04/30	24.75	17.77	1.71×10^{-2}	7.32×10^{-2}	1.24×10^{-4}
7	2005/07/15–2005/10/12	10.25	7.29	7.72×10^{-3}	4.48×10^{-2}	4.04×10^{-4}
8	2006/12/25–2007/03/28	12.84	10.10	9.86×10^{-3}	4.82×10^{-2}	1.92×10^{-4}
9	2008/06/13–2008/09/05	23.57	18.57	2.05×10^{-2}	6.74×10^{-2}	2.16×10^{-4}
10	2009/07/20–2009/09/17	15.12	12.53	1.55×10^{-2}	4.09×10^{-2}	3.24×10^{-4}
Overall		19.44	14.20	1.44×10^{-2}	7.32×10^{-2}	1.56×10^{-5}

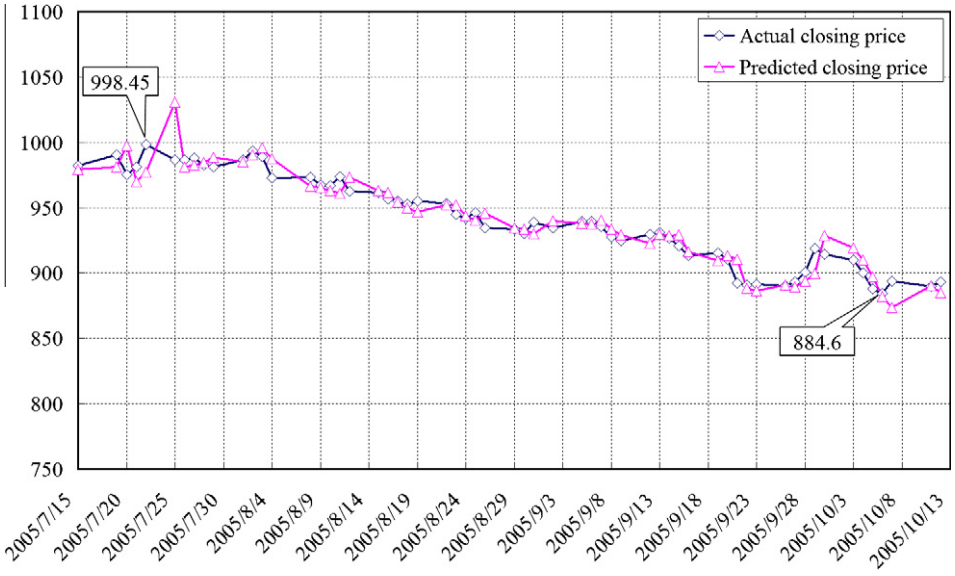


Fig. 5. Predicted and actual values of the closing prices during the period from July 15, 2005 to October 12, 2005.

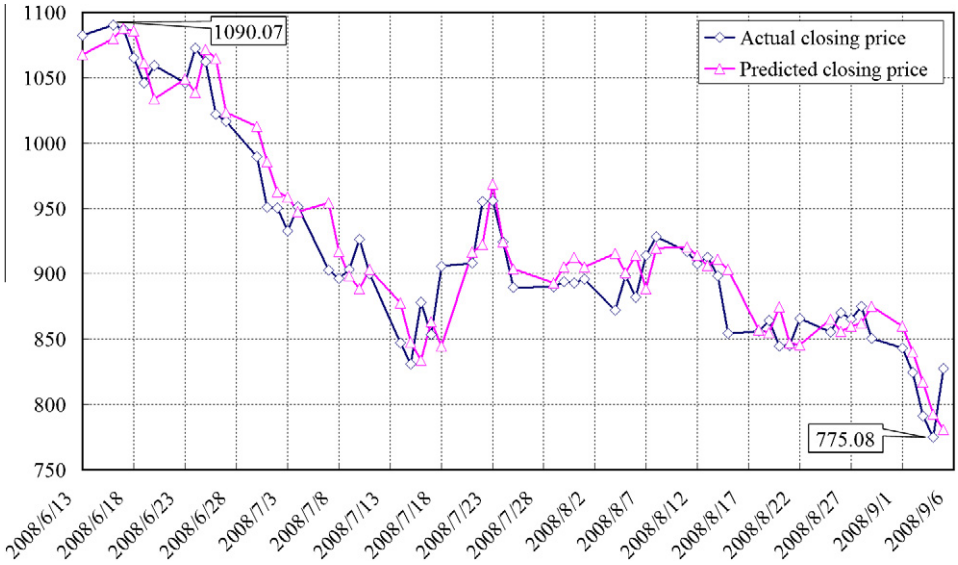


Fig. 6. Predicted and actual values of the closing prices during the period from June 13, 2008 to September 5, 2008.

