International Journal of Hybrid Information Technology
Vol. 9, No.6 (2016), pp. 443-454
http://dx.doi.org/10.14257/ijhit.2016.9.6.39

# An Application of Genetic Programming in Nonlinear Combining Forecasting

Yingxiao Zhou and Peng Zhao*

*School of Traffic and Transportation, Beijing Jiaotong University, China*
*E-mail: 13114207@bjtu.edu.cn; pzhao@bjtu.edu.cn*
**Corresponding author*

***Abstract***

*It has been deemed as an effective tool of forecasting performance improvement to combine different component forecasting models. However, current nonlinear combining models are not able to meet the requirement of high forecasting accuracy in practice. To tackle this challenge, this paper constructs a hybrid, named genetic programming and least squared estimation based nonlinear combining method (GPLSE-NC), of a standard genetic programming (GP) algorithm and the least square estimation (LSE) method, based on which a new nonlinear combined forecasting model is proposed. To verify the feasibility of the proposed model, based on the container throughput data of Shanghai Port from January 2004 to November 2015, 4 different forecasting models are constructed and compared with the proposed GPLSE-NC combining model in terms of three forecasting performance evaluation criteria. The empirical results show significant superiority of the GPLSE-NC model over its rivals, which reveals that the proposed model has a great potential to be a powerful nonlinearly combine forecasting approach.*

***Keywords****: Container throughput forecasting, nonlinear combining forecasting, genetic programming, least square estimation*

## 1. Introduction

The real economy system is characteristic of high complexity, volatility and uncertainty, moreover, every forecasting model has its unique precondition and scope of application, therefore it is impossible to find a universal model that can generate perfect forecasts in all kinds of backgrounds. Given this fact, it is one of the hottest and most important issues for academia and industry in terms of how to appropriately combine different component forecasting models.

In essence, all kinds of forecasting models are based on two assumptions: first, the model is able to concisely reflect the reality with an acceptable deviation; secondly, the development trend of economy is stable to some degree in the forecasting horizon. However, there exists no such a model that is able to cover all aspects of the real economy. Besides, the economy itself always keep changing. Therefore, every forecasting model suffers from uncertainty to some extent, which leads to that a model probably performs better than others in a period under some circumstance, but worse if the forecasting horizon or circumstance is changed. The critical challenge is that people cannot distinguish which is the best model in the forecasting. Consequently, it is reasonable and necessary to scientifically combine different models to obtain higher forecasting performance.

ISSN: 1738-9968 IJHIT
Copyright © 2016 SERSC

The remainder of this paper is organized as follows. Section 2 reviews the current studies of combining forecasting models. Section 3 elaborates the GP-LSE combining model. Section 4 simply describes 4 benchmark models. It is followed by an empirical study to demonstrate the feasibility of the proposed model in Section 5. Section 6 concludes this paper.

## 2. Literature Review

In order to obtain higher forecasting performance, Bates and Granger 1 first proposed that idea that combing two different forecasting models should generate more satisfactory forecasts, and they further provided the principles and formula of configuring the weights between different models by minimizing the variance of forecasting errors.

Ref. 2 extended the approach proposed in Ref. 1 to include more component models. Ref. 3 regarded the essence of the method in Refs.1~2 as the Least Squares Estimation subject to some special constraints, and argued that the sum of the weights between different component models was not necessarily equal to 1. Ref. 4 applied 4 models respectively developed by Wharton Econometrics, Chase Econometrics, Bureau of Economic Analysis and Data Resources, Inc. to GNP (Gross National Product) forecast, and the results showed the superiority of equal weighting over other more complex weight distribution approaches. Ref. 5 proposed a regression based combining method, where real observations are regarded as the response variable, outputs of the different component models are deemed as explanatory variables, and the weights of component models are determined by the regression coefficients. Ref. 6 proposed a superior matrix combining method with more robustness and demonstrated its advantage over the equal weighting, maximizing forecasting variance and regression based weighting in terms of the forecasting performance. Ref. 7 developed a combining method called AFTER that automatically calibrated the weights of component models in the light of new observed information. Refs. 8-9 addressed nonlinear combining methods from the perspective of artificial neural network (ANN).

To best of our knowledge, although the combined model generally performs better than its component models, [10-12] current combining methods are confronted with two main challenges, including (a) majority of current combining methods are linear, but relationships among variables in reality are frequently nonlinear; (b) some nonlinear combining approaches like the ANN-based are able to simulate the nonlinearity, they cannot generate the analytic expression of the combined model, though. To tackle these challenges and following Ref. 13, this paper proposes a nonlinear combining method called GP-LSE that has the powerful nonlinear fitting capacity and is able to provide the analytic expression of the combined model.

