J. of Mult.-Valued Logic & Soft Computing, Vol. 16, pp. 567–584
Reprints available directly from the publisher
Photocopying permitted by license only

An Improved PSO for Parameter Determination and Feature Selection of SVR and its Application in STLF

DONG-XIAO NIU[1] AND YING-CHUN GUO[1,2]

[1] *School of Business Administration, North China Electric Power University, Baoding 071003, China*
[2] *College of Mathematics and Computer Science, Hebei University, Baoding 071002, China*
E-mail: guoyc@hbu.cn

Received: October 22, 2008. Accepted: April 21, 2009.

A novel support vector regression (SVR) optimized by an improved particle swarm optimization (PSO) combined with simulated annealing algorithm (SA) was proposed. The optimization mechanism also combined the discrete PSO with the continuous-valued PSO to simultaneously optimize the input feature subset selection and the SVR kernel parameter setting. By incorporating with SA, the global searching capacity of the proposed model was enhanced. The improved SAPSO was used to optimize the parameters of SVR and select the input features simultaneously. Based on the operational data provided by a regional power grid in north China, the method was used in short-term load forecasting (STLF). The experimental results showed the proposed approach can correctly select the discriminating input features and compared to the PSO-SVR and the traditional SVR, the average time of the proposed method in the experimental process reduced and the forecasting accuracy increased respectively. So, the improved method is better than the other two models.

Keywords: Support vector regression (SVR), particle swarm optimization (PSO), simulated annealing (SA), parameter determination, feature selection, short-term load forecasting (STLF).

1 INTRODUCTION

Support vector machines (SVM) which were suggested by Vapnik [1], have recently been used in a range of problems. Classification and prediction are the

two most important tasks of SVM. To build a SVM-based regression (SVR) model, feature subset selection is an important issue in building regression systems. It is advantageous to limit the number of input features to produce a good predictive and less computationally intensive model [2]. With a small and appropriate feature subset, the rationale for the regression decision can be realized easier. In addition to feature selection, proper model parameter setting can improve the SVR prediction accuracy. To design a SVR, one must choose a kernel function, set the kernel parameters and determine a soft margin constant C. The parameters that should be optimized include the penalty parameter C and the kernel function parameters such as the σ for the radial basis function (RBF) kernel. Thus, appropriate feature subset selection and model parameter setting have a heavy impact on the regression accuracy [3].

As feature subset selection infuences the appropriate kernel parameters, obtaining optimal feature subset and SVR parameters must occur simultaneously. The grid algorithm is an alternative to find the SVR parameters when using the RBF kernel function [4]; however, this approach does not simultaneously perform feature subset selection. Huang and Wang [3] adopted the genetic algorithm to optimize the feature subset and model parameter selection for the SVM. They achieved a promising result.

This study tries a new technology, swarm intelligence: particle swarm optimization (PSO) to optimize the feature subset and parameter selection for SVR. Proposed by Kennedy and Eberhart [5, 6] and inspired by social behavior in nature, PSO is a population-based search algorithm that is initialized with a population of random solutions, called particles. Each particle in the PSO fies through the search space at a velocity which dynamically adjusted according to its own and its companion's historical behavior. Recently, numerous researches on PSO theories or applications have been reported [7–14]. Because particle swarm optimization is powerful, easy to implement, and computationally efficient [6], this study introduces PSO as an optimization technique to simultaneously optimize both the optimal feature subset and SVR parameters.

A novel SVR optimized by an improved PSO combined with simulated annealing algorithm (SA) was proposed. By incorporating with the SA, the global searching capacity of the proposed model (SAPSO) was enhanced. At the same time, we use two types of PSO: the continuous-valued version and discrete version. The continuous-valued version is used to optimize the best SVR model parameters, while the discrete version is used to search the optimal feature subset. Combining SAPSO with SVR, this novel hybrid approach is proposed here to optimize the parameters and feature subset simultaneously, without degrading the SVR prediction accuracy. Based on the electric load data provided by a regional power grid in north China, the method was used in the actual short-term load forecasting. The experimental results showed the proposed approach can correctly select input features. Compared with

the traditional SVR and PSO-SVR, the average time of the proposed method in the experimental process reduced and the forecasting accuracy increased respectively.

This paper is organized as follows. Section 2 describes the related works including the basic SVR, PSO and SA concepts. Section 3 describes the improved SAPSO method. Section 4 describes the hybrid system which improved SAPSO for parameter determination and feature selection of SVR. Section 5 implements the proposed model in short-term load forecasting. Section 6 gives remarks and provides conclusions.