challenging task. The fundamental analysis, technical analysis, and traditional time series forecasting, which have their respective merits and limitations, are the three main categories of stock prediction methodologies. In this study, a self-organizing map (SOM) neural network and genetic programming (GP) were utilized to develop an integrated approach, called the SOM-GP procedure, for predicting stock prices. An SOM neural network was applied to split the sample data into several clusters in such a way that the objects within each cluster were highly similar, which aims to facilitate the construction of the approximation functions that describe the implicit mathematical relationship between the technical indicators and the closing prices. In addition, this study introduced the clustering efficiency (CE) index that measures clustering performance, in order that the optimal number of clusters for SOM clustering could be determined. The GP algorithm was then used to construct the prediction models for the sample data of the clusters, as previously formed through SOM, thus, the closing price of the next day can be predicted based on the technical indicators of a certain day. The feasibility and effectiveness of the proposed hybrid SOM-GP prediction procedure were verified through conducting experimental predictions of the finance and insurance sub-index of TAIEX over the period from January 4, 1996 to September 18, 2009. The obtained results delivered the overall root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), as 19.44, 14.20, and 1.44×10^{-2}, respectively. Specifically, the MAPE with 1.44×10^{-2} indicates that the absolute percentage of the differences between the actual and predicted closing prices was only 1.44%, on average. The minimum absolute percentage error (APE_{min}) can attain the excellent level of 0.00156%, and the maximum absolute percentage (APE_{max}) of the differences between the actual and predicted closing prices was mere 7.32%. Based on the above information, the proposed SOM-GP prediction procedure can be considered as a feasible and effective tool for stock price prediction. Through further observations of the distribution of actual closing prices, and their corresponding prediction performance indices for the different periods, this study concluded that the frequent, alternating rise and fall, as well as the range of the daily closing prices during a period, can significantly increase the difficulty of prediction. Further research directions suggested from this study might include using feature selection techniques to choose the most important technical indicators as the input variables of the mathematical prediction models, and optimizing the parameters of GP through other soft computing methods, e.g. particle swarm optimization or ant colony optimization.

Appendix A. Descriptions and definitions of technical indicators used in this study

Notations:

i: the day i
HP_i: the highest price of day i
LP_i: the lowest price of day i
OP_i: the opening price of day i
CP_i: the closing price of day i
TV_i: the trade volume of day i

