A Combination of PSO and SVM for Road Icing Forecast

LI Jian
Department of Computer Engineering, Hubei University of Education, Wuhan, China
Email: leejan4ever@gmail.com

***Abstract*—The road icing is an adverse weather condition lead to dangerous driving conditions with consequential effects on road transportation. A numerical road icing predication approach is employed for automatic prediction of road icing conditions in three cities of Hubei Province, viz., Wuhan, Shiyan and Xianning. The approach is derived from the support vector machine (SVM). To improve the classification accuracy for road icing prediction, a modified particle swarm optimization (PSO) is employed to simultaneously optimize the SVM kernel function parameter and the penalty parameter. The modified PSO is derived from the genetic PSO, and employs the crossover operator and the mutation operator derived from the differential evolution to enhance searching performance. With the data from 1980 to 2006, using the proposed approach, the road icing models for the three cities in Hubei province are created, which have been used for the prediction from 2007 to 2008. The results have shown feasibility and effectiveness of the forecast approach.**

***Index Terms*—particle swarm optimization, support vector machine, evolutionary algorithm, weather forecasting**

I. Introduction

The Hubei Province locates in the middle and lower Yangtze which belongs to the north subtropical region monsoon climate region. Consequently, in the winter and the early spring, it experiences frequent snow, sleet, ice, and frost. Such adverse weather condition leads to road icing, a dangerous driving condition with consequential effects on road transportation.

To forecast the road icing conditions, various approaches have been proposed. Norrman proposed an expert system approach for classifying different types of slipperiness on roads in Sweden [1]. Alexander evaluated the road icing conditions in the Danish road network which was conducted based on observational data (road surface, air and dew point temperatures) from 2003-2007 to improve the quality of road forecasts [2]. An automated system for prediction of icing on the road was proposed by Konstantin, where a numerical forecasting system was developed. The system is based on a road conditions model forced by input from an operational atmospheric limited area model [3]. Weather prediction was employed in a road-maintenance decision-support system to provide real-time treatment guidance [4]. Liu studied the temperature factors which influenced the road icing conditions remarkably [5]. The relationship between developmental tendency of ice roads and the temperature in Heilongjiang was studied by Xu [6].

To provide accurate prediction, a numerical road icing forecast approach is employed which was derived from the support vector machine (SVM). SVM were suggested by Vapnik [7], and have recently been used in a range of problems. To build a SVM, one must choose a kernel function; set the kernel parameters such as the gamma (γ) for the radial basis function (RBF) kernel. Moreover, for the road icing prediction problem, a linearly inseparable problem, and a soft margin constant C has to be set too [8]. These parameters have a heavy impact on the classification accuracy [9]. To provide a promising result for the parameters the genetic algorithm (GA), the particle swarm optimization (PSO) and the grid algorithm were employed to optimize the parameters [8], [9].

PSO was proposed by Kennedy and Eberhart [11],[12]. And it is inspired by the social behavior of bird flocking, fish schooling and swarm theory, etc. The theoretical framework of PSO is very simple, and PSO possesses the properties of easy implementation and fast convergence. PSO was customized to continuous function optimization; hence PSO has gained much attention and been successfully applied in various fields. To enhance the search performance many technologies were employed, such as parameter adjust strategies [13], mutation operators [14], and hybrid PSO methods. In hybrid PSOs, CPSO methods were integrated with one or more assistant optimization methods such as the linear interior [15], the differential evolution [16], the genetic algorithm [17], etc. The simulation results have shown that incorporated with the mechanisms of other algorithms, the searching performance of CPSO could be enhanced significantly. Instead of traditional PSO, this study employs a modified PSO method which was derived from the traditional PSO, genetic PSO [18], and the differential evolution [19],[20]. The modified PSO method is employed as an optimization technique to simultaneously optimize the SVM parameters.

This paper is organized as follows. Section II describes the road icing prediction problem and SVM. Section III describes the modified PSO. Section IV presents the PSO-SVM hybrid system. Experimental results are given in Section V. And Section VI gives the conclusion.