2 RELATED WORKS

2.1 Review of Support Vector Regression

The basic principle of SVR estimates regression function is to map data in the input space to a high dimensional feature space by using a nonlinear mapping [15, 16]. Then, a linear mapping is made in the high dimensional space. Set the training sample $\{x_i, y_i\}_{i=1}^{N}$, $x_i \in R^m$as the input vector, and $y_i \in R$ as the corresponding export value. N is the number of training sample. SVR regression estimates the unknown function by the following Equation:

$$f(x) = \langle w, \psi(x) \rangle + b \tag{1}$$

Here $\psi(x)$ presents the mapped high dimensional feature space, $w \in R^m$, $b \in R$, $\langle w, \psi(x) \rangle$ presents the dot product. The following formulae can be obtained by minimizing the risk function to coefficient w and b:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} |y_i - \langle w, \varphi(x_i) \rangle - b|_\varepsilon \tag{2}$$

$$|y_i - \langle w, \varphi(x_i) \rangle - b|_\varepsilon = \begin{cases} 0, & |y_i - \langle w, \varphi(x_i) \rangle - b| \le \varepsilon \\ |y_i - \langle w, \varphi(x_i) \rangle - b| - \varepsilon, & otherwise \end{cases} \tag{3}$$

Eq. (2), the first item $\frac{1}{2}\|w\|^2$ is called the model complexity item; the second item is the empirical error term determined by insensitive loss which defined by Eq. (3). ε and C are the adjustment balanced parameters. The loss function can be used to express the decision function given by Eq. (1) in the form of scatter data point.

Due to high dimensions in feature space and the undifferentiable objective functions, the direct solution to Eq. (2) is nearly unfeasible. SVR regression method avoids these problems ingeniously through introducing dot product nuclear function and using the Wolfe antithesis skills, and thus transforms the

above problems to the following antithesis question.

$$\begin{aligned} \max w(\alpha_i, \alpha_i^*) = & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) \\ & -\varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{j=1}^{N} y_i(\alpha_i - \alpha_i^*) \\ s.t. & \begin{cases} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \end{aligned} \tag{4}$$

Finally, the corresponding regression function Eq. (1) can be directly expressed as follows:

$$f(x) = \sum_{j=1}^{N}(\alpha_i - \alpha_i^*)K(x_i, x) + b \tag{5}$$

In Eq. (4), $\alpha_i^{(*)}$ is so-called Lagrange multiplier which satisfies the equation: $\alpha_i \times \alpha_i^* = 0$, $\alpha_i^{(*)} \geq 0$. $K(x_i, x_j)$ is the nuclear function, and its value is inner product of vector x_i and x_j in feature space, namely

$$K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle \tag{6}$$

Any function satisfying the Mercer condition can be taken as the nuclear function. The study selects the Gauss nuclear function:

$$K(x_i, x_j) = \exp\left[-\left(\frac{1}{\sigma^2}\right)\right] \|x_i - x_j\|^2 \tag{7}$$

where σ is nuclear parameter.

Based on KKT condition [17], only a part of coefficient $(\alpha_i - \alpha_i^*)$ is non-zero value, and training error is bigger than ε or is equal to ε, these training samples are support vector. Obviously regression function is completely decided by it.

It is well known that SVR generalization performance (estimation accuracy) depends on a good setting of parameters C, ε and the kernel parameter [18]. Parameter C determines the trade off between the model complexity and the degree to which deviations larger than ε are tolerated. C does not appear in Eq. (5) but as an upper bound of α_i, α_i^*, the larger C is, the better performance SVR has. But if C is too large (infinity), then the objective is to minimize the empirical risk only, without regard to model complexity part in the optimization Eq. (2). Parameter ε controls the width of the ε -insensitive zone, used to fit the training data. The value of ε can affect the number of SVs used to construct the regression function. Larger ε value results in fewer SVs

selected, and also result in more 'flat' (less complex) regression estimates. The kernel parameters implicitly define the nonlinear mapping from input space to some high dimensional feature space and affect the complex of data distribution in high dimensional feature space. However, we must notice that each of them reaching an optimal point sometimes does not lead to a good performance for SVR, so just when the combination of them arrives an optimal value, we may get a nice performance.

2.2 Particle swarm optimization

Basic concept of particle swarm optimization

Particle swarm optimization (PSO) is an evolutionary computation technique. Similar to genetic algorithms, PSO is a population based optimization tool [5, 6]. It is inspired by social behavior among individuals. Particles (individuals) representing a potential problem solution move through an n-dimensional search space. Each particle i maintains a record of the position of its previous best performance in a vector called *pbest*. When a particle takes the entire population as its topological neighbors, the best value is a global best and is called *gbest*. All particles can share information about the search space.

Representing a possible solution to the optimization problem, each particle moves in the direction of its best solution and the global best position discovered by any particles in the swarm. Each particle calculates its own velocity and updates its position in each iteration. Let $p_{i,d}$ denote the best previous position encountered by the ith particle. $p_{g,d}$ denotes the global best position thus far, and t denotes the iteration counter. The current velocity of the dth dimension of the ith particle at time t+1 is

$$\begin{aligned} v_{i,d}(t+1) = w v_{i,d}(t) + c_1 r_1 (p_{i,d}(t) - x_{i,d}(t)) \\ + c_2 r_2 (p_{g,d}(t) - x_{i,d}(t)) \end{aligned} \tag{8}$$

$$v_{i,d} \in [-v_{\max}, v_{\max}] \tag{9}$$

In the above formula, r_1 and r_2 is a random function in the range [0, 1], positive constant c_1 and c_2 are personal and social learning factors, and w is the inertia weight. Inertia weight was first introduced by Shi and Eberhart [19]. Inertia weight balances the global exploration and local exploitation. The velocity is restricted to the $[-v_{\max}, v_{\max}]$ range in which $v_{\max}$ is a predefined boundary value. The value of $v_{\max}$ determines the resolution of the search regions between the present and target position. Eberhart and Shi [20] suggested that $v_{\max}$ be set at about 10–20% of the dynamic range of the variable in each dimension. The new position of a particle is calculated using the following formula:

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \tag{10}$$

Binary PSO

Many optimization problems are set in a space featuring discrete, qualitative distinctions between variables and between levels of variables. Typical examples include problems which require the ordering or arranging of discrete elements. Kennedy and Eberhart [21] proposed a binary PSO in which a particle moves in a state space restricted to zero and one on each dimension, in terms of the changes in probabilities that a bit will be in one state or the other.

