Modified Parallel Cat Swarm Optimization in SVM Modeling for Short-term Cooling Load Forecasting

Yuanmei Wen
School of Information Engineering, Guangdong University of Technology, Guangzhou, China
Email: ym0218@gdut.edu.cn

Yanyu Chen
School of Information Engineering, Guangdong University of Technology, Guangzhou, China
Email: andybilly5@hotmail.com

***Abstract*—In order to improve forecasting accuracy of cooling load, this paper applies support vector machine (SVM) model with modified parallel cat swarm optimization (MPCSO) to forecast next-day cooling load in district cooling system(DCS). By extracting the Eigen value of the input historical load data, principal component analysis (PCA) algorithm is used to reduce the complexity of the data sequence. Based on cats' cooperation and competition, an MPCSO algorithm is proposed to optimize the hyper parameters for the SVM model. Finally, the SVM model with MPCSO (namely MPCSO-SVM) is established to conduct the short-term cooling load forecasting. Numerical example results show that the proposed model outperforms the existing alternative models. Thus, the proposed model is effective and applicable to cooling load forecasting.**

***Index Terms*—load forecasting model, support vector machine, parallel cat swarm optimization, principal component analysis**

I. INTRODUCTION

Load shifting in ice-storage district cooling system (DCS) is achieved through energy storing and releasing in the form of ice. In the power troughs hours, energy is stored by produce ice and it is released by melting the ice in the peak hours. To do so, the ice production and the operation strategy adopted by the chillers should be determined by the ice cooling demands that are subject to dynamic variation. Thus, it is crucial to accurately predict the cooling load such that the system can operate in a smooth and economical way. Every day, it needs to predict the cooling load for the next day, which is called short-term cooling load prediction. With historical cooling load data, this is done by statistical analysis in considering some key factors, such as temperature and humidity.

Currently, there are mainly two categories of techniques for cooling load prediction: mathematics and knowledge-based ones [1-4]. The first one is based on mathematical determination approach, such as regression analysis and time series method. Regression analysis method formulates the cooling load prediction problem by an empirically mathematical function of the main parameters that affect the cooling load, such as temperature, humidity [5, 6]. Generally, such a method is suitable for long-term forecasting with large samples. For instance, Yoshida [7] applies this method to predict air-conditioning load based on an ARX model derived from building load simulation. Results show that its average prediction error is 29% in summer and 12% in winter. Based on the continuity of time and the cooling load inertia, time series method can be used to predict the future cooling load [8-12]. Therefore, it is applicable to short-term forecasting of cooling load if the load variation is small. In [13], a modified seasonal exponential smoothing method is used to predict the cooling load of office buildings such that the average error is 8.8%. Nevertheless, due to the nonlinearity, time-varying, and uncertainty of the cooling load, although the mathematical determination approach has been used to predict cooling load, the complex relationship between predicted data cannot be described by strict mathematical equations. Moreover, by using such a method, even if a model is established, it is very challenging to solve it due to that one needs to identify the parameters in the model. Another disadvantage is that, by such methods, it requires to process a large number of sample data, leading to computationally inefficiency.

Artificial neural network (ANN) and support vector machine (SVM) belong to the knowledge-based intelligent models. It is known that SVM can be used to approximate any sufficiently complex nonlinear relationship well and is adaptable to unknown or uncertain systems. Ferrano [14] uses ANN to predict the required ice production for the thermal storage system in a building. It is shown that, for cooling load prediction, ANN is better than the traditional approaches in terms of both prediction accuracy and computational efficiency [15, 16]. However, its error rate is too high such that it is not applicable to real-life applications. To improve the prediction accuracy, modified ANN techniques are presented by combining with robust filter, wavelet, and fuzzy theory [17-26]. It is reported that, in forecasting a building air conditioning load, compared with back

doi:10.4304/jsw.9.8.2093-2104

propagation neural network (BPNN), the root mean square error and the average relative error obtained by radial basis function neural network (RBFNN) are reduced by 36% [27]. Although, to some extent, ANN and modified ANN can improve the prediction accuracy, they require large number of samples for learning and are not computationally efficient.

With the structure risk minimization criterion, a support vector machine (SVM) has some advantages over an ANN model. SVM can learn with small and high-dimensional samples, also, it can avoid a local minimum solution. Hence, it has been widely applied to forecasting, such as power grid load forecasting, natural gas usage trend forecasting, and etc. [28] Besides, it has been applied to predict energy consumption in a building [29, 30]. In addition, it has high prediction accuracy. It is reported that, for building air conditioning load prediction, the root mean square error obtained by SVM is about 50% lower than that obtained by BP neural network [31]. Nevertheless, prediction accuracy by using SVM is sensitive to its parameters. To solve this problem, in recent years, swarm intelligence and bionic algorithm are used to optimize the parameters of SVM [32-36]. The applicability of different algorithms for parameter optimization of SVM in building cooling load prediction is evaluated by Li [37-39]. It is shown that the prediction accuracy obtained by using an optimized SVM is improved.

Due to that a DCS works for 24 hours every day, the variation range of the cooling load is large. Hence, it is very difficult to predict the next day load [17, 40, 42-48]. To further improve the prediction accuracy of the unsteady cooling load in a DCS, support vector machine (SVM) model with modified parallel cat swarm optimization (MPCSO) is proposed in this paper. Its forecasting ability is studied by a case problem in Guangzhou, China. In this study, 4 months cooling load data from Jul.1 to Oct.31 are used for testing the proposed MPCSO-SVM forecasting model.

The rest of this paper is organized as follows. Section 2 presents the method of SVM parameters optimization with MPCSO. Section 3 provides the prediction modeling processes based on MPCSO-SVM. Section 4 illustrates a real case that reveals the forecasting performance of the proposed MPCSO-SVM model with comparing to the existed forecasting models. The conclusions are given in Section 5.

II. SVM Parameter Optimization With MPCSO

SVM is widely used for non-linear regression. However, the regression quality depends on the SVM regression parameters [49-53]. To improve the regression forecasting accuracy, an MPCSO is presented to optimize the SVM regression parameters.

A. SVM Parameter Selection

To use SVM for the short-term cooling load forecasting, it needs a training data set denoted as $\left(x_i, y_i\right)$, $i = 1, 2, \cdots, n$, where represents the days during which the data are collected. In $\left(x_i, y_i\right)$, $x_i \subset \mathbb{R}^M$ is the M-dimensional cooling load forecasting feature extracted from the historical operating data, while $y_i \subset \mathbb{R}$ there is the corresponding target output, the real value of the cooling load from the historical operating data. Then, the cooling load regression forecasting problem can be formulated as

$$f\left(x\right) = \left[\omega \cdot \varphi\left(x\right)\right] + b \qquad (1)$$

where $f\left(x\right)$ is the regression estimation function which constructed through learning by using the cooling load training data set, ω is weight vector, b is the threshold value, and $\varphi\left(x\right)$ is the nonlinear function that maps the input cooling load prediction feature space to a high-dimensional feature space which is the only hidden space. This regression problem is equivalent to solve the following optimization problem.

