

Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting

Wei-Chiang Hong

Received: 15 April 2010 / Accepted: 2 October 2010 / Published online: 16 October 2010
© Springer-Verlag London Limited 2010

Abstract Accurate forecasting of inter-urban traffic flow has been one of the most important issues globally in the research on road traffic congestion. Because the information of inter-urban traffic presents a challenging situation, the traffic flow forecasting involves a rather complex nonlinear data pattern, particularly during daily peak periods, traffic flow data reveals cyclic (seasonal) trend. In the recent years, the support vector regression model (SVR) has been widely used to solve nonlinear regression and time series problems. However, the applications of SVR models to deal with cyclic (seasonal) trend time series had not been widely explored. This investigation presents a traffic flow forecasting model that combines the seasonal support vector regression model with chaotic immune algorithm (SSVRCIA), to forecast inter-urban traffic flow. Additionally, a numerical example of traffic flow values from northern Taiwan is used to elucidate the forecasting performance of the proposed SSVRCIA model. The forecasting results indicate that the proposed model yields more accurate forecasting results than the seasonal autoregressive integrated moving average, back-propagation neural network, and seasonal Holt–Winters models. Therefore, the SSVRCIA model is a promising alternative for forecasting traffic flow.

Keywords Traffic flow forecasting · Seasonal adjustment · Support vector regression (SVR) · Chaotic immune algorithm (CIA) · SARIMA · Seasonal Holt–Winters (SHW)

1 Introduction

The effective capacity of inter-urban motorway networks is an essential component of traffic control and information systems, particularly during daily peak periods. Since slightly inaccurate capacity predictions will lead to congestion with huge social costs in terms of travel time, fuel costs and environment pollution, accurate forecasting of the traffic flow during peak periods is a very topic attracted interest in the literature.

There has been a wide variety of forecasting approaches applied to forecast the traffic flow of inter-urban motorway networks. Those approaches could be classified according to the type of data, forecast horizon, and potential end-use [1], including Kalman state space filtering models [2–5] and system identification models [6]. However, traffic flow data are in the form of spatial time series and are collected at specific locations at constant intervals of time. The above-mentioned studies and their empirical results have indicated that the problem of forecasting inter-urban motorway traffic flow is multi-dimensional, including relationships among measurements made at different times and geographical sites. In addition, these methods have difficulty coping with observation noise and missing values while modeling. Therefore, Danech-Pajouh and Aron [7] employed a layered statistical approach with a mathematical clustering technique to group the traffic flow data and a separately tuned linear regression model for each cluster. Their experimental results revealed that the proposed model is superior to the other forecasting approach—autoregressive integrated moving average models (ARIMA). Based on the multi-dimensional pattern recognition requests, such as intervals of time, geographical sites, and the relationships between dependent variable and independent variables, non-parametric regression models [8–10] have also successfully been employed to forecast motorway traffic flow.

W.-C. Hong (✉)
Department of Information Management,
Oriental Institute of Technology, 58 Sec. 2, Sichuan Rd.,
Panchiao, Taipei 220, Taiwan
e-mail: samuelsonhong@gmail.com

Furthermore, the ARIMA models, initially developed by Box and Jenkins [11], are one of the most popular alternatives in traffic flow forecasting [10, 12–15]. For example, Kamarianakis and Prastacos [13] successfully employed the ARIMA model with space and time factors to forecast space–time stationary traffic flow. However, the limitation of ARIMA models is that their natural tendency to concentrate on the mean values of the past series data seems unable to capture the rapid variational process underlying of traffic flow [16]. Recently, as an extension of ARIMA model, Williams [17] applied seasonal ARIMA (SARIMA) model to traffic flow forecasting. The proposed model considered the peak/non-peak flow periods by seasonal differencing and forecasting results reported that it significantly outperformed the heuristic forecast generation method in terms of forecasting accuracy. However, it is quite time-consuming to detect the outlier required and to estimate the parameter of SARIMA model. These new findings are also encouraging the author to employ the SARIMA model as the bench model in this study.

As mentioned above that the process underlying inter-urban traffic flow is complicated to be captured by a single linear statistical algorithm, the artificial neural networks (ANN) models, able to approximate any degree of complexity and without prior knowledge of problem solving, have received much attention and been considered as alternatives for traffic flow forecasting models [14, 18–23]. ANN is based on a model of emulating the processing of the human neurological system to determine related numbers of vehicle and temporal characteristics from the historical traffic flow patterns, especially for nonlinear and dynamic evolutions. Therefore, ANN is widely applied in traffic flow forecasting. Recently, Yin et al. [24] developed a fuzzy-neural model (FNM) to predict traffic flow in an urban street network. The FNM contains two modules: gate network (GN) and expert network (EN). The GN classifies the input data using fuzzy approach, and the EN identifies the input–output relationship by neural network approaches. The empirical results showed that the FNM model provides more accurate forecasting results than the BPNN model. Vlahogianni et al. [22], successfully considering based on the proper representation of traffic flow data with temporal and spatial characteristics, employed a genetic algorithm-based, multilayered, structural optimization strategy to determine the appropriate neural network structure. Their results show that the capabilities of a simple static neural network, with genetically optimized step size, momentum, and number of hidden units, are very satisfactory when modeling both univariate and multivariate traffic data. Even though ANN-based forecasting models could approximate any function particularly for nonlinear function, the limitations are not only difficult to explain the operations of the so-called black-box (such as how to determine suitable

network structure), but also the problem of any ANN algorithm minimizing network training errors is non-convex and it is hard to find the global optimum.