The velocity formula (8) remains unchanged except that $x_{i,d}$, $p_{i,d}$ and $p_{g,d}$ are integers in 0, 1 and $v_{i,d}$ must be constrained to the interval [0.0, 1.0]. This can be accomplished by introducing a sigmoid function $S(v)$, and the new particle position is calculated using the following rule:

$$\textit{if} \quad \frac{1}{2} < S(v_{i,d}), \quad \textit{then} \quad x_{i,d} = 1; \quad \textit{else} \quad x_{i,d} = 0 \tag{11}$$

The function $S(v)$ is a sigmoid limiting transformation defined by Eq. (12).

$$S(v) = \frac{1}{1 + e^{-v}} \tag{12}$$

2.3 Simulation annealing algorithm

The simulation annealing algorithm (SA) is one kind of global intelligence optimization algorithms, whose principle is the simulation of metal annealing process. In the optimum searching process of the SA, it accepts both the optimal solution and the suboptimal solution with a certain probability. This enables it to search optimal solution in more new partial regions of the solution space. The probability is related to the initial temperature of the annealing process, and the higher the temperature, the greater the probability of receiving a suboptimal solution. The possibility will finally vanish along with the gradual reduction of the temperature. Due to this mechanism, SA has better capability of avoiding the partial extreme value compared to other optimization algorithms [11].

3 PARTICLE SWARM ALGORITHM OPTIMIZED BY SA

This study uses SA to optimize the iteration process of each particle in the particle swarm according to the following formulae (13)–(15). Let $\Delta E = f(pbest) - f(gbest)$, Here f is fitting function, *pbest* is the best local particle in the particle swarm, and *gbest* is the global one. Regarding each particle, if $\Delta E \leq 0$, $gbest = pbest$ is accepted; else $gbest = pbest$ is accepted with the probability p defined by Eq. (13).

$$p = \exp(-\Delta E / T) \tag{13}$$

$$T_0 = \textit{initial } T \tag{14}$$

$$T_t = \lambda T_{t-1} \tag{15}$$

where p is the probability that the particle in the particle swarm accepts a suboptimal solution, T_0 is the initial temperature of the SA when particle swarm algorithm starts, T is the corresponding simulation annealing temperature to the current particle swarm iteration, λ is the temperature attenuation coefficient $\lambda \in (0, 1)$. p reduces gradually to 0 according to Eq. (15) along with the reduction of the temperature. Through the introduction of the SA process into the optimum searching process of the PSO (named SAPSO), the capability of avoiding the partial optimal value is enhanced theoretically.

The SAPSO algorithm in our investigation is described as follows:

Step 1: Initialization of each individual. Initialize randomly the element particles of the swarm including swarm size n, iteration N, acceleration constant c_1 and c_2, inertia weight w, the position vector X and the velocity vector V and so on;

Step 2: Calculate the evaluation value of each particle using the fitness function $f(x_i(t))$ $(i = 1, 2, \ldots, n)$;

Step 3: To each particle, Compare each particle's fitness value $f(x_i(t))$ with its personal best position's fitness value; the better is the new personal best position $pbest_i(t)$;

Step 4: To each particle, Compare each particle's fitness value $f(x_i(t))$ with its global best position's fitness value; the better is the new global best position $gbest_i(t)$;

Step 5: Update the position and the velocity of each particle according to Eq. (8)–(10);

Step 6: Calculate the new evaluation function for each particle $f(x_i(t+1))$;

Step 7: Search for the best position among all particles' shown personal best position by SA: calculate the change of the evaluation function $\Delta E = f(pbest_i(t)) - f(gbest_i(t))$, if $\Delta E \leq 0$, accept new position with the probability 1; if $p = \exp(-\Delta E/T) > r$ (r is a random number in [0, 1]), accept new position with the probability p; if accept new position, then let $T \leftarrow \lambda T$; otherwise refuse new position, go to step 5; it will generate a set of new particles, denote the global best position as *gbest*;

Step 8: If the number of iterations reaches the maximum or reaching the stop condition, then stop. The latest *gbest* is regarded as the optimal solution. Otherwise, go to step 2.

4 SAPSO FOR PARAMETER DETERMINATION AND FEATURE SELECTION OF SVR

4.1 Particle representation

To implement our proposed approach, this research used the RBF kernel function (defined by Eq. (7)), because the RBF kernel function can analyze higher dimensional data and requires that only three parameters, C, σ and ε. When the RBF kernel is selected, the parameters and features of SVR used as input attributes can be optimized by our proposed SAPSO approach.