$$\begin{aligned} &\min_{\omega, b, \xi, \xi^*} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \\ &\text{subject to} \quad \begin{array}{l} f\left(x_i\right) - y_i \le \varepsilon + \xi_i \\ y_i - f\left(x_i\right) \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0, i = 1, 2, \cdots, n \end{array} \end{aligned} \qquad (2)$$

where $\|\omega\|^2$ is the weight vector norm called confidence risk, which is used to constrain the model structure capacity in order to obtain better generalization performance; ξ and ξ^* are two positive slack variables which measure the deviation $\left(y_i - f\left(x_i\right)\right)$ from the boundaries of the ε -insensitive zone with $\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right)$ called experience risk. To decide the balance between confidence risk and experience risk, penalty coefficient C , cost of error, is used. Insensitive loss coefficient ε , the width of the tubes, is one of the key factors in experience risk. By constructing the Lagrange function, the dual problem can be given as

$$\begin{aligned} &\min_{\alpha, \alpha^*} \frac{1}{2}\left(\alpha - \alpha^*\right)^T K\left(x_i, x_j\right)\left(\alpha - \alpha^*\right) \\ &\quad + \varepsilon\sum_{i=1}^{n}\left(\alpha_i + \alpha_i^*\right) - \sum_{i=1}^{n} y_i\left(\alpha_i - \alpha_i^*\right) \\ &\text{subject to} \quad \begin{array}{l} \sum_{i=1}^{n}\alpha_i = \sum_{i=1}^{n}\alpha_i^* \\ 0 \le \alpha_i, \alpha_i^* \le C, i = 1, 2, \cdots, n \end{array} \end{aligned} \qquad (3)$$

where α and α^* are Lagrange multiplier coefficients obtained by solving the dual optimization problem, and $K\left(x_i, x_j\right) = \varphi\left(x_i\right) \times \varphi\left(x_j\right)$ is the kernel function. It is a nonlinear function that maps the input feature space to a high dimensional space for regression forecasting calculation. The most widely used kernel function is the radial basis function. This is because that it can well map the historical data to the high-dimensional feature space in a nonlinear way with just a width parameter to be optimized. With δ being the width, the radial basis function is

$$K\left(x_i, x_j\right) = e^{-\frac{\left\|x_i - x_j\right\|^2}{\delta^2}} \tag{4}$$

Finally, the regression forecasting function is

$$y = f\left(x\right) = \sum_{i=1}^{n}\left(\alpha_i - \alpha_i^*\right) K\left(x_i, x\right) + b \tag{5}$$

The parameters that determine the regression quality are the penalty coefficient C , the insensitive loss coefficient ε , and the width of radial basis function kernel function δ . The penalty coefficient determines the forecasting model training error in the proportion of the objective function. The insensitive loss coefficient controls the regression function such that it is within the insensitive domain width of cooling load historical data. Hence, it affects the number of support vectors. The width of radial basis function controls radial scope of the kernel function and it affects the SVM regression complexity. Owing to the importance of the parameters, they require to be optimized. The MPCSO is used to optimize these SVM regression parameters next.

B. SVM Parameter Optimization with MPCSO

In this subsection, we present an MPCSO to search the optimal penalty coefficient C_{best} , insensitive loss coefficient ε_{best} and the width of radial basis function kernel function δ_{best} in the SVM prediction parameters space $\left(C, \varepsilon, \delta\right)$.

Cat Swarm Optimization (CSO) is first proposed by Chu and Tsai[54, 55] via observing the behavior of cats in daily life in 2006. It is mainly applied to solve the optimization problem and the satisfied results have been achieved. Parallel Cat Swarm Optimization (PCSO) [56] and Enhanced Parallel Cat Swarm Optimization (EPCSO) [41] are proposed by Tsai in 2008 and 2012, respectively. These methods can obtain higher accuracy and faster computation speed in cluster analysis, expression recognition, and multi-objective optimization [57-61].

Strong curiosity to moving objects and the outstanding skill of hunting are the two distinctive features of a cat. These two behavioral traits of cats are modeled by CSO: seeking mode and tracing mode, which reflects the cooperation between "cats". However, in order to further improve the CSO optimization speed and prediction accuracy, PCSO absorbs the advantage of parallel computing to improve the tracing mode such that a parallel tracing mode is adopted. PCSO establishes a plurality of CSO to search the best parameters in the prediction parameter space independently and simultaneously by dividing the "cat swarm" into some groups. At the same time, it adds information exchanging mode such that the CSOs can exchange information occasionally, which reflects the cooperation between groups. Hence, PCSO is particularly suitable for multi-parameter optimization and multi-threaded computing, because it makes full use of computer resources and obtain the optimal result quickly. When PCSO is running, the "cats" are randomly distributed in the prediction parameter space. Inevitability, it results in a state such that there more "cats" in some areas and less in others. Hence, one obtains the optimal parameters in the more "cats" areas, because lots of optimization is executed in the more "cats" areas. This process reduces the SVM prediction accuracy and robustness of optimal parameters. Thus, an MPCSO algorithm is proposed to solve the problem so as to sprinkle the "cats" evenly in the prediction parameter space to lead the PCSO to search the best parameters evenly in the whole SVM prediction parameter space $\left(C, \varepsilon, \delta\right)$.

By MPCSO, the procedure for the parameter optimizing of SVM is as follows.

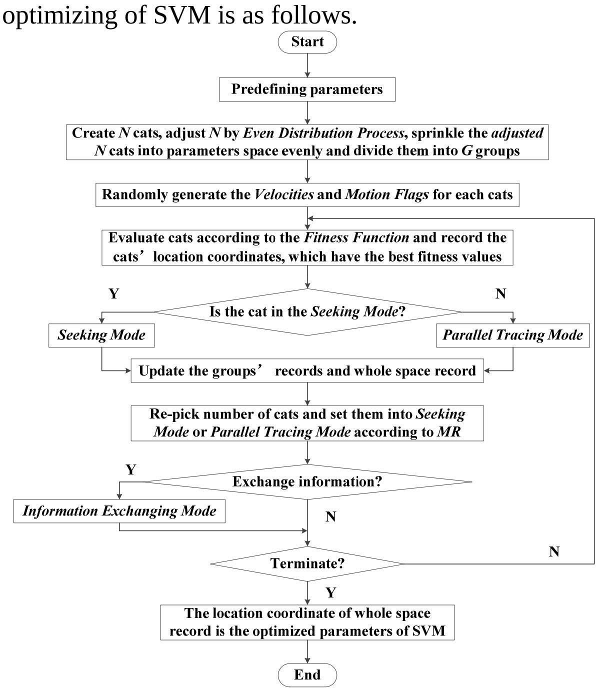


Figure 1. The flow chart of prediction parameters optimization with MPCSO-SVM.

Step 1: Create N cats, adjust N and sprinkle the *adjusted* N cats into the 3 dimensional SVM prediction parameter space $\left(C, \varepsilon, \delta\right)$ evenly by the Even Distribution Process and divide the *adjusted* N cats into G groups.

Step 2: Randomly generate the *Velocities* for each dimension V_C , V_ε and V_δ of each cat, which should be in the predefined range. Set the *Motion Flag* of each cat to make them move into the Parallel Tracing Mode or the Seeking Mode according to the predefined value of MR, where $MR \in \left[0,1\right]$ denotes the ratio of cats working in Seeking Mode and Parallel Tracing Mode.

Step 3: For each cat, take location coordinates $\left(C_i, \varepsilon_i, \delta_i\right)$ into the *Fitness Function* of SVM, calculate the fitness values respectively, and record the location coordinates and the fitness values.

Now, the structure of cats in the cat swarm has been built as

Cat(i)={
Location $\left[C,\varepsilon,\delta\right]$,
Velocity $\left[C,\varepsilon,\delta\right]$,
Fitness,
Motion Flag,
Group
}

Step 4: Move the cats into the Seeking Mode or the Parallel Tracing Mode according to the status of the *Motion Flags*. Select the cat which has the best fitness value from each group to update the corresponding group record, and select the group which has the best fitness value from the whole cat swarm to update the whole space record.

Step 5: Re-pick a number of cats and set them into Seeking Mode or Parallel Tracing Mode according to *MR*.

Step 6: Check whether the number of iterations reaches a predefined *ECH* (a threshold to exchange the information of groups). If the condition is satisfied, move into the Information Exchanging Mode.

Step 7: Check whether the process satisfies the termination condition. If yes, output location coordinate of the whole space record, which represents the best SVM prediction parameters $\left(C_{best},\varepsilon_{best},\delta_{best}\right)$, and stop the MPCSO. Otherwise, go to **Step 3**.