Support vector machines (SVM) were originally developed to solve pattern recognition and classification problems. With the introduction of Vapnik's ϵ -insensitive loss function, SVMs have been extended to solve nonlinear regression estimation problems, i.e., the so-called support vector regression (SVR), and have been successfully applied to solve forecasting problems in many fields in many fields, such as financial time series (stocks index and exchange rate) forecasting [25–29] engineering and software field (production values and reliability) forecasting [30, 31], atmospheric science forecasting [32–35], electric load forecasting [36–40], and so on. The practical results indicated that poor forecasting accuracy is suffered from the lack of knowledge of the selection of the three parameters (σ , C , and ϵ) in an SVR model. However, the structured ways in determining three free parameters in an SVR model is poor. Recently, some major nature-inspired evolutionary algorithms are applied to solve optimization problems, immune algorithm (IA) is one among them. IA, proposed by Mori et al. [41] is used in this study and is based on the learning mechanism of natural immune systems. Similar to GA, SA, and PSO, IA is also a population-based evolutionary algorithm; therefore, it provides a set of solution for exploration and exploitation of search space to obtain optimal/near optimal solution [42]. In addition, the diversity of the employed population set will determine the searching results, the desired solution, or premature convergence (trapped into local minimum). As special mechanism to avoid being trapped in local minimum, the ergodicity property of chaotic sequences has been used as an optimization technique to hybridize with evolutionary algorithms. In this investigation, the chaotic immune algorithm (CIA) is tried to determine the values of three parameters in an SVR model. On the other hand, as mentioned that the traffic flow data not only involves a complicated nonlinear data pattern, but also reveals cyclic (seasonal) trend during daily peak periods (morning/evening commute peak time). However, the applications of SVR models to deal with cyclic (seasonal) trend time series had not been widely explored. Therefore, this paper also attempts to apply the seasonal adjustment method [43, 44] to deal with seasonal trend time series problem. Thus, the proposed SSVRCIA model is applied to forecast inter-urban motorway traffic flow in Panchiao city of Taipei County, Taiwan. The rest of this paper is organized as follows. Section 2 presents the models for comparing forecast performance and SVR models. Section 3 introduces the proposed SSVRCIA forecasting model. Section 4 illustrates a numerical example that reveals the forecasting performance of the proposed models. Conclusions are finally made in Sect. 5.

2 Forecasting methodology

In this investigation, two models, the seasonal ARIMA (SARIMA), seasonal Holt–Winters (SHW), back-propagation neural network (BPNN) models and the SSVRCIA model, are used to compare the forecasting performance of traffic flow.

2.1 Seasonal autoregressive integrated moving average (SARIMA) model

Proposed by Box and Jenkins [11], the seasonal ARIMA process has been one of the most popular approaches in time series forecasting, particularly for strong seasonal component. The SARIMA process is often referred to as the SARIMA(p, d, q) \times (P, D, Q) $_S$ model. Similar to the ARIMA model, the forecasting values are assumed to be a linear combination of past values and past errors. A time series $\{X_t\}$ is a SARIMA process with seasonal period length S if d and D are nonnegative integers and if the differenced series $W_t = (1 - B)^d(1 - B^S)^D X_t$ is a stationary autoregressive moving average process. In symbolic terms, the model can be written as

$$\phi_p(B)\Phi_P(B^S)W_t = \theta_q(B)\Theta_Q(B^S)\varepsilon_t, \quad t = 1, 2, \dots, N \quad (1)$$

where N is the number of observations up to time t ; B is the backshift operator defined by $B^a W_t = W_{t-a}$; $\phi_p(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ is called a regular (non-seasonal) autoregressive operator of order p ; $\Phi_P(B^S) = 1 - \Phi_1 B^S - \dots - \Phi_P B^{PS}$ is a seasonal autoregressive operator of order P ; $\theta_q(B) = 1 - \theta_1 B - \dots - \theta_q B^q$ is a regular moving average operator of order q ; $\Theta_Q(B^S) = 1 - \Theta_1 B^S - \dots - \Theta_Q B^{QS}$ is a seasonal moving average operator of order Q ; ε_t is identically and independently distributed as normal random variables with mean zero, variance σ^2 and $\text{cov}(\varepsilon_t, \varepsilon_{t-k}) = 0, \forall k \neq 0$.

In the definition above, the parameters p and q represent the autoregressive and moving average order, respectively; and the parameters P and Q represent the autoregressive and moving average order at the model’s seasonal period length, S , respectively. The parameters d and D represent the order of ordinary and seasonal differencing, respectively.

Basically, when fitting a SARIMA model to data, the first task is to estimate values of d and D , the orders of differencing needed to make the series stationary and to remove most of the seasonality. The values of p, P, q , and Q then need to be estimated by the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the differenced series. Other model parameters may be estimated by suitable iterative procedures.

2.2 Seasonal Holt–Winters (SHW) model

To consider the seasonal effect, the second employed model is the seasonal Holt–Winters’ linear exponential smoothing (SHW) approach, which is extended from the Holt–Winters model [45, 46]. The Holt–Winters method cannot be extended to accommodate additive seasonality if the magnitude of the seasonal effects does not change with the series or multiplicative seasonality if the amplitude of the seasonal pattern changes over time. The forecast for SHW model is as follows:

$$s_t = \alpha \frac{a_t}{I_{t-L}} + (1 - \alpha)(s_{t-1} + b_{t-1}) \quad (2)$$

$$b_t = \beta(s_t - s_{t-1}) + (1 - \beta)b_{t-1} \quad (3)$$

$$I_t = \gamma \frac{a_t}{s_t} + (1 - \gamma)I_{t-L} \quad (4)$$

$$f_t = (s_t + ib_t)I_{t-L+i} \quad (5)$$

where a_t is the actual value at time t ; s_t is the smoothed estimate at time t ; b_t is the trend value at time t ; α is the level smoothing coefficient; and β is the trend smoothing coefficient. L is the length of seasonality; I is the seasonal adjustment factor; and γ is the seasonal adjustment coefficient.

Equation (2) lets the actual value be smoothed in a recursive manner by weighting the current level (α), and then adjusts s_t directly for the trend of the previous period, b_{t-1} , by adding it to the last smoothed value, s_{t-1} . This helps to eliminate the lag and brings s_t to the approximate base of the current data value. In addition, the first term of (2) is divided by the seasonal number I_{t-L} ; this is done to de-seasonalize a_t (eliminate seasonal fluctuations from a_t). Equation (3) updates the trend, which is expressed as the difference between the last two smoothed values. It modifies the trend by smoothing with β in the last period ($s_t - s_{t-1}$) and adding that to the previous estimate of the trend multiplied by $(1 - \beta)$. Equation (4) is comparable to a seasonal index that is found as a ratio of current values of the series, a_t , divided by the smoothed value for the series, s_t . If a_t is larger than s_t , the ratio will be greater than 1, else, the ratio will be less than 1. In order to smooth the randomness of a_t , (4) weights the newly computed seasonal factor with γ and the most recent seasonal number corresponding to the same season with $(1 - \gamma)$. Equation (5) is used to forecast ahead. The trend, b_t , is multiplied by the number of periods ahead to be forecast, i , and added to the base value, s_t , finally, the summation of s_t and ib_t is multiplied by the seasonal number I_{t-L+i} . The forecast error (e_t) is defined as the actual value minus the forecast (fitted) value for time period t , that is:

$$e_t = a_t - f_t \quad (6)$$

The forecast error is assumed to be an independent random variable with zero mean and constant variance. Values of smoothing coefficients, α and β , and seasonal adjustment coefficient, γ , are determined to minimize the forecasting error.

