

# A hybrid procedure with feature selection for resolving stock/futures price forecasting problems

Chih-Ming Hsu

Received: 11 March 2011 / Accepted: 3 August 2011 / Published online: 20 August 2011  
© Springer-Verlag London Limited 2011

**Abstract** Stock/futures price forecasting is an important financial topic for individual investors, stock fund managers, and financial analysts and is currently receiving considerable attention from both researchers and practitioners. However, the inherent characteristics of stock/futures prices, namely, high volatility, complexity, and turbulence, make forecasting a challenging endeavor. In the past, various approaches have been proposed to deal with the problems of stock/futures price forecasting that are difficult to resolve by using only a single soft computing technique. In this study, a hybrid procedure based on a backpropagation (BP) neural network, a feature selection technique, and genetic programming (GP) is proposed to tackle stock/futures price forecasting problems with the use of technical indicators. The feasibility and effectiveness of this procedure are evaluated through a case study on forecasting the closing prices of Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) futures of the spot month. Experimental results show that the proposed forecasting procedure is a feasible and effective tool for improving the performance of stock/futures price forecasting. Furthermore, the most important technical indicators can be determined by applying a feature selection method based on the proposed simulation technique, or solely on the preliminary GP forecast model.

**Keywords** Stock/futures price forecasting · Backpropagation neural network · Feature selection · Genetic programming

---

C.-M. Hsu (✉)  
Department of Business Administration, Minghsin University  
of Science and Technology, 1 Hsin-Hsing Road, Hsin-Fong,  
Hsinchu 304, Taiwan, ROC  
e-mail: cmhsu@must.edu.tw

## 1 Introduction

Stock/futures price forecasting is an important issue in investment/financial decision-making among individual investors, stock fund managers, and financial analysts and is currently receiving considerable attention from both researchers and practitioners. Stock/futures price forecasting is considered a challenging task due to the fact that the prices are highly volatile, complex, and dynamic. Therefore, various approaches for tackling the problems of stock/futures price forecasting have been proposed. These approaches can be broadly classified into three categories: fundamental analysis, technical analysis, and traditional time series forecasting. Fundamental analysis examines the statistics of macroeconomics data such as interest rates, money supply, and foreign exchange rates, as well as the basic financial information of a corporation, in order to forecast stock price movements [1, 2]. Technical analysis involves studying stock prices, recent and historical price trends and cycles, factors beyond stock price, such as dividend payments, trading volume, group trends and popularity, and the volatility of a stock to forecast future stock prices [3]. The third approach, traditional time series forecasting, uses statistical tools to forecast future values of a stock based on its past values and other variables. These tools include moving average, autoregressive integrated moving average (ARIMA) [4], generalized autoregressive conditional heteroskedasticity (GARCH) [5], and multivariate regression, etc.

In recent years, data mining/computational intelligence techniques have become rapidly growing alternative methods for resolving, or assisting other approaches in resolving, the problems of stock/futures price forecasting. For example, Pai and Lin [6] exploited the strengths of the autoregressive integrated moving average (ARIMA)

and support vector machine (SVM) to develop a hybrid methodology. The ARIMA technique was used to model the linear dataset, while the SVMs were employed to deal with the nonlinear data pattern in time series forecasting. The performance of the proposed model was evaluated by examining real datasets of ten stocks and adequate results were obtained. Furthermore, comparison results showed that their presented methodology can significantly improve the forecasting performance of the single ARIMA or SVM model. Chang and Liu [7] presented a Takagi–Sugeno–Kang (TSK) type fuzzy rule-based system to tackle stock price forecasting problems. They used stepwise regression to analyze and select technical indicators that are more influential to the stock price. The *K*-means clustering technique was then applied to partition the dataset into several clusters. Finally, a simplified fuzzy rule inference system was set up with the optimal rule parameters trained by simulated annealing to forecast 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 backpropagation neural networks and multiple regression analysis. Ince and Trafalis [8] proposed an approach for stock price forecasting by using kernel principal component analysis (kPCA) and support vector regression (SVR), based on the assumption that the future value of a stock price depends on its financial indicators. The kPCA was utilized to identify the most important technical indicators, and the SVR was employed to construct the relationship between the stock price and selected technical indicators. For comparison, they also applied factor analysis to determine the most influential input technical indicators for a forecasting model and used multilayer perceptron (MLP) neural networks to build the forecasting model. The experiments were conducted on the daily stock prices of ten companies traded on the NASDAQ. The comparison results indicated that proposed heuristic models produce better results than the studied kPCA-SVR model. Furthermore, there is no significant difference in forecasting performance between the MLP neural model and SVR technique in terms of the mean square error. Huang and Tsai [9] proposed a hybrid procedure using filter-based feature selection, a self-organizing feature map (SOFM), and support vector regression (SVR), in order to forecast the stock market price index. The filter-based feature selection was first used to determine important input technical indicators thus reducing data complexity. The SOFM algorithm was then applied to cluster the training data into several disjointed clusters such that the elements in each cluster are similar. Finally, the SVR technique was employed to construct an individual forecasting model for each cluster. Their proposed model was demonstrated through a case

study on forecasting the next day's price index for Taiwan index futures (FITX), and the experiment results showed that the proposed approach can improve forecasting accuracy and reduce the training time over the traditional single SVR model. Liang et al. [10] presented a two-stage approach to improve option price forecasting by using modified conventional option pricing methods, neural networks (NNs), and support vector regressions (SVRs). In their study, three improved modified conventional parametric methods, the binomial tree method, the finite difference method, and the Monte Carlo method, were utilized to forecast the option prices in the first stage. Then, the NNs and SVRs were employed to carry out nonlinear curve approximation to further reduce the forecasting errors in the second stage. The proposed approach was demonstrated by experimental studies on the data taken from the Hong Kong options market, and the results showed that the NN and SVR approaches can significantly shrink the average forecast errors, thus improving forecasting accuracy. In yet another approach to stock price forecasting, Hadavandi et al. [11] presented a hybrid model by integrating stepwise regression analysis, artificial neural networks, and genetic fuzzy systems. They first used stepwise regression analysis to select factors that can significantly affect stock prices. A self-organizing map (SOM) neural network was then used to divide the raw data into several clusters such that the elements in each cluster are more homogeneous. Finally, data in each cluster were fed into a genetic fuzzy system to forecast stock prices. Their proposed approach was demonstrated through an experiment on forecasting the next day's closing prices of two IT corporations and two airlines, and comparison results indicated that the proposed approach outperforms other previous methods. Lu [12] proposed an integrated stock price forecasting model by using independent component analysis (ICA) and a backpropagation (BP) neural network. The ICA was first applied to the forecasting variables in order to filter out the noise contained in these variables, thus generating the independent components (ICs). The filtered forecasting variables then served as the input variables of the BP neural network to build the forecasting model. The proposed approach was illustrated by a case study on forecasting the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) closing cash index and Nikkei 225 opening cash index. The comparison results revealed that the proposed forecasting model outperforms the integrated wavelet-BP model, the BP model without filtered forecasting variables, and the random walk model. Cheng et al. [13] employed the cumulative probability distribution approach (CDPA), minimize entropy principle approach (MEPA), rough sets theory (RST), and genetic algorithms (GAs) to develop a hybrid approach for stock

price forecasting. The CDPA and MEPA were first utilized to partition technical indicator values and daily price fluctuation into linguistic values according to the characteristics of data distributions. The RST was then employed to generate linguistic rules from the linguistic technical indicator dataset. Finally, the GAs were applied to refine the extracted rules, thus improving the forecasting accuracy and stock return modeling. Their proposed methodology was verified through a case study on forecasting the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) and adequate results were obtained.

The above-mentioned studies prove that using neural network techniques is an effective way to address the problems of stock/futures price forecasting. Furthermore, forecasting performance can be further improved by using feature selection methods to find out the most important factors that serve as input variables for a forecasting model. These methods include stepwise regression, kernel principal component analysis, factor analysis, and independent component analysis. The current study proposes a systematic procedure based on a backpropagation (BP) neural network, a feature selection technique, and genetic programming (GP) to tackle stock/futures price forecasting problems with the use of technical indicators. First, the BP neural network is used to construct the preliminary forecasting model that describes the complex nonlinear relationship between the technical indicators and future stock/futures prices. Next, the feature selection technique based on simulation is utilized to explore the preliminary forecasting model in order to select the most important technical indicators for forecasting stock/futures prices. In addition, the GP is also employed to build another forecasting model, thus automatically screening the vital technical indicators that closely correlate with future stock/futures prices. The BP neural network is then applied again to develop the final forecasting model by using the technical indicators that are produced by the feature selection technique based on simulation or screened by the GP program as the input forecasting variables. Finally, the performance of the final forecasting model is evaluated and compared with the performance of the preliminary BP and GP forecasting models.

The remainder of this paper is organized as follows. In Sect. 2, BP neural networks, feature selection, and GP are discussed. The proposed integrated approach is presented in Sect. 3. Section 4 evaluates the feasibility and effectiveness of the proposed approach by presenting and analyzing a case study on forecasting the closing prices of Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) futures of the spot month. Finally, Sect. 5 concludes the paper.

## 2 Methodologies

Three methods are used in this study to develop the integrated procedure for solving a stock/futures price forecasting problem. This section briefly introduces these methods, beginning with the backpropagation neural network.

### 2.1 Backpropagation neural network

The backpropagation (BP) neural network is a widely used network that is trained through a supervised learning algorithm. The design of the network is a multi-layered neural model that consists of one input layer, at least one hidden layer, and one output layer. Each layer, which comprises several nodes called neurons, is fully connected to the succeeding layer. The number of neurons in the input and output layers can be accurately determined by the number of input and output variables in a problem. On the other hand, the optimal number of hidden layers and the neurons in each hidden layer are usually determined through a trial-and-error method based on root mean squared errors (RMSEs) found in the training and test data. Theoretically, one hidden layer is sufficient for approximating any continuous functional relationship between input and output variables with an arbitrary degree of accuracy [14]. When applying a BP neural network to a specific problem, the neural model must be learned (trained) in advance. The learning process includes three stages: (1) feed-forward of the input training pattern, (2) calculation and backpropagation of the associated error, and (3) weight and bias adjustment. The learning process proceeds repeatedly until the RMSE associated with the training data sufficiently converge to a global minimum. Furthermore, the optimal number of learning iterations is usually determined by observing the progress of test RMSE by feeding independent test data into the trained BP neural model in order to prevent over-training. During the learning process of a BP neural network, two major parameters, including the learning rate and momentum, must be specified by the users. The appropriate setting of the learning rate and momentum can help the BP neural model achieve the global minimum training and test RMSEs. However, the values of both parameters are problem dependent [15] and are usually determined through trial and error. Once the learning process is accomplished, the well-trained BP neural network can be utilized to estimate the unknown values of output variables by the recall process of feeding the input pattern into this BP model.

BP neural networks have attracted substantial research interest from a wide range of applications. As these applications demonstrate, adequate estimation results can be obtained through these neural networks [16–19].