Particle type				
	Input feature mask (discrete)	C (continuous)	σ (continuous)	ϵ (continuous)
Representation	$x_{i,1}, x_{i,2}, \ldots, x_{i,n_F}$	x_{i,n_F+1}	x_{i,n_F+2}	x_{i,n_F+3}

TABLE 1
Particle i is comprised of four parts: input feature mask,C, σ and ε

The particle therefore, is comprised of four parts: the features mask (discrete-valued), C, σ and ε (continuous-valued), when the RBF kernel is selected. Table 1 shows the representation of particle i with dimension of $n_F + 3$, where n_F is the number features that varies from different datasets. The feature mask is Boolean that "1" represents the feature is selected, and "0" indicates feature is not selected.

4.2 Fitness definition

Prediction accuracy or prediction error and the number of selected features are the two criteria used to design a evaluation function. Thus, for the particle with low prediction error and a small number of features produce a good performance. We solve the multiple criteria problem by creating a single objective function that combines the two goals into one. As defined by formula (16), the objective function has two predefined weights: (i) w_E for the prediction error; and (ii) w_F for the selected feature. In the investigation, the value of the mean absolute percent error (MAPE), shown as (17), serves as one part of criterion. The weight MAPE can be adjusted to a high value (such as 100%) if it is the most important. The particle with low objective function (Error) value has high probability to effect the other particles' positions of the next iteration.

$$Error_i = w_E \times MAPE_i + w_F \times \frac{\sum_{j=1}^{n_F} f_j}{n_F} \tag{16}$$

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{17}$$

f_j the value of feature mask– "1" represents that feature j is selected and "0" represents that feature j is not selected, and n_F is the total number of features.

4.3 The SAPSO algorithm of SVR

To guarantee valid results for making predictions regarding new data, the data set was further randomly partitioned into training sets and independent test sets via a k-fold cross validation. For example $k = 10$ which means that all of the data will be divided into ten parts, each of which will take turns being

the test data set. The other nine data parts serve as the training data set for adjusting the model prediction parameters.

All of the input variables are scaled during the data preprocessing stage. Generally, each feature can be linearly scaled to the [0, 1] range using the following formula (18), where x is original value, x' is scaled value, $\max_a$ is the maximum value of feature a, and $\min_a$ is the minimum value of feature a.

$$x' = \frac{x - \min_a}{\max_a - \min_a} \tag{18}$$

Randomly split the data into k groups using k-fold cross validation. Each group contains training, validation and test sets. The training set is used to build the SVR model. The validation set is used to determine the proper training iteration to avoid overtraining. The test set is used to evaluate the model's prediction accuracy. For each dataset group, detailed experimental procedure for training and testing is as follows:

Step 1. Data preparation: Training, validation, and test sets are represented as Tr, Va, and Te, respectively.

Step 2. Particle initialization and SAPSO parameters setting: Generate initial particles comprised of the feature mask, C, σ and ε. Set the SAPSO parameters including number of iterations, velocity limitation, number of particles, particle dimension, and weight for accuracy calculation. Set *iteration*=0, and perform the training process from step 3–8.

Step 3. Set *iteration* $i = i + 1$.

Step 4. SVR model training:

(a) Training and validation set preprocessing: select input features for training and validation data sets according to the feature mask which is represented in the first part of a particle.

(b) SVR regression accuracy calculation: for the training set Tr, conduct k-fold cross validation (CV) on the training set, and calculate the average CV accuracy based on the (C, σ, ε) which is represented in the second, third and fourth part of a particle.

(c) Evaluated the prediction accuracy on validation set Va using the trained model based on the (C, σ, ε)and the whole training set Tr.

Step 5. Fitness evaluation: For each particle, evaluate its evaluate function by formula (16). Note that the $Error_i$ in formula (16) is set as average CV error obtained in the previous step.

Step 6. Update the global and personal best according to the evaluation results by SAPSO. Record the average training CV accuracy and validation accuracy for the global and personal best.

Step 7. Particle manipulations: Each particle moves to its next position using formula (8)–(10).

Step 8. Stop condition checking: If stopping criteria (maximum iterations predefined) are met, go to the next step, otherwise, go to step 3.

Particle type (application in STLF)							
	Input feature mask (discrete)				C (continuous)	σ (continuous)	ϵ (continuous)
Representation	T_{ave}, Wp,	$T_{\max}$, $D(d)$,	$T_{\min}$, $W(d)$,	H, Rf	$x_{i,9}$	$x_{i,10}$	$x_{i,11}$

TABLE 2
Particle i is comprised of two parts: input feature masks and parameters

Step 9. To avoid overtraining, we observe the validation accuracy curve, and stop training when the iteration has the best validation accuracy during the training process.

Step 10. With the stopping training iteration determined in the previous step, recall the recorded (in database) feature mask, C, σ, and ε in the stopping iteration. Retrain and build the SVR on the larger set Tr + Va based on the selected feature subset and SVR model parameters. Finally, measure testing accuracy on test set Te via the trained SVR.

Step 11. End the training and testing procedure.