II. SVM for Road Icing Prediction

A. The Road Icing Prediction

Ice on road surfaces is one of the most serious dangerous meteorological hazardous phenomenons, and it is well known that annually it causes serious injuries even

doi:10.4304/jcp.5.9.1372-1379

deaths in road accidents. The objective of the road icing prediction is to create a model based on the previously data to forecast if a region will occur road icing or not in a given day for the reduction of the threat of the ice-related accidents. Consequently, it is very important to select proper meteorological factors which influence the road icing remarkably.

In order for ice to be formed on the road surface the following is required: temperatures near and below 0°C as well as presence of water (moisture) on the surface of the road [2]. Consequently, the meteorological factors which may influence the temperatures and the presence of water should be considered. In [2], the latitude and longitude, time, road surface temperature, air temperature, dew point temperature, height of the station above sea level, etc were considered. Cloud cover and observations of temperature, humidity, water and ice on the road from the road station sites are taken into account in [3]. The temperature factors were discussed by Liu [5] and Xu [6].

In our study, based on the experienced forecaster and the statistical results, for each day twelve meteorological factors are selected. Mean air temperature, maximum temperature, minimum temperature, and minimum surface temperature are the factors impact the temperatures. Average relative humidity, precipitation, minimum relative humidity, mean vapor pressure, and the 24 hours precipitation of the passed day are the factors impact the water or moisture. Moreover, average air pressure, average total cloud cover, minimum total cloud cover are the factors impact both temperature and water, potentially. Moreover, in the three cities in the Hubei Province, viz., Wuhan, Shiyan and Xianning, a road icing season is considered to continue from November through March. The reason for this period is that based on the historical observation data, in the other periods the temperatures were higher than 0°C in almost all the days, and there were seldom road icing cases.

B. Support Vector Machine

The SVM is a new and promising technique for classification [7]. Given training vectors $x_i \in \boldsymbol{R}^n$, $i = 1,\ldots, l$, in two classes, and a vector $y \in \{1, -1\}$, the support vector technique requires the solution of the following optimization problem:

$$\min \quad \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i \tag{1}$$

$$\begin{aligned} &y_i(\omega^T\phi(x_i)+b) \ge 1-\xi_i \\ &\xi_i \ge 0, \quad i = 1,\ldots,l. \end{aligned} \tag{2}$$

Training vectors x_i are mapped into a higher (maybe infinite) dimensional space by the function ϕ . The existing common method to solve (1) is through its dual, a finite quadratic programming problem.

$$\min \ \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \tag{3}$$

$$\begin{aligned} &0 \le \alpha_i \le C, \quad i = 1,\ldots,l \\ &y^T\alpha = 0, \end{aligned} \tag{4}$$

where e is the vector of all ones, C is the upper bound of all variables, Q is an l by l positive semi definite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ and $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$ is the kernel. In the linear case, $K(x_i, x_j) \equiv x_i^T x_j$, but the extension to the nonlinear case is straightforward, by using any Mercer kernel $K(\cdot,\cdot)$.

After learning, the feed forward phase of the SVM is computed by

$$y(x) = \sum_{i=1}^{l}(y_i\alpha_i K(x_i, x)) + b \tag{5}$$

where x is a new sample, which is classified according to the sign of y.

The model of rainstorm forecast is nonlinear, so a Mercer kernel is needed. This paper employed the radial basis function (RBF) kernel as follows.

$$K(x_i, x_j) \equiv e^{-\gamma\sum_{i=1}^{n}(x_i - x_j)} \tag{6}$$

Consequently, it can be observed that there are two parameters influence the performance of SVM viz., C and γ. C is employed to handle the linear inseparable cases as a penalty coefficient to punish those wrong classifications which is an important parameter for road icing forecast. Firstly, the road icing model is proved to be a complex nonlinear system. Secondly, the errors in the measured data are unavoidable because of the system errors and the artificial factors. γ is another important parameter which influence the hyper plane remarkably.