2.3 Back-propagation neural networks (BPNN) model

The multi-layer back-propagation neural network (BPNN) is one of the most widely used neural network models. Consider the simplest BPNN architecture including three layers: an input layer (x), an output layer (o), and a hidden layer (h). The computational procedure of this network is described below:

$$o_i = f\left(\sum_j g_{ij}x_{ij}\right) \tag{7}$$

where o_i denotes the output of node i , $f(\cdot)$ represents the activation function, g_{ij} is the connection weight between nodes i and j in the lower layer which can be replaced with v_{ji} and w_{kj} , and x_{ij} denotes the input signal from the node j in the lower layer.

The BPNN algorithm attempts to improve neural network performance by reducing the total error through changing the gradient weights. The BPNN algorithm minimizes the sum-of-error-square, which can be calculated by:

$$E = \frac{1}{2} \sum_{p=1}^P \sum_{j=1}^K (d_{pj} - o_{pj})^2 \tag{8}$$

where E denotes the square errors, K represents the output layer neurons, P is the training data pattern, d_{pj} denotes the actual output and o_{pj} represents the network output. The BPNN algorithm is expressed as follows. Let Δv_{ji} denote the weight change for any hidden layer neuron and Δw_{kj} for any output layer neuron,

$$\Delta v_{ji} = -\eta \frac{\partial E}{\partial v_{ji}} \quad i = 1, \dots, I, \quad j = 1, \dots, J - 1, \tag{9}$$

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} \quad j = 1, \dots, J - 1, \quad k = 1, \dots, K \tag{10}$$

where η represents the learning rate parameter, specified at the start of training cycle and determining the training speed and stability of the network. Notably, the J th node is the bias neuron without weight. Equations (11) and (12) express the signal (s_j) to each hidden layer neuron and the signal (u_k) to each neuron in the output layer are expressed as $s_j = \sum_{i=1}^I v_{ji}x_i$ and $u_k = \sum_{j=1}^{J-1} w_{kj}y_j$, respectively.

The error signal terms for the j th hidden neuron δ_{yj} , and for the k th output neuron δ_{ok} are defined as $\delta_{yj} = -\frac{\partial E}{\partial s_j}$ and $\delta_{ok} = -\frac{\partial E}{\partial u_k}$, respectively.

Applying the chain rule, the gradients of the cost function with respect to weights v_{ji} and w_{kj} are $\frac{\partial E}{\partial v_{ji}} = \frac{\partial E}{\partial s_j} \frac{\partial s_j}{\partial v_{ji}}$ and $\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial u_k} \frac{\partial u_k}{\partial w_{kj}}$, respectively. Then, obviously, the gradients of s_j and u_k with respect to weights v_{ji} and w_{kj} are $\frac{\partial s_j}{\partial v_{ji}} = x_i$ and $\frac{\partial u_k}{\partial w_{kj}} = y_j$, respectively. By combining above mention equations, we will obtain $\frac{\partial E}{\partial v_{ji}} = -\delta_{yj}x_i$ and $\frac{\partial E}{\partial w_{kj}} = -\delta_{ok}y_j$. Finally, the weight change from (9) and (10) can now be written as $\Delta v_{ji} = -\eta \frac{\partial E}{\partial v_{ji}} = \eta \delta_{yj}x_i$ and $\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = \eta \delta_{ok}y_j$, respectively. The weights, v_{ji} and w_{kj} , are changed as (11) and (12),

$$w_{kj} = w_{kj} + \Delta w_{kj} = w_{kj} + \eta \delta_{ok}y_j \tag{11}$$

$$v_{ji} = v_{ji} + \Delta v_{ji} = v_{ji} + \eta f'_j(u_j)x_i \sum_{k=1}^K \delta_{ok}w_{kj} \tag{12}$$

The most common activation functions are the squashing sigmoid function, such as the logistic and tangent hyperbolic functions.

2.4 Support vector regression (SVR) model

The brief ideas of SVMs for the case of regression are introduced. A nonlinear mapping $\varphi(\cdot) : \mathfrak{R}^n \rightarrow \mathfrak{R}^{n_h}$ is defined to map the input data (training data set) $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ into a so-called high dimensional feature space (which may have infinite dimensions), \mathfrak{R}^{n_h} . Then, in the high dimensional feature space, there theoretically exists a linear function, f , to formulate the nonlinear relationship between input data and output data. Such a linear function, namely SVR function, is as (13),

$$f(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) + b \tag{13}$$

where $f(\mathbf{x})$ denotes the forecasting values; the coefficients \mathbf{w} ($\mathbf{w} \in \mathfrak{R}^{n_h}$) and b ($b \in \mathfrak{R}$) are adjustable. As mentioned above, SVM method one aims at minimizing the empirical risk by employing the ε -insensitive loss function to find out an optimum hyper plane on the high dimensional feature space to maximize the distance separating the training data into two subsets. Thus, the SVR focuses on finding the optimum hyper plane and minimizing the training error between the training data and the ε -insensitive loss function. Then, the SVR minimizes the overall errors,

$$\text{Min}_{\mathbf{w}, b, \xi^*, \xi} R_\varepsilon(\mathbf{w}, \xi^*, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N (\xi_i^* + \xi_i) \tag{14}$$

with the constraints

$$\begin{aligned} \mathbf{y}_i - \mathbf{w}^T \varphi(\mathbf{x}_i) - b &\leq \varepsilon + \zeta_i^*, & i = 1, 2, \dots, N \\ -\mathbf{y}_i + \mathbf{w}^T \varphi(\mathbf{x}_i) + b &\leq \varepsilon + \zeta_i, & i = 1, 2, \dots, N \\ \zeta_i^* &\geq 0, & i = 1, 2, \dots, N \\ \zeta_i &\geq 0, & i = 1, 2, \dots, N \end{aligned}$$

After the quadratic optimization problem with inequality constraints is solved, the parameter vector \mathbf{w} in (13) is obtained,

$$\mathbf{w} = \sum_{i=1}^N (\beta_i^* - \beta_i) \varphi(\mathbf{x}_i) \tag{15}$$

where β_i^*, β_i are obtained by solving a quadratic program and are the Lagrangian multipliers. Finally, the SVR regression function is obtained as (16) in the dual space,

$$f(\mathbf{x}) = \sum_{i=1}^N (\beta_i^* - \beta_i) K(\mathbf{x}_i, \mathbf{x}) + b \tag{16}$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ is called the kernel function, and the value of the Kernel equals the inner product of two vectors, \mathbf{x}_i and \mathbf{x}_j , in the feature space $\varphi(\mathbf{x}_i)$ and $\varphi(\mathbf{x}_j)$, respectively; that is, $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \circ \varphi(\mathbf{x}_j)$. Any function that meets Mercer’s condition [47] can be used as the Kernel function.