5 APPLICATION OF SAPSO-SVR IN SHORT-TERM LOAD FORECASTING

5.1 Data descriptions

Short-term load forecasting (STLF) is a multivariable forecasting problem. It can be treated as a function regression problem. The load of next day is output of regression model and the corresponding load influencing factors such as historical load data, temperature information and meteorological information are the input data of regression model. So the dimension of solution space of the SAPSO is comprised of two parts: the first part is feature mask, according to the experience and historical database, there are eight features including average temperature (*Tave*), highest temperature (*Tmax*), lowest temperature (*Tmin*), humidity index (H), wind power (*Wp*), type of the day *D(d)*, type of the week *W(d)* and the rain fall (Rf). The second part is the three parameters C, σ, and ε of the Gaussian kernel function. So the dimension of solution space is 11 which showed in Table 2.

The improved SAPSO algorithm is employed to feature selection and parameters determination of SVR for STLF. The training data is supplied by historical database. The final target is to find a mapping function from influencing factors to future load with a good generalization capability. While using SAPSO-SVR model for short-term load forecasting, the historical data is divided into two data sets. One is the training data and the other is the testing data. The training data is used to train the SVR, and the testing data is used to evaluate the trained SVR.

5.2 Experimental procedures

The procedures for the proposed SAPSO-SVR in STLF are presented as follows:

Step 1. Determine learning range of the proposed SAPSO-SVR. The model is trained and tested by using the similar day's data which in same season of forecast day, such as past 30 days from the day before forecast day and past 60 days before and after forecast day in previous year.

Step 2. Treat disordered samples. Checking every point, pick out and correct all disordered data which contravenes the general rule of the load series.

Step 3. Scale the load data. Using Eq. (19), the learning data are scaled into [0, 1]. Where L_t is original value, $\hat{L}$ is scaled value, $L_{\max}$ is the maximum value of load, and $L_{\min}$ is the minimum value of load.

$$\hat{L} = \frac{L_t - L_{\min}}{L_{\max} - L_{\min}} \tag{19}$$

Step 4. Train the proposed model. Take the parameters of these similar days as training samples to train SVR. Select features and determine parameters of SVR by SAPSO described in previous section of this paper.

Step 5. Test error calculation. Based on the trained SVR, the estimation result of the testing samples is gained. Then, the testing error is calculated using Eq. (16)–(17).

Step 6. Select parameters. The result of step (5) serves as the subjective function for identifying suitable features and parameters employing the proposed SAPSO algorithm. This iterative process is presented in detail in Fig. 1.

FIGURE 1
Framework of improved SAPSO-SVR for STLF.

Step 7. Forecast next day's load. Taking the influencing factor vector of forecasting day into the trained SVR with proper parameters, the given point load of next day will be forecasted.

Step 8. Correct result. Because load influenced by a series of uncertain factors, it is hard to get a satisfy result absolutely using historical data. Operators should correct the forecasting result from experience.

5.3 System implementation details

This investigation uses the historical load data in a practical electric system from 2005 to 2006 in China. The data includes the 24 points load and the features of each day. The features include weather conditions such as average temperature, highest temperature, lowest temperature, humidity index, wind power, rain fall and type of date etc. Using these historical data, 24 load points of next day are forecasted. This paper adopts the improved SAPSO algorithm to select features and determine parameters of SVR and the improved SMO algorithm train SVR [25]. The Gaussian kernel is selected as the kernel function.

The swarm size is set to 40 particles; the dimension of solution space is $3 + 8$. The dimensions are correspond to three parameters C, σ, and ε and eight features including average temperature (*Tave*), highest temperature (*Tmax*), lowest temperature (*Tmin*), humidity index (H), wind power (*Wp*), type of the day *D(d)*, type of the week *W(d)*, and rain fall (*Rf*). The features should be discreted before applying SAPSO algorithm in feature selection and parameters determination of SVR.

In training stage, the initial inertia weight is one, and the highest iterative number is 100. The range constrains of the there parameters are set as $C \in [0.01, 1000]$, $\sigma \in [0.01, 10]$, and $\varepsilon \in [0.01, 0.9]$. This study set $v_{\max}$ at about 2–10% of the dynamic range of the variable on each dimension for the continuous type of dimensions, achieving a better and stable convergence in our preliminary experiments. Therefore, $[-v_{\max}, v_{\max}]$ is predefined as $[-20, 20]$ for parameter C, $[-0.2, 0.2]$ for parameter σ, and $[-0.02, 0.02]$ for parameter ε. For the discrete type particle for feature mask, we set $[-v_{\max}, v_{\max}] = [-1, 1]$. Preliminary experiments also let this study set the personal and social learning factors $(c_1, c_2) = (2, 2)$ that achieves better prediction accuracy. The MAPE's weight w_E is adjusted to 95%, and the feature size's weight w_F is set to 5%. These system parameter settings are summarized in Table 3.

The simulator is programmed by using Matlab7.0. The training data set are fed into the SAPSO-SVR model. With the trained SVR model, the prediction result of testing data is gained. Then, the adjusted parameters C, σ, ε and the appropriate features with minimum testing error are selected. The parameters determination and feature selection of SVR are adjusted by SAPSO algorithm. Finally, the trained SVR model with appropriate features and parameters is built to forecast electricity load of next similar day.