## 3. GPLSE-NC

### 3.1. Genetic Programming Based On Least Square Estimation (GPLSE)

GP, first proposed in Ref. 14, has the powerful ability of fitting nonlinearity and is able to generate the analytic expression, therefore has been widely applied to a wide range of areas including optimum control, symbol regression, solving partial differential equations, solving the equilibriums of a game, evolution of spontaneous behaviors and so on. [15-21] Considering the advantage of genetic programming, this paper takes the lead to apply it to the combined forecasting.

One of the key problems of the standard genetic programming algorithm is that it converges at a very slow speed. Although value mutation [22] and Gaussian mutation

 Copyright © 2016 SERSC

[23-26] can speed up the convergence of GP, these two algorithms are inclined to get trapped in local optimums. To overcome this problem, this section proposes a LSE based GP algorithm named GPLSE, which replaces with LSE the standard manner of searching the optimum in the parameter space by using crossover and mutation operators. GPLSE enjoys a relatively higher convergence speed and avoid getting to the local optimum. Besides, GPLSE applies a new fitness function to control the complexity of the forecasting formular, adopts a self-adaptive selection approach to enhance the evolution efficiency of the population, employs a dynamic crossover and mutation strategy to balance the convergence speed and the occurrence probability of morbid convergence. Main steps of GPLSE is presented in Figure 1.

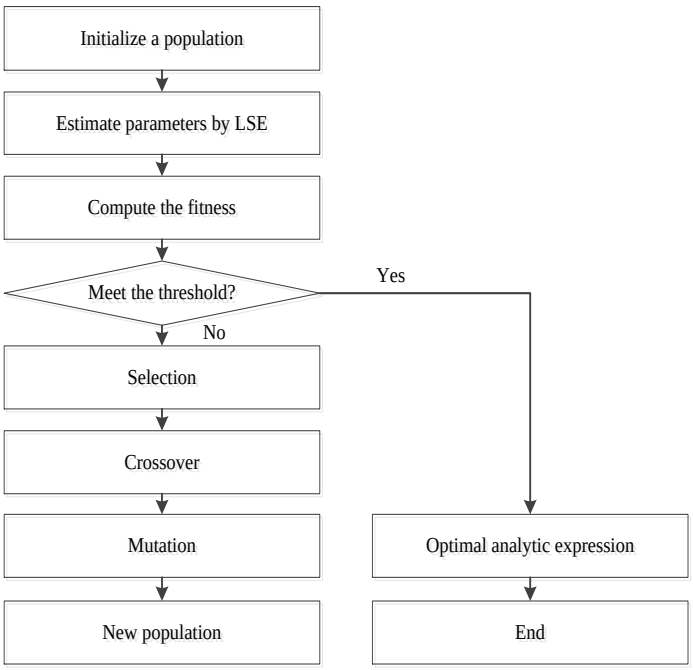


**Figure 1. Steps of GPLSE**

### 3.1.1. Population Initialization:

Denote a function set $F=\{+,-,\times,\div\}$ and a variable set $V=\{x_1,x_2,\cdots,x_m\}$, where $x_i$ is a variable of the target function to be generated. The population initialization algorithm constructs $n$ mathematical expressions by randomly selecting elements from the function set $F$ and variable set $V$, where $n$ stands for the population size. Every mathematic expression in the population is represented by a binary tree.

The detailed procedure of population initialization proceeds as follows. Firstly, randomly select $n$ operators from $F$ as the roots of $n$ binary trees. Secondly, keep selecting elements from the function set $F$ and variable set $V$ as leaf nodes for every root until all the leaves are from $V$. Figure 2 presents an example of the procedure of constructing a binary tree.