C. *Even Distribution Process*

Even Distribution Process can change the number of cats N, and sprinkle the "cats" evenly in the SVM prediction parameter space $\left(C,\varepsilon,\delta\right)$ to achieve the objective of seeking the best parameters in the whole space evenly and increasing the robustness of PCSO.

The detailed steps are as the follows:

Step 1: Calculate the cube root of the predefined number of cats *N*, and round it to the positive direction.

Step 2: Record the rounded number *D* and let the cube of *D* be the new number of cats *N*.

Step 3: In the SVM prediction parameter space $\left(C,\varepsilon,\delta\right)$, each dimension is divided into D+1 equal parts such that the space is divided into $(D+1)^3$ equal parts, and the endpoints of each equal part, excluding the points on the boundary of the space, are the location coordinates of the cats.

Now, the cats have been sprinkled evenly in the SVM prediction parameter space $\left(C,\varepsilon,\delta\right)$.

D. *Seeking Mode*

In the Seeking Mode of MPCSO, one of the cat's behaviors is simulated such that the cat keeps alertly looking around for its next movement during a period of resting. There are three essential factors need to be predefined in the Seeking Mode, they are Seeking Memory Pool (*SMP*), Seeking Range of the selected Dimension (*SRD*), and Counts of Dimension to Change (*CDC*).

The detailed steps for the Seeking Mode are shown in Fig.2.

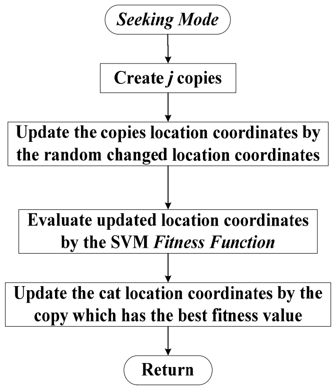


Figure 2. The flow chart of seeking mode in MPCSO-SVM.

Step 1: Create *j* copies of the cat's current position, where *j=SMP-1* and retain the present position as one of the candidates.

Step 2: For each copy, according to *CDC*, plus or minus the *SRD* percent of the present location and replace the old ones with the following equation:

$$\forall j, L_{j,d}\left(t\right)=\left(1+rand\times SRD\right)\times L_{j,d}\left(t-1\right), \quad d=C,\varepsilon,\delta \tag{6}$$

where *j* is the *j*-th copy of the current location, d is the dimension of the prediction parameters' space and rand is a random variable in the range $\left[0,1\right]$. $L_{j,d}\left(t-1\right)$ is the last location coordinate of *j* copy *d* dimension and $L_{j,d}\left(t\right)$ is the updated value.

Step 3: Calculate the fitness values of all candidate points respectively. If all the fitness values are not exactly equal, calculate the selecting probability P_i of each candidate point as

$$P_i=\frac{\left|FS_i-FS_b\right|}{FS_{\max}-FS_{\min}}, where\ 0<i<j \tag{7}$$

If the goal of the fitness function is to find the minimum value, then let $FS_b=FS_{\max}$. Otherwise, let $FS_b=FS_{\min}$, where $FS_{\max}$ denotes the largest FS in the candidates and $FS_{\min}$ denotes the smallest one.

Step 4: Select the copy which has the best fitness value from the *j* copies and record its corresponding location coordinate.

Step5: Compare this copy's fitness value with the original fitness value. If the copy's fitness value is better than the original one, move the cat to the new position, otherwise, keep the cat stay on the original position.

E. Parallel Tracing Mode

The Parallel Tracing Mode applies the group's recorded location coordinate which has the best fitness value to update the location coordinate and velocity of the cat in the group. It can direct the cat to approach the corresponding group's best location coordinate and search the location coordinate on the path.

The parallel tracing mode is described in Fig.3.

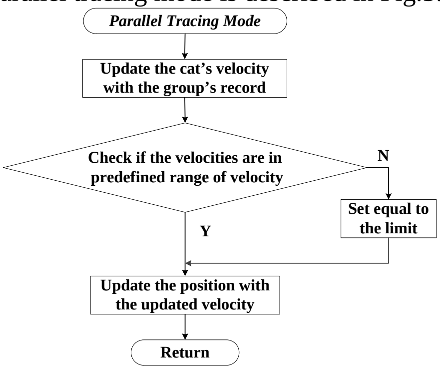


Figure 3. The flow chart of Parallel tracing mode in MPCSO-SVM.

Step 1: Update the velocity for every dimension $V_d(t)$ for the cat at the current iteration according to following equation.

$$\begin{aligned} &V_d(t) = V_d(t-1) + \\ &\qquad r \times c \times \left[L_{G_{best},d}(t-1) - L_d(t-1) \right], \quad (8) \\ &d = C, \varepsilon, \delta \end{aligned}$$

where $L_{G_{best},d}(t-1)$ is the d-dimension location coordinate of the group which the cat belongs to and has the best fitness value at the previous iteration. $L_d(t-1)$ is the *d*-dimension location coordinate of the cat at the previous iteration. $V_d(t-1)$ is the *d*-dimension velocity of the cat at the previous iteration. c is a constant and r is a random value in the range of $[0,1]$.

Step 2: Check if the velocities are in predefined range of velocity. In case the new velocity is over-range, it is set equal to the limit.

Step 3: Update the position of cat according to the following equation.

$$L_d(t) = L_d(t-1) + V_d(t) \tag{9}$$

F. Information Exchanging Mode

Information Exchanging Mode forces the groups to exchange their information and achieves the group cooperation. It predefines a parameter *ECH* to control when this mode is executed. The information exchanging mode is applied once per *ECH* iteration. It consists of the following four steps (shown in Fig.4).

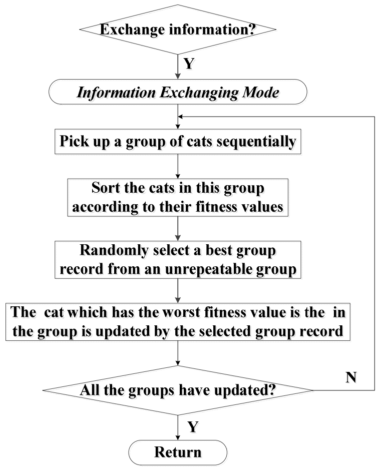


Figure 4. The flow chart of the information exchanging mode in MPCSO-SVM.

Step 1: Pick up a group of cats sequentially and sort the cats in this group according to their fitness values.

Step 2: Randomly select a best group record from an unrepeatable group.

Step 3: The location coordinate of the cat whose fitness value is the worst in the group is replaced by the selected group record's location coordinate.

Step 4: Repeatedly perform **Steps 1-3** several times to let every group receives a best location coordinate from the others.

III. The Forecasting Processes Of MPCSO-SVM Method

The modeling process of short-term cooling load forecasting based on MPCSO-SVM method is shown in Fig.5.

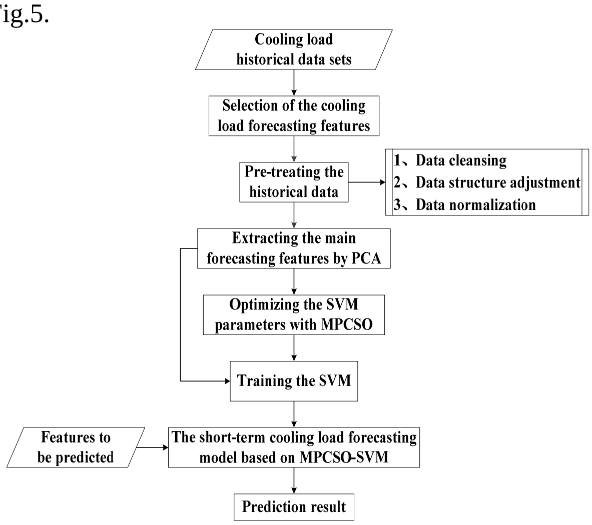


Figure 5. The short-term cooling load forecasting model flow chart.