III THE MODIFIED PARTICLE SWARM OPTIMIZATION

A. The Classical Particle Swarm Optimization

The classical particle swarm optimization (CPSO) consists of a swarm of particles flying through the search space. Each particle i in the swarm contains parameters for position (X_i^t) and velocity (V_i^t) where $X_i^t \in R^n$, $V_i^t \in R^n$ ($t \in N$, $i \in N$), and n is the dimensionality of the search space, t is the number of current iteration. The position of each particle represents a potential solution to the optimization problem. At each iteration step, each particle adjusts its velocity based on its momentum and the influence of its best position (P_i^t) tracked and the best position found by the neighbors (P_G^t) so far, after which it computes a new position to examine. The CPSO formulas are given as follows [12].

$$\begin{aligned} V_i^t = {}& \omega\cdot V_i^{t-1} + c_1\cdot r_1\cdot(P_i^{t-1} - X_i^{t-1}) \\ & + c_2\cdot r_2\cdot(P_G^{t-1} - X_i^{t-1}) \end{aligned} \tag{7}$$

$$X_i^t = X_i^{t-1} + V_i^t \tag{8}$$

Where c_1 and c_2 are constants knows as the cognitive and social acceleration coefficients, respectively, ω is the inertia weight, r_1 and r_2 are random numbers uniformly distributed .between 0 and 1, and the position is clamped as $X_i \in [X_{\min}, X_{\max}]$.

The first part of (7) represents the previous velocity, which provides the necessary momentum for particles to fly across the search space. The second part is known as the "cognitive" component, which represents the personal thinking of each particle. This component encourages the particles to fly toward their own best positions found so far ($k = 2$). The third part is known as the "social" component, which represents the collaborative effect of the particles in finding the global optimum. This component always pulls the particles toward the best positions found by their neighbors so far [13].

B. The Genetic Particle Swarm Optimization

To solve combinatorial optimization problems, Yin [18] proposed a genetic particle swarm optimization with genetic reproduction mechanisms, namely crossover and mutation. Denote by D the dimension of the particle, the GPSO with genetic recombination for the d-th component of particle i is described as follows.

$$X_{i,d}^{t} = \omega(0,\omega_1).rand(X_{i,d}^{t-1}) + \omega(\omega_1,\omega_2).rand(P_{i,d}^{t-1}) + \omega(\omega_2,1).rand(P_{G,d}^{t-1}) \tag{9}$$

$$\omega(a,b) = \begin{cases} 1(if: a \le R_1 < b) \\ 0, otherwise \end{cases} \tag{10}$$

$$rand(y) = \begin{cases} \bar{y}, if: (R_2 < p_m) \\ y, otherwise \end{cases} \tag{11}$$

Where R_1 and R_2 are the random numbers uniformly distributed in [0, 1], thus with $0 < \omega_1 < \omega_2 < 1$, only one of the three terms on right hand side of (3) will remain, and $rand()$ mutates y with a small mutation probability p_m. And then ω_1 and ω_2 are used to select components from X_i^{t-1} , P_i^{t-1} and P_G^{t-1} . They work similarly to what c_1 and c_2 do in the CPSO.

There are four neighborhood topology types were proposed for PSO methods in [11] viz., circles (*lbest*), wheels, stars (*gbest*) and random edges. To avoid premature convergence, our approach employed the circle topology, in which each particle only communicates with its two immediate neighbors. Consequently, the particle holding the neighborhood best position can only influence its two neighbors in each iteration step, while its neighbors may transfer its information to the whole swarm step by step during the generation.

C. The Modified Genetic Particle Swarm Optimization

To apply GPSO to continuous function optimization problems, a modified genetic particle swarm optimization (MGPSO) was introduced in [20] which modified the crossover and the mutation operators with the differential evolution and the genetic algorithm mechanisms.

In GPSO, each component of the newborn is selected randomly from one of the three positions, which introduces poor diversities for continuous variables. Consequently, the heuristic crossover (HC) used in GA was employed [21], which is more specific to floating point representation. HC works as follows: let the parents be *X* and *Y* such that *Y* is not worse in fitness than *X*. The newborn is $Y + r(Y\text{-}X)$, where r is a random value selected uniformly in the interval [0, 1]. Since in GPSO, the newborn is made of X_i^{t-1} , P_i^{t-1} and P_G^{t-1} , the heuristic crossover has to be modified. Differential evolution (DE) is a simple population based stochastic parallel search evolutionary algorithm for global optimization [19], in which for an individual *i*, the weighted difference of two randomly chosen population vectors, X_{r2} and X_{r3} , is added to another randomly selected population member X_{r1} , to build a mutated vector as $M_i = X_{r1} + \mathrm{Pr}(X_{r2} \text{ - } X_{r3})$. Based on which, a modified heuristic crossover (MHC) in MATLAB style was given in Fig.1.