	C	σ	ϵ	Feature mask
Search range	[0.01,1000]	[0.01, 10]	[0.01,0.9]	0,1
Velocity$[-v_{\max}, v_{\max}]$	$[-20, 20]$	$[-0.2, 0.2]$	$[-0.02, 0.02]$	$[-1, 1]$
Learning factors(c_1, c_2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)

TABLE 3
Setting of the system parameters

Time	C	σ	ϵ	MAPE
1:00	1000.0	1.6382	0.0529	0.0278
2:00	570.8	8.9540	0.0573	0.0146
3:00	495.9	9.0025	0.0618	0.0149
4:00	469.3	6.1143	0.0690	0.0281
5:00	109.0	9.3608	0.0580	0.0044
6:00	573.5	7.4710	0.0734	0.0158
7:00	395.8	9.4505	0.0382	0.0234
8:00	857.5	2.4018	0.0314	0.0049
9:00	400.1	8.2110	0.0277	0.0298
10:00	986.3	8.5740	0.0527	0.0276
11:00	939.1	7.8096	0.0256	0.0202
12:00	556.9	4.7601	0.0685	0.0382
13:00	482.1	9.4326	0.0341	0.0046
14:00	248.1	4.2845	0.0604	0.0275
15:00	1.3	10.0000	0.0298	0.0009
16:00	18.6	10.0000	0.0213	0.0164
17:00	20.9	7.5608	0.0474	0.0174
18:00	404.0	10.0000	0.0704	0.0041
19:00	252.6	6.9232	0.0411	0.0477
20:00	639.4	4.3367	0.0641	0.0189
21:00	582.7	7.1128	0.0945	0.0370
22:00	683.9	9.9070	0.0790	0.0022
23:00	1000.0	2.0933	0.0606	0.0232
24:00	2.3	0.5025	0.0309	0.0188

TABLE 4
Parameters and MAPE of SAPSO-SVR with feature selection

5.4 Experimental results

The proposed method is used to forecast the load of each hour in April 1, 2006. The forecasting accuracy is illustrated in Table 4. From Table 4, it can be observed that the large gap of parameters emerges, especially the parameter C. To a giving precision constrain of testing results, it is hard to find a uniform parameters compatible to all points. On the other hand, the forecasting

accuracy is influenced by factor selection. The experiment results showed that the influenced factors are different according to the season of the forecast day. That is to say, different season should select different influence factors of SVR for forecasting. For example, the rain fall is an important factor in summer and not important in winter.

Figure 2 showed that the forecasting capacity of the improved SAPSO-SVM model with feature selection better than the PSO-SVM model. In Figure 3, it is showed that the corresponding forecasting error of the proposed model is smaller than the PSO-SVR.

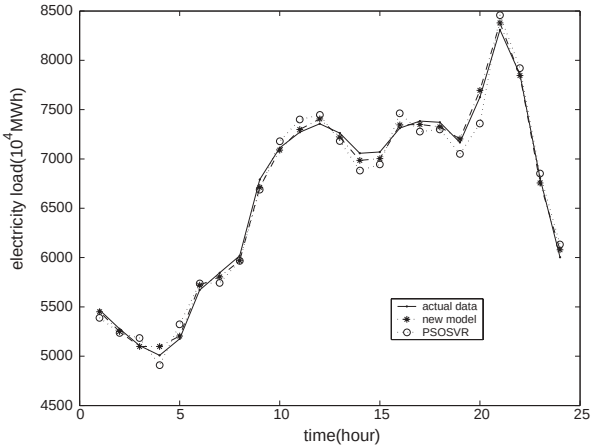

FIGURE 2
The contrast of the forecasting results of new model and PSO-SVR.

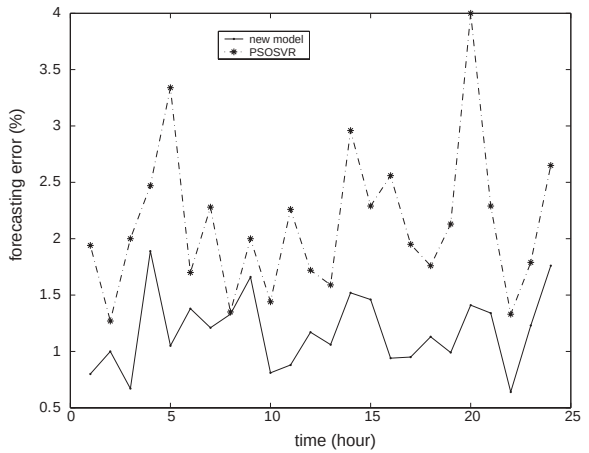

FIGURE 3
The contrast of the forecasting errors of new model and PSO-SVR.