1. MA_10: 10-day moving average.
   The 10-day moving average is the mean price of a security over the most recent 10 days, and is calculated by:
   $$\text{MA\_10}_i = \frac{\sum_{j=i-9}^{i} CP_j}{10}. \quad (9)$$
2. BIAS_20: 20-day bias.
   The 20-day bias is the deviation between the closing price and the 20-day moving average (MA_20), and is calculated by:
   $$\text{BIAS\_20}_i = \frac{CP_i - \text{MA\_20}_i}{\text{MA\_20}_i}. \quad (10)$$
3. MACD: moving average convergence/divergence.
   The moving average convergence/divergence is a momentum indicator that shows the relationship between two moving averages. First, define the demand index (DI) as:
   $$\text{DI}_i = (HP_i + LP_i + 2 \times CP_i)/4. \quad (11)$$
   Next, define the 12-day exponential moving average (EMA_12) and 26-day exponential moving average (EMA_26) as:
   $$\text{EMA\_12}_i = \frac{11}{13} \times \text{EMA\_12}_{i-1} + \frac{2}{13} \times \text{DI}_i \quad (12)$$
   and
   $$\text{EMA\_26}_i = \frac{25}{27} \times \text{EMA\_26}_{i-1} + \frac{2}{27} \times \text{DI}_i, \quad (13)$$
   respectively. Then, the difference between EMA_12 and EMA_26 can be calculated by:
   $$\text{DIF}_i = \text{EMA\_12}_i - \text{EMA\_26}_i. \quad (14)$$
   Hence, the moving average convergence/divergence can be defined by:
   $$\text{MACD}_i = \frac{8}{10} \times \text{MACD}_{i-1} + \frac{2}{10} \times \text{DIF}_i. \quad (15)$$
4. K_9: 9-day stochastic indicator K.
   The 9-day stochastic indicator K is defined as:
   $$\text{K\_9}_i = \frac{2}{3} \times \text{K\_9}_{i-1} + \frac{1}{3} \times \frac{CP_i - LP\_9_i}{HP\_9_i - LP\_9_i} \times 100. \quad (16)$$
   where LP_9_i and HP_9_i are the lowest and highest prices of the previous 9 days, i.e. days $i, i-1, \ldots, i-7$ and $i-8$, respectively.
5. D_9: 9-day stochastic indicator D.
   The 9-day stochastic indicator D is defined as:
   $$\text{D\_9}_i = \frac{2}{3} \times \text{D\_9}_{i-1} + \frac{1}{3} \times \text{K\_9}_i, \quad (17)$$
   where K_9_i is the 9-day stochastic indicator K of day i, as previously defined.
6. WMS%R_9: 9-day Williams overbought/oversold index.
   The 9-day Williams overbought/oversold index is a momentum indicator that measures overbought and oversold levels, and is calculated by:
   $$\text{WMS\% R\_9}_i = \frac{HP\_9_i - CP_i}{HP\_9_i - LP\_9_i}, \quad (18)$$
   where LP_9_i and HP_9_i are the lowest and highest prices of the previous 9 days, i.e. days $i, i-1, \ldots, i-7$ and $i-8$, respectively.
7. +DI_14: 14-day plus directional indicator.
   First, define plus directional movement (+DM) and minus directional movement (−DM) as:
   $$+\text{DM}_i = HP_i - HP_{i-1} \quad (19)$$
   and
   $$-\text{DM}_i = LP_{i-1} - LP_i, \quad (20)$$
   respectively. The plus true directional movement (+TDM) can be calculated by:
   $$+\text{TDM}_i = \begin{cases} +\text{DM}_i, & \textit{if } +\text{DM}_i > -\text{DM}_i \text{ and } +\text{DM}_i > 0, \\ 0, & \textit{otherwise}. \end{cases} \quad (21)$$
   Similarly, the minus true directional movement (-TDM) can be calculated by:

$$-\mathrm{TDM}_i = \begin{cases} -\mathrm{DM}_i, & \text{if } +\mathrm{DM}_i < -\mathrm{DM}_i \text{ and } -\mathrm{DM}_i > 0, \\ 0, & \text{otherwise}. \end{cases} \tag{22}$$

Hence, the 14-day plus directional movement (+DM_14) can be calculated by:

$$+\mathrm{DM\_14}_i = \frac{13}{14} \times (+\mathrm{DM\_14}_{i-1}) + \frac{1}{14} \times (+\mathrm{TDM}_i). \tag{23}$$

Similarly, the 14-day minus directional movement (−DM_14) can be calculated by:

$$-\mathrm{DM\_14}_i = \frac{13}{14} \times (-\mathrm{DM\_14}_{i-1}) + \frac{1}{14} \times (-\mathrm{TDM}_i). \tag{24}$$

Next, define the true range (TR) as:

$$\mathrm{TR}_i = Max\{HP_i - LP_i, |HP_i - CP_{i-1}|, |LP_i - CP_{i-1}|\}. \tag{25}$$

The 14-day true range (TR_14) can be calculated by:

$$\mathrm{TR\_14}_i = \frac{13}{14} \times \mathrm{TR\_14}_{i-1} + \frac{1}{14} \times \mathrm{TR}_i. \tag{26}$$

Therefore, the 14-day plus directional indicator can be defined as:

$$+\mathrm{DI\_14}_i = \frac{+\mathrm{DM\_14}_i}{\mathrm{TR\_14}_i}. \tag{27}$$

8. −DI_14: 14-day minus directional indicator.
The 14-day minus directional indicator is defined as:

$$-\mathrm{DI\_14}_i = \frac{-\mathrm{DM\_14}_i}{\mathrm{TR\_14}_i}. \tag{28}$$

where $-\mathrm{DM\_14}_i$ and $\mathrm{TR}/14_i$ are the 14-day minus directional movement and 14-day true range of day i, respectively, as previously defined.