## 2.2 Feature selection

Feature selection is a process of selecting a minimal subset of features based on certain reasonable criteria in order to remove irrelevant and redundant features and hence noise. In this way, the original problem can be tackled equally well, if not better, than by using a full set of features [20]. Feature selection can prevent the selection of too many or too few features (i.e., more or less than necessary) and achieve data reduction for accelerating training and increasing computational efficiency [21]. If too many (irrelevant) features are selected, the effects of noise may duplicate or shadow the valuable information in data. However, if too few features are chosen, the information contained in the selected subset of features might be low. Feature selection approaches can be classified into two categories: the wrapper approach and the filter approach. In the wrapper approach, the feature selection algorithm acts as a wrapper around the induction algorithm [22] and searches for a good subset by evaluating feature subsets via the induction algorithm. In other words, the feature selection and induction algorithms interact while searching for the best feature subset. Comparatively, the filter approach utilizes a preprocessing step to filter features and thus find a good feature subset. This method entirely ignores the performance of the selected feature subset on the induction algorithm. That is, the feature selection algorithm proceeds independently of the induction algorithm [22].

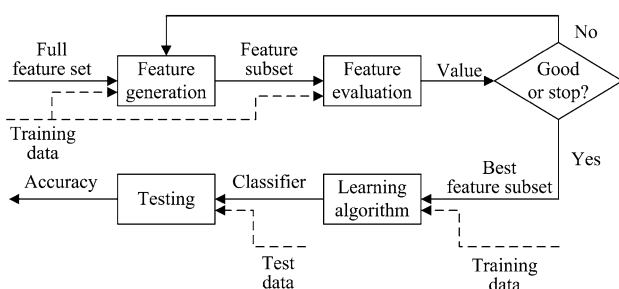
Liu and Motoda [20] provided a unified model of feature selection, which is shown in Fig. 1. First, the feature generation produces a feature subset each time using a certain search strategy based on the training data. The generated feature subset along with the training data are then evaluated by a measuring function. The feature generation mechanism stops searching once the stopping criteria are fulfilled. While the stopping criteria vary case by case, a stop may be activated by any one of the following conditions: (1) the feature subset is good enough, (2) some bound is reached, and (3) the search completes [20]. Notably, the feature subset is determined to be adequate if its estimated accuracy, i.e., classification or forecasting

accuracy, is better than the accuracy obtained by using a full set of features. Next, the training data with a specified feature subset, i.e., the best feature subset generated, are fed into the learning algorithm in order to construct the classifier (or forecaster). Finally, the classifier (or forecaster) is tested on the test data with the selected features, thus obtaining the classification accuracy (or forecasting errors).

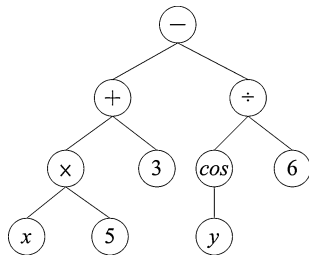
Researchers have attempted to resolve problems of feature selection using various techniques. These include the following: information-theoretic measures [23], rough sets and ant colony optimization [24], genetic algorithms [25], and mathematical programming [26]. Further discussions on feature selection can be found in the studies reported by Kohavi and John [22], Liu and Motoda [20], Piramuthu [21], and Mladenic [27].

## 2.3 Genetic programming

Genetic programming (GP) is an evolutionary methodology that can automatically create computer programs to solve a user-defined problem by genetically breeding a population of computer programs using the principles of Darwinian natural selection and biologically inspired operations. The solution technique of GP closely relates to the field of genetic algorithms (GAs). However, there exist three major differences between GAs and GP. First, GP usually evolves a tree-based structure, while GAs evolve binary or real number strings. Second, the length of the binary or real number string is fixed in traditional GAs, while the parse trees can vary in length throughout the run in GP. Finally, the solutions of GP are active structures; this means that they can be directly executed without post-processing, whereas the binary or real number strings in GAs are passive structures which require post-processing [28]. Figure 2 illustrates a tree-based representation of the solution  $[(x \times 5) + 3] - [\cos(y) \div 6]$  in GP. As seen, the terminal components in the branches, i.e.,  $x$ ,  $5$ ,  $3$ ,  $y$ , and  $6$ , are called terminal elements. The terminal elements (e.g., the independent variables of the problem, zero-argument functions, random constants, etc.) available for each branch of the to-be-evolved computer program in GP are defined by a terminal set. On the other hand, the remaining components in Fig. 2, i.e.,  $+$ ,  $-$ ,  $\times$ ,  $\div$ , and  $\cos$ , are called primitive functions. The primitive functions available to the medium elements of the to-be-evolved computer program, such as addition, square root, multiplication, sine, etc., are defined by a function set. Like GAs, the fitness (adaptability) of a solution in the GP population is explicitly or implicitly measured via a predefined fitness function. Furthermore, some parameters and the termination criterion must be specified in advance in order to apply GP to resolve a problem. The major parameters that control



**Fig. 1** A unified model of feature selection



**Fig. 2** Tree-based representation of a solution in GP

the run in GP include population size, maximum size of programs, crossover rate, and mutation rate. In addition, the time to stop the evolutionary procedure of GP is determined by the termination criterion. The general steps of GP are presented in the studies of Koza et al. [29, 30] and Ciglaric and Kidric [31] and are summarized as follows:

*Step 1* Generating an initial population This step generates an initial population of computer programs, which can be of different sizes and different shapes, appropriate to a problem subject to a prespecified maximum size.

*Step 2* Evaluating computer programs Each program in the population is executed and measured by using a predefined fitness function to obtain the fitness value.

*Step 3* Creating the next generation A pool of programs is first selected from the population using a probability based on the fitness value. Next, the genetic operations, including reproduction, crossover, mutation, and architecture-altering operations, are applied to the selected programs thus producing the offspring. Finally, the current population is replaced with the population of offspring based on a certain strategy, e.g., the elitist strategy, in order to create the next generation.

*Step 4* Examining the termination criterion When the termination criterion is satisfied, the outcome is designated as the final results of the run. Otherwise, Steps 2–4 are executed iteratively.

As shown in the studies of Etemadi et al. [32], Hwang et al. [33], and Bae et al. [34], GP has produced many novel and outstanding results in resolving problems from numerous fields. Further discussions on GP, its applications and related resources can be found in the studies of Langdon and Poli [35] and Koza et al. [29, 30].

### 3 Proposed forecasting procedure

In this study, a systematic procedure based on a back-propagation (BP) neural network, a feature selection technique, and genetic programming (GP) was proposed to

tackle stock/futures price forecasting problems with the use of technical indicators. The proposed forecasting procedure comprises three stages and is described as follows:

#### 3.1 Construction of preliminary forecasting models

In the first stage, the essential historical stock/futures trading data, e.g., opening price, highest price, lowest price, closing price, trade volume, etc. in each trading day are first collected. Next, the required technical indicators, e.g., moving average, Williams overbought/oversold index, psychological line, commodity channel index, etc. are calculated. To avoid variables with larger numeric ranges from dominating those with smaller numeric ranges, we normalize the technical indicators into a range between 0 and 1 using the following formula:

$$x_{ij} = \frac{ti_{ij} - ti_j^{\min}}{ti_j^{\max} - ti_j^{\min}} \quad \text{for } i = 1, 2, \dots, n; j = 1, 2, \dots, m \tag{1}$$

where  $ti_{ij}$  and  $x_{ij}$  are the original and normalized technical indicators, respectively;  $ti_j^{\max}$  and  $ti_j^{\min}$  are the maximum and minimum value of the  $j$ th technical indicator, respectively; and  $n$  and  $m$  are the total number of trading days and technical indicators, respectively. These normalized technical indicators, which take the full feature set as shown previously in Fig. 1, are the independent input variables in the stock/futures price forecasting model. In addition, the closing price of the stock/futures in each trading day is also normalized into a range between 0 and 1, based on the following formula:

$$y_i = \frac{p_i - p^{\min}}{p^{\max} - p^{\min}} \quad \text{for } i = 1, 2, \dots, n \tag{2}$$

where  $p_i$  and  $y_i$  are the original and normalized closing price of the stock/futures, respectively;  $p^{\max}$  and  $p^{\min}$  are the maximum and minimum value of the closing price of the stock/futures, respectively; and  $n$  is the total number of trading days. The normalized technical indicators in each trading day along with the normalized closing price in the future, e.g., next trading day or next second trading day, denoted by  $(x_{i1}, x_{i2}, \dots, x_{im}, y_i)$ , are then partitioned into training, test, and validation data, based on a prespecified proportion, e.g., 4:1:1. Next, a BP neural network is applied to the training and test data to construct several forecasting models. Then, a preliminary optimal forecasting model, named  $BP\_Model_{pre}$ , is selected based on simultaneously minimizing the root mean squared errors (RMSEs) of the training and test data. For comparison, the GP algorithm is also utilized to construct several forecasting models, and a preliminary optimal forecasting model, denoted as  $GP\_Model_{pre}$ , is determined according

to the training and test RMSEs to describe the functional relationship between the normalized technical indicators of each trading day and the normalized closing price of stock/futures as forecasted.

### 3.2 Feature selection

In the second stage, the feature selection technique based on simulation is utilized to determine the impact of each technical indicator on forecasting the closing price of the stock/futures. Suppose the total numbers of trading days in the training, test, and validation data are  $n_{tr}$ ,  $n_{te}$ , and  $n_{va}$ , respectively. First, the mean of the  $j$ th normalized technical indicator in successive  $t$  trading days, starting from the  $i$ th trading day in the training and test periods, is calculated by:

$$\bar{x}_{ij}^{(t)} = \frac{\sum_{k=0}^{t-1} x_{(i+k)j}}{t} \quad \text{for } i = 1, 2, \dots, n_{tr} + n_{te};$$

$$j = 1, 2, \dots, m. \tag{3}$$

Next, the standard deviation of the  $j$ th normalized technical indicator in successive  $t$  trading days, starting from the  $i$ th trading day in the training and test periods, can be calculated by:

$$s_{ij}^{(t)} = \sqrt{\frac{\sum_{k=0}^{t-1} (x_{(i+k)j} - \bar{x}_{ij}^{(t)})^2}{t-1}} \quad \text{for } i = 1, 2, \dots, n_{tr} + n_{te};$$

$$j = 1, 2, \dots, m. \tag{4}$$

Therefore, the mean of the standard deviation of the  $j$ th normalized technical indicator in successive  $t$  trading days, denoted by  $\bar{s}_j^{(t)}$ , during the training and test periods can be expressed as:

$$\bar{s}_j^{(t)} = \frac{\sum_{i=1}^{n_{tr}+n_{te}} s_{ij}^{(t)}}{n_{tr} + n_{te}} \quad \text{for } j = 1, 2, \dots, m. \tag{5}$$

To assess the impact of the  $j$ th technical indicator on forecasting the closing price of the stock/futures, we assume that the  $j$ th technical indicator in the  $i$ th trading day during the training and test periods can vary, based on a normal distribution, with its mean of  $x_{ij}$  and standard deviation of  $\bar{s}_j^{(t)}$ . Hence, the probability with which the value  $x_{ij}^*$  will lie in the range from  $x_{ij} - z_{\alpha/2}\bar{s}_j^{(t)}$  to  $x_{ij} + z_{\alpha/2}\bar{s}_j^{(t)}$  is  $1 - \alpha$ . That is,

$$P\left(x_{ij} - z_{\alpha/2}\bar{s}_j^{(t)} \leq x_{ij}^* \leq x_{ij} + z_{\alpha/2}\bar{s}_j^{(t)}\right) = 1 - \alpha$$

$$\text{for } i = 1, 2, \dots, n_{tr} + n_{te}; j = 1, 2, \dots, m \tag{6}$$

where  $z_{\alpha/2}$  is the critical value such that the probability that the standard normal distribution  $z$  is greater than  $z_{\alpha/2}$  is  $\alpha/2$ ;  $1 - \alpha$  is the confidence level set by the user, e.g., 0.95 or 0.99.