There are several types of kernel function. The most used kernel functions are the Gaussian RBF with a width of $\sigma : K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-0.5\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2\right)$ and the polynomial kernel with an order of d and constants a_1 and a_2 : $K(\mathbf{x}_i, \mathbf{x}_j) = (a_1 \mathbf{x}_i \mathbf{x}_j + a_2)^d$. Till now, it is hard to determine the type of kernel functions for specific data patterns [48, 49]. However, the Gaussian RBF kernel is not only easier to implement, but also capable to nonlinearly map the training data into an infinite dimensional space; thus, it is suitable to deal with nonlinear relationship problems. Therefore, the Gaussian RBF kernel function is specified in this study.

3 Chaotic immune algorithm (CIA) in selecting parameters and seasonal adjustment

3.1 CIA in selecting parameters

The selection of the three parameters, σ , ε , and C , of an SVR model influence the accuracy of forecasting. However, structural methods for confirming efficient selection of parameters efficiently are lacking. Recently, Hong [38] applied immune algorithm (IA) to determine parameters of an SVR model and found that the proposed model is superior to other competitive forecasting models (ANN and regression models). However, based on the operation procedure of IA, if the population diversity of an initial population cannot be maintained under selective pressure,

i.e., the initial individuals are not necessarily fully diversified in the search space, then IA could only seek for the solutions in the narrow space and the solution is far from the global optimum (premature convergence). To overcome the shortcoming, it is necessary to find some effective approach and improved design or procedure on IA to track in the solution space effectively and efficiently. One feasible approach is focused on the chaos approach, due to its easy implementation and special ability to avoid being trapped in local optimum [50]. The application of chaotic sequences can be a good alternative to diversify the initial definition domain in stochastic optimization procedures, i.e., small changes in the parameter settings or the initial values in the model. Due to the ergodicity property of chaotic sequences, it will lead to very different future solution finding behaviors; thus, chaotic sequences can be used to enrich the searching behavior and to avoid being trapped into local optimum [51]. There are lots of applications in optimization problem using chaotic sequences [52–56]. Coelho and Mariani [57] recently apply chaotic artificial immune network (chaotic opt-aiNET) to solve the economic dispatch problem (EDP), which are based on Zaslavsky’s map by its spread-spectrum characteristic and large Lyapunov exponent to successfully escape from local optimum and to converge to a stable equilibrium. Therefore, it is believable that applying chaotic sequences to diversify the initial definition domain in IA’s initialization procedure (CIA) is a feasible approach to optimize the parameter selection in an SVR model.

To design the CIA, many principal factors like identification of the affinity, selection of antibodies, crossover and mutation of antibody population are similar to the IA factors; more procedure details of the CIA on this study is as follows, and the flowchart is shown as Fig. 1.

Step 1 Initialization of antibody population

The values of the three parameters in an SVR model in the i th iteration can be represented as $X_k^{(i)}, k = C, \sigma, \varepsilon$. Set $i = 0$, and employ (17) to map the three parameters among the intervals ($\text{Min}_k, \text{Max}_k$) into chaotic variable $x_k^{(i)}$ located in the interval (0, 1).

$$x_k^{(i)} = \frac{X_k^{(i)} - \text{Min}_k}{\text{Max}_k - \text{Min}_k}, \quad k = C, \sigma, \varepsilon \tag{17}$$

Then, employ the chaotic sequence, defined as (18), with $\mu = 4$ to compute the next iteration chaotic variable, $x_k^{(i+1)}$.

$$x_k^{(i+1)} = \mu x_k^{(i)} (1 - x_k^{(i)}) \tag{18}$$

$$x_k^{(i)} \in (0, 1), \quad i = 0, 1, 2, \dots,$$

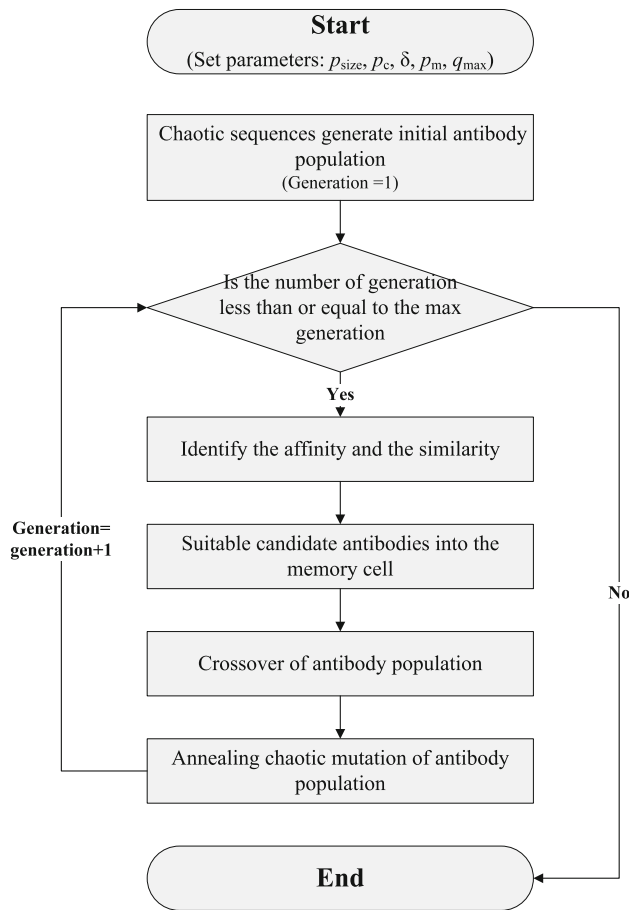


Fig. 1 Chaotic immune algorithm (CIA) flowchart

where $x^{(i)}$ is the value of the chaotic variable x at the i th iteration, μ is the so-called bifurcation parameter of the system, $\mu \in [0, 4]$.

And, transform $x_k^{(i+1)}$ to obtain three parameters for the next iteration, $X_k^{(i+1)}$, by the following (19).

$$X_k^{(i+1)} = \text{Min}_k + x_k^{(i+1)}(\text{Max}_k - \text{Min}_k) \quad (19)$$

After this transformation, the three parameters, C , σ , and ε , are constituted the initial antibody population and then will be represented by binary-code string. For example, assume that an antibody contains 12 binary codes to represent three SVR parameters. Each parameter is thus expressed by four binary codes. Assume the set-boundaries for parameters σ , C , and ε are 2, 10, and 0.5, respectively, then, the antibody with binary-code “1 0 0 1 0 1 0 1 0 0 1 1” implies that the real values of the three parameters σ , C , and ε are 1.125, 3.125, and 0.09375, respectively. The number of initial antibodies is the same as the size of the memory cell. The size of the memory cell is set to ten in this study.