9. MTM_10: 10-day momentum.
The 10-day momentum measures the price changes of a security during a period of 10 days, and is calculated by:

$$\mathrm{MTM\_10}_i = CP_i - CP_{i-10}. \tag{29}$$

10. ROC_10: 10-day rate of change.
The 10-day rate of change measures the percent changes of the current price relative to the price of 10 days ago, and is calculated by:

$$\mathrm{ROC\_10}_i = \frac{CP_i - CP_{i-10}}{CP_{i-10}} \times 100. \tag{30}$$

11. RSI_5: 5-day relative strength index.
The relative strength index is a momentum oscillator that compares the magnitude of recent gains to the magnitude of recent losses. First, define the gain of day i as:

$$G_i = \begin{cases} CP_i - CP_{i-1}, & \text{if } CP_i > CP_{i-1}, \\ 0, & \text{otherwise}. \end{cases} \tag{31}$$

Similarly, the loss of day i is calculated by:

$$L_i = \begin{cases} CP_i - CP_{i-1}, & \text{if } CP_i < CP_{i-1}, \\ 0, & \text{otherwise}. \end{cases} \tag{32}$$

Next, the 5-day average gain (AG_5) and 5-day average loss (AL_5), which can be calculated by:

$$\mathrm{AG\_5}_i = \frac{4}{5} \times \mathrm{AG\_5}_{i-1} + \frac{1}{5} \times G_i \tag{33}$$

and

$$\mathrm{AL\_5}_i = \frac{4}{5} \times \mathrm{AL\_5}_{i-1} + \frac{1}{5} \times L_i, \tag{34}$$

respectively. Hence, the 5-day relative strength index can be defined by:

$$\mathrm{RSI\_5}_i = \frac{\mathrm{AG\_5}_i}{\mathrm{AG\_5}_i + \mathrm{AL\_5}_i} \times 100. \tag{35}$$

12. CCI_24: 24-day commodity channel index.
The commodity channel index is used to identify cyclical turns in commodities. First, define the typical price (TP) as:

$$\mathrm{TP}_i = \frac{HP_i + LP_i + CP_i}{3}. \tag{36}$$

Next, calculate the 24-day simple moving average of the typical price (SMATP_24) by:

$$\mathrm{SMATP\_24}_i = \frac{\sum_{j=i-23}^{i} \mathrm{TP}_j}{24}. \tag{37}$$

Then, the 24-day mean deviation (MD_24) can be calculated by:

$$\mathrm{MD\_24}_i = \frac{\sum_{j=i-23}^{i} |\mathrm{TP}_j - \mathrm{SMATP\_24}_i|}{24}. \tag{38}$$

Hence, the 24-day commodity channel index can be defined as:

$$\mathrm{CCI\_24}_i = \frac{\mathrm{TP}_i - \mathrm{SMATP\_24}_i}{0.015 \times \mathrm{MD\_24}_i}. \tag{39}$$

13. AR_26: 26-day buying/selling momentum indicator.
The 26-day buying/selling momentum indicator is defined as:

$$\mathrm{AR\_26}_i = \frac{\sum_{j=i-25}^{i} (HP_j - OP_j)}{\sum_{j=i-25}^{i} (OP_j - LP_j)}. \tag{40}$$

14. BR_26: 26-day buying/selling willingness indicator.
The 26-day buying/selling willingness indicator is defined as:

$$\mathrm{BR\_26}_i = \frac{\sum_{j=i-25}^{i} (HP_j - CP_{j-1})}{\sum_{j=i-25}^{i} (CP_{j-1} - LP_j)}. \tag{41}$$

15. VR_26: 26-day volume ratio.
The 26-day volume ratio is defined by:

$$\mathrm{VR\_26}_i = \frac{\mathrm{TVU\_26}_i - \mathrm{TVF\_26}_i/2}{\mathrm{TVD\_26}_i - \mathrm{TVF\_26}_i/2} \times 100\%. \tag{42}$$

where $\mathrm{TVU\_26}_i$, $\mathrm{TVD\_26}_i$, and $\mathrm{TVF}/26_i$ represent the total trade volumes of stock prices rising, falling, and holding, respectively, from the previous 26 days, i.e. days $i, i-1, \ldots, i-24$ and $i-25$.

16. PSY_13: 13-day psychological line.
The psychological line is a volatility indicator based on the number of time intervals that the market was up during the preceding period. The 13-day psychological line is defined by:

$$\mathrm{PSY\_13}_i = \frac{\mathrm{TDU\_13}_i}{13} \times 100\%, \tag{43}$$

where $\mathrm{TDU\_13}_i$ is the total number of days regarding stock price rises of the previous 13 days, i.e. days $i, i-1, \ldots, i-11$ and $i-12$.