Next, sample points of a total number of  $s$ , denoted by  $x_{ij(l)}^*$ , are drawn from the range  $\left(x_{ij} - z_{\alpha/2}\bar{s}_j^{(t)}, x_{ij} + z_{\alpha/2}\bar{s}_j^{(t)}\right)$ , based on the following formulas:

$$x_{ij(l)}^* = x_{ij} + z_{\alpha^*}\bar{s}_j^{(t)} \quad \text{for } i = 1, 2, \dots, n_{tr} + n_{te};$$

$$j = 1, 2, \dots, m \tag{7}$$

where  $z_{\alpha^*}$  is the critical value such that the probability that the standard normal distribution  $z$  is greater than  $z_{\alpha^*}$  is  $\alpha^*$ , and

$$\alpha^* = \frac{\alpha}{2} + (l - 1) \times \frac{1 - \alpha}{s - 1} \quad \text{for } l = 1, 2, \dots, s. \tag{8}$$

Then, the normalized technical indicators for the  $i$ th trading day, not including the  $j$ th technical indicator which is replaced by the sample point  $x_{ij(l)}^*$ , are fed into the well-trained BP neural network constructed in the previous stage (i.e., BP\_Model<sub>pre</sub>) to obtain the forecasted normalized closing price of the stock/futures, denoted by  $\hat{y}_{i(l)}$ . Hence, the absolute percentage error (APE) of the forecast can be calculated by:

$$APE_{i(l)} = \left| \frac{\hat{y}_{i(l)} - y_i}{y_i} \right| \times 100\%$$

$$\text{for } i = 1, 2, \dots, n_{tr} + n_{te}; l = 1, 2, \dots, s \tag{9}$$

where  $y_i$  is the normalized actual closing price of the stock/futures as described in the first stage. Therefore, the impact of the  $j$ th normalized technical indicator on forecasting the normalized closing price of the stock/futures, denoted by  $IMP_j$ , can be evaluated through the mean absolute percentage error (MAPE) expressed as:

$$IMP_j = \frac{\sum_{i=1}^{n_{tr}+n_{te}} \sum_{l=1}^s APE_{i(l)}}{s \times (n_{tr} + n_{te})} \quad \text{for } j = 1, 2, \dots, m. \tag{10}$$

Next, starting from one, the normalized technical indicators are ranked in ascending order based on their  $IMP_j$ s. The normalized technical indicators for the  $i$ th trading day are then fed into the well-trained BP\_Model<sub>pre</sub> model, while the technical indicators whose ranks are not greater than  $r$  ( $r = 0, 1, \dots, m - 1$ ) are replaced by 0.5, to acquire the forecasted normalized closing price of the stock/futures, denoted by  $\hat{y}_{i(r)}$ , during the training and test periods. The reason for replacing a normalized technical indicator with 0.5, which is the mean of its maximum and minimum values, is that we intend to weaken its effect on forecasting the closing price; this will assist us to determine the best feature subset. The mean absolute percentage error (MAPE) of the forecast, while weakening the effect of normalized technical indicators whose ranks are not greater than  $r$ , can then be calculated by:

$$MAPE_{(r)} = \sum_{i=1}^{n_{tr}+n_{te}} \left| \frac{\hat{y}_{i(r)} - y_i}{y_i \times (n_{tr} + n_{te})} \right| \times 100\%$$

for  $r = 0, 1, \dots, m - 1$ . (11)

The direction of change (CD) regarding the MAPE of the forecast while weakening the effect of one more normalized technical indicator can then be expressed by:

$$CD_{(r)} = \begin{cases} +1 & \text{if } MAPE_{(r)} > MAPE_{(r-1)} \\ 0 & \text{if } MAPE_{(r)} = MAPE_{(r-1)} \\ -1 & \text{if } MAPE_{(r)} < MAPE_{(r-1)} \end{cases}$$

for  $r = 1, 2, \dots, m - 1$ . (12)

We then find a certain rank, say  $r^*$ , such that  $CD_{(r^*)} = -1$  and  $CD_{(r)} = +1$  (for  $r = r^* + 1, r^* + 2, \dots, m - 1$ ). This implies that the absolute percentage error of the forecast always increases while weakening the effects of normalized technical indicators with ranks  $r^* + 1, r^* + 2, \dots$ , and  $m - 1$ . Hence, the normalized technical indicators with ranks greater than  $r^*$ , denoted by  $(f_{i1}, f_{i2}, \dots, f_{i(m-r^*)})$  (for  $i = 1, 2, \dots, n$ , where  $n$  is the total number of trading days), are retained as input variables for constructing the final BP forecasting model. This completes the tasks of feature selection and finding the best feature subset, which were described by Fig. 1.

### 3.3 Construction and validation of final forecasting models

In the final stage, BP neural networks are re-applied to the training and test data, whose input variables are the retained normalized technical indicators in the previous stage, i.e.,  $(f_{i1}, f_{i2}, \dots, f_{i(m-r^*)})$ , to construct several forecasting models. Based on the simultaneous optimization of the training and test RMSEs, a final optimal BP neural model, named  $BP\_BP\_Model_{final}$ , is selected to represent the functional relationship between the retained normalized technical indicators and the normalized stock/futures closing price as forecasted. Furthermore, the preliminary optimal GP model constructed in the first stage, i.e.,  $GP\_Model_{pre}$ , is actually a computer program that is constructed automatically by using the most important normalized technical indicators. Hence, BP neural networks are also applied to the training and test data whose input variables are the normalized technical indicators that appeared in the  $GP\_Model_{pre}$  model, and another final optimal BP neural model is determined, namely  $GP\_BP\_Model_{final}$ . This latter model is selected based on the training and test RMSEs and used for comparison to the  $BP\_BP\_Model_{final}$  model. Therefore, the forecasted closing price of the stock/futures, denoted by  $\hat{p}_i$ , can be obtained by inputting the retained normalized technical indicators  $(f_{i1}, f_{i2}, \dots, f_{i(m-r^*)})$  for each trading day to

$BP\_BP\_Model_{final}$ , or by inputting the normalized technical indicators which appeared in the  $GP\_Model_{pre}$  model in each trading day to  $GP\_BP\_Model_{final}$ , and denormalizing the output. Finally, through statistical metrics of the root mean squared error (RMSE) and mean absolute percentage error (MAPE), the effectiveness of the proposed forecasting procedure can be evaluated by applying the  $BP\_BP\_Model_{final}$  and  $GP\_BP\_Model_{final}$  models to the validation data that were not previously used. The RMSE and MAPE regarding the forecast for the validation data are defined as follows:

$$RMSE_{forecast} = \sqrt{\sum_{i=1}^{n_a} \frac{(p_i - \hat{p}_i)^2}{n_{va}}}$$

(13)

$$MAPE_{forecast} = \sum_{i=1}^{n_a} \frac{|p_i - \hat{p}_i|/p_i}{n_{va}} \times 100\%$$

(14)

where  $p_i$  and  $\hat{p}_i$  are the actual and forecasted closing price of the stock/futures, respectively, and  $n_{va}$  is the total number of trading days in the validation data. At this stage, the tasks shown in the lower part of Fig. 1 are completed.

### 3.4 Proposed forecasting procedure

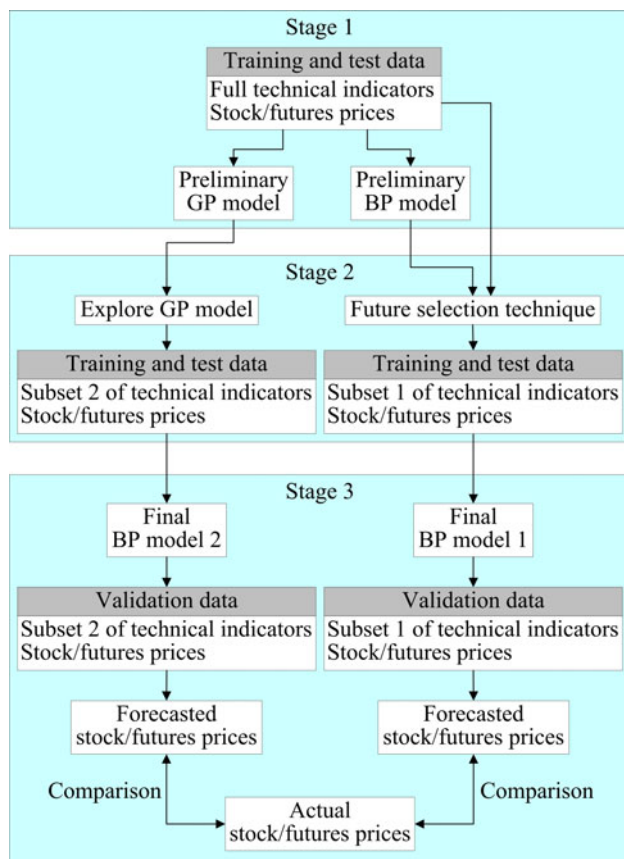
The proposed forecasting procedure in this study is conceptually illustrated in Fig. 3 and summarily restated as follows:

Stage 1: Construction of preliminary forecasting models

1. Collect the essential historical stock trading data and calculate the required technical indicators.
2. Normalize the technical indicators and closing prices of the stock/futures using (1) and (2), respectively.
3. Partition the normalized technical indicators along with the normalized closing price in the future into training, test, and validation data.
4. Apply the BP neural network and GP algorithm to the training and test data to construct preliminary optimal forecasting models.

Stage 2: Feature selection

1. Calculate the mean and standard deviation of each normalized technical indicator during a number of successive trading days through (3) and (5), respectively.
2. Assess the impact of each technical indicator on forecasting the closing price of the stock/futures through (10).
3. Determine the best subset of normalized technical indicators according to the direction of change regarding the mean absolute percentage error of the forecast, while weakening the effect of one more normalized technical indicator, as shown in (12).



**Fig. 3** Proposed forecasting procedure

Stage 3: Construction of final forecasting models and validation

1. Apply the BP neural network to the training and test data with input normalized technical indicators retained in Stage 2 to construct the final optimal forecasting model.
2. Using as inputs the normalized technical indicators that appeared in the preliminary optimal GP forecasting model in Stage 1, apply the BP neural network to the training and test data to construct another final optimal forecasting model.
3. Evaluate via (13) and (14) the effectiveness of the proposed forecast procedure by applying the two final optimal forecasting models constructed in Steps 1 and 2 of Stage 3 to the validation data.

## 4 Case study

In this section, a case study on forecasting the closing prices of Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) futures of the spot month is presented to demonstrate the feasibility and effectiveness of

the proposed forecasting procedure. The details of the case study and the procedure are described below.