Copyright © 2016 SERSC

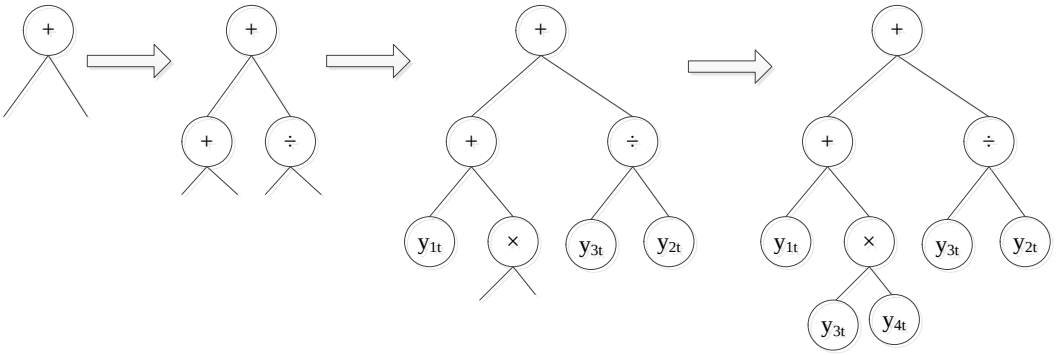


**Figure 2. The Procedure of Constructing a Binary Tree**

### 3.1.2. Using LSE to Estimate Parameters:

The corresponding mathematic expression of Figure 2 can be written as

$$w_t = \beta_0 + \beta_1 y_{1t} + \beta_2 y_{3t} \times y_{4t} + \beta_3 \left( y_{3t} \div y_{2t} \right). \tag{1}$$

Parameter estimation for Equation (1) using LSE proceeds as follows. (a) Rewrite Equation (1) as

$$\mathbf{W} = \mathbf{Y}\boldsymbol{\beta}, \tag{2}$$

where

$$\mathbf{W}^T = \left[ w_1, w_2, \cdots, w_m \right], \tag{3}$$

$$\boldsymbol{\beta}^T = \left[ \beta_0, \beta_1, \beta_2, \beta_3 \right], \tag{4}$$

$$\mathbf{Y} = \begin{bmatrix} 1 & y_{11} & y_{31} \times y_{41} & y_{31} \div y_{21} \\ 1 & y_{12} & y_{32} \times y_{42} & y_{32} \div y_{22} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_{1m} & y_{3m} \times y_{4m} & y_{3m} \div y_{2m} \end{bmatrix}. \tag{5}$$

In the light of the econometrics theory of LSE, the parameter vector $\boldsymbol{\beta}$ can be estimated by the formula

$$\boldsymbol{\beta} = \left( \mathbf{Y}^T \mathbf{Y} \right)^{-1} \left( \mathbf{Y}^T \mathbf{W} \right). \tag{6}$$

### 3.1.3 Computing the Fitness:

The fitness is a criterion used to compare forecasting performances of different models. In GP algorithm, the model with the highest fitness will be regarded as the best one. The mean square error (MSE) is one of the most frequently employed in the current literature, but it tends to complicate the procedure of determining the selection probability. Therefore, this section introduces the goodness-of-fit as the fitness criterion, which is written as

$$R^2 = \frac{\sum_{i=1}^{m} \left( w_i' - \overline{w} \right)^2}{\sum_{i=1}^{m} \left( w_i - \overline{w} \right)^2}, \tag{7}$$

where $w_i$ is the $i$-th observation, $w_i'$ is the forecasted value corresponding to $w_i$ and $\overline{w}$ is the mean of observations.

In order to decrease the computational complexity and avoid overgrowth of mathematical expressions, the length of expressions denoted by $l$ has to be controlled. Correspondingly, the new fitness function is proposed hereby and can be written as

 Copyright © 2016 SERSC

$$R^L = \frac{\sum_{i=1}^{m}\left(w_i - \bar{w}\right)^2}{\sum_{i=1}^{m}\left(w_i - \bar{w}\right)^2} + \frac{\lambda}{L}, \tag{8}$$

where $\lambda$ is a nonnegative scalar representing the weight of the formula's length. The bigger the value of $\lambda$, the heavier punishment will be imposed on a long mathematic expression.

Check the fitness of every expression in the population. If the fitness of an expression reaches the predefined threshold, this expression is considered the target expression. Otherwise, according to some rules, some expressions with higher fitness will be selected to go through the crossover and mutation procedures and consequently transformed to a new generation of expressions. iteratively, until there emerges an expression whose fitness meets the threshold.

### 3.1.4. Selection:

If there is no expression meeting the fitness threshold in the present generation of the population, the selection operator has to be implemented. The contribution of selection lies in that it can increase the occurrence probability of high-fitness expressions and decrease that of low-fitness expressions, by which the structure of the population is optimized.

The selection algorithm proposed in this section first selects expressions with the highest fitness to directly enter the next generation, and then select other candidates from the remainder by using the roulette wheel selection method. This strategy can insure that the fitness of next generation is no less, if not larger, than that of the present. It should be noted that standard GP algorithms are inclined to apply a fixed selection probability, however, for higher evolution efficiency, this paper refers to a new self-adaptive selection probability control strategy, which proceeds as follows.

Step 1. Compute the fitness of the $i$-th expression denoted by $R_i^L$ using Equation (8) and the selection probability $p_i$ using Equation (9)

$$p_i = \frac{R_i^L}{\sum_{i=1}^{n} R_i^L}, \tag{9}$$

where $n$ is the size of the present generation of population, *i.e.*, the total number of expressions in the population.