This modeling processes consist of the following steps: selecting the cooling load forecasting features, historical data pre-treating, the main forecasting features extracting by PCA (Principal Component Analysis) method, SVM parameter optimization with MPCSO, SVM training to obtain the short-term cooling load forecasting model based on MPCSO-SVM.

A. *Selection of the Cooling Load Forecasting Features*

It follows from the results obtained in [4, 13, 20, 27, 31] that the closer the days during which the data are collected are to the days during which cooling load is to be predict, the more the prediction accuracy is impacted. In addition, the environment temperature and humidity are also the predominant influencing factors. Therefore this paper selects 14 features as the inputs of the proposed SVM model, which are the cooling load of day 1 and 2 before the day to be predicted, the maximum and minimum temperature & humidity of day1 and 2 before the day to be predicted, the maximum and minimum temperature & humidity of the day to be predicted. The selected cooling load forecasting features are showed in Fig. 6.

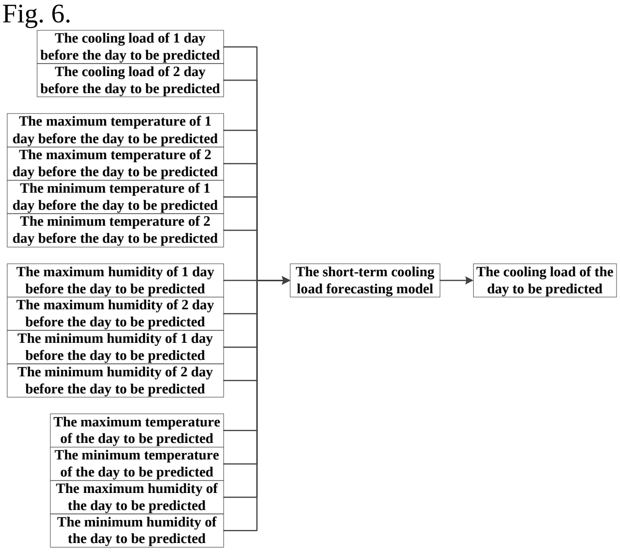


Figure 6. The selected cooling load forecasting features.

B. *Historical Data Pre-treating*

Historical forecasting feature data pre-treating includes data cleansing, data structure adjustment, and data normalization. Data cleansing corrects obvious mistakes in the data and fills in missing ones by doing the arithmetic mean of the two adjacent data. Data structure adjustment makes the data fit to train the SVM model. This paper applies the following function to normalize the training data set and predicting set.

$$y = \frac{(y_{\max} - y_{\min}) \times (x - x_{\min})}{x_{\max} - x_{\min}} + y_{\min} \tag{10}$$

where x is the original data, $x_{\max}$ and $x_{\min}$ are the minimum and maximum value of the original data, respectively. y , $y_{\max}$ and $y_{\min}$ are the expected normalized value of x , $x_{\max}$ and $x_{\min}$ respectively.

C. *Main Forecasting Feature Extracting by PCA*

The larger the sample dimension, the more computational time of SVM spends. To solve the problem, PCA is applied to extract the principal components of the forecasting features and reduce the number of dimensions accordingly. PCA decrease the number of dimensions of data with the information loss minimization by selecting maximal variance samples [62]. Then, the feature extracting can be eventuated to the eigenvalue computing of covariance matrix.

The steps of extracting the principal components of forecasting features by PCA are shown as follows.

Step 1: Calculate the mean, variance matrix, and scatter matrix of the forecasting feature matrix.

Step 2: Calculate the variance matrix's eigenvalues and corresponding eigenvectors.

Step 3: Sort the eigenvalues in the descending order. Select the largest *n* eigenvalues and corresponding eigenvectors. In which, *n* must be set by the contribution value which is the ratio of the sum of the *n* eigenvalues and the sum of all the eigenvalues. Optimum contribution value is more than 90%.

Step 4: Calculate the product of forecasting feature matrix and the difference of eigenvector matrix and the average. Then, the *n*-dimensional matrix with the number of dimensions being reduced is obtained.

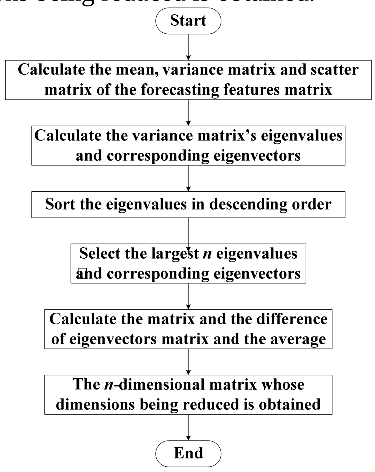


Figure 7. The flow chart of extracting the main forecasting features by PCA.

D. *MPCSO-SVM-based Short-term Cooling Load Forecasting Model*

After extracting the main forecasting features of historical data, the feature data and the MPCSO optimized SVM parameters are imported to SVM, then the SVM is trained to generate the MPCSO-SVM-based short-term cooling load forecasting model. Therefore the forecasting model consists of three sub-modules: database module, prediction module, and result displaying module.

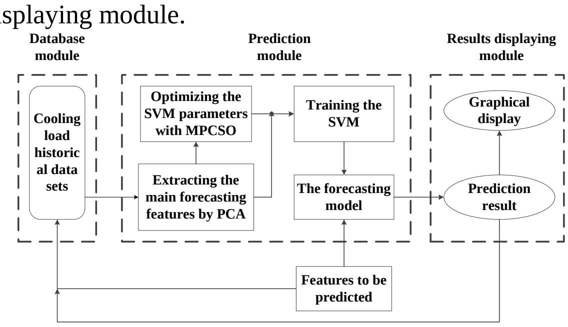


Figure 8. The structure of the MPCSO-SVM based short-term cooling load forecasting model.

1) Database module stores all the data needed by MPCSO-SVM-based short-term cooling load forecasting modeling, i.e., the historical data, the forecasting result,

and the real operation data. When a new forecasting process is finished, all the data are recorded in the database module. In this way, the database is updated constantly.

2) Prediction module is the core of the short-term cooling load forecasting model. It consists of forecasting feature selection, historical data pre-treating, feature extracting of historical data by PCA, SVM parameter optimization by MPCSO, and SVM training to generate the MPCSO-SVM based short-term cooling load forecasting model. When the prepared forecasting features are input, an excellent forecasting result is expected.

3) Results displaying module is mainly used for showing the forecasting results and the cooling load real values by data visualization technology.

E. Evaluation Index

In order to test the prediction accuracy of the short-term cooling load forecasting model, relative error (E_{RE}), root mean square relative error (E_{RMSRE}), and average relative error (E_{ARE}) are applied to evaluate the precision of prediction in this paper.

$$E_{RE} = \frac{y_i^{'} - y_i}{y_i} \times 100\% \tag{11}$$

$$E_{RMSRE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i^{'} - y_i}{y_i}\right)^2} \times 100\% \tag{12}$$

$$E_{ARE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i^{'} - y_i}{y_i}\right| \times 100\% \tag{13}$$

where y_i is the actual cooling load values, $y_i^{'}$ is the corresponding predictive cooling load values, n is the number of days in the testing set.

Mean square error (E_{MSE}) is selected to be the SVM's fitness function which is

$$E_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{'} - y_i\right)^2 \tag{14}$$

It is applied to calculate the location coordinate $\left(C_i, \varepsilon_i, \delta_i\right)$ of each cat in according with the fitness value.

IV. Case Study

A. Data Preparation

To verify the effectiveness of the short-term cooling load forecasting model based on MPCSO-SVM method proposed in this paper, cooling system in a subway station district with ice storage air-conditioning technology is taken as a case study. A totally 121 days' data set collected from Jul. 3, 2012 to Oct. 31, 2012, is used to verity the proposed MPCSO-SVM forecasting model. Three months data is collected as the train data set, the rest of month data is used as a test set to verify the MPCSO-SVM forecasting model. For instance, the months' data from Aug. 1 to Oct. 31 is collected to train the model, so the month data from Jul. 3 to Jul. 31 is used to test the model.