### 4.1 Constructing preliminary forecasting models

#### 4.1.1 Preparing the experimental data

A total of 2,353 pairs of daily trading data, including opening price, highest price, lowest price, closing price, and trade volume, are first collected from the Taiwan Stock Exchange Corporation (TWSE) over a period of approximately nine-and-a-half years, from January 2, 2001, to June 30, 2010. Fifteen technical indicators are then selected as the input variables for forecasting the closing price of futures. This is in line with previous studies by Kim and Han [36], Kim and Lee [37], Tsang et al. [38], Chang and Liu [8], Ince and Trafalis [9], Huang and Tsai [10], and Lai et al. [39]. The formulas for these technical indicators are presented in “Appendix”, and they include the following: (1) 10-day moving average, (2) 20-day bias, (3) moving average convergence/divergence, (4) 9-day stochastic indicator K, (5) 9-day stochastic indicator D, (6) 9-day Williams overbought/oversold index, (7) 10-day rate of change, (8) 5-day relative strength index, (9) 24-day commodity channel index, (10) 26-day volume ratio, (11) 13-day psychological line, (12) 14-day plus directional indicator, (13) 14-day minus directional indicator, (14) 26-day buying/selling momentum indicator, and (15) 26-day buying/selling willingness indicator. The fifteen technical indicators are calculated according to the formulas in “Appendix” and the initial 2,353 pairs of daily trading data. Notably, the technical indicators for the first few days, beginning with January 2, 2001, are not available due to the technical indicator definitions. For example, the 9-day stochastic indicator K can be obtained only for the period after the 9th trading day from January 2, 2001. After this limitation and the validity of the technical indicators are taken into account, there are 2,318 pairs of technical indicator data. These data, along with the closing prices from March 1, 2001, to June 30, 2010, are used as the sample data in the follow-up forecasting procedure.

#### 4.1.2 Constructing models for a forecast of 1 day ahead

The above fifteen technical indicators and the closing price in each trading day are normalized into a range between 0 and 1 using the formulas in (1) and (2), respectively. These normalized technical indicators in each trading day from March 1, 2001, to June 29, 2010, along with the normalized closing price of the next trading day from March 2, 2001, to June 30, 2010, are then partitioned into training, test, and validation sample data groups, based on the proportion of 4:1:1, as shown in Table 1. Notably, training data are



**Table 1** Data partitioned into training, test, and validation data groups

Dataset	Training period	Test period	Validation period	Dataset size
1	2001/03/01–2001/07/09	2001/07/10–2001/08/21	2001/08/22–2001/10/08	150
2	2001/10/09–2002/02/22	2002/02/25–2002/04/09	2002/04/10–2002/05/22	150
3	2002/05/23–2002/09/26	2002/09/27–2002/11/08	2002/11/11–2002/12/20	150
4	2002/12/23–2003/05/08	2003/05/09–2003/06/20	2003/06/23–2003/08/01	150
5	2003/08/04–2003/12/09	2003/12/10–2004/01/29	2004/01/30–2004/03/11	150
6	2004/03/12–2004/07/16	2004/07/19–2004/08/31	2004/09/01–2004/10/13	150
7	2004/10/14–2005/02/25	2005/03/01–2005/04/12	2005/04/13–2005/05/25	150
8	2005/05/26–2005/10/03	2005/10/04–2005/11/15	2005/11/16–2005/12/27	150
9	2005/12/28–2006/05/15	2006/05/16–2006/06/27	2006/06/28–2006/08/08	150
10	2006/08/09–2006/12/14	2006/12/15–2007/01/26	2007/01/29–2007/03/20	150
11	2007/03/21–2007/07/27	2007/07/30–2007/09/07	2007/09/10–2007/10/24	150
12	2007/10/25–2008/03/10	2008/03/11–2008/04/22	2008/04/23–2008/06/04	150
13	2008/06/05–2008/10/13	2008/10/14–2008/11/24	2008/11/25–2009/01/07	150
14	2009/01/08–2009/05/21	2009/05/22–2009/07/03	2009/07/06–2009/08/17	150
15	2009/08/18–2009/12/21	2009/12/22–2010/02/02	2010/02/03–2010/03/24	150
16	2010/03/25–2010/05/21	2010/05/24–2010/06/09	2010/06/10–2010/06/29	67
Total number of sample data	1,391	463	463	2,317

usually given preference over test data. Thus, the size of training data is set as four times the size of test data. In addition, validation data of a size equal to the size of the test data are designed to verify the capability of the well-trained BP neural network or GP models obtained by using training and test data to deal with data not previously used. With this approach, the influence of partitioned sample data on the performance of the obtained forecasting model can be presumed to be eliminated. Next, Table 1 contains 16 subsets, which are achieved by slicing periods of time in order to ensure that the training, test, and validation sample data groups can cover the entire period of research. Through this manner, it is believed that the functional relationship between the normalized technical indicators and the normalized closing price of the next trading day can be estimated more accurately by using the BP neural network or GP models constructed later. Based on the training and test data in Table 1, a three-layered BP neural network is then designed by using NeuralWorks Professional II/Plus (<http://www.neuralware.com>) software to create a mathematical forecasting model. The total number of neurons in the input and output layers is set to 15, which is the total number of technical indicators, and 1, which is the closing price of futures on the next trading day, respectively. In addition, the learning rate and momentum are set, through trial and error, to 0.25 and 0.75, respectively, and the default values in software are applied for the remaining parameters. A hidden layer analysis is attempted where the learning process is terminated when the training RMSE sufficiently converges and over-training is prevented by observing the test RMSE. The hidden layer has a

total number of neurons from 1 to 30. Table 2 summarizes the execution results. The structure 15-X-1 indicates that the BP neural model consists of X neurons in the hidden layer. Based on the simultaneous consideration of the training and test RMSEs, the neural model with the structure 15-20-1 is selected as the preliminary optimal BP forecasting model for forecasting the normalized closing price of the next trading day. It is named BP\_Model<sub>pre\_1\_day</sub>. For comparison, the GP algorithm is also applied to the training and test data in Table 1 to establish a futures price forecasting model. Here, the Discipulus 4.0 (<http://www.rmltech.com>) software is employed where the fitness of a program is evaluated through RMSE. The termination criterion is that the generations without improvement have reached 100 and the remaining parameters are set as their default values in the software. The Discipulus 4.0 software is executed 5 times, and Table 3 summarizes the obtained results. Based on the training and test RMSEs, the 2nd model, described as GP\_Model<sub>pre\_1\_day</sub>, is selected as the preliminary optimal GP forecasting model.

#### 4.1.3 Constructing models for a forecast of 2 and 3 days ahead

Similarly, we intend to construct the BP neural network and GP models for forecasting the closing futures prices of the next second and third trading days. For forecasting the closing price of the next second trading day, the fifteen normalized technical indicators in each trading day from March 1, 2001, to June 28, 2010, along with the normalized

**Table 2** Execution results of BP neural networks (forecast of 1 day ahead)

Structure (15-X-1)	Training RMSE	Test RMSE	Structure (15-X-1)	Training RMSE	Test RMSE	Structure (15-X-1)	Training RMSE	Test RMSE
15-1-1	0.0147	0.0157	15-11-1	0.0116	0.0123	15-21-1	0.0116	0.0123
15-2-1	0.0137	0.0159	15-12-1	0.0114	0.0122	15-22-1	0.0117	0.0124
15-3-1	0.0121	0.0127	15-13-1	0.0113	0.0119	15-23-1	0.0112	0.0118
15-4-1	0.0121	0.0131	15-14-1	0.0112	0.0120	15-24-1	0.0116	0.0126
15-5-1	0.0118	0.0128	15-15-1	0.0113	0.0119	15-25-1	0.0111	0.0116
15-6-1	0.0117	0.0126	15-16-1	0.0111	0.0119	15-26-1	0.0114	0.0119
15-7-1	0.0116	0.0124	15-17-1	0.0113	0.0118	15-27-1	0.0111	0.0119
15-8-1	0.0114	0.0122	15-18-1	0.0113	0.0115	15-28-1	0.0112	0.0117
15-9-1	0.0117	0.0129	15-19-1	0.0111	0.0117	15-29-1	0.0114	0.0116
15-10-1	0.0117	0.0124	15-20-1	0.0111	0.0115	15-30-1	0.0118	0.0130

**Table 3** Execution results of GP algorithms (forecast of 1 day ahead)

Model no.	Training RMSE	Test RMSE
1	0.0113	0.0118
2	0.0112	0.0116
3	0.0114	0.0117
4	0.0116	0.0116
5	0.0118	0.0121

closing price of the second trading day, i.e., from March 5, 2001, to June 30, 2010, are partitioned into training, test, and validation sample data groups based on the proportion of 4:1:1. The total numbers of paired data in the training, test, and validation sample data groups are 1,391, 463, and 462, respectively. With regard to the forecasting of the closing price of the next third trading day, the sample data are formed by the paired normalized technical indicators from March 1, 2001, to June 25, 2010, along with the normalized closing price of the third trading day, i.e., from March 6, 2001, to June 30, 2010. The sample data are also partitioned into training, test, and validation sample data groups, with sample sizes of 1,391, 463, and 461, respectively, based on the proportion of 4:1:1.

Next, a three-layered BP neural network using Neural-Works Professional II/Plus software is applied to the above training and test data regarding the closing price forecast of the next second and third trading days, where the BP parameters are set to those used previously. Table 4 summarizes the partial execution results. According to this table, the neural models with the structures 15-20-1 and 15-25-1 are selected as the preliminary optimal BP forecasting models, based on the training and test RMSEs, for forecasting the normalized closing prices of the next second and third trading days; they are identified, respectively, as BP\_Model<sub>pre\_2\_day</sub> and BP\_Model<sub>pre\_3\_day</sub>. Finally, the GP algorithm using Discipulus 4.0 software is reapplied to

the above training and test data concerning the closing price forecast of the next second and third trading days, where the fitness function, termination criterion, and the other remaining parameters are set to those used previously. Table 5 summarizes the results obtained by executing the Discipulus 4.0 software 5 times. Based on the training and test RMSEs, the 1st and 4th models are selected as the preliminary optimal GP forecasting models for forecasting the normalized closing prices of the next second and third trading days; they are identified, respectively, as GP\_Model<sub>pre\_2\_day</sub> and GP\_Model<sub>pre\_3\_day</sub>.

## 4.2 Selecting features

### 4.2.1 Evaluating the impact of technical indicators

Consider the training and test periods. Starting from a certain trading date, the mean of each normalized technical indicator in successive trading days is first calculated by (3). Notably, parameters  $n_{tr}$  and  $n_{te}$  are set to 1,391 and 463, which are, respectively, the total number of training and test paired data, as shown in Table 1. In addition, parameter  $m$  is set to 15, which is the total number of normalized technical indicators, and parameter  $t$  is set to 10, which stands for the successive 10 trading days considered in this study. Next, starting from a certain trading date, the standard deviation of each normalized technical indicator in the successive 10 trading days is calculated using (4). Thus, the mean of the standard deviation of each normalized technical indicator in the successive 10 trading days can be obtained by (5). Table 6 lists these values.