Step 2. Select an expressions with the fitness value equal to $p_{\max} = \max_i\left(p_i\right)$, put it into the next generation and keep it off the crossover and mutation operator.

Step 3. Use the roulette wheel selection method to choose $n-1$ expressions through $n-1$ rounds from the remainder of the present generation with a dynamically-adjusted probability. Denote $s_k^r$ $\left(r = n-1, n-2, \cdots, 1\right)$ the times of the $k$-th expression having been selected in the past $r$ rounds, then the selection probability of the $k$-th expression in the $\left(n-r\right)$-th round can be computed by Equation (10)

$$p_{n-r}^k = \frac{n \times p_k - s_k^r}{r}. \tag{10}$$

### 3.1.5. Crossover:

The crossover operator exchanges two parts of two different binary trees and is a key approach to the evolution of the present generation. Generally speaking, a crossover operator first selects two expressions from the present generation with a crossover probability $p_{cross}$, respectively. Then, randomly select a node in each of the trees. Finally,

Copyright © 2016 SERSC

exchange two sub-trees with the roots of the above randomly selected nodes, in this way two new trees are generated.

Figure 3 vividly describes an example of the crossover operator, where Tree1 and Tree2 are selected with the selection probability, Node1 and Node2 are randomly selected, and two sub-trees with the roots of Node1 and Node2 are exchanged with each other to generate the two new trees, *i.e.*, New Tree1 and New Tree2.

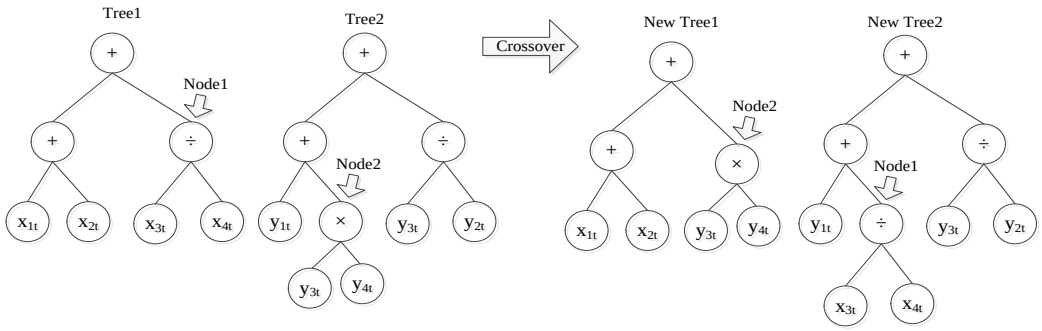


**Figure 3. An Example of Crossover**

### 3.1.6. Mutation:

Mutation is an indispensable tool to help with avoiding getting trapped into the local optimum, keeping the diversity of individual expressions, and decreasing the occurrence probability of morbid convergence. Except the individual with the highest fitness in the present generation, all of the remainder will go through the mutation procedure.

The mutation procedure proceeds as follows. (a) Select individuals for mutation from the present generation with the mutation probability. (b) Create a new binary tree. (c) Randomly select a non-leaf node, e.g., Node*, in the selected tree and replace the sub-tree with the root of Node* with the newly created tree. Figure 4 gives an example of mutation.

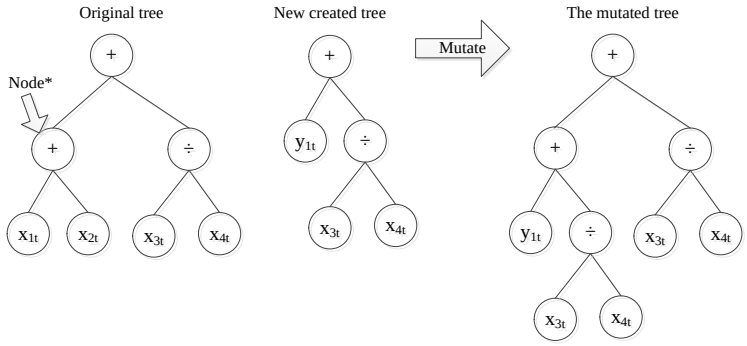


**Figure 4. An Example of Mutation**

### 3.1.7. Dynamically Adjusting Crossover and Mutation:

According to GP theory, individuals with higher fitness (good individuals) should be remained and those with low fitness (bad individuals) should be weeded out or changed into better ones so as to keep the population improving. Correspondingly, the bad individuals should mutate with higher probability, while the good ones with lower probability. This paper applies a dynamical adjustment policy, which automatically calibrates the crossover and mutation probability according to the individuals' fitness.