The forecasting features which selected from the relevant data of Jul. 1, 2012 to Oct. 31, 2012 are illustrated in Section III.A.

B. Parameter Predefinition for the Model

There are a few parameters needed to be predefined before running the model. They include the range of the SVM prediction parameter space $\left(C, \varepsilon, \delta\right)$, the range of the optimization velocity, the maximum number of the optimization iteration, the number of cats, *SMP*, *SRD*, *MR* and *ECH*.

The predefined values of these parameters in this case are shown in Tab. I.

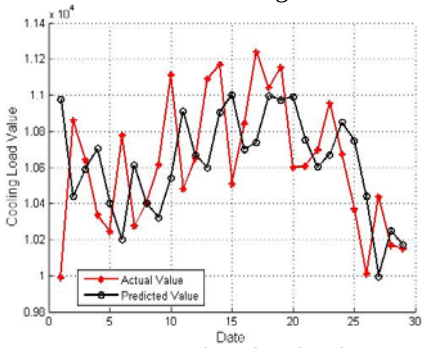
TABLE I.
PARAMETER TABLE.

The range of the SVM prediction parameters space						The range of the optimization velocity						*iter*	*CatNum*	*SMP*	*SRD*	*MR*	*ECH*
C_{max}	C_{min}	ε_{max}	ε_{min}	δ_{max}	δ_{min}	V_{Cmax}	V_{Cmin}	$V_{\varepsilon max}$	$V_{\varepsilon min}$	$V_{\delta max}$	$V_{\delta min}$						
10^3	10^{-2}	1	10^{-4}	10^3	10^{-2}	25	-25	10^{-3}	-10^{-3}	20	-25	100	200	5	0.2	0.05	5

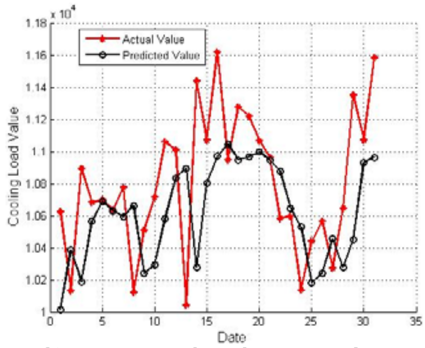
C. Prediction Results and Error Analysis

The 4 months forecasting cooling load results obtained by using the proposed model and the 4 months real values are respectively shown in Fig. 9.

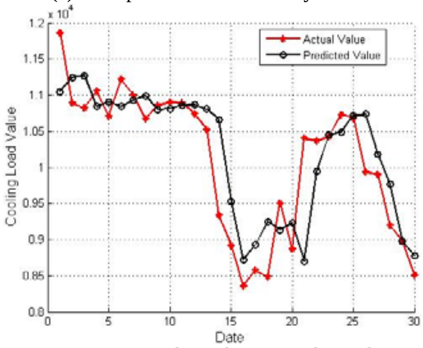
In Fig. 9, the red lines represent the every month's real cooling load of the subway station and the black line is the forecasting cooling load values. It can be seen that the variation range of this subway station's real cooling load values is great, and the linear regression methods are not suitable to predict the next day cooling load value. Fig. 9 shows that when the changes of the real values are very large, the forecasting values are much closer to the real values. That is to say, the short-term cooling load forecasting model based on MPCSO-SVM method proposed in this paper is proved to be applicable and feasible. In order to show the errors between the real values and the forecasting values more clearly, each day's relative errors between the real values and the forecasting values, root mean square error of each month, and average relative error of each month are recorded in Tab. II. It shows that the maximum relative error is 17.5391% and the minimum is 0.0006% (taken the absolute). In other words, the relative errors are less than 18% and the median of the relative errors is 2.7706%. The root mean square relative errors of the forecasting values respectively are 3.5583%, 4.1441%, 5.5473% and 5.9936%.The average of them is 4.8108%. The average relative errors of the four months are 2.8549%, 3.2742%, 4.0192% and 4.1937% respectively. The average of them is 3.5855%. This implies that the MPCSO-SVM method proposed in this paper is applicable to cooling load forecasting in DCS. The forecasting values from this model can meet the engineering requirements, and achieve the objective of storing ice reasonably, using energy effectively.



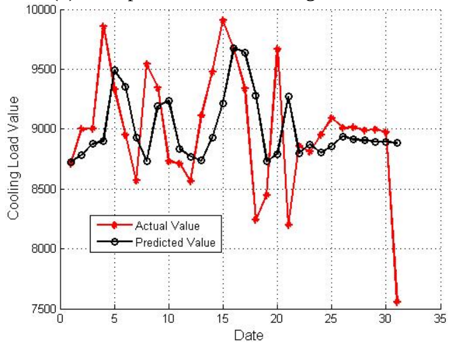
(a). Comparison about the July's data.



(b). Comparison about the August's data.



(c). Comparison about the September's data.



(d). Comparison about the October's data.

Figure 9. Comparison of predicted and actual values.

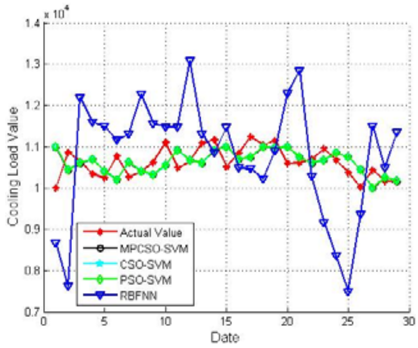
TABLE II.
THE SHORT-TERM COOLING LOAD FORECASTING MODEL PREDICTION ERRORS TABLE.

	Date		Date		Date		Date	
E_{RE}	3-Jul	10.0191%	1-Aug	-5.7473%	1-Sep	-6.8946%	1-Oct	0.1284%
	4-Jul	-3.8270%	2-Aug	2.5341%	2-Sep	3.2892%	2-Oct	-2.4023%
	5-Jul	-0.3359%	3-Aug	-6.4801%	3-Sep	4.1840%	3-Oct	-1.3963%
	6-Jul	3.5575%	4-Aug	-1.0664%	4-Sep	-1.8769%	4-Oct	-9.7092%
	7-Jul	1.6085%	5-Aug	-0.1061%	5-Sep	1.8819%	5-Oct	1.7174%
	8-Jul	-5.3103%	6-Aug	-0.0529%	6-Sep	-3.3728%	6-Oct	4.4992%
	9-Jul	3.3722%	7-Aug	-1.6804%	7-Sep	-0.5539%	7-Oct	4.2205%
	10-Jul	-0.0006%	8-Aug	5.3647%	8-Sep	2.9951%	8-Oct	-8.4477%
	11-Jul	-2.7073%	9-Aug	-2.5309%	9-Sep	-0.4869%	9-Oct	-1.6410%
	12-Jul	-5.0691%	10-Aug	-3.9337%	10-Sep	-0.7804%	10-Oct	5.8448%
	13-Jul	4.1576%	11-Aug	-4.3284%	11-Sep	-0.2596%	11-Oct	1.3848%
	14-Jul	0.1749%	12-Aug	-1.5765%	12-Sep	1.1292%	12-Oct	2.3574%
	15-Jul	-4.4033%	13-Aug	8.5139%	13-Sep	2.7253%	13-Oct	-4.1197%
	16-Jul	-2.3021%	14-Aug	-10.1740%	14-Sep	14.1250%	14-Oct	-5.7192%
	17-Jul	4.7031%	15-Aug	-2.4092%	15-Sep	6.7477%	15-Oct	-6.9775%
	18-Jul	-1.3005%	16-Aug	-5.5884%	16-Sep	4.3672%	16-Oct	0.1018%
	19-Jul	-4.3924%	17-Aug	0.9222%	17-Sep	4.1129%	17-Oct	3.2536%
	20-Jul	-0.3968%	18-Aug	-2.9524%	18-Sep	9.0839%	18-Oct	12.5718%
	21-Jul	-1.6213%	19-Aug	-2.2677%	19-Sep	-3.9718%	19-Oct	3.3233%
	22-Jul	3.7071%	20-Aug	-0.6418%	20-Sep	4.0764%	20-Oct	-9.0653%
	23-Jul	1.4084%	21-Aug	-0.1615%	21-Sep	-16.4215%	21-Oct	13.1103%
	24-Jul	-0.8642%	22-Aug	2.7560%	22-Sep	-4.0758%	22-Oct	-0.6523%
	25-Jul	-2.5389%	23-Aug	0.4715%	23-Sep	0.2332%	23-Oct	0.7010%
	26-Jul	1.6951%	24-Aug	3.8636%	24-Sep	-2.2830%	24-Oct	-1.6735%
	27-Jul	3.6888%	25-Aug	-2.4849%	25-Sep	0.3085%	25-Oct	-2.5692%
	28-Jul	4.3167%	26-Aug	-3.0599%	26-Sep	8.1119%	26-Oct	-0.7809%
	29-Jul	-4.1767%	27-Aug	1.7915%	27-Sep	2.8086%	27-Oct	-1.1136%
	30-Jul	0.8761%	28-Aug	-3.4728%	28-Sep	6.2539%	28-Oct	-0.9419%
	31-Jul	0.2618%	29-Aug	-7.9320%	29-Sep	-0.0675%	29-Oct	-1.1499%
			30-Aug	-1.2639%	30-Sep	3.0982%	30-Oct	-0.8928%
			31-Aug	-5.3703%			31-Oct	17.5391%
E_{RMSRE}	3.5583%		4.1441%		5.5473%		5.9936%	
E_{ARE}	2.8549%		3.2742%		4.0192%		4.1937%	