Step 2 Identification of the affinity and the similarity

A higher affinity value implies that an antibody has a higher activation with an antigen. To continue keeping the diversity of the antibodies stored in the memory cells, the antibodies with lower similarity have higher probability of being included in the memory cell. Therefore, an antibody with a higher affinity value and a lower similarity value has a good likelihood of entering the memory cells. The affinity between the antibody and antigen is defined as (20).

$$Ag_k = 1/(1 + d_k) \quad (20)$$

where d_k denotes the SVR forecasting errors obtained by the antibody k .

The similarity between antibodies is expressed as (21).

$$Ab_{ij} = 1/(1 + T_{ij}) \quad (21)$$

where T_{ij} denotes the difference between the two SVR forecasting errors obtained by the antibodies inside (existed) and outside (will be entering) the memory cell.

Step 3 Selection of antibodies in the memory cell

Antibodies with higher values of Ag_k are considered to be potential candidates for entering the memory cell. However, the potential antibody candidates with Ab_{ij} values exceeding a certain threshold are not qualified to enter the memory cell. In this investigation, the threshold value is set to 0.9.

Step 4 Crossover of antibody population

New antibodies are created via crossover and mutation operations. To perform crossover operation, strings representing antibodies are paired randomly. Moreover, the proposed scheme adopts the single-point-crossover principle. Segments of paired strings (antibodies) between two determined break-points are swapped. In this investigation, the probability of crossover (p_c) is set as 0.5. Finally, the three crossover parameters are decoded into a decimal format.

Step 5 Annealing chaotic mutation of antibody population

For the i th iteration (generation), crossover antibody population ($\hat{X}_k^{(i)}, k = C, \sigma, \varepsilon$) of current solution space ($\text{Min}_k, \text{Max}_k$) are mapped to chaotic variable interval $[0, 1]$ to form the crossover chaotic variable space $\hat{x}_k^{(i)}, k = C, \sigma, \varepsilon$, as (22),

$$\hat{x}_k^{(i)} = \frac{\hat{X}_k^{(i)} - \text{Min}_k}{\text{Max}_k - \text{Min}_k}, \quad k = C, \sigma, \varepsilon, \quad i = 1, 2, \dots, q_{\max} \quad (22)$$

where q_{\max} is the maximum evolutionary generation of the population. Then, the i th chaotic variable $x_k^{(i)}$ is summed up

to $\hat{x}_k^{(i)}$, and the chaotic mutation variable are also mapped to interval $[0, 1]$ as in (23),

$$\tilde{x}_k^{(i)} = \hat{x}_k^{(i)} + \delta x_k^{(i)} \tag{23}$$

where δ is the annealing operation. Finally, the chaotic mutation variable obtained in interval $[0, 1]$ is mapped to the solution interval ($\text{Min}_k, \text{Max}_k$) by definite probability of mutation (p_m), thus completing a mutative operation.

$$\tilde{X}_k^{(i)} = \text{Min}_k + \tilde{x}_k^{(i)} (\text{Max}_k - \text{Min}_k). \tag{24}$$

Step 6 Stopping criteria

If the number of generations equals a given scale, then the best antibody is a solution, otherwise return to Step 2.

The CIA is used to seek a better combination of the three parameters in SVR. The value of the normalized root mean square error (NRMSE) is used as the criterion (the smallest value of NRMSE) of forecasting errors to determine the suitable parameters used in SVR model in this investigation, which is given by (25).

$$\text{NRMSE} = \sqrt{\frac{\sum_{i=1}^n (a_i - f_i)^2}{\sum_{i=1}^n a_i^2}} \tag{25}$$

where n is the number of forecasting periods; a_i is the actual traffic flow value at period i ; and f_i is the forecasting traffic flow value at period i .

3.2 Seasonal adjustment

As mentioned that during daily peak periods, traffic flow data reveals cyclic (seasonal) trend, any model attempts to accomplish the goal of high accurate forecasting performance, it is necessary to estimate this seasonal component. There are several approaches to estimate the seasonal index of data series [44, 58, 59], including product-model type and non-product-model type. Based on the data series type consideration, this investigation employed Deo and Hurvich’s [58] approach to compute the seasonal index, as shown in (26),

$$\text{peak}_t = \frac{a_t}{f_t} = \frac{a_t}{\sum_{i=1}^N (\beta_i^* - \beta_i) K(\mathbf{x}_i, \mathbf{x}) + b} \tag{26}$$

where $t = j, l + j, 2l + j, \dots, (m - 1)l + j$ only for the same peak time point in each period. Then, the seasonal index (SI) for each peak time point j is computed as (27),

$$\text{SI}_j = \frac{1}{m} (\text{peak}_j + \text{peak}_{l+j} + \dots + \text{peak}_{(m-1)l+j}) \tag{27}$$

Eventually, the forecasting value of the SSVRCIA is obtained by (28),

$$f_{N+k} = \left(\sum_{i=1}^N (\beta_i^* - \beta_i) K(\mathbf{x}_i, \mathbf{x}_{N+k}) + b \right) \times \text{SI}_k \tag{28}$$

where $k = j, l + j, 2l + j, \dots, (m - 1)l + j$ implies the peak time point in another period (for forecasting period).

4 A numerical example and experimental results

The traffic flow data sets were originated from three civil motorway detector sites. The civil motorway is the busiest inter-urban motorway networks in Panchiao city, the capital of Taipei County, Taiwan. The major site was located at the center of Panchiao City, where the flow intersects an urban local street system, and it provided one way traffic volume for each hour in weekdays. Therefore, one way flow data for peak traffic are employed in this investigation, which includes the morning peak period (from 6:00 to 10:00) and the evening peak period (from 16:00 to 20:00). The data collection is conducted from February 2005 to March 2005. During the observation period, the number of traffic flow data available for the morning and evening peak periods are 45 and 90 h, respectively. For convenience, the traffic flow data are converted to equivalent of passengers (EOP), and both of these two peak periods show the seasonality of traffic data. In addition, traffic flow data are divided into three parts: training data, validation data, and testing data. For the morning peak period, the training data set, validation data set, and testing data set are 30, 10, and 10 h accordingly. For the evening peak period, the experimental data are arranged as training data (60 h), validation data (15 h), and testing data (15 h).