Where *rand* is a random value selected uniformly in the interval [0, 1]; D is the number of the dimension; and *Pr* is a user defined parameter. Consequently, each component is selected randomly from one of the three positions with the GPSO rules, and then a weighted difference of the other two positions' component is added to the selected component with DE rules. The sequence of the two components in the weighted difference is based on the HC rule. Consequently, MGPSO was derived from GPSO, incorporated with DE and HC rules for floating point representation. And in Fig.1, α , β , and γ is named as X_i , P_i , and P_G dominated crossover, respectively [20].

$for\ \ d = 1:D$
$\quad XR = rand;$
$\quad if\ \ XR < \omega_1$
$\qquad X_{i,d}^{t} = X_{i,d}^{t-1} + Pr(P_{G,d}^{t-1} - P_{i,d}^{t-1}); \quad (\alpha)$
$\quad elseif\ \ XR < \omega_2$
$\qquad X_{i,d}^{t} = P_{i,d}^{t-1} + Pr(P_{G,d}^{t-1} - X_{i,d}^{t-1}); \quad (\beta)$
$\quad else$
$\qquad X_{i,d}^{t} = P_{G,d}^{t-1} + Pr(P_{i,d}^{t-1} - X_{i,d}^{t-1}); \quad (\gamma)$
end

Figure 1. the modified heuristic crossover

As described in (11), the mutation operator of GPSO is based on the binary representation. To introduce diversity, the uniform mutation operator (UM) used in GA [21] was incorporated to MGPSO as follows [20].

$for\ \ d = 1:D$
$\quad if\ \ r_1 < m_r$
$\qquad x_{i,d}^{t} = rand[x_{\min,d}, x_{\max,d}]$
$\quad end$
end

Figure 2. the uniform mutation operator (UM)

Where r_1 is a random value selected uniformly in the interval [0, 1]; m_r is the mutation ratio; $rand[x_{\min,d}, x_{\max,d}]$ is used to generate a new value between the boundaries of the component. The uniform mutation is implemented after MHC.

In MGPSO, ω_1 and ω_2 are two key parameters used to set the probabilities to generate $X_{i,d}^t$ based on X_i, P_i, and P_G dominated crossovers for the balance of the local and the global search. Consequently, this paper will study the parameters at first. Three new parameters Px, Pp, and Pg are defined as $Px=\omega_1$, $Pp=\omega_2-\omega_1$, and $Pg=1-\omega_2$. And then, $0\le Px,Pp,Pg\le 1$, and $Px+Pp+Pg=1$. Consequently, Px, Pp, and Pg denotes the probability to generate a component with X_i, P_i, and P_G dominated crossover, respectively. And the three new parameters are relatively clearer to represent the probabilities of the dominated crossovers than ω_1 and ω_2.

D. The Modified GPSO with Inertia Weight

In the original GPSO, the concept of inertia weight is absent, while it is a particular mechanism of CPSO and provides necessary momentum and consistency. To employ the mechanism for MGPSO, a modified velocity was employed.

Initialization:

1. Randomly initialize the population X_i^0 *and calculate the fitness* $F_i^0=f(X_i^0)$

2. Set the velocity to zero $V_i^0=0$

3. Calculate P_i^0 *and* P_G^0 *based on* F_i

Repeat a predetermined number of maximum generations

1 Save a backup copy of each particle using $B_i^t=X_i^{t-1}$

2. Generate a new position for each particle with MHC using $X_i^t=MHC(P_i^{t-1},P_G^{t-1},X_i^{t-1})$

3. Implement the uniform mutation to each particle using $X_i^t=UM(X_i^t)$

4. Add the inertia to each particle $X_i^t=X_i^t+\omega g V_i^{t-1}$ *and calculate the fitness* $F_i^t=f(X_i^t)$

5. Calculate the velocity of each particle using $V_i^t=X_i^t-B_i^t$

6. Calculate P_i^t *and* P_G^t *based on* F_i^t

End repeat

Figure 3. Main algorithm of MGPSO-IW

In CPSO, based on the particle's previous velocity, the new velocity combines P_i^{t-1} and P_G^{t-1}, while in MGPSO, the combination is performed by MHC. Consequently, this paper will employ a new format velocity. Since the velocity is the displacement of a particle at each iteration step in CPSO, in MGPSO after the new position is generated, the velocity is defined as follows.