 Copyright © 2016 SERSC

Simply speaking, the policy increases the crossover and mutation probability when an individual's fitness is low and decrease the probability when an individual's fitness is high.

It is notable that a trade-off has to be taken into account between the diversity of population and the probability of morbid convergence, considering that lack of diversity frequently tends to result in higher risk of morbid convergence, but too high diversity will lead to the slow convergence. To tackle this problem, this paper dynamically adjusts the crossover and mutation probability in the light of a newly proposed criterion named 'generation growth margin (GGM)', *i.e.*, the growth of the highest individual fitness in terms of two consecutive generation. A value of GGM beyond the predefined reasonable range will trigger the adjustment of the crossover and mutation probability.

### 3.2. The Framework of GPLSE-NC

GPLSE-NC can be described as the following steps: (a) collect the data and split it into the training set and the testing set; (b) estimate the parameters of a variety of different component models using the training data set and validate these model using the testing data set; (c) following Ref. 27, the method proposed by select the component models to be combined. It should be noted that this step is crucial to the final forecasting performance, because if unsuitable component models are select, the combined results may even be worse; (d) apply GPLSE-NC to generate the combining model of the selected component models using the testing data set; (3) forecast the target using the selected component models and combine them into the final results by applying the combining model generated in step (d).

Figure 5 vividly describes the framework of GPLSE-NC.

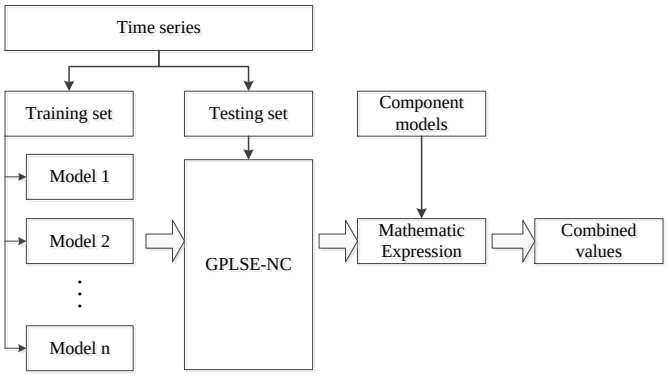


**Figure 5. The Framework of GPLSE-NC**

## 4. Component Models and Benchmark Combined Models

### 4.1. Component Models

#### 4.1.1. SARIMA:

As is widely known, ARIMA is a frequently used econometric model for forecasting purposes. However, considering that real economic time series often comprise seasonality, SARIMA is developed based on ARIMA for solving this concern. Generally, SARIMA model is frequently specified as SARIMA (p, d, q)(P,D,Q), which is written as

$$\varphi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D x_t = \theta_q(B)\Theta_Q(B^s)\varepsilon_t, \quad (11)$$

where

$$\varphi_p(B)=1-\varphi_1(B)-\varphi_2(B)-\cdots-\varphi_p(B),$$
$$\theta_q(B)=1+\theta_1(B)+\theta_2(B)+\cdots+\theta_q(B),$$
$$\Phi_P(B^S)=1-\phi_1(B^S)-\phi_2(B^S)-\cdots-\phi_P(B^S),$$
$$\Theta_Q(B^S)=1+\psi_1(B^S)+\psi_2(B^S)+\cdots+\psi_Q(B^S)$$
$$E(\varepsilon_t,\varepsilon_s)=0\ (t\neq s),\ D(\varepsilon_t)=\sigma^2,$$
$$E(\varepsilon_t)=0,\ E(x_t,\varepsilon_s)=0\ (t<s).$$

In the above equations, B denotes the backshift operator subjected to $B(x_t)=x_{t-1}$, while $x_t$ and $\varepsilon_t$ respectively denote the observation and the random term, at time point t. Parameters including p, P, d, D, q, Q and S represent the auto regressive order, seasonal auto regressive order, difference order, seasonal difference order, moving average order and seasonal moving average order, cycle length of seasonality, respectively.

#### 4.1.2. LSSVM:

Least square support vector machine (LSSVM), first proposed in Ref. 30, is an extension of SVM. LSSVM is still based on the VC dimension and structural risk theory, but its main difference from standard SVM is that LSSVM applies a least square linear system instead of a quadratic programming problem in SVM as the loss function. Therefore, LSSVM performs better than SVM in terms of the computation speed and meanwhile its forecasting accuracy is no worse than SVM.