4.1 Parameter determination of different comparative forecasting models

The parameter selection of forecasting models is important for obtaining good forecasting performance. For the SAR-IMA model, the parameters are determined by taking the first-order regular difference and first seasonal difference to remove non-stationary and seasonality characteristics. Using statistical packages, with no residuals autocorrelated and approximately white noise residuals, the most suitable models for these two morning/evening peak periods for the traffic data are SARIMA(1, 0, 1) × (0, 1, 1)₅ with non-constant item and SARIMA(1, 0, 1) × (1, 1, 1)₅ with constant item, respectively. The equations used for the SARIMA models are presented as (29) and (30), respectively.

$$(1 - 0.5167B)(1 - B^5)X_t = (1 + 0.3306B) \times (1 - 0.9359B^5)\varepsilon_t \tag{29}$$

$$(1 - 0.5918B)(1 - B^5)X_t = 2.305 + (1 - 0.9003B^5)\varepsilon_t \tag{30}$$

Table 1 CIA's parameters setting in each peak period

Traffic peak periods	Population size (p_{size})	Maximal generation (q_{max})	Probability of crossover (p_c)	The annealing operation parameter (δ)	Probability of mutation (p_m)
Morning peak	200	500	0.5	0.9	0.1
Evening peak	200	500	0.5	0.9	0.1

For the seasonal Holt–Winters method, by Minitab 14 statistic software, the appropriate parameters (L , α , β , and γ) for morning peak period are determined 5, 0.15, 0.72, and 0.73, correspondingly; for evening peak period as 5, 0.48, 0.04, and 0.13, correspondingly. For the BPNN model, Matlab 6.5 computing software is employed to implement the forecasting procedure. The number of nodes in the hidden layer is used as a validation parameter of the BPNN model. The most suitable number of hidden nodes of a BPNN model is three.

4.2 SSVRCIA traffic forecasting model

Before conducting the seasonal adjustment for the SSVRCIA model, it is necessary to implement the CIA algorithm to determine suitable values of the three parameters in an SVR model. The parameters of the CIA in the proposed models for both traffic peak periods are experimentally set respectively as shown in Table 1. For the SVRCIA modeling procedure, in the training stage, a rolling-based forecasting procedure was conducted, and, in the validation and testing stage, a 1-h-ahead forecasting policy adopted. Then, several types of data-rolling are considered to forecast traffic flow in the next hour. Different numbers of the traffic flow in a time series were fed into the SVRCIA model to forecast the traffic flow in the next validation period. While training errors improvement occurs, the three kernel parameters, σ , C , and ε of the SVRCIA model adjusted by CIA algorithm, are employed to calculate the validation error. Then, the adjusted parameters with minimum validation error are selected as the most appropriate parameters. Table 2 indicates that SVRCIA models perform the best when 15 and 35 input data are used for morning/evening traffic forecast respectively.

Now the seasonal term is considered. For the morning peak period, there are five peak time points in each cycle, from 6:00 to 10:00. The seasonal indexes for each peak time point are calculated based on the 40 forecasting values of the SVRCIA model in training (30 forecasting values) and validation (10 forecasting values) stages, as shown in Table 3. Similarly in the evening peak period, there are also five peak time points in each cycle, from 16:00 to 20:00. The seasonal indexes for each peak time point are also shown in Table 3.

Table 2 Forecasting results and associated parameters of the SVRCIA models

Nos. of input data	Parameters			NRMSE of testing
	σ	C	ε	
Morning peak period				
5	0.5089	1,013	0.2290	0.3265
10	1.0479	1,421	0.5395	0.3108
15	1.0502	2,522	0.6872	0.2602
20	1.0826	3,885	0.8626	0.2629
25	0.9668	4,518	0.3377	0.2634
Evening peak period				
5	0.5038	1,220	0.2412	0.1190
10	0.8876	10,628	0.6664	0.1267
15	0.1293	10,477	0.4463	0.3905
20	1.0315	10,340	0.6668	0.1092
25	0.7472	10,215	0.8739	0.1112
30	0.6498	10,948	0.9494	0.1091
35	0.6717	9,792	0.7517	0.1004
40	0.1605	8,774	0.2110	0.1206
45	0.1372	10,791	0.6252	0.1402
50	0.1338	9,560	0.2565	0.1212
55	0.1903	9,359	0.5196	0.1017

Bold values indicate better forecasting results

Table 3 The seasonal indexes for each peak time point

Peak periods	Seasonal index
Morning peak	
6:00	0.6530
7:00	1.0294
8:00	1.0580
9:00	0.9451
10:00	0.9351
Evening peak	
16:00	0.8708
17:00	0.9989
18:00	1.1131
19:00	1.0572
20:00	0.9887

The well-trained models, SARIMA, BPNN, SHW, SVRCIA, and SSVRCIA, are applied to forecast the traffic flow during the morning/evening peak period. Tables 4 and 5

Table 4 Morning peak period traffic flow forecasting results (unit: EOP)

Peak periods	Actual	SARIMA (1,0,1) × (0,1,1) ₅	BPNN	SHW ($L = 5, \alpha = 0.15,$ $\beta = 0.72, \gamma = 0.73$)	SVRCIA	SSVRCIA
031106	1,317.5	1,363.87	2,239.5	1,361.3	2,418.3	1,402.8
031107	2,522.0	2,440.1	1,270.0	2,573.9	2,010.5	2,069.5
031108	2,342.0	2,593.9	2,364.5	2,677.6	2,013.6	2,130.4
031109	2,072.0	2,422.1	2,483.0	2,351.2	2,364.2	2,234.4
031110	1,841.5	2,459.9	2,188.5	2,256.9	2,065.3	1,931.3
031206	995.5	1,578.3	2,091.5	1,387.8	1,820.8	1,188.9
031207	1,457.0	2,569.9	1,277.0	2,656.8	1,663.5	1,712.3
031208	1,899.0	2,690.4	2,507.0	2,586.4	1,441.3	1,524.9
031209	1,870.5	2,505.4	2,627.0	2,329.1	1,419.8	1,341.9
031210	2,151.5	2,538.7	2,264.5	2,180.2	1,813.5	1,695.8
NRMSE		0.3039	0.3682	0.2706	0.2602	0.1681

“031106” denotes the 6 o’clock on 11 March 2005, and so on

Table 5 Evening peak period traffic flow forecasting results (unit: EOP)