$$V_i^t=X_i^t-X_i^{t-1} \tag{12}$$

The weighted velocity is added to the position generated by MHC and the mutation operator as flows.

$$X_i^t=X_i^t+\omega\cdot V_i^{t-1} \tag{13}$$

Where ω is the inertia weight, it was derived from CPSO to maintain the inertia to provide necessary momentum and consistency. The flow chart of MGPSO with inertia weight (MGPSO-IW) is given as in Fig.3.

By comparisons with (7) and (8), three differences can be found. Firstly, in CPSO, V_i^t is the only momentum for a particle, while in MGPSO-IW, the momentum comes from MHC, the mutation operator and the previous velocity. Secondly, in CPSO, V_i^t is the combination of V_i^{t-1}, X_i^{t-1}, P_i^{t-1} and P_G^{t-1}, and added to the previous position, while in MGPSO-IW, V_i^t is simply set to the momentum of a particle in current iteration step and added to its position in the next iteration step. Finally, in CPSO the velocity is commonly limited between the lower boundary and the upper boundary, without which particles could essentially fly out of the physically meaningful solution space. While in MGPSO-IW, the velocity needs no boundary, because in MGPSO-IW, the velocity is consequentially clamped in the range.

To validate the feasibility and the effectiveness of the velocity strategy, MGPSO-IW with the inertia weight was implemented to the four benchmark functions given as follows [13].

f1: Sphere, the global optimum is 0 with $x^*=0$

$$f_1(x)=\sum_{i=1}^n x_i^2, -100\le x_i\le 100,$$

f2: Rosenbrock, the global optimum is 0 with $x^*=1$

$$f_2(x)=\sum_{i=1}^n[100(x_{i+1}-x_i^2)^2+(x_i-1)^2], -100\le x_i\le 100$$

f3: Rastrigin, the global optimum is 0 with $x^*=0$

$$f_3(x)=\sum_{i=1}^n(x_i^2-10\cos(2\pi x_i)+10), -10\le x_i\le 10,$$

f4: Griewank, the global optimum is 0 with $x^*=0$

$$f_4(x)=\sum_{i=1}^n x_i^2/4000-\prod_{i=1}^n\cos(x_i/\sqrt{i})+1, -600\le x_i\le 600$$

The first two functions are simple unimodal functions whereas the next two functions are multimodal functions designed with a considerable amount of local minima.

All the benchmarks are tested with dimensions 10, 20, and 30, and 50 trials are carried out with the same random initial populations with each dimension. Mutation rate m_r decreased from 1e-2 to 1e-4. For all the benchmarks, the stopping criteria are set to 0.01. The population size is 40. Table I lists the average of the optimal value (standard deviations) of MGPSO with (*Px, Pp, Pg*) = (0.2, 0.8, 0) as well as various inertia weight, where Dim is the dimension, Gen is the maximum generation, Fun denotes the function. Table II lists the number of trails that converges to the stopping criteria and average number of generations for convergence for the 50 trials.

It can be seen in Table I that with (*Px*, *Pp*, *Pg*) = (0.2, 0.8, 0), in general, the results with proper inertia weight are relative better. With ω is 0.35, 0.45 and 0.55, MGPSO-IW has consistently converged to the stopping criteria for f1, f2 and f3 under all the three dimensions, while for f4, with inertia weight between 0.00 and 0.65, MGPSO has provided comparative results, and the results are deteriorated with higher inertia weight.