The mechanism of LSSVM can be simply described as: given m observations denoted by $(x_{m\times s}, y_{m\times 1})$, select a kernel function $k(\cdot)$ to map the observations to the points in a high dimensional space, and then transform the nonlinear problem in the original space to a linear one in the new space by constructing a decision function of $y(x)=\omega\cdot k(x)+b$, correspondingly the solutions can be estimated by solving the following optimization problem

$$\min(\omega,\xi)=\frac{1}{2}\|\omega\|^2+c\sum_{i=1}^{m}\xi^2 \tag{12}$$
$$\text{s.t.}\quad y=\omega\cdot\phi(x)+b.$$

Details of LSSVM can be found in Ref. 31.

### 4.2. Benchmark Combined Models

#### 4.2.1. Minimizing Error (ME) Method:

ME is essentially a weighted average approach summing up different forecasts from various component models with a weight configuration that minimizes errors between the combined forecasted values and the true observed values.

Given forecasted values from n component models denoted by $\{\hat{y}_{it}\}^n$, the optimal weight configuration $\{\omega_i\}^n$ can be obtained by solving the following programming problem

$$\min\ \sum_{i,t}(\omega_i\hat{y}_{it}-y_t)^2 \tag{13}$$
$$\text{s.t}\quad \sum_i\omega_i=1$$

where $\hat{y}_i(t)$ stands for the forecasted value of the i-th component model, $y_t$ denotes the observed value, at period t. It should be noted that although $\omega_i$ is the weight of the i-th component model, it could be negative or larger than 1 in some special conditions, according to Ref. 32.

 Copyright © 2016 SERSC

#### 4.2.2 BP-ANN based Method:

Generally, a BP-ANN comprises 1 input layer, 1 or more hidden layers and 1 output layer. Neurons in adjacent layers are connected but those in the same layer are not. An example is graphically presented in Figure 6.

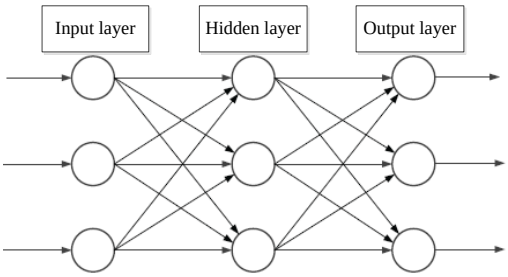


**Figure 6. A Three Layer BP-ANN**

It is notable that quantities of neurons in input layer, hidden layer and output layer can be different, especially for forecasting, the number of out-layer neuron is set to be 1.

BP-ANN has powerful ability of dealing with nonlinearity, considering that it is able to approach to any continuous integrable function with high enough accuracy on the condition that BP-ANN possesses enough neurons in the hidden layer. [32] Details about BP-ANN based integration can be found in Ref. 33.

## 5. Empirical Study

### 5.1. Data Description and Evaluation Criteria

In this study, the monthly data of container throughput of Shanghai port, from January 2004 to November 2015, are downloaded from the CEIC Macroeconomic Database. The data from January 2004 to November 2013 are used as training set, those belonging to 2014 the validation set, and the remainder the testing set. Considering that time series data frequently comprises different components, including the long-term trend, seasonality, cyclicity and random term, X12 algorithm is used to process the original data, as described by Figure 7. After that, the processed data is used to construct models.

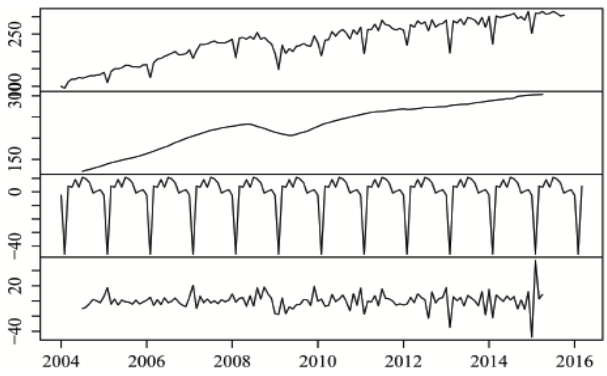


**Figure 7. Components of the Data of Shanghai Port**

To evaluate the forecasting performance of models, three most-frequently-used criteria are employed, including root mean square error (RMSE), total percentage error (TPE) and direction index (DI), written as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{x}_t - x_t)^2} \ , \tag{13}$$

$$TPE = \left|1 - \frac{\sum_{t=1}^{n}\hat{x}_t}{\sum_{t=1}^{n}x_t}\right| \ , \tag{14}$$

$$DI = \frac{\sum_{t=1}^{n} dr_t}{n} \tag{15}$$

where $x_t$ is the observed value, $\hat{x}_t$ is the simulated value, at time t, and n is the length of forecasting horizon. $dr_t$ is defined as

$$dr_t = \begin{cases} 1, if\ (y_t - y_{t-1})(\hat{y}_t - \hat{y}_{t-1}) > 0 \\ 0, if\ (y_t - y_{t-1})(\hat{y}_t - \hat{y}_{t-1}) < 0 \end{cases} . \tag{16}$$

RMSE evaluates the monthly average error of a model's forecasted values, TPE stands for the total forecasting error in the whole year, and DI calculates the probability where the forecasted values vibrate in the same direction as the real observations.