Peak periods	Actual	SARIMA (1,0,1) × (1,1,1) ₅	BPNN	SHW ($L = 5, \alpha = 0.48,$ $\beta = 0.04, \gamma = 0.13$)	SVRCIA	SSVRCIA
031016	2,310.5	2,573.84	2,844.5	1,999.5	2,209.8	1,924.3
031017	2,618.0	2,821.57	2,110.5	2,260.0	2,321.2	2,318.7
031018	2,562.0	3,107.01	2,620.0	2,475.5	2,364.4	2,631.9
031019	2,451.5	3,103.66	2,853.5	2,433.2	2,569.9	2,716.9
031020	2,216.5	3,011.80	2,873.0	2,312.3	2,423.7	2,396.4
031116	2,175.5	2,611.58	2,624.5	2,004.6	2,369.9	2,063.7
031117	2,577.0	2,859.31	1,999.0	2,254.4	2,153.1	2,150.8
031118	2,879.5	3,144.75	2,191.5	2,444.4	2,431.8	2,706.9
031119	2,693.0	3,141.40	2,500.5	2,404.0	2,643.9	2,795.1
031120	2,640.0	3,049.54	2,386.0	2,281.8	2,729.3	2,698.5
031216	2,146.5	2,649.32	2,203.5	2,000.5	2,618.8	2,280.5
031217	2,544.5	2,897.05	2,116.0	2,251.7	2,596.6	2,593.8
031218	2,873.0	3,182.49	2,522.5	2,438.7	2,534.4	2,821.1
031219	2,567.5	3,179.13	2,620.0	2,383.7	2,623.9	2,774.0
031220	2,660.5	3,087.28	2,512.0	2,268.2	2,579.3	2,550.2
NRMSE		0.1821	0.1636	0.1142	0.1004	0.0828

“031106” denotes the 6 o’clock on 11 March 2005, and so on

show the actual values and the forecast values obtained using various forecasting models in the morning peak and the evening peak, respectively. The NRMSE values for each peak hour are calculated to compare fairly the proposed models with other alternative models. The proposed SSVRCIA model has smaller NRMSE values than the SARIMA, BPNN, SHW, and SVRCIA models to capture the traffic flow patterns on hourly average basis. Clearly, the seasonal adjustment employed here is proficient in dealing with such cyclic peak data type of forecasting problems.

5 Conclusions

Accurate traffic forecast is crucial for the inter-urban traffic control system, particularly for avoiding congestion and for increasing efficiency of limited traffic resources during peak periods. The historical traffic data of Panchiao City in northern Taiwan show a seasonal fluctuation trend which occurs in many inter-urban traffic systems. Therefore, over-prediction or under-prediction of traffic flow influences the transportation capability of an inter-urban system. This study introduces the application of forecasting techniques,

SSVRCIA, to investigate its feasibility for forecasting inter-urban motorway traffic. The experimental results indicate that the SSVRCIA model has better forecasting performance than the SARIMA, BPNN, SHW, and SVR-CIA models. The superior performance of the SSVRCIA model is due to the generalization ability of SVR model for forecasting and the proper selection of SVR parameters by CIA and effective seasonal adjustments. In addition, SVR method employs the quadratic programming technique which is based on the assumptions of convex set and existence of global optimum solution. Thus, it should be theoretically approximated to the global optimum solution if superior searching algorithms are employed. In the contrast, SARIMA, SHW models employ the parametric technique which is based on specific assumptions, such as linear relationship between the current value of the underlying variables and previous values of the variable and error terms, and these assumptions are not completely tallied with real world problems.

This investigation is the first to apply the SVR with CIA and seasonal adjustment for forecasting inter-urban motorway traffic flow. Many forecasting methodologies have been proposed to deal with the seasonality of traffic flow. However, most models are time-consuming in verifying the suitable time-phase divisions, particularly when the sample size is large. In this investigation, the SSVRCIA model provides a convenient and valid alternative for traffic flow forecasting. The SSVRCIA model directly uses historical observations from traffic control systems and then determines suitable parameters by efficient optimization algorithms. The next step would be to develop trading strategies to involve other factors and meteorological control variables during peak periods, such as driving speed limitation, important social events, the percentage of heavy vehicles, bottleneck service level, and waiting time during intersection traffic signals can be included in the traffic forecasting model. In addition, even the proposed SSVRCIA model is one of the hybrid forecasting models; some other advanced optimization algorithms for parameters selection can be applied for the SVR model to satisfy the requirement of real-time traffic control systems. The goal of the author is to show that combination of novel techniques is as good as pure techniques.

Acknowledgments This research was conducted with the support of National Science Council, Taiwan (NSC 99-2410-H-161-001, NSC 98-2410-H-161-001, and NSC 98-2811-H-161-001).

References

- Dougherty MS (1996) Investigation of network performance prediction literature review (Technical Note 394). Institute for Transport Studies, University of Leeds
- Okutani I, Stephanedes YJ (1984) Dynamic prediction of traffic volume through Kalman filtering theory. *Transp Res B* 18(1):1–11
- Stathopoulos A, Karlaftis GM (2003) A multivariate state space approach for urban traffic flow modeling and prediction. *Transp Res C* 11(2):121–135
- Van Arem B, Van der Vlist M, de Ruiter J, Muste M, Smulders S (1994) Travel time estimation in the GERDIEN project. In: *Proceedings of the 2nd DRIVE II workshop on short-term traffic forecasting*, Delft
- Whittaker J, Garside S, Lindveld K (1994) Tracking and predicting a network traffic process. In: *Proceedings of the 2nd DRIVE II workshop on short-term traffic forecasting*, Delft
- Vythoulkas PC (1993) Alternative approaches to short term forecasting for use in driver information systems. In: Daganzo CF (ed) *Transportation and traffic theory*. Elsevier, Amsterdam
- Danech-Pajouh M, Aron M (1991) ATHENA: a method for short-term inter-urban motorway traffic forecasting. *Recherche Transports Sécurité* 6:11–16
- Eubank RL (1988) *Spline smoothing and nonparametric regression*. Marcel Dekker, New York
- Smart BL, Demetsky MJ (1997) Traffic flow forecasting: comparison of modeling approaches. *J Transp Eng* 123(4):261–266
- Smith BL, Williams BM, Oswald RK (2002) Comparison of parametric and nonparametric models for traffic flow forecasting. *Transp Res C* 10(4):303–321
- Box GEP, Jenkins GM (1976) *Time series analysis: forecasting and control*. Holden-Day, San Francisco
- Hamed MM, Al-Masaeid HR, Bani-Said ZM (1995) Short-term prediction of traffic volume in urban arterials. *ASCE J Transp Eng* 121(3):249–254
- Kamarianakis Y, Prastacos P (2005) Space-time modeling of traffic flow. *Comput Geosci* 31(2):119–133
- Kirby HR, Watson SM, Dougherty MS (1997) Should we use neural networks or statistical models for short-term motorway traffic forecasting? *Int J Forecast* 13(1):43–50
- Williams BM (2001) Multivariate vehicular traffic flow prediction: an evaluation of ARIMAX modeling. *Transp Res Rec* 1776:194–200
- Angeline PJ, Saunders GM, Pollack JB (1994) An evolutionary algorithm that constructs recurrent neural networks. *IEEE Trans Neural Netw* 5(1):54–65
- Williams BM (1999) *Modeling and forecasting vehicular traffic flow as a seasonal stochastic time series process*. Doctoral dissertation, Department of Civil Engineering, University of Virginia, Charlottesville
- Chen H, Grant-Muller S, Mussone L, Montgomery F (2001) A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. *Neural Comput Appl* 10:277–286
- Dougherty MS, Cobbett MR (1997) Short-term inter-urban traffic forecasts using neural networks. *Int J Forecast* 13(1):21–31
- Florio L, Mussone L (1996) Neural network models for classification and forecasting of freeway traffic flow stability. *Control Eng Pract* 4(2):153–164
- Ledoux C (1997) An urban traffic flow model integrating neural networks. *Transp Res C* 5(5):287–300
- Vlahogianni EI, Karlaftis MG, Golias JC (2005) Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. *Transp Res C* 13(3):211–234
- Zhang G, Patuwo BE, Hu MY (1998) Forecasting with artificial neural networks: the state of art. *Int J Forecast* 14:35–62
- Yin H, Wong SC, Xu J, Wong CK (2002) Urban traffic flow prediction using a fuzzy-neural approach. *Transp Res C* 10(2): 85–98
- Tay FEH, Cao LJ (2001) Application of support vector machines in financial time series forecasting. *Omega* 29(4):309–317