Day	Model 1	Model 2	Model 3
2006.4.1	3.45	2.47	2.20
2006.4.2	3.22	2.13	2.00
2006.4.3	2.54	2.59	2.46
2006.4.4	2.55	1.29	1.25
2006.4.5	1.78	1.74	1.71
2006.4.6	2.48	2.91	2.53
2006.4.7	2.62	2.30	2.01
2006.4.8	3.56	4.18	3.55
2006.4.9	3.21	1.63	1.55
2006.4.10	2.58	2.60	2.63
2006.4.11	1.33	1.21	1.12
2006.4.12	1.58	1.70	1.57
2006.4.13	1.27	1.11	1.22
2006.4.14	1.59	1.27	1.22
2006.4.15	3.52	3.26	3.01
2006.4.16	3.25	4.01	3.15
2006.4.17	3.09	2.30	2.18
2006.4.18	1.59	1.68	1.51
2006.4.19	2.25	2.51	2.36
2006.4.20	2.43	2.49	2.42
2006.4.21	1.92	1.93	1.82
2006.4.22	2.46	2.41	2.23
2006.4.23	3.24	3.24	3.23
2006.4.24	2.86	1.61	1.48
2006.4.25	1.69	1.78	1.66
2006.4.26	2.30	2.47	2.32
2006.4.27	1.96	1.98	1.72
2006.4.28	2.66	2.41	2.23
2006.4.29	3.56	2.98	2.82
2006.4.30	4.25	3.79	3.73
Mean	2.56	2.34	2.16

TABLE 5
Forecasting results for different models

In addition, the proposed method is used to forecast the load of each day in a month. It is used to forecast the load data from April 1, 2006 to April 30, 2006.The performance of the proposed method is also compared with the forecasting results of the PSO-SVR model and the simple SVR in Table 5.

Figure 4 showed that the improved SAPSO-SVR model with feature selection has a more accurate forecasting capacity than the other two models.

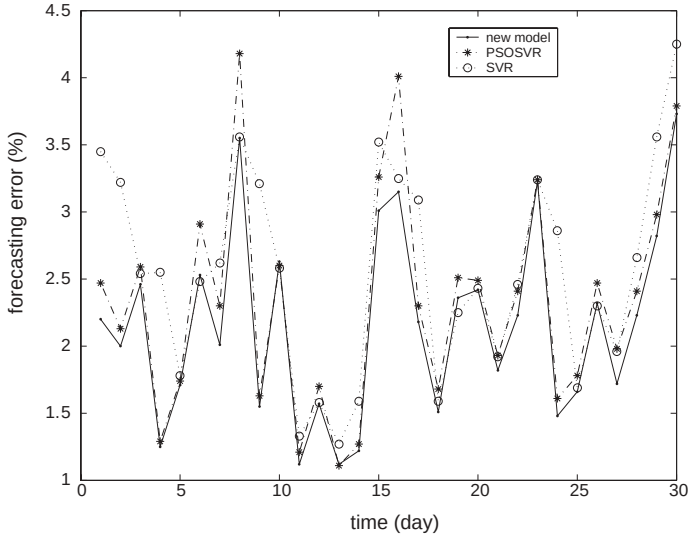

FIGURE 4
The contrast of the forecasting errors of three models which Model 1 is simple SVR, Model 2 is PSO-SVR without feature selection and Model 3 is SAPSO-SVR with feature selection.

The results indicate that the improved SAPSO-SVR model with feature selection is superior to the PSO-SVR model and the model without feature selection. Model 1 is simple SVR, Model 2 is PSO-SVR without feature selection, Model 3 is SAPSO-SVR with feature selection.

6 CONCLUSIONS

For SVR, input feature subset selection and the kernel parameters setting are crucial problems. This study proposed a novel hybrid SAPSO-SVR system to solve these two problems. First, two types of PSO versions, namely continuous valued and discrete version were combined to optimize the best SVR model parameters and the optimal feature subset. The proposed system can optimize the model parameters and search the discriminating feature subset simultaneously. Second, by incorporating with the simulated annealing method, the global searching capacity of the particle swarm optimization was enhanced. The improved particle swarm optimization algorithm (SAPSO) was used to optimize the parameters and select the input features of SVR.

Based on the operational data provided by a regional power grid in north China, the improved SAPSO-SVR system was used in the actual short-term load forecasting. The experimental results showed the proposed approach can correctly select the discriminating input features and compared to the PSO-SVR and the traditional SVR, the average time of the proposed method

in the experimental process reduced and the forecasting accuracy increased respectively. So, the improved method is better than the other two methods. The test results showed the proposed forecasting method could provide a considerable improvement of the forecasting accuracy. In the future, how to use advanced techniques to updating the suitable features and parameters of the proposed model in time is a very important direction.

7 ACKNOWLEDGMENTS

The authors would like to thank the National Science Council of the Republic of China, for financially supporting this research under National Natural Science Foundation No. 70671039.

REFERENCES

[1] V. N. Vapnik. The Nature of Statistical Learning Theory, Springer-Verlag, NY, USA, 1995.

[2] G. P. Zhang. Neural networks for classification: A survey. *IEEE Trans. Syst. Man Cybern.-Part C: Appl. Rev*. **30**(4) (2000), 451–462.

[3] C.-L. Huang, C.-J. Wang. A GA-based attribute selection and parameter optimization for support vector machine. *Expert Syst. Appl*. **31**(2) (2006), 231–240.

[4] C. W. Hsu, C. J. Lin. A simple decomposition method for support vector machine, *Mach. Learn.* **46**(1–3) (2002), 219–314.

[5] J. Kennedy, R. C. Eberhart. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*. Piscataway, NJ, pp. 1942–1948, 1995.

[6] J. Kennedy, R. C. Eberhart, and Y. Shi. Swarm Intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2001.