Next, we intend to assess the impact of each technical indicator on forecasting the closing price of the next trading day. Consider the first trading day in our study, March 1, 2001. The first normalized technical indicator, i.e., the 10-day moving average, is 0.381902. Hence, 200 sample points are initially drawn using the formulas in (7) and (8)

**Table 4** Partial execution results of BP neural networks (forecast of 2 and 3 days ahead)

BP models for forecast of 2 days ahead			BP models for forecast of 3 days ahead		
Structure (15-X-1)	Training RMSE	Test RMSE	Structure (15-X-1)	Training RMSE	Test RMSE
15-17-1	0.0146	0.0145	15-21-1	0.0171	0.0182
15-18-1	0.0144	0.0144	15-22-1	0.0173	0.0177
15-19-1	0.0144	0.0144	15-23-1	0.0171	0.0170
15-20-1	0.0144	0.0143	15-24-1	0.0172	0.0181
15-21-1	0.0146	0.0156	15-25-1	0.0170	0.0169
15-22-1	0.0148	0.0153	15-26-1	0.0171	0.0172
15-23-1	0.0145	0.0146	15-27-1	0.0170	0.0171
15-24-1	0.0147	0.0156	15-28-1	0.0171	0.0170
15-25-1	0.0144	0.0144	15-29-1	0.0171	0.0172
15-26-1	0.0145	0.0148	15-30-1	0.0173	0.0186

**Table 5** Execution results of GP algorithms (forecast of 2 and 3 days ahead)

GP models for forecast of 2 days ahead			GP models for forecast of 3 days ahead		
Model no.	Training RMSE	Test RMSE	Model no.	Training RMSE	Test RMSE
1	0.0147	0.0151	1	0.0176	0.0172
2	0.0162	0.0156	2	0.0177	0.0174
3	0.0155	0.0156	3	0.0177	0.0178
4	0.0148	0.0155	4	0.0176	0.0170
5	0.0158	0.0153	5	0.0176	0.0174

**Table 6** Summary of  $\bar{s}_j^{(10)}$ s

$j$	$\bar{s}_j^{(10)}$
1	0.014178
2	0.054988
3	0.032307
4	0.166352
5	0.134182
6	0.244025
7	0.062844
8	0.149794
9	0.076420
10	0.062330
11	0.089601
12	0.073836
13	0.076264
14	0.064473
15	0.054763

$\bar{s}_j^{(10)}$  represents the mean of the standard deviation of the  $j$ th normalized technical indicator in the successive 10 trading days during the training and test periods

where parameters  $\alpha$  and  $s$  are set to 0.01 and 200, respectively. For example, the first and fifth sample points are, respectively, obtained by (15) and (16), as shown:

$$0.381902 + z_{0.005}\bar{s}_1^{(10)} = 0.381902 + 2.576 \times 0.014178 = 0.418425 \tag{15}$$

$$0.381902 + z_{0.0249}\bar{s}_1^{(10)} = 0.381902 + 1.962 \times 0.014178 = 0.409719. \tag{16}$$

Then, the normalized technical indicators for the first trading day, not including the first technical indicator which is replaced by the first sample point, i.e., 0.418425, are fed into the neural network BP\_Model<sub>pre\_1\_day</sub> model. Thus, a forecasted value  $\hat{y}_{1(1)}$  regarding the normalized closing price of the next trading day, March 2, 2001, is obtained. The absolute percentage error (APE) of the forecast can be calculated based on (9) as follows:

$$APE_{1(1)} = \left| \frac{\hat{y}_{1(1)} - y_1}{y_1} \right| \times 100\% \tag{17}$$

where  $y_1$  and  $\hat{y}_{1(1)}$  are the normalized actual and forecasted closing price of futures on March 2, 2001. Similarly, the values of  $APE_{1(2)}$ ,  $APE_{1(3)}$ , ..., and  $APE_{1(200)}$  can be obtained by feeding the normalized technical indicators for the first trading day, not including the first technical indicator which is replaced by the remaining 199 sample points, into the BP\_Model<sub>pre\_1\_day</sub> neural model and calculating, in order, the forecast errors based on (9). Thus, the impact of the first normalized technical indicator on forecasting the normalized closing price of the next trading day can be evaluated through (10). By following the above

**Table 7** Summary of IMP<sub>j</sub>s

<i>j</i>	IMP <sub><i>j</i></sub>	Rank
1	0.015918	14
2	0.015729	12
3	0.014997	5
4	0.015760	13
5	0.015953	15
6	0.015044	10
7	0.015046	11
8	0.014999	8
9	0.015023	9
10	0.014993	1
11	0.014994	2
12	0.014999	7
13	0.014995	4
14	0.014995	3
15	0.014997	6

IMP<sub>*j*</sub> represents the impact of the *j*th normalized technical indicator on forecasting the normalized closing price of the next trading day

procedure, the impact of the remaining normalized technical indicators on forecasting the normalized closing price of the next trading day can be obtained, and Table 7 summarizes these values. This table also shows the ranks of IMP<sub>*j*</sub>s based on their values in ascending order.

4.2.2 Determining the best subset of technical indicators

Next, we intend to determine the best subset of technical indicators by evaluating the effect of weakening normalized technical indicators on forecasting the closing price. For example, the normalized technical indicators for the first trading day are fed into the BP\_Model<sub>pre\_1\_day</sub> neural model to acquire the forecasted normalized closing price of the next trading day  $\hat{y}_{1(3)}$ , while the technical indicators whose ranks are not greater than 3, i.e., the 10th, 11th, and 14th, are replaced by 0.5. Similarly, the values  $\hat{y}_{2(3)}$ ,  $\hat{y}_{3(3)}$ , ..., and  $\hat{y}_{1,854(3)}$  are acquired by feeding the normalized technical indicators for each trading day during the training and test periods into the BP\_Model<sub>pre\_1\_day</sub> neural model, while replacing the technical indicators with ranks of 1, 2, or 3 by 0.5 (the subscript 1,854 in  $\hat{y}_{1,854(3)}$  is the total number of sample data in the training and test periods, as shown in Table 1). Then, while weakening the effect of normalized technical indicators whose ranks are not greater than 3, the mean absolute percentage error (MAPE<sub>(3)</sub>) of the forecast can be obtained by (11) where parameters *n*<sub>tr</sub>, *n*<sub>te</sub>, and *m* are 1,391, 463, and 15, as described previously. Similarly, the values MAPE<sub>(0)</sub>, MAPE<sub>(1)</sub>, ..., and MAPE<sub>(14)</sub> can be

**Table 8** Summary of MAPE<sub>(*r*)</sub>s and CD<sub>(*r*)</sub>s

<i>r</i>	MAPE <sub>(<i>r</i>)</sub>	CD <sub>(<i>r</i>)</sub>
0	0.014993	NA
1	0.015110	+1
2	0.015459	+1
3	0.016111	+1
4	0.016801	+1
5	0.0172843	+1
6	0.0181275	+1
7	0.0215102	+1
8	0.0218827	+1
9	0.0205938	-1
10	0.0183378	-1
11	0.0274889	+1
12	0.1091111	+1
13	0.1402894	+1
14	0.4469097	+1

(1) MAPE<sub>(*r*)</sub> represents the mean absolute percentage error of the forecast while weakening the effect of normalized technical indicators whose ranks are not greater than *r*. (2) CD<sub>(*r*)</sub> represents the direction of change (CD) regarding the MAPE of the forecast while weakening the effect of one more normalized technical indicator of rank *r*, as illustrated in (10)

acquired by applying the above procedure, and Table 8 summarizes these values. Hence, the direction of change (CD) regarding the MAPE of the forecast, while weakening the effect of one more normalized technical indicator, can be obtained through (12) as shown in Table 8.

In Table 8, we can observe that CD<sub>(10)</sub> is -1 and all of CD<sub>(11)</sub>, CD<sub>(12)</sub>, ..., and CD<sub>(14)</sub> are +1. Thus, the normalized technical indicators with ranks 11–15 in Table 7, i.e., the 7th, 2nd, 4th, 1st, and 5th normalized technical indicators, denoted by  $(f_{i1}^{(1)}, f_{i2}^{(1)}, \dots, f_{i5}^{(1)})$ , are retained as input variables for forecasting the futures closing price of the next trading day later. Then, the above procedure is applied to the training and test sample data groups for the forecasting of the closing prices of the next second and third trading days, using the neural network models previously constructed, namely, BP\_Model<sub>pre\_2\_day</sub> and BP\_Model<sub>pre\_3\_day</sub>. Thus, the impact of each normalized technical indicator on forecasting the normalized closing prices of the next second and third trading days can be obtained, as summarized in Tables 9 and 10. Furthermore, the MAPE of the forecast while weakening the effect of a certain normalized technical indicator and the direction of change regarding the MAPE of the forecast while weakening the effect of one more normalized technical indicator can be acquired. This is shown in Tables 11 and 12. According to Table 11, CD<sub>(10)</sub> is -1 and CD<sub>(11)</sub>, CD<sub>(12)</sub>, ..., and CD<sub>(14)</sub> are all +1. Therefore, the normalized technical indicators

**Table 9** Summary of IMP<sub>j</sub>s (forecast of 2 days ahead)

<i>j</i>	IMP <sub><i>j</i></sub>	Rank
1	0.020175	14
2	0.019910	12
3	0.019478	4
4	0.020020	13
5	0.020234	15
6	0.019514	9
7	0.019538	11
8	0.019485	8
9	0.019531	10
10	0.019477	1
11	0.019479	5
12	0.019481	7
13	0.019478	2
14	0.019478	3
15	0.019480	6

The definition of IMP<sub>*j*</sub> is the same as that in Table 7

**Table 10** Summary of IMP<sub>j</sub>s (forecast of 3 days ahead)

<i>j</i>	IMP <sub><i>j</i></sub>	Rank
1	0.023764	13
2	0.023595	12
3	0.023190	5
4	0.023798	14
5	0.023996	15
6	0.023206	9
7	0.023230	10
8	0.023193	7
9	0.023262	11
10	0.023187	1
11	0.023200	8
12	0.023189	3
13	0.023189	4
14	0.023189	2
15	0.023191	6

The definition of IMP<sub>*j*</sub> is the same as that in Table 7

with ranks 11–15 in Table 9, i.e., the 7th, 2nd, 4th, 1st, and 5th normalized technical indicators denoted by  $(f_{i1}^{(2)}, f_{i2}^{(2)}, \dots, f_{i5}^{(2)})$ , are retained as input variables for forecasting the futures closing price of the next second trading day later. Similarly, it is found that CD<sub>(9)</sub> is −1 and CD<sub>(10)</sub>, CD<sub>(11)</sub>, ..., and CD<sub>(14)</sub> are all +1. Therefore, the normalized technical indicators with ranks 10–15 in Table 10, i.e., the 7th, 9th, 2nd, 1st, 4th, and 5th normalized technical indicators denoted by  $(f_{i1}^{(3)}, f_{i2}^{(3)}, \dots, f_{i6}^{(3)})$ , are retained as input variables for forecasting the futures closing price of the next third trading day later.