### 5.2. Empirical Results

This section applies an SARIMA model and a LSSVM model to forecasting the container throughput volume of Shanghai Port lying in the Yangtz River Delta, one of the most economic dynamic regions of China. Next, three combining methods including ME, the ANN-based and the newly proposed GPLSE-NC are employed to integrate the above two single models, *i.e.* SARIMA and LSSVM. After that, the above 5 models are compared with regard to their forecasting performance on the validation set and the testing set. The detailed results are presented in Table 1.

From Table 1, we can draw the following conclusions: (a) compared with the other models, GPLSE-NC performs best in terms of MSE, TPE, and DI on both the validation set and the testing set; (b) Although two combining models, *i.e.*, ME and ANN-based, perform better than two single models, *i.e.*, SARIMA and LSSVM, in terms of MSE on the validation set, both ME and ANN-based suffer from a lower DI, which implies that combining models will not necessarily perform better with reference to some evaluation criterion if the component models or the combining method is not suitably selected; (c) the values of MSE of the SARIMA and the LSSVM models go up significantly on the testing set, showing that the performance of single models tends to be subjected to high fluctuations. In contrast, the value of MSE of GPLSE-NC only increases a little bit, which indicates that GPLSE-NC has is able to consistently improve the forecasting performance.

**Table 1. Comparison of 5 Models' Forecasting Performance**

| Model | Validation set | | | Testing set | | |
|---|---|---|---|---|---|---|
| | *MSE* | *TPE* | *DI* | *MSE* | *TPE* | *DI* |
| SARIMA | 8.76 | 4.18% | 83% | 13.07 | 5.44% | 75% |
| LSSVM | 9.83 | 4.41% | 92% | 12.87 | 5.87% | 83% |
| ME | 8.08 | 3.93% | 83% | 9.24 | 5.01% | 83% |
| ANN-based | 8.42 | 4.78% | 83% | 9.51 | 4.85% | 83% |
| GPLSE-NC | 7.47 | 1.12% | 92% | 8.33 | 2.34% | 92% |