TABLE I.
AVERAGE OF THE OPTIMAL VALUE(STANDARD DEVIATION) FOR 50 TRIALS FOR MGPSO-IW

Fun	Gen(Dim)	inertia weight ω								
		0.00	0.05	0.15	0.25	0.35	0.45	0.55	0.65	0.75
f1	1000 (10)	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	2000 (20)	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	3000 (30)	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
f2	3000 (10)	0.028 (0.037)	0.035 (0.085)	0.057 (0.099)	0.015 (0.018)	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	4000 (20)	0.025 (0.041)	0.059 (0.140)	0.039 (0.048)	0.014 (0.023)	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	5000 (30)	0.033 (0.050)	0.021 (0.031)	0.029 (0.040)	0.034 (0.102)	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
f3	3000 (10)	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	0.304 (0.483)
	4000 (20)	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	0.300 (0.447)
	5000 (30)	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	0.010 (0.0004)	0.159 (0.344)
f4	3000 (10)	0.024 (0.012)	0.023 (0.011)	0.025 (0.015)	0.027 (0.014)	0.026 (0.013)	0.024 (0.011)	0.023 (0.011)	0.028 (0.012)	0.065 (0.024)
	4000 (20)	0.024 (0.012)	0.024 (0.014)	0.028 (0.014)	0.028 (0.015)	0.021 (0.011)	0.021 (0.010)	0.027 (0.012)	0.034 (0.015)	0.066 (0.027)
	5000 (30)	0.023 (0.014)	0.025 (0.015)	0.030 (0.016)	0.024 (0.012)	0.022 (0.011)	0.021 (0.011)	0.025 (0.011)	0.033 (0.012)	0.064 (0.025)

TABLE II.
NUMBER OF THAT CONVERGED TO THE STOPPING CRITERIA AND AVERAGE NUMBER OF GENERATIONS FOR CONVERGENCE FOR 50 TRIALS

Fun	Gen(Dim)	ω=0	ω=0.05	ω=0.15	ω=0.25	ω=0.35	ω=0.45	ω=0.55	ω=0.65	ω=0.75
f1	3000 (10)	50 (354.9)	50 (303.6)	50 (270.5)	50 (232.7)	50 (255.2)	50 (304.7)	50 (427.4)	50 (573.4)	50 (848.0)
	2000 (20)	50 (358.8)	50 (302.5)	50 (270.2)	50 (238.6)	50 (249.2)	50 (290.5)	50 (418.4)	50 (563.6)	50 (858.7)
	3000 (30)	50 (358.8)	50 (300.0)	50 (271.9)	50 (242.7)	50 (249.4)	50 (303.8)	50 (419.4)	50 (554.0)	50 (850.4)
f2	3000 (10)	38 (627.0)	38 (450.6)	26 (451.4)	44 (517.9)	50 (556.1)	50 (582.9)	50 (827.9)	50 (1336.9)	50 (2187.3)
	4000 (20)	34 (529.4)	31 (482.5)	28 (732.7)	47 (556.7)	50 (502.3)	50 (627.7)	50 (960.6)	50 (1283.0)	50 (2224.4)
	5000 (30)	35 (521.8)	38 (448.0)	33 (472.9)	42 (375.0)	50 (491.8)	50 (694.2)	50 (916.9)	50 (1248.3)	50 (2180.8)
f3	3000 (10)	50 (1070.3)	50 (734.4)	50 (968.2)	50 (935.8)	50 (846.3)	50 (1104.6)	50 (1685.0)	50 (1930.1)	27 (2686.9)
	4000 (20)	50 (1099.0)	50 (754.3)	50 (1043.5)	50 (938.7)	50 (856.0)	50 (1071.5)	50 (1558.0)	50 (1938.9)	25 (2662.1)
	5000 (30)	50 (1075.0)	50 (773.1)	50 (956.7)	50 (953.8)	50 (811.9)	50 (1047.1)	50 (1691.7)	49 (1972.1)	33 (2685.6)
f4	3000 (10)	10 (1482.5)	7 (986.6)	13 (991.4)	10 (885.2)	6 (816.0)	6 (1421.5)	8 (1864.8)	0 (n/a)	1 (2743.0)
	4000 (20)	7 (995.7)	9 (1311.2)	4 (726.0)	12 (1184.8)	12 (1071.3)	10 (1498.6)	6 (2297.7)	3 (2580.0)	0 (n/a)
	5000 (30)	14 (1513.4)	13 (1093.8)	6 (723.7)	12 (983.9)	11 (1239.8)	11 (1197.3)	5 (2293.0)	2 (2531.0)	0 (n/a)

From Table II, it is clear that with proper inertia weight the convergence speeds of MGPSO-IW are faster than those of MGPSO for f1 (with ω from 0.05 to 0.45), f2 (with ω =0.35) and f3 (with ω from 0.05 to 0.35), and for f2 the convergence rate is higher. While for f4, inertia weight provides poor improvement. Moreover, with ω more than 0.55, the convergence speed decreases for all the functions under the three dimensions, markedly.