### 4.3 Constructing the final forecasting models

To construct the final BP neural model for forecasting the futures closing price of the next trading day, a three-layered BP neural network is applied again to the training and test data in Table 1, where the input variables are the retained normalized technical indicators  $(f_{i1}^{(1)}, f_{i2}^{(1)}, \dots, f_{i5}^{(1)})$  and the output variable is the normalized closing price of the next trading day. Hence, the neural model consists of 5 and 1 neurons in the input and output layers, respectively, and the remaining parameters are the same as those in Sect. 4.1.2. Table 13 summarizes the partial execution results. The 5-24-1 neural model, named BP\_BP\_Model<sub>final\_1\_day</sub>, is selected as the final optimal forecasting model. Similarly, BP neural networks can be applied to the training and test data to construct the final forecasting models for the forecast of the closing prices in

**Table 11** Summary of MAPE<sub>(*r*)</sub>s and CD<sub>(*r*)</sub>s (forecast of 2 days ahead)

<i>r</i>	MAPE <sub>(<i>r</i>)</sub>	CD <sub>(<i>r</i>)</sub>
0	0.019477	NA
1	0.019533	+1
2	0.019695	+1
3	0.020208	+1
4	0.020441	+1
5	0.020610	+1
6	0.022273	+1
7	0.023354	+1
8	0.023650	+1
9	0.022935	−1
10	0.022821	−1
11	0.027522	+1
12	0.054160	+1
13	0.070317	+1
14	0.258963	+1

The definitions of MAPE<sub>(*r*)</sub> and CD<sub>(*r*)</sub> are the same as those in Table 8

the next second and third trading days, as described in Sect. 4.1.3. When forecasting the closing price of the next second trading day, the normalized technical indicators  $(f_{i1}^{(2)}, f_{i2}^{(2)}, \dots, f_{i5}^{(2)})$  retained in the previous section serve as the input variables and the normalized closing price of the second trading day serves as the output variable. On the other hand, when creating the BP model for the forecast of the closing price of the next third trading day, the input

**Table 12** Summary of MAPE<sub>(r)</sub>s and CD<sub>(r)</sub>s (forecast of 3 days ahead)

<i>r</i>	MAPE <sub>(r)</sub>	CD <sub>(r)</sub>
0	0.023186	NA
1	0.023258	+1
2	0.024192	+1
3	0.025207	+1
4	0.025772	+1
5	0.025132	-1
6	0.027753	+1
7	0.028092	+1
8	0.027474	-1
9	0.027024	-1
10	0.028532	+1
11	0.032618	+1
12	0.061546	+1
13	0.255470	+1
14	0.263035	+1

The definitions of MAPE<sub>(r)</sub> and CD<sub>(r)</sub> are the same as those in Table 8

variables are the retained normalized technical indicators  $(f_{i1}^{(3)}, f_{i2}^{(3)}, \dots, f_{i6}^{(3)})$  and the output variable is the normalized closing price of the third trading day. Table 13 summarizes the partial execution results. The 5-26-1 and 6-35-1 neural models are selected as the final optimal model for forecasting the closing prices of the next second and third trading days and are, respectively, named BP\_BP\_Model<sub>final\_2\_day</sub> and BP\_BP\_Model<sub>final\_3\_day</sub>.

In addition, the preliminary optimal GP forecasting model GP\_Model<sub>pre\_1\_day</sub> contains the 1st, 2nd, 4th, 5th, and 13th normalized technical indicators, denoted by

$(f_{i1}^{(1)}, f_{i2}^{(1)}, \dots, f_{i5}^{(1)})$ . Hence, to construct the final optimal forecasting model, a BP neural network is used again based on the training and test data in Table 1, where the input variables are  $(f_{i1}^{(1)}, f_{i2}^{(1)}, \dots, f_{i5}^{(1)})$  and the output is the normalized closing price of the next trading day. Similarly, a BP neural network can be applied to the training and test data, as described in Sect. 4.1.2, to construct the final forecasting models for the forecast of the closing prices in the next second trading day, where the input variables are the 1st, 2nd, 5th, and 6th normalized technical indicators, denoted by  $(f_{i1}^{(2)}, f_{i2}^{(2)}, f_{i5}^{(2)}, f_{i6}^{(2)})$ , that appeared in the preliminary optimal GP forecasting model (i.e., GP\_Model<sub>pre\_2\_day</sub>) and the output is the normalized closing price of the second trading day. When forecasting the closing price of the next third trading day, the 1st, 2nd, 4th, 5th, 9th, and 15th normalized technical indicators, denoted by  $(f_{i1}^{(3)}, f_{i2}^{(3)}, \dots, f_{i6}^{(3)})$ , that are contained in the preliminary optimal GP forecasting model (i.e., GP\_Model<sub>pre\_3\_day</sub>), serve as the input variables, and the normalized closing price of the third trading day serves as the output variable. Table 14 summarizes the partial execution results. The 5-23-1, 4-28-1, and 6-28-1 neural models are selected as the final optimal models for forecasting the closing prices of the next, next second, and next third trading days, and they are, respectively, named GP\_BP\_Model<sub>final\_1\_day</sub>, GP\_BP\_Model<sub>final\_2\_day</sub>, and GP\_BP\_Model<sub>final\_3\_day</sub>.

Next, the retained normalized technical indicators  $(f_{i1}^{(1)}, f_{i2}^{(1)}, \dots, f_{i5}^{(1)})$ ,  $(f_{i1}^{(2)}, f_{i2}^{(2)}, \dots, f_{i5}^{(2)})$ , and  $(f_{i1}^{(3)}, f_{i2}^{(3)}, \dots, f_{i6}^{(3)})$  for each trading day are fed into the appropriate corresponding forecast models, namely, BP\_BP\_Model<sub>final\_1\_day</sub>, BP\_BP\_Model<sub>final\_2\_day</sub>, and

**Table 13** Partial execution results of BP neural networks (with feature selection by BP)

BP models for forecast of 1 day ahead			BP models for forecast of 2 days ahead			BP models for forecast of 3 days ahead		
Structure (5-X-1)	Training RMSE	Test RMSE	Structure (5-X-1)	Training RMSE	Test RMSE	Structure (6-X-1)	Training RMSE	Test RMSE
5-21-1	0.0114	0.0118	5-21-1	0.0147	0.0145	6-30-1	0.0170	0.0177
5-22-1	0.0108	0.0110	5-22-1	0.0143	0.0141	6-31-1	0.0170	0.0178
5-23-1	0.0109	0.0110	5-23-1	0.0144	0.0140	6-32-1	0.0169	0.0176
5-24-1	0.0105	0.0109	5-24-1	0.0145	0.0142	6-33-1	0.0169	0.0175
5-25-1	0.0111	0.0114	5-25-1	0.0145	0.0143	6-34-1	0.0171	0.0169
5-26-1	0.0107	0.0109	5-26-1	0.0142	0.0140	6-35-1	0.0170	0.0167
5-27-1	0.0111	0.0114	5-27-1	0.0146	0.0143	6-36-1	0.0173	0.0170
5-28-1	0.0111	0.0112	5-28-1	0.0145	0.0141	6-37-1	0.0169	0.0172
5-29-1	0.0111	0.0113	5-29-1	0.0146	0.0142	6-38-1	0.0170	0.0179
5-30-1	0.0112	0.0114	5-30-1	0.0146	0.0143	6-39-1	0.0169	0.0177

**Table 14** Partial execution results of BP neural networks (with feature selection by GP)

BP models for forecast of 1 day ahead			BP models for forecast of 2 days ahead			BP models for forecast of 3 days ahead		
Structure (5-X-1)	Training RMSE	Test RMSE	Structure (4-X-1)	Training RMSE	Test RMSE	Structure (6-X-1)	Training RMSE	Test RMSE
5-21-1	0.0114	0.0121	4-23-1	0.0149	0.0148	6-25-1	0.0172	0.0166
5-22-1	0.0110	0.0115	4-24-1	0.0148	0.0148	6-26-1	0.0173	0.0166
5-23-1	0.0110	0.0112	4-25-1	0.0144	0.0143	6-27-1	0.0171	0.0169
5-24-1	0.0113	0.0116	4-26-1	0.0146	0.0143	6-28-1	0.0172	0.0165
5-25-1	0.0112	0.0116	4-27-1	0.0142	0.0144	6-29-1	0.0172	0.0165
5-26-1	0.0112	0.0117	4-28-1	0.0143	0.0142	6-30-1	0.0174	0.0172
5-27-1	0.0111	0.0115	4-29-1	0.0148	0.0148	6-31-1	0.0172	0.0174
5-28-1	0.0111	0.0115	4-30-1	0.0146	0.0146	6-32-1	0.0171	0.0171
5-29-1	0.0112	0.0115	4-31-1	0.0148	0.0148	6-33-1	0.0174	0.0171
5-30-1	0.0113	0.0116	4-32-1	0.0147	0.0146	6-34-1	0.0174	0.0171

BP\_BP\_Model<sub>final\_3\_day</sub>, to obtain the normalized forecasted futures prices for the next, next second, and next third trading days. The forecasted closing prices of futures can then be acquired by denormalizing these normalized forecasted futures prices. Similarly, the closing prices of the next, next second, and next third trading days can be acquired by inputting the normalized technical indicators  $(f_{i1}^{(1)}, f_{i2}^{(1)}, \dots, f_{i5}^{(1)})$ ,  $(f_{i1}^{(2)}, f_{i2}^{(2)}, f_{i2}^{(2)}, f_{i4}^{(2)})$ , and  $(f_{i1}^{(3)}, f_{i2}^{(3)}, \dots, f_{i6}^{(3)})$  for each trading day to the appropriate corresponding final optimal BP models, namely, GP\_BP\_Model<sub>final\_1\_day</sub>, GP\_BP\_Model<sub>final\_2\_day</sub>, and GP\_BP\_Model<sub>final\_3\_day</sub>, and denormalizing these output values.

#### 4.4 Measuring the forecast performance

To evaluate the overall performance of the proposed forecasting procedure, the statistics, including root mean squared error (RMSE) and mean absolute percentage error (MAPE), as shown in (13) and (14), are used to assess forecast errors, as shown in Table 15. For comparison, this table also lists the performance evaluation regarding the preliminary optimal BP and GP forecasting models. In addition, the maximum and minimum absolute percentage errors (APEs), denoted by APE<sub>max</sub> and APE<sub>min</sub>, respectively, are also available in this table. As shown in Table 15, some findings can be obtained from the forecasting results using the BP\_BP\_Model<sub>final\_1\_day</sub>, BP\_BP\_Model<sub>final\_2\_day</sub>, and BP\_BP\_Model<sub>final\_3\_day</sub> models. The overall MAPEs of forecasting the closing futures prices of the next, next second, and next third trading days are a mere 1.36, 1.86, and 2.26%, respectively. The absolute percentage of the difference between the actual and forecasted closing prices is, on average, only 2.26%, even with respect to the most difficult case of

forecasting, which is the forecast of the closing futures price of the next third trading day. This indicates that the proposed procedure in this study can yield a forecast of great precision for the closing futures prices over a given period of three trading days. In forecasting the closing futures price of the next trading day, the minimum APE attains the excellent level of 0.0001%, and the maximum APE is around 10.5%. In forecasting the closing futures price of the next third trading day, the maximum APE grows to 15.2% and the minimum APE still attains the excellent level of 0.0025%. The performance of forecasting the closing futures prices of the next 3 days via the BP\_BP\_Model<sub>final\_1\_day</sub>, BP\_BP\_Model<sub>final\_2\_day</sub>, and BP\_BP\_Model<sub>final\_3\_day</sub> models is almost better than that obtained through the GP\_BP\_Model<sub>final\_1\_day</sub>, GP\_BP\_Model<sub>final\_3\_day</sub>, and GP\_BP\_Model<sub>final\_3\_day</sub> models, with respect to RMSE, MAPE, the minimum APE, and the maximum APE. The only exception to this statement is that the maximum APE of BP\_BP\_Model<sub>final\_1\_day</sub> is slightly larger than that acquired by GP\_BP\_Model<sub>final\_1\_day</sub>. Hence, we can generally conclude that the feature selection method that uses the simulation technique based on the preliminary BP neural model is more effective in selecting important technical indicators for forecasting closing futures prices than the method that determines these indicators solely on the basis of the preliminary GP forecast models.