 Copyright © 2016 SERSC

# References

[1] J M. Bates and C. W. J. Granger, "The combination of forecasts. Operations Research Quarterly", vol. 20, **(1969)**, pp. 451–468.
[2] P. Newbold and C. W. J. Granger, "Experience with forecasting univariate time series and the combination of forecasts", Journal of the Royal Statistical Society: Series A, vol.137, **(1974)**, pp. 131–165.
[3] C. W J. Granger and R. Ramanathan, "Improved methods of forecasting", Journal of Forecasting, vol. 3, **(1984)**, pp. 197–204.
[4] R. T. Clemen and R. L. Winkler, "Combining economic forecasts. Journal of Business and Economic Statistics", vol. 4, **(1986)**, pp. 39–46.
[5] C. R. Nelson, "Applied Time Series Analysis for Managerial Forecasting", San Francisco: Holden-day, **(1973)**.
[6] S. Gupta and P. C. Wilton, "Combination of forecasts: An extension", Management Science, vol. 33, **(1987)**, pp. 356–372.
[7] Y H. Yang, "Combining forecasting procedures some theoretical results", Econometric Theory, vol. 20, **(2004)**, pp. 176–222.
[8] L. A. Yu, S. Y. Wang and K. K. Lai, "A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates", Computers & Operations Research, vol. 32, **(2004)**, pp. 2523–2541.
[9] L. A. Yu, S. Y. Wang and K. K. Lai, "A hybrid econometric-AI ensemble learning model for Chinese foreign trade prediction", Lecture Notes in Computer Science, vol. 4490, **(2007)**, pp. 106–113.
[10] R. L. Winkler and S. Makridakis, "The combination of forecasts", Journal of the Royal Statistical Society: Series A, vol. 146, **(1983)**, pp. 150–157.
[11] A. E. Bopp, "On combining forecasts: Some extensions and results", Management Science, vol. 31, **(1985)**, pp. 1492–1498.
[12] M. J. Lawrence, R. H. Edmundson and M. J. O'Connor, "The accuracy of combining judgemental and statistical forecasts". Management Science, vol. 32, **(1986)**, pp. 1521–1532.
[13] F. M. Yang, M. Li, A. Q. Huang and J. Li, "Forecasting time series with genetic programming based on least square method", Journal of Systems Science and Complexity, vol. 27, **(2014)**, pp. 117–129.
[14] J. R. Koza, "Genetic Programming: On the Program of Computer by Natural Selection", Cambridge Massachusetts, USA: MIT Press, **(1992)**.
[15] J. R. Koza, W. Mydlowec, G. Lanza, J, Yu and M. A. Keane, "Reverse engineering and automatic synthesis of metabolic pathways from observed data using genetic programming", Stanford Medical Informatics Technical Report, **(2000)**.
[16] J. R. Koza, W, Mydlowec, G. Lanza, J. Yu and M A Keane, "Reverse engineering of metabolic pathways from observed data using genetic programming", Pacific Symposium on Biocomputing, **(2001)**, pp. 434–445.
[17] C. Fonlupt, "Solving the ocean color problem using a genetic programming approach", Applied Soft Computing, vol. 1, **(2001)**, pp. 63–72.
[18] M. Sugimoto, S. Kikuchi and M. Tomita, "Reverse engineering of biochemical equations from time course data by means of genetic programming", BioSystems, vol. 80, **(2005)**, pp. 155–164.
[19] W. P. Worzel, J. J. Yu, A. A. Almal and A M Chinnaiyan, "Applications of genetic programming in cancer research", The International Journal of Biochemistry & Cell Biology, vol. 41, **(2009)**, pp. 405–413.
[20] H. C. Tsai, "Using weighted genetic programming to program squat wall strengths and tune asso ciated formulas", Engineering Applications of Artificial Intelligence, vol. 24, **(2011)**, pp. 526–533.
[21] M. Forouzanfar, A. Doustmohammadi, S. Hasanzade and G. H. Shakouri, "Transport energy demand forecast using multi-level genetic programming", Applied Energy, vol. 91, **(2012)**, pp. 496–503.
[22] M. Evett and T. Fernandez, "Numeric mutation improves the discovery of numeric constants in genetic programming", Proceedings of the Genetic and Evolutionary Computation Conference, **(1998)**, pp. 66–71.
[23] D. B. Fogel and J. W. Atmar, "Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems", Biological Cybernetics, vol. 63, **(1990)**, pp. 111–114.
[24] P. J. Angeline, "Advances in Genetic Programming", Volume 2, Cambridge, USA: MIT Press, **(1996)**.
[25] M, Sebag, M. Schoenauer and C. Ravise, "Inductive learning of mutation step-size in evolutionary parameter optimization", Proceedings of the Conference of Evolutionary Programming, **(1997)**, pp. 247–261.
[26] C. Ryan and M. Keijzer, "An analysis of diversity of constants of genetic programming", Proceedings of the European Conference of Genetic Programming, **(2003)**, pp. 404–413.
[27] A. Q. Huang, J. Xiao and S. Y. Wang, "A combined forecast method integrating contextual knowledge", International Journal of Knowledge and Systems Science, vol. 2, **(2011)**, pp. 39–53.

Copyright © 2016 SERSC 

[28] G. Box, G. M. Jenkins and G. C. Reinsel, "Time Series Analysis: Forecasting and Control", Third edition, New Jersey: Prentice-Hall, **(1994)**.
[29] G. E. P. Box and G. M. Jenkins, "Time Series Analysis: Forecasting and Control", San Francisco: Holden Day, **(1976)**.
[30] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle, "Least squares support vector machine classifiers", Neural Processing Letters, vol. 9, **(1999)**,pp. 293–300.
[31] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle, "Least Squares Support Vector Machines", Singapore: World Scientific, **(2002)**.
[32] S. M. Shi, L. D. Xu and B. Liu, "Improving the accuracy of nonlinear combined forecasting using neural networks", Expert Systems with Application, vol. 16, **(1999)**, pp. 49–54.
[33] R. Hecht-Nielsen, "Theory of the back propagation neural network", Proceedings of International Joint Conference on Neural Networks, **(1989)**, pp. 593–605.

## Authors

**Zhou Yingxiao**, She was born in 1989. Dr. Major is control science and engineering, the research direction is logistics management.

**Zhao Peng**, He was born in 1967. Prof.Dr. Major is transportation planning and management, control science and engineering.

 Copyright © 2016 SERSC