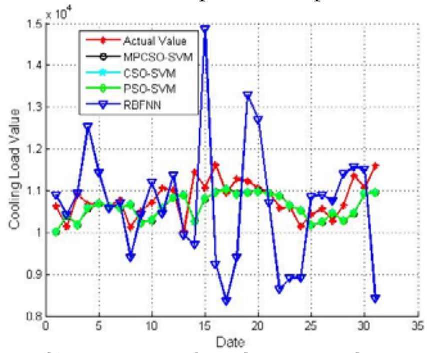
V. Comparative Study

To show the advantages of the short-term cooling load forecasting model based on MPCSO-SVM method, comparison of the forecast accuracy of the MPCSO-SVM, CSO-SVM (SVM forecasting model with cat swarm optimization), PSO-SVM (SVM forecasting model with particle swarm optimization)[63], and RBFNN (radial basis function neural network) [64,65]is made in this section.

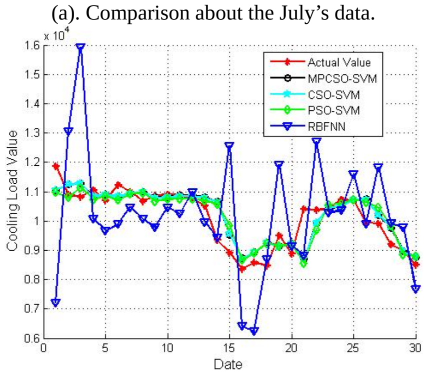
The predicted values of the aforementioned methods and the real values are shown in the Fig. 10. It can be clearly seen that the line of RBFNN method is the most far away from the line of the real values. At the same time, the lines of other methods are very close and have the same trends. That is to say that all of these three methods have an excellent prediction performance.



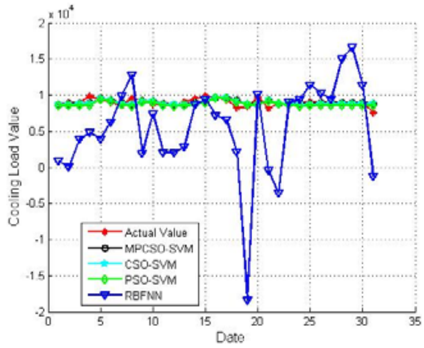
(a). Comparison about the July's data.



(b). Comparison about the August's data.



(c). Comparison about the September's data.



(d). Comparison about the October's data.

Figure 10. Comparison of MPCSO-SVM, CSO-SVM, PSO-SVM and RBFNN.

In order to show the errors of all the methods more clearly, Tab. III shows each month's relative errors of all the four methods. The root mean square relative errors and average relative errors of all the methods' are also recorded in Tab. III. From Tab. III, we can find that the RBFNN method have the maximum error as shown in Fig. 10. The other methods' relative errors are very close such that we can't distinguish clearly from the lines in Fig. 10.

TABLE III.
PREDICTION ERRORS TABLE.

	July				August			
	MPCSO-SVM	CSO-SVM	PSO-SVM	RBFNN	MPCSO-SVM	CSO-SVM	PSO-SVM	RBFNN
Maximum of E_{RE}	10.0191%	10.1118%	10.2086%	29.8568%	10.1740%	10.3942%	10.3717%	34.3196%
Minimum of E_{RE}	0.0006%	0.0271%	0.0332%	2.0281%	0.0529%	0.0136%	0.0049%	0.4031%
Median of E_{RE}	2.7073%	2.7367%	2.6983%	10.1722%	2.5341%	2.7208%	2.6990%	5.4861%
E_{RMSRE}	3.5583%	3.5688%	3.5779%	13.7240%	4.1441%	4.1669%	4.1669%	13.0963%
E_{ARE}	2.8549%	2.8575%	2.8573%	11.4012%	3.2742%	3.2803%	3.2832%	9.6186%
	September				October			
	MPCSO-SVM	CSO-SVM	PSO-SVM	RBFNN	MPCSO-SVM	CSO-SVM	PSO-SVM	RBFNN
Maximum of E_{RE}	16.4215%	16.5997%	17.5866%	47.1061%	17.5391%	16.3661%	13.6816%	318.1571%
Minimum of E_{RE}	0.0675%	0.2052%	0.1481%	0.2408%	0.1018%	0.0107%	0.2438%	2.4121%
Median of E_{RE}	3.1937%	3.1437%	3.1422%	8.8715%	2.4023%	3.1827%	4.2032%	51.2033%
E_{RMSRE}	5.5473%	5.5587%	5.8642%	18.0346%	5.9936%	6.0280%	6.2962%	83.8624%
E_{ARE}	4.0192%	4.0389%	4.2826%	13.1823%	4.1937%	4.4207%	5.0150%	58.2712%

However, from the records in Tab. III, especially, for the root mean square relative errors and the average relative errors of the MPCSO-SVM, CSO-SVM and PSO-SVM, we can find the differences between them. In July, the root mean square relative error of MPCSO-SVM is 0.0105% lower than CSO-SVM, 0.0196% lower than PSO-SVM and 10.1657% lower than RBFNN. In August, the root mean square relative error of MPCSO-SVM is 0.0228% lower than CSO-SVM, 0.0228% lower than PSO-SVM and 8.9522% lower than RBFNN. In September, the root mean square relative error of MPCSO-SVM is 0.0114% lower than CSO-SVM, 0.3169% lower than PSO-SVM and 12.4873% lower than RBFNN. In October, the root mean square relative error of MPCSO-SVM is 0.0344% lower than CSO-SVM, 0.3026% lower than PSO-SVM and 77.8688% lower than RBFNN. In summary, the average of the MPCSO-SVM's root mean square relative error is 0.0198% lower than CSO-SVM, 0.1655% lower than PSO-SVM and 27.3685% lower than RBFNN. The average relative error of MPCSO-SVM is 0.0026% lower than CSO-SVM, 0.0024% lower than PSO-SVM and 8.5463% lower than RBFNN in July. The average relative error of MPCSO-SVM is 0.0061% lower than CSO-SVM, 0.0090% lower than PSO-SVM and 6.3444% lower than RBFNN in August. The average relative error of MPCSO-SVM is 0.0197% lower than CSO-SVM, 0.2634% lower than PSO-SVM and 9.1631% lower than RBFNN in September. The average relative error of MPCSO-SVM is 0.2270% lower than CSO-SVM, 0.8213% lower than PSO-SVM and 54.0775% lower than RBFNN in October. To sum up, the average of the MPCSO-SVM's average relative error is 0.0639% lower than CSO-SVM, 0.2740% lower than PSO-SVM and 19.5328% lower than RBFNN. The Fig. 10 and Tab. III can fully explain the MPCSO-SVM is better than the other three approaches.