In addition, the final BP forecast models that use as input variables the retained normalized technical indicators based on the preliminary BP neural model and simulation technique perform better than their corresponding preliminary BP forecast models. Similarly, the final BP forecast models with input variables that appear in the preliminary GP models perform better than their corresponding preliminary GP forecast models, with one exception: the minimum APE of GP\_BP\_Model<sub>final\_2\_day</sub> is

**Table 15** Performance of the proposed forecasting procedure

Forecast model	Period	RMSE <sub>forecast</sub>	MAPE <sub>forecast</sub>	APE <sub>max</sub>	APE <sub>min</sub>
<i>(A) Forecast of 1 day ahead</i>					
BP_BP_Model <sub>final_1_day</sub>	Training period	113.039	0.013410	0.086722	0.000019
	Test period	116.705	0.014831	0.105037	0.000100
	Validation period	115.998	0.013014	0.086584	0.000001
	Overall	114.368	0.013615	0.105037	0.000001
GP_BP_Model <sub>final_1_day</sub>	Training period	117.949	0.014211	0.090430	0.000021
	Test period	120.158	0.015444	0.100601	0.000095
	Validation period	118.464	0.013633	0.080611	0.000040
	Overall	118.496	0.014342	0.100601	0.000021
BP_Model <sub>pre_1_day</sub>	Training period	119.064	0.014537	0.097810	0.000005
	Test period	123.621	0.016365	0.095604	0.000051
	Validation period	117.048	0.013395	0.084650	0.000020
	Overall	119.589	0.014674	0.097810	0.000005
GP_Model <sub>pre_1_day</sub>	Training period	120.499	0.014697	0.091853	0.000027
	Test period	124.635	0.016157	0.103731	0.000011
	Validation period	119.401	0.014314	0.073579	0.000038
	Overall	121.120	0.014912	0.103731	0.000011
<i>(B) Forecast of 2 days ahead</i>					
BP_BP_Model <sub>final_2_day</sub>	Training period	152.258	0.018586	0.133330	0.000011
	Test period	150.425	0.019126	0.120727	0.000095
	Validation period	155.303	0.018257	0.117606	0.000020
	Overall	152.505	0.018628	0.133330	0.000011
GP_BP_Model <sub>final_2_day</sub>	Training period	153.041	0.018914	0.131404	0.000124
	Test period	152.709	0.019974	0.127455	0.000063
	Validation period	157.469	0.018857	0.102810	0.000175
	Overall	153.867	0.019115	0.131404	0.000063
BP_Model <sub>pre_2_day</sub>	Training period	154.842	0.019179	0.137388	0.000012
	Test period	153.256	0.020370	0.133930	0.000026
	Validation period	155.539	0.018499	0.115810	0.000017
	Overall	154.671	0.019282	0.137388	0.000012
GP_Model <sub>pre_2_day</sub>	Training period	157.734	0.019717	0.131219	0.000040
	Test period	162.360	0.021397	0.148554	0.000029
	Validation period	156.344	0.018699	0.101481	0.000015
	Overall	158.395	0.019850	0.148554	0.000015
<i>(C) Forecast of 3 days ahead</i>					
BP_BP_Model <sub>final_3_day</sub>	Training period	182.333	0.022873	0.152231	0.000025
	Test period	179.406	0.022755	0.139770	0.000035
	Validation period	180.349	0.021591	0.112186	0.000039
	Overall	181.357	0.022594	0.152231	0.000025
GP_BP_Model <sub>final_3_day</sub>	Training period	184.531	0.023195	0.163091	0.000008
	Test period	176.586	0.023737	0.144942	0.000015
	Validation period	179.642	0.021784	0.108256	0.000055
	Overall	182.000	0.023023	0.163091	0.000008
BP_Model <sub>pre_3_day</sub>	Training period	182.322	0.022847	0.155054	0.000010
	Test period	181.389	0.024204	0.173746	0.000102
	Validation period	182.011	0.022119	0.117606	0.000018
	Overall	182.075	0.022974	0.173746	0.000010



**Table 15** continued

Forecast model	Period	RMSE <sub>forecast</sub>	MAPE <sub>forecast</sub>	APE <sub>max</sub>	APE <sub>min</sub>
GP_Model <sub>pre_3_day</sub>	Training period	188.505	0.023779	0.173600	0.000008
	Test period	181.884	0.023680	0.134497	0.000027
	Validation period	187.746	0.022407	0.096142	0.000058
	Overall	187.048	0.023486	0.173600	0.000008

somewhat larger than that of GP\_Model<sub>pre\_2\_day</sub>. Therefore, we can generally conclude that the technical indicators that are critical to the forecast of futures prices can be determined by applying a feature selection method that uses either a simulation technique based on the preliminary BP neural model or the screening technique of the preliminary GP forecast model. Based on the above analysis, the proposed forecasting procedure in this study can be considered an effective tool for forecasting futures prices in the next three trading days with the use of technical indicators.

## 5 Conclusions

With the inherent high volatility, complexity, and turbulence of stock/futures prices, financial forecasting is a challenging task that continues to be investigated by researchers and practitioners. The approaches undertaken in the past for tackling the stock/futures price forecasting problems can be broadly classified into three main categories: fundamental analysis, technical analysis, and traditional time series. Each of these approaches has its merits and limitations. In this study, a backpropagation (BP) neural network, a feature selection technique, and genetic programming (GP) were utilized to develop an integrated approach to deal with the stock/futures price forecasting problems. First, the BP neural network was used to construct a preliminary forecasting model that describes the complex nonlinear relationship between technical indicators and future stock/futures prices. Next, the feature selection technique based on simulation was utilized to explore the preliminary forecasting model; thus, the most important technical indicators for forecasting stock/futures prices were determined. For comparison, the GP algorithm was also employed to build another forecasting model that automatically screens the vital technical indicators which closely correlate with future stock/futures prices. Then, the BP neural network was applied again to construct the final stock/futures price forecasting model. At this stage, the technical indicators retained by the feature selection technique or GP algorithm were used as the input variables. Finally, the performance of the final forecasting model was evaluated and compared with that of the preliminary BP and GP forecasting models.

The feasibility and effectiveness of the proposed forecasting procedure were verified through a case study on forecasting the closing prices of Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) futures of the spot month over the period from January 2, 2001, to June 30, 2010. The obtained results showed that the overall mean absolute percentage errors (MAPEs) of forecasting the closing futures prices of the next and next second trading days were only 1.36 and 1.86%, respectively. Even in the most difficult case of forecasting, i.e., the forecast of the closing futures price in the next third trading day, the absolute percentage of the difference between the actual and forecasted closing prices was only 2.26%, on average. The analysis also indicated that the technical indicators which are critical to the forecast of futures prices can be determined by applying one of the following: a feature selection method that uses a simulation technique based on the preliminary BP neural model or the screening technique of the preliminary GP forecast model. Based on the above information, the proposed forecasting procedure can be considered a feasible and effective tool for stock/futures price forecasting.

However, the proposed forecasting procedure still has some limitations. First, during the construction of both the preliminary and final forecasting models, there are no absolute criteria based on the RMSE of the training and test data that will enable the selection of the optimal BP neural network or GP models. Next, the key parameter settings in BP neural networks or GP algorithms may influence the final training results; however, there are no exact rules for setting such key parameters. Finally, this study proposed the direction of change (CD) regarding the MAPE of the forecast and constructed the final BP forecasting models by developing a rule based on a simulation technique for selecting important normalized technical indicators as input variables. However, this is only a heuristic method, and the feasibility and effectiveness of the approach can only be verified through practical implementation. Further research directions suggested by this study might include using clustering techniques to divide the sample data into clusters such that the approximate functional model that describes an implicit mathematical relationship between the technical indicators and future closing stock/futures price can be

constructed more easily and precisely. In addition, optimizing the parameters of BP neural networks or GP algorithms through other soft computing methods (e.g., particle swarm optimization or ant colony optimization) might be another future research direction.

**Appendix: 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. 10-day moving average

The 10-day moving average is the mean price of the futures over the most recent 10 days and is calculated by:

$$MA_{10}_i = \frac{\sum_{j=i-9}^i CP_j}{10}. \tag{18}$$

2. 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:

$$BIAS_{20}_i = \frac{CP_i - MA_{20}_i}{MA_{20}_i}. \tag{19}$$

3. 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:

$$DI_i = (HP_i + LP_i + 2 \times CP_i)/4. \tag{20}$$

Next, define the 12-day exponential moving average (EMA\_12) and 26-day exponential moving average (EMA\_26) as:

$$EMA_{12}_i = \frac{11}{13} \times EMA_{12}_{i-1} + \frac{2}{13} \times DI_i \tag{21}$$

and

$$EMA_{26}_i = \frac{25}{27} \times EMA_{26}_{i-1} + \frac{2}{27} \times DI_i, \tag{22}$$

respectively. Then, the difference between EMA\_12 and EMA\_26 can be calculated by:

$$DIF_i = EMA_{12}_i - EMA_{26}_i. \tag{23}$$

Hence, the moving average convergence/divergence can be defined by:

$$MACD_i = \frac{8}{10} \times MACD_{i-1} + \frac{2}{10} \times DIF_i. \tag{24}$$

4. 9-day stochastic indicator K

The 9-day stochastic indicator K is defined as:

$$K_{9}_i = \frac{2}{3} \times K_{9}_{i-1} + \frac{1}{3} \times \frac{CP_i - LP_{9}_i}{HP_{9}_i - LP_{9}_i} \times 100. \tag{25}$$

where LP\_9<sub>*i*</sub> and HP\_9<sub>*i*</sub> are the lowest and highest prices of the previous 9 days, i.e., days  $i, i - 1, \dots, i - 7$ , and  $i - 8$ , respectively.

5. 9-day stochastic indicator D

The 9-day stochastic indicator D is defined as:

$$D_{9}_i = \frac{2}{3} \times D_{9}_{i-1} + \frac{1}{3} \times K_{9}_i. \tag{26}$$

where K\_9<sub>*i*</sub> is the 9-day stochastic indicator K of day  $i$ , as previously defined.

6. 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:

$$WMS\%R_{9}_i = \frac{HP_{9}_i - CP_i}{HP_{9}_i - LP_{9}_i} \tag{27}$$

where LP\_9<sub>*i*</sub> and HP\_9<sub>*i*</sub> are the lowest and highest prices of the previous 9 days, i.e., days  $i, i - 1, \dots, i - 7$ , and  $i - 8$ , respectively.

7. 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:

$$ROC_{10}_i = \frac{CP_i - CP_{i-10}}{CP_{i-10}} \times 100. \tag{28}$$

8. 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:

$$G_i = \begin{cases} CP_i - CP_{i-1} & \text{if } CP_i > CP_{i-1} \\ 0 & \text{otherwise} \end{cases}. \tag{29}$$

Similarly, the loss of day  $i$  is calculated by:

$$L_i = \begin{cases} CP_{i-1} - CP_i & \text{if } CP_i < CP_{i-1} \\ 0 & \text{otherwise} \end{cases}. \tag{30}$$

Next, the 5-day average gain (AG<sub>5</sub>) and 5-day average loss (AL<sub>5</sub>) can be calculated by:

$$AG_{5i} = \frac{4}{5} \times AG_{5i-1} + \frac{1}{5} \times G_i \tag{31}$$

and

$$AL_{5i} = \frac{4}{5} \times AL_{5i-1} + \frac{1}{5} \times L_i, \tag{32}$$

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

$$RSI_{5i} = \frac{AG_{5i}}{AG_{5i} + AL_{5i}} \times 100. \tag{33}$$

9. 24-day commodity channel index

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

$$TP_i = \frac{HP_i + LP_i + CP_i}{3}. \tag{34}$$

Next, calculate the 24-day simple moving average of the typical price (SMATP<sub>24</sub>) by:

$$SMATP_{24i} = \frac{\sum_{j=i-23}^i TP_j}{24}. \tag{35}$$

Then, the 24-day mean deviation (MD<sub>24</sub>) can be calculated by:

$$MD_{24i} = \frac{\sum_{j=i-23}^i |TP_j - SMATP_{24i}|}{24}. \tag{36}$$

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

$$CCI_{24i} = \frac{TP_i - SMATP_{24i}}{0.015 \times MD_{24i}}. \tag{37}$$

10. 26-day volume ratio

The 26-day volume ratio is defined by:

$$VR_{26i} = \frac{TVU_{26i} - TVF_{26i}/2}{TVD_{26i} - TVF_{26i}/2} \times 100\%. \tag{38}$$

where TVU<sub>26i</sub>, TVD<sub>26i</sub>, and TVF<sub>26i</sub> represent the total trade volumes of stock prices rising, falling, and holding, respectively, from the previous 26 days, i.e., days *i*, *i* - 1, ..., *i* - 24, and *i* - 25.