26. Huang W, Nakamori Y, Wang SY (2005) Forecasting stock market movement direction with support vector machine. *Comput Oper Res* 32(10):2513–2522
27. Hung WM, Hong WC (2009) Application of SVR with improved ant colony optimization algorithms in exchange rate forecasting. *Control Cybern* 38(3):863–891
28. Pai PF, Lin CS (2005) A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega* 33(6):497–505
29. Pai PF, Lin CS, Hong WC, Chen CT (2006) A hybrid support vector machine regression for exchange rate prediction. *Int J Inf Manag Sci* 17(2):19–32
30. Pai PF, Hong WC (2006) Software reliability forecasting by support vector machines with simulated annealing algorithms. *J Syst Softw* 79(6):747–755
31. Hong WC, Pai PF (2006) Predicting engine reliability by support vector machines. *Int J Adv Manuf Technol* 28(1–2):154–161
32. Hong WC (2008) Rainfall forecasting by technological machine learning models. *Appl Math Comput* 200(1):41–57
33. Hong WC, Pai PF (2007) Potential assessment of the support vector regression technique in rainfall forecasting. *Water Resour Manag* 21(2):495–513
34. Wang W, Xu Z, Lu JW (2003) Three improved neural network models for air quality forecasting. *Eng Comput* 20(2):192–210
35. Mohandes MA, Halawani TO, Rehman S, Hussain AA (2004) Support vector machines for wind speed prediction. *Renew Energy* 29(6):939–947
36. Hong WC (2009) Hybrid evolutionary algorithms in a SVR-based electric load forecasting model. *Int J Electr Power Energy Syst* 31(7–8):409–417
37. Hong WC (2009) Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. *Energy Convers Manag* 50(1):105–117
38. Hong WC (2009) Electric load forecasting by support vector model. *Appl Math Model* 33(5):2444–2454
39. Pai PF, Hong WC (2005) Support vector machines with simulated annealing algorithms in electricity load forecasting. *Energy Convers Manag* 46(17):2669–2688
40. Pai PF, Hong WC (2005) Forecasting regional electric load based on recurrent support vector machines with genetic algorithms. *Electr Power Syst Res* 74(3):417–425
41. Mori K, Tsukiyama M, Fukuda T (1993) Immune algorithm with searching diversity and its application to resource allocation problem. *Trans Inst Electr Eng Jpn* 113-C(10):872–878
42. Prakash A, Khilwani N, Tiwari MK, Cohen Y (2008) Modified immune algorithm for job selection and operation allocation problem in flexible manufacturing system. *Adv Eng Softw* 39(3):219–232
43. Xiao Z, Ye SJ, Zhong B, Sun CX (2009) BP neural network with rough set for short term load forecasting. *Expert Syst Appl* 36(1):273–279
44. Wang J, Zhu W, Zhang W, Sun D (2009) A trend fixed on firstly and seasonal adjustment model combined with the ϵ -SVR for short-term forecasting of electricity demand. *Energy Policy* 37(11):4901–4909
45. Holt CC (1957) Forecasting seasonal and trends by exponentially weighted averages. Carnegie Institute of Technology, Pittsburgh
46. Winters PR (1960) Forecasting sales by exponentially weighted moving averages. *Manag Sci* 6:324–342
47. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
48. Amari S, Wu S (1999) Improving support vector machine classifiers by modifying kernel functions. *Neural Netw* 12(6):783–789
49. Vojislav K (2001) Learning and soft computing—support vector machines, neural networks and fuzzy logic models. The MIT Press, Massachusetts
50. Wang L, Zheng DZ, Lin QS (2001) Survey on chaotic optimization methods. *Comput Technol Autom* 20(1):1–5
51. Pan H, Wang L, Liu B (2008) Chaotic annealing with hypothesis test for function optimization in noisy environments. *Chaos Solitons Fractals* 35(5):888–894
52. Zuo XQ, Fan YS (2006) A chaos search immune algorithm with its application to neuro-fuzzy controller design. *Chaos Solitons Fractals* 30(1):94–109
53. Liu B, Wang L, Jin YH, Tang F, Huang DX (2005) Improved particle swarm optimization combined with chaos. *Chaos Solitons Fractals* 25(5):1261–1271
54. Yang D, Li G, Cheng G (2007) On the efficiency of chaos optimization algorithms for global optimization. *Chaos Solitons Fractals* 34(4):1366–1375
55. Li L, Yang Y, Peng H, Wang X (2006) Parameters identification of chaotic systems via chaotic ant swarm. *Chaos Solitons Fractals* 28(5):1204–1211
56. Tavazoei MS, Haeri M (2007) Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Appl Math Comput* 187(2):1076–1085
57. LdS Coelho, Mariani VC (2009) Chaotic artificial immune approach applied to economic dispatch of electric energy using thermal units. *Chaos Solitons Fractals* 40(5):2376–2383
58. Deo R, Hurvich C (2006) Forecasting realized volatility using a long-memory stochastic volatility model: estimation, prediction and seasonal adjustment. *J Econom* 131:29–58
59. Azadeh A, Ghaderi SF (2008) Annual electricity consumption forecasting by neural network in high energy consuming industrial sectors. *Energy Convers Manag* 49:2272–2278