In summary, MPCSO-SVM is the best approach of the four methods, because it has the minimum errors. Hence, the short-term cooling load forecasting model based on MPCSO-SVM method proposed in this paper is feasible and highly practical.

VI. Summary And Conclusion

To predict the cooling load by SVM model, the selected three prediction parameters, namely the penalty coefficient C, the insensitive loss coefficient ε and the width of radial basis function kernel function δ, play a key role. An inappropriate penalty coefficient results in the phenomenon of under-learning or over-learning. An unsuitable insensitive loss coefficient weakens the robustness of SVM, because it affects the number of support vectors. If the width of radial basis function kernel function is not proper, the generalization ability of SVM is reduced. The optimization of these SVM prediction parameters improves SVM's prediction accuracy, enhances the robustness and generalization ability of SVM method.

In this paper, the short-term cooling load is used as the forecast target to verify the usefulness of this proposed MPCSO-SVM forecasting method. It is tested by case study from a subway station district cooling system and preferable prediction results are obtained. For the case problem, the average of predicted root mean square relative error is 4.8108%, 0.0198% lower than CSO-SVM, 0.1655% lower than PSO-SVM and 27.3685% lower than RBFNN. The average of predicted average relative error is 3.5855%, 0.0639% lower than CSO-SVM, 0.2740% lower than PSO-SVM and 19.5328% lower than RBFNN. These results show that the proposed model is effective in the cooling load forecasting.

In future work, we will study how to further improve the SVM's prediction accuracy, generalization ability, and robustness, and reduce the SVM's training time. All of these contribute to further optimize and improve the short-term cooling load forecasting. In addition, we will promote the MPCSO method and the MPCSO-SVM forecasting model such that that it can be applied to wider field.

References

[1] J. W. Yan, "Research and implement on optimal operating and control of district cooling system for energy efficiency," *South China University of Technology*, 2012.

[2] Y. Y. Duan, "Current status and development of air-conditioning load predicting," *Refrigeration and Air Conditioning*, 2012(03): 300-304.

[3] Z. J. Wang, "Study on hourly building cooling load prediction method during the urban energy planning stage," *Da Lian University of Technology*, 2010.

[4] Y. Liao, "Optimal control research based on cooling load prediction in ice storage air-conditioning system," *Guangdong University of Technology*, 2008.

[5] J. R. Forrester, W. J. Wepfer, "Formulation of a load prediction algorithm for a large commercial building," *ASHRAE transactions*, 1984, 90(2): 536-551.

[6] A. Q. Li, "Study on residential building thermal environment and energy prediction for small towns in chongqing," *Chongqing University*, 2006.

[7] H. Yoshida, T. Inooka, "Rational operation of a thermal storage tank with load prediction scheme by ARX model approach," *Proceedings Build Simulate*. 1997, 97.

[8] J. W. Macarthur, A. Mathur, J. Zhao, "On-line recursive estimation for load profile prediction," *ASHRAE transactions*, 1989, 95: 621-628.

[9] J. E. Seem, J. E. Braun, "Adaptive methods for real-time forecasting of building electrical demand," *Ashrae Transactions*, 1991, 97(1): 710-721.

[10] A. Kimbara, S. Kurosu, R. Endo, et al, "On-line prediction for load profile of an air-conditioning system," *ASHRAE TRANS*, 1995, 101: 198-207.

[11] J. Q. Xu, R. Xiao, C. Huang, et al, "ARMAX model of ice-storage air conditioning system load based on temperature interval," *Journal of Wuhan University of Technology*, 2009(10): 109-112.

[12] L. Chen, "Application of Wavelet Analysis and Time Series in air conditioning load forecasting," *Fluid Machinery*, 2008(02): 83-86.

[13] D. S. He, X. Zhang, "Analysis of air conditioning load prediction by modified seasonal exponential smoothing model," Journal of Tongji University (Natural Science). 2005(12): 1672-1676.

[14] F. J. Ferrano, K. Wong, "Prediction of thermal storage loads using a neural network," *ASHRAE Transactions (American Society of Heating, Refrigerating and Air-Conditioning Engineers);(United States)*, 1990, 96.

[15] M. Kawashima, C. E. Dorgan, J. W. Mitchell, "Hourly thermal load prediction for the next 24 hours by ARIMA, EWMA, LR, and an artificial neural network," *Chicago, IL, USA: ASHRAE*, 1995.

[16] J. F. Kreider, J. S. Haberl, "Predicting hourly building energy usage," *ASHRAE Journal (American Society of Heating, Refrigerating and Air-Conditioning Engineers);(United States)*, 1994, 36(6).

[17] M. Sakawa, S. Ushiro, K. Kato, et al, "Cooling load prediction in a district heating and cooling system through simplified robust filter and multi-layered neural network," *Tokyo*: 1999.