11. 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:

$$PSY_{13i} = \frac{TDU_{13i}}{13} \times 100\%. \tag{39}$$

where TDU<sub>13i</sub> is the total number of days with regard to stock price rises of the previous 13 days, i.e., days *i*, *i* - 1, ..., *i* - 11, and *i* - 12.

12. 14-day plus directional indicator

First, define plus directional movement (+DM) and minus directional movement (-DM) as:

$$+DM_i = HP_i - HP_{i-1} \tag{40}$$

and

$$-DM_i = LP_{i-1} - LP_i, \tag{41}$$

respectively. The plus true directional movement (+TDM) can be calculated by:

$$+TDM_i = \begin{cases} +DM_i & \text{if } +DM_i > -DM_i \text{ and } +DM_i > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{42}$$

Similarly, the minus true directional movement (-TDM) can be calculated by:

$$-TDM_i = \begin{cases} -DM_i & \text{if } +DM_i < -DM_i \text{ and } -DM_i > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{43}$$

Hence, the 14-day plus directional movement (+DM<sub>14</sub>) can be calculated by:

$$+DM_{14i} = \frac{13}{14} \times (+DM_{14i-1}) + \frac{1}{14} \times (+TDM_i). \tag{44}$$

Similarly, the 14-day minus directional movement (-DM<sub>14</sub>) can be calculated by:

$$-DM_{14i} = \frac{13}{14} \times (-DM_{14i-1}) + \frac{1}{14} \times (-TDM_i). \tag{45}$$

Next, define the true range (TR) as:

$$TR_i = \text{Max}\{HP_i - LP_i, |HP_i - CP_{i-1}|, |LP_i - CP_{i-1}|\}. \tag{46}$$

The 14-day true range (TR<sub>14</sub>) can be calculated by:

$$TR_{14i} = \frac{13}{14} \times TR_{14i-1} + \frac{1}{14} \times TR_i. \tag{47}$$

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

$$+DI_{14_i} = \frac{+DM_{14_i}}{TR_{14_i}}. \quad (48)$$

13. 14-day minus directional indicator

The 14-day minus directional indicator is defined as:

$$-DI_{14_i} = \frac{-DM_{14_i}}{TR_{14_i}}. \quad (49)$$

where  $-DM_{14_i}$  and  $TR_{14_i}$  are, respectively, the 14-day minus directional movement and 14-day true range of day  $i$ , as previously defined in (45) and (47).

14. 26-day buying/selling momentum indicator

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

$$AR_{26_i} = \frac{\sum_{j=i-25}^i (HP_j - OP_j)}{\sum_{j=i-25}^i (OP_j - LP_j)}. \quad (50)$$

15. 26-day buying/selling willingness indicator

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

$$BR_{26_i} = \frac{\sum_{j=i-25}^i (HP_j - CP_{j-1})}{\sum_{j=i-25}^i (CP_{j-1} - LP_j)}. \quad (51)$$

## References

- Zarandi MHF, Rezaee B, Turksen IB, Neshat E (2009) A type-2 fuzzy rule-based expert system model for stock price analysis. *Expert Syst Appl* 36(1):139–154. doi:10.1016/j.eswa.2007.09.034
- Thomsett MC (1998) *Mastering fundamental analysis*. Dearborn Financial Publishing, Chicago
- Thomsett MC (1999) *Mastering technical analysis*. Dearborn Financial Publishing, Chicago
- Box GEP, Jenkins GM (1970) *Time series analysis, forecasting and control*. Holden-Day, San Francisco
- Bollerslev T (2001) Generalized autoregressive conditional heteroskedasticity. *J Econ* 100(1):A307–A327
- Pai PF, Lin CS (2005) A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega-Int J Manag Sci* 33(6):497–505. doi:10.1016/j.omega.2004.07.024
- Chang PC, Liu CH (2008) A TSK type fuzzy rule based system for stock price prediction. *Expert Syst Appl* 34(1):135–144. doi:10.1016/j.eswa.2006.08.020
- Ince H, Trafalis TB (2008) Short term forecasting with support vector machines and application to stock price prediction. *Int J Gen Syst* 37(6):677–687. doi:10.1080/03081070601068595
- Huang CL, Tsai CY (2009) A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Syst Appl* 36(2):1529–1539. doi:10.1016/j.eswa.2007.11.062
- Liang X, Zhang HS, Mao JG, Chen Y (2009) Improving option price forecasts with neural networks and support vector regressions. *Neurocomputing* 72(13–15):3055–3065. doi:10.1016/j.neucom.2009.03.015
- Hadavandi E, Shavandi H, Ghanbari A (2010) Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. *Knowl Based Syst* 23(8):800–808. doi:10.1016/j.knsys.2010.05.004
- Lu CJ (2010) Integrating independent component analysis-based denoising scheme with neural network for stock price prediction. *Expert Syst Appl* 37(10):7056–7064. doi:10.1016/j.eswa.2010.03.012
- Cheng CH, Chen TL, Wei LY (2010) A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting. *Inf Sci* 180(9):1610–1629. doi:10.1016/j.ins.2010.01.014
- Fausett L (1994) *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Englewood Cliffs
- Patterson DW (1996) *Artificial neural networks: theory and applications*. Prentice Hall, Singapore
- Lin CT, Yeh HY (2009) Empirical of the Taiwan stock index option price forecasting model—applied artificial neural network. *Appl Econ* 41(15):1965–1972. doi:10.1080/00036840601131672
- Yazici B, Memmedli M, Aslanargun A, Asma S (2010) Analysis of international debt problem using artificial neural networks and statistical methods. *Neural Comput Appl* 19(8):1207–1216. doi:10.1007/s00521-010-0422-4
- Bilgili M, Sahin B (2010) Comparative analysis of regression and artificial neural network models for wind speed prediction. *Meteorol Atmos Phys* 109(1–2):61–72. doi:10.1007/s00703-010-0093-9
- Hashemi MR, Ghadampour Z, Neill SP (2010) Using an artificial neural network to model seasonal changes in beach profiles. *Ocean Eng* 37(14–15):1345–1356. doi:10.1016/j.oceaneng.2010.07.004
- Liu H, Motoda H (1998) *Feature selection for knowledge discovery and data mining*. Kluwer, Boston
- Piramuthu S (2004) Evaluating feature selection methods for learning in data mining applications. *Eur J Oper Res* 156(2):483–494. doi:10.1016/s0377-2217(02)00911-6
- Kohavi R, John GH (1997) Wrappers for feature subset selection. *Artif Intell* 97(1–2):273–324. doi:10.1016/s0004-3702(97)00043-x
- Battiti R (1994) Using mutual information for selecting features in supervised neural-net learning. *IEEE Trans Neural Netw* 5(4):537–550
- Jensen R, Shen Q (2005) Fuzzy-rough data reduction with ant colony optimization. *Fuzzy Sets Syst* 149(1):5–20. doi:10.1016/j.fss.2004.07.014
- Song XP, Ding YS, Huang JW, Ge Y (2010) Feature selection for support vector machine in financial crisis prediction: a case study in China. *Expert Syst* 27(4):299–310. doi:10.1111/j.1468-0394.2010.00546.x
- Bradley PS, Mangasarian OL, Street WN (1998) Feature selection via mathematical programming. *INFORMS J Comput* 10(2):209–217
- Mladenic D (2006) Feature selection for dimensionality reduction. In: Saunders G, Grobelnik M, Gunn S, ShaweTaylor J (eds) *Subspace, latent structure and feature selection*, vol 3940. Lecture notes in computer science, pp 84–102
- Sivanandam SN, Deepa SN (2008) *Introduction to genetic algorithms*. Springer, Berlin
- Koza JR, Keane MA, Streeter MJ, Mydlowec W, Yu J, Lanza G (2005) *Genetic programming IV: routine human-competitive machine intelligence*
- Koza JR, Streeter MJ, Keane MA (2008) Routine high-return human-competitive automated problem-solving by means of genetic programming. *Inf Sci* 178(23):4434–4452. doi:10.1016/j.ins.2008.07.028
- Ciglaric I, Kidric A (2006) Computer-aided derivation of the optimal mathematical models to study gear-pair dynamic by using genetic programming. *Struct Multidiscip Optim* 32(2):153–160. doi:10.1007/s00158-006-0004-3

32. Etemadi H, Rostamy AAA, Dehkordi HF (2009) A genetic programming model for bankruptcy prediction: empirical evidence from Iran. *Expert Syst Appl* 36(2):3199–3207. doi:[10.1016/j.eswa.2008.01.012](https://doi.org/10.1016/j.eswa.2008.01.012)
33. Hwang TM, Oh H, Choung YK, Oh S, Jeon M, Kim JH, Nam SH, Lee S (2009) Prediction of membrane fouling in the pilot-scale microfiltration system using genetic programming. *Desalination* 247(1–3):285–294. doi:[10.1016/j.desal.2008.12.031](https://doi.org/10.1016/j.desal.2008.12.031)
34. Bae H, Jeon TR, Kim S, Kim HS, Kim D, Han SS, May GS (2010) Optimization of silicon solar cell fabrication based on neural network and genetic programming modeling. *Soft Comput* 14(2):161–169. doi:[10.1007/s00500-009-0438-9](https://doi.org/10.1007/s00500-009-0438-9)
35. Langdon WB, Poli R (2002) *Foundations of genetic programming*. Springer, New York
36. Kim KJ, Han I (2000) Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Syst Appl* 19(2):125–132
37. Kim KJ, Lee WB (2004) Stock market prediction using artificial neural networks with optimal feature transformation. *Neural Comput Appl* 13(3):255–260. doi:[10.1007/s00521-004-0428-x](https://doi.org/10.1007/s00521-004-0428-x)
38. Tsang PM, Kwok P, Choy SO, Kwan R, Ng SC, Mak J, Tsang J, Koong K, Wong TL (2007) Design and implementation of NN5 for Hong Kong stock price forecasting. *Eng Appl Artif Intell* 20(4):453–461. doi:[10.1016/j.engappai.2006.10.002](https://doi.org/10.1016/j.engappai.2006.10.002)
39. Lai RK, Fan CY, Huang WH, Chang PC (2009) Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Syst Appl* 36(2):3761–3773. doi:[10.1016/j.eswa.2008.02.025](https://doi.org/10.1016/j.eswa.2008.02.025)