[7] Zhihua Cui, Xingjuan Cai, Jianchao Zeng, and Guoji Sun. Particle Swarm Optimization with FUSS and RWS for High Dimensional Functions, *Applied Mathematics and Computation* **205**(1) (2008), 98–108.

[8] Ramana Murthy, G, Senthil Arumugam, M., and Loo, C. K. Hybrid particle swarm optimization algorithm with fine tuning operators. *Int. J. Bio-Inspired Computation* **1**(1/2) (2009), 14–31.

[9] Cai, X., Cui, Y., and Tan, Y. Predicted modified PSO with time-varying accelerator coefficients. *Int. J. Bio-Inspired Computation* **1**(1/2) (2009), 50–60.

[10] F. van den Bergh, A. P. Engelbrecht. A study of particle swarm optimization particle trajectories. *Inf. Sci.* **176**(8) (2006), 937–971.

[11] A. Chatterjee, P. Siarry. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Comput. Operations Res.* **33**(3) (2006), 859–871.

[12] D. A. Yi, G. E. Xiu-run. An improved PSO-based ANN with simulated annealing technique. *Neurocomputing* **63**(1) (2005), 527–533.

[13] P.-Y. Yin, S.-S. Yu, P.-P. Wang, and Y.-T. Wang. A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems. *Comput. Stand. Interfaces* **28**(4) (2006), 441–450.

[14] Parsopoulos, K. E., Kariotou, F., Dassios, G., and Vrahatis, M. N. Tackling magnetoencephalography with particle swarm optimization. *Int. J. Bio-Inspired Computation* **1**(1/2) (2009), 32–49.

[15] N. Cristianini, J. Shawe-Taylor. An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, UK, 2000.

[16] U. KreBel. Pairwise Classification and Support Vector Machines. *Advances in Kernel Methods-Support Vector Learning*. MIT Press, Cambridge, MA, USA, pp. 254–268, 1999.

[17] Peressini, A. L., Sullivan, F. E. and Uhl, J. J. The mathematics of nonlinear programming. Springer, New York, 1988.

[18] Vladimir Cherkassky, Yunqian Ma. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks* **17**(1) (2004), 113–126.

[19] Y. Shi, R. C. Eberhart. A modified particle swarm optimizer. In: *Proceedings of the IEEE International Conference on Evolutionary Computation*, Piscataway, NJ, pp. 69–73, 1998.

[20] R. C. Eberhart, Y. Shi. Particle swarm optimization: Developments, application and resources. In: *Proceedings of the 2001 Congress on Evolutionary Computation* vol. 1, pp. 81–86, 2001.

[21] J. Kennedy, R. C. Eberhart. A discrete binary version of the particle swarm algorithm. In: *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*. Piscataway, NJ, pp. 4104–4109, 1997.

[22] V. Kecman. Learning and Soft Computing, MIT Press, Cambridge, MA, USA, 2001.

[23] D. K. Agrafiotis, W. Cedeno. Feature selection for structure-activity correlation using binary particle swarms. *J. Med. Chem.* **45**(5) (2002), 1098–1107.

[24] C. W. Hsu, C. C. Chang, C. J. Lin. (2003). A practical guide to support vector classification, available at: http://www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf.

[25] H. T. Lin, C. J. Lin. (2003). A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods, Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, available at: http://www.csie.ntu.edu.tw/cjlin/papers/tanh.pdf.

[26] W. F. Punch, E. D. Goodman, M. Pei, C.-S. Lai, P. Hovland, and R. Enbody. Further research on feature selection and classification using genetic algorithms, in: Stephanie Forrest (ed.), Fifth International Conference on Genetic Algorithms. San Mateo, CA, Morgan Kaufmann, pp. 557–564, 1993.

[27] S. L. Salzberg. On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Min. Knowl. Discov.* **1** (1997), 317–327.

[28] C.-C. Chang, C.-J. Lin. (2001). LIBSVM: A library for support vector machines, Software available at: http://www.csie.ntu.edu.tw/cjlin/libsvm.

[29] P. M. Murphy, D. W. Aha. (2001). UCI Repository of machine learning databases, Department of Information and Computer Science. University of California, Irvine, CA, http://www.ics.uci.edu/mlearn/MLRepository.html .

[30] Tay, F. E. H., Cao, L. J. Modified support vector machines in financial time series forecasting. *Neurocomputing* **48**(10) (2002), 847–861.

[31] PAI Ping-feng , HONG Wei-chiang. Support vector machines with simulated annealing algorithms in electricity load forecasting. *Energy Conversion and Management* **46**(17) (2005), 2669–2688.

[32] PAI Ping-feng , HONG Wei-chiang. Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. *Electric Power Systems Research* **74**(3) (2005), 417–425.

[33] CAO Li-juan. Support vector machines experts for time series forecasting. *Neurocomputing* **51**(4) (2003), 321–339.

[34] HUANG Wei, Nakamori Y, WANG Shou-yang. Forecasting stock market movement direction with support vector machine. *Computers and Operations Research* **32**(10) (2005), 2513–2522.

[35] Frauke Friedrichs, Christian Igel. Evolutionary tuning of multiple SVM parameters. *Neurocomputing* **64** (2005), 107–117.