[18] N. Zhu, X. Y. Shi, J. J. Liu, et al, “The optimum study of neural network control in air conditioning system load prediction,” *Journal of Refrigeration*, 2002(02): 35-38.
[19] X. K. Chang, Q. Xia, X. Q. Jin, “A load prediction based on improved algorithms of BP model,” *Building Energy & Environment*, 2003(01): 5-7.
[20] H. J. Li, D. S. He, “Application of artificial neural network model to improve the generalization ability in air conditioning load prediction,” *Building Science*, 2009(06): 90-94.
[21] X. J. Dong, X. L. Zhou, S. X. Zhu. “Cooling load fuzzy-forecast theory and software in ice thermal storage air condition system,” *Refrigeration Air Conditioning & Electric Power Machinery*, 2006, 27(2): 22-25.
[22] Y. Yao, Z. W. Lian, S. Q. Liu, et al, “Hourly cooling load prediction by a combined forecasting model based on analytic hierarchy process,” *International Journal of Thermal Sciences*, 2004, 43(11): 1107-1118.
[23] X. Li, L. Ding, M. Shao, et al, “A Novel Air-Conditioning Load Prediction Based on ARIMA and BPNN Model,” *Shenzhen*: 2009.
[24] J. Liu, “Research on the ice-storage air-conditioning load forecasting based on Wavelet Analysis and Neural Network,” *Harbin Institute of Technology*, 2007.
[25] J. Liu, Y. T. Jiang, X. D. Li, “Load forecasting of the ice-storage air-conditioning based on Wavelet Neural Network,” *Low Temperature Architecture Technology*, 2009(03): 93-95.
[26] H. Q. Liu, G. F. Zhou, X. Q. Zhang, “Analysis of chaos characteristic and forecasting on air conditioning load,” *Sichuan Building Science*, 2009(05): 299-302.
[27] Q. Li, Q. L. Meng, “Prediction model of hourly air conditioning load of building based on RBF Neural Network,” *Journal of South China University of Technology (Natural Science Edition)*, 2008(10): 25-30.
[28] G. M. Xu, S. G. Huang, “Runway Incursion Event Forecast Model based on LS-SVR with Multi-kernel,” *Journal of Computers*, 2011(6):1346-1352
[29] L. Chen, D. M. Wu, “Discussing of air conditioning load predicting,” *Refrigeration*, 2007(03): 32-34.
[30] F. Lai, F. Magoules, F. Lherminier, “Vapnik's learning theory applied to energy consumption forecasts in residential buildings,” *International Journal of Computer Mathematics*, 2008, 85(10): 1563-1588.
[31] Q. Li, Q. L. Meng, Y. Hiroshi, et al, “Building air conditioning load prediction model based on support vector machine,” *Heating Ventilating & Air Conditioning*, 2008(01): 14-18.
[32] S. Y. Ohn, H. N. Nguyen, D. S. Kim, et al, “Determining optimal decision model for support vector machine by genetic algorithm,” *Computational & Information Science, Proceedings*, 2004, 3314: 895-902.
[33] T. Lee, M. Cho, C. Shieh, et al, “Particle swarm optimization-based SVM for incipient fault classification of power transformers,” *Foundations of Intelligent Systems, Springer*, 2006, 84-90.
[34] S. F. Yuan, F. L. Chu, “Fault diagnosis based on support vector machines with parameter optimization by artificial immunization algorithm,” *Mechanical Systems & Signal Processing*, 2007, 21(3): 1318-1330.
[35] X. J. Chen, J. Z. Wang, D. H. Sun, et al, “Time series forecasting based on novel support vector machine using Artificial Fish Swarm Algorithm,” *Fourth International Conference on Natural Computation*: 2008.
[36] X. D. Zhang, F. Hu, L. Zhao, “Recognition of practical speech emotion based on improved shuffled frog leaping algorithm and support vector machine,” *Signal Processing*, 2011, 27(5).
[37] J. H. Lv, X. M. Li, L. X. Ding, et al, “Applying principal component analysis and weighted support vector machine in building cooling load forecasting,” *International Conference on Computer and Communication Technologies in Agriculture Engineering*: 2010.
[38] X. M. Li, L. X. Ding, J. H. Lv, et al, “A novel hybrid approach of KPCA and SVM for building cooling load prediction,” *Third International Conference on Knowledge Discovery and Data Mining*: 2010.
[39] X. M. Li, L. X. Ding, Y. Y. Deng, et al, “Hybrid support vector machine and ARIMA model in building cooling prediction,” *International Symposium on Computer, Communication, Control and Automation*: 2010.
[40] F. Yik, J. Burnett, I. Prescott, “Predicting air-conditioning energy consumption of a group of buildings using different heat rejection methods,” *Energy & Buildings*, 2001, 33(2): 151-166.
[41] P. W. Tsai, J. S. Pan, S. M. Chen, et al, “Enhanced parallel cat swarm optimization based on the Taguchi method,” *Expert Systems with Applications*, 2012, 39(7): 6309-6319.
[42] T. T. Chow, K. F. Fong, A. Chan, et al, “Energy modeling of district cooling system for new urban development,” *Energy & Buildings*, 2004, 36(11): 1153-1162.
[43] T. T. Chow, G. Q. Zhang, Z. Lin, et al, “Global optimization of absorption chiller system by genetic algorithm and neural network,” *Energy & Buildings*, 2002, 34(1): 103-109.
[44] M. Sakawa, K. Kato, S. Ushiro, et al, “Operation planning of district heating and cooling plants using genetic algorithms for mixed integer programming,” *Applied Soft Computing*, 2001, 1(2): 139-150.
[45] M. Sakawa, K. Kato, S. Ushiro. “Operational planning of district heating and cooling plants through genetic algorithms for mixed 0--1 linear programming,” *European Journal of Operational Research*, 2002, 137(3): 677-687.
[46] A. L. Chan, T. Chow, S. K. Fong, et al. “Performance evaluation of district cooling plant with ice storage,” *Energy*, 2006, 31(14): 2750-2762.
[47] J. Ortiga, J. C. Bruno, A. Coronas, et al, “Review of optimization models for the design of polygene ration systems in district heating and cooling networks,” *Computer Aided Chemical Engineering*, 2007, 24: 1121-1126.
[48] A. E. Ben-Nakhi, M. A. Mahmoud. “Cooling load prediction for buildings using general regression neural networks,” *Energy Conversion and Management*, 2004, 45(13): 2127-2141.
[49] D. Niu, Y. Wang, D. D. Wu. “Power load forecasting using support vector machine and ant colony optimization,” *Expert Systems with Applications*, 2010, 37(3): 2531-2539.
[50] G. Ren, Z. P. Zhou, “Traffic safety forecasting method by particle swarm optimization and support vector machine,” *Expert Systems with Applications*, 2011, 38(8): 10420-10424.
[51] N. Anand, B. K. Panigrahi, S. Mathur, et al, “Stream flow forecasting by SVM with quantum behaved particle swarm optimization,” *Neurocomputing*, 2012.
[52] S. Chatterjee, “Vision-based rock-type classification of limestone using multi-class support vector machine,” *Applied Intelligence*, 2013: 1-14.
[53] V. Vapnik, *The Nature of Statistical Learning Theory*, *Springer*, 2000.
[54] S. C. Chu, P. W. Tsai, J. S. Pan, “Cat Swarm Optimization,” *RICAI 2006: Trends in Artificial Intelligence, Proceedings*, 2006, 4099: 854-858.

[55] S. Chu, P. Tsai, "Computational intelligence based on the behavior of cats," *International Journal of Innovative Computing, Information and Control*, 2007, 3(1): 163-173.

[56] P. Tsai, J. Pan, S. Chen, et al, "Parallel cat swarm optimization," *The 7th International Conference on Machine Learning and Cybernetics, Proceeding*, 2008.

[57] B. Santosa, M. K. Ningrum, "Cat Swarm Optimization for Clustering," *International Conference of Soft Computing and Pattern Recognition*, 2009.

[58] Y. Liu, Y. Shen, "Data clustering with CAT swarm optimization," *Journal of Convergence Information Technology*, 2010, 5(8).

[59] Z. H. Wang, C. C. Chang, M. C. Li. "Optimizing least-significant-bit substitution using cat swarm optimization strategy," *Information Sciences*, 2012, 192: 98-108.

[60] P. M. Pradhan, G. Panda, "Solving multi-objective problems using cat swarm optimization," *Expert Systems with Applications*, 2012, 39(3): 2956-2964.

[61] P. M. Pradhan, G. Panda, "Pareto optimization of cognitive radio parameters using multi-objective evolutionary algorithms and fuzzy decision making," *Swarm and Evolutionary Computation*, 2012.

[62] H. W. Shi, W. Q. Li, "Risk Assessment for Construction Projects Contracting Based on Unascertained Sets," Journal of Computers, 2011(6):2446-2453

[63] J. Yan, F. C. Tian, J. W. Feng, P. F. Jia, Q. H. He, Y. Shen, "A PSO-SVM Method for Parameters and Sensor Array Optimization in Wound Infection Detection based on Electronic Nose," Journal of Computers, 2012(7):2663-2670

[64] H. Q. Zhang, J. B. Li, "Prediction of Tourist Quantity Based on RBF Neural Network," Journal of Computers, 2012(7):965-970

[65] Z. C. Yang, "User-Online Load Movement Forecasting for Social Network Site Based on BP Artificial Neural Network," Journal of Computers, 2013(8):3176-3183

Yuanmei Wen was born in Hubei province, China, in 1968. She received the B.S degree in electrical engineering from Chongqing University (CU), Chongqing, China, in 1989, the M.S degree in power electronics from Huazhong University of Science and Technology (HUST), Wuhan, China in 1995, and the PH.D degree in material processing engineering from South China University of Technology (SCUT), Guangzhou, China, in 2009.

She was with Shashi Refrigerator Made Factory, Shashi, China, from 1989 to 1992. She joined Guangdong University of Technology, Guangzhou, China, in 1995, and is currently an associate professor of Information Engineering. She has published more than 20 articles in journals and also published many papers in conference proceedings. Her main research interests include intelligent information process and data mining.

Yanyu Chen was born in Guangdong Province, China, in 1990. He received the B.S degree in communication engineering from Ocean University of China, Qingdao College, Qingdao, China, in 2012. He is currently a M.S degree candidate in the school of information engineering, Guangdong University of Technology, Guangzhou, China. His main research interests include data mining.