References

Bae, H., Jeon, T. R., Kim, S., Kim, H. S., Kim, D., Han, S. S., et al. (2010). Optimization of silicon solar cell fabrication based on neural network and genetic programming modeling. *Soft Computing, 14*(2), 161–169.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics, 31*(3), 307–327.

Box, G. E. P., & Jenkins, G. M. (1970). *Time series analysis, forecasting, and control*. San Francisco: Holden-Day.

Chang, P.-C., & Liu, C.-H. (2008). A TSK type fuzzy rule based system for stock price prediction. *Expert Systems with Applications, 34*(1), 135–144.

Ciglarič, I., & Kidrič, A. (2006). Computer-aided derivation of the optimal mathematical models to study gear-pair dynamic by using genetic programming. *Structural and Multidisciplinary Optimization, 32*(2), 153–160.

Etemadi, H., Rostamy, A. A. A., & Dehkordi, H. F. (2009). A genetic programming model for bankruptcy prediction: empirical evidence from Iran. *Expert Systems with Applications, 36*(2), 3199–3207.

Fausett, L. (1994). *Fundamentals of neural networks: Architectures, algorithms, and applications*. New Jersey: Prentice-Hall.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Huang, C.-L., & Tsai, C.-Y. (2009). A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Systems with Applications, 36*(2), 1529–1539.

Hwang, T.-M., Oh, H., Choung, Y.-K., Oh, S., Jeon, M., Kim, J. H., et al. (2009). Prediction of membrane fouling in the pilot-scale microfiltration system using genetic programming. *Desalination, 247*(1–3), 285–294.

Ince, H., & Trafalis, T. B. (2008). Short term forecasting with support vector machines and application to stock price prediction. *International Journal of General Systems, 37*(6), 677–687.

Kim, K.-J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1-2), 307–319.

Kim, K.-J., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications, 19*(2), 125–132.

Kim, K.-J., & Lee, W. B. (2004). Stock market prediction using artificial neural networks with optimal feature transformation. *Neural Computing and Applications, 13*(3), 255–260.

Kohonen, T. (1995). *Self-organizing maps*. Berlin: Springer-Verlag.

Kohonen, T. (1989). *Self-organization and associative memory*. Berlin: Springer-Verlag.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*. Cambridge, MA: MIT Press.

Koza, J. R., Keane, M. A., Streeter, M. J., Mydlowec, W., Yu, J., & Lanza, G. (2005). *Genetic programming IV: Routine human-competitive machine intelligence*. New York: Springer.

Koza, J. R., Streeter, M. J., & Keane, M. A. (2008). Routine high-return human-competitive automated problem-solving by means of genetic programming. *Information Sciences, 178*(23), 4434–4452.

Lai, R. K., Fan, C.-Y., Huang, W.-H., & Chang, P.-C. (2009). Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications, 36*(2), 3761–3773.

Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications, 36*(8), 10896–10904.

Liang, X., Zhang, H., Xao, J., & Chen, Y. (2009). Improving option price forecasts with neural networks and support vector regressions. *Neurocomputing, 72*(13–15), 3055–3065.

Lin, G.-F., & Wu, M.-C. (2009). A hybrid neural network model for typhoon-rainfall forecasting. *Journal of Hydrology, 375*(3–4), 450–458.

Pai, P.-F., & Lin, C.-S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega, 33*(6), 497–505.

Szczurowska, I., Kuniszyk-Jozkowiak, W., & Smolka, E. (2009). Speech nonfluency detection using Kohonen networks. *Neural Computing and Applications, 18*(7), 677–687.

Thomsett, M. C. (1998). *Mastering fundamental analysis*. Chicago: Dearborn Financial Publishing.

Thomsett, M. C. (1999). *Mastering technical analysis*. Chicago: Dearborn Financial Publishing.

Tsang, P. M., Kwok, P., Choy, S. O., Kwan, R., Ng, S. C., Mak, J., et al. (2007). Design and implementation of NN5 for Hong Kong stock price forecasting. *Engineering Applications of Artificial Intelligence, 20*(4), 453–461.

Yu, L., Chen, H., Wang, S., & Lai, K. K. (2009). Evolving least squares support vector machines for stock market trend mining. *IEEE Transactions on Evolutionary Computation, 13*(1), 87–102.

Zhang, J., An, L., Tang, T., & Hong, Y. (2009). Visual health subject directory analysis based on users' traversal activities. *Journal of the American Society for Information Science and Technology, 60*(10), 1977–1994.