Consequently, in general, the inertia weight is effective for MGPSO-IW, and with ω between 0.35 and 0.45, the performance is relatively consistent for the four functions. While the typical value of the inertia weight in classical PSO is bigger, in the fixed inertia weight strategies it is typically set to 0.9, and in the time-varying strategy, it decreases from 0.9 to 0.4 during the generation [13]. One of the reasons is that, in CPSO the velocity is the only momentum of a particle containing the previous “social” and “cognitive” information, consequently, the higher inertia weight means more momentum and more useful information. While in MGPSO-IW, a particle mainly moves with the modified heuristic crossover with the inertia weight as a secondary part in the momentum. Consequently, in MGPSO-IW the inertia weight should be less, or the inertia will disturb the consistency and continuity.

MGPSO-IW ($\omega = 0.45$) are compared with the PSO methods discussed in [13] with the same population, maximal generation and stopping criteria. The averages of the optimal values (standard deviations) are listed in Table III, where PSO-TVIW, PSO-RANDIW, PSO-TVAC, MPSO-TVAC, MPSO-FAC, HPSO-TVAC, and HPSO-FAC denote that PSO with time-varying inertia weigh, with rand inertia weigh, with time-varying acceleration coefficients, with both mutation and time-varying acceleration coefficients, with both mutation and fixed acceleration coefficients, with both self-organizing hierarchical strategy and time-varying acceleration coefficients, and with both self-organizing hierarchical strategy and fixed acceleration coefficients, respectively.

TABLE III.
AVERAGE OF THE OPTIMAL VALUE(STANDARD DEVIATION) FOR 50 TRIALS FOR MGPSO-IW WITH $\omega = 0.45$ AND THE OTHER PSO METHODS

Fun	Gen(Dim)	MGPSO-IW	PSO-TVIW	PSO-RANDIW	PSO-TVAC	MPSO-TVAC	MPSO-FAC	HPSO-TVAC	HPSO-FAC
f1	3000 (10)	**0.01**	**0.0 1**	**0.01**	**0.0 1**	**0.0 1**	**0.0 1**	**0.0 1**	**0.0 1**
	2000 (20)	**0.01**	**0.0 1**	**0.01**	**0.0 1**	**0.0 1**	**0.0 1**	**0.0 1**	**0.0 1**
	3000 (30)	**0.01**	**0.0 1**	**0.01**	**0.0 1**	**0.0 1**	**0.0 1**	**0.0 1**	0.2 30 (0. 173)
f2	3000 (10)	**0.01**	27.11 (58.312)	2.102 (3.218)	0.946 (32.127)	4.247 (7.961)	11.12 (14.243)	12.967 (11.538)	12.963 (14.397)
	4000 (20)	**0.01**	51.56 (119.79)	28.1788 (73.072)	17.944 (46.296)	17.7148 (60.306)	54.402 (92.88)	14.093 (9,641)	101.126 (129.56)
	5000 (30)	**0.01**	63.35 (71.210)	35.277 (55.751)	28.97 (51.638)	18.633 (25.122)	135.08 (306.07)	13.666 (11.006)	706.28 (951.95)
f3	3000 (10)	**0.01**	2.069 (1.152)	4.63 (2.366)	2.268 (1.333)	0.01 (0.0033)	19.54 (36.577)	**0.01**	0.0671 (0.237)
	4000 (20)	**0.01**	11.74 (3.673)	26.293 (8.176)	15.323 (5.585)	0.3415 (0.588)	2.786 (1.808)	**0.01**	11.391 (6.489)
	5000 (30)	**0.01**	29.35 (6.578)	69.7266 (20.700)	36.236 (8.133)	2.050 (1.910)	12.477 (5.990)	0.044 (0.196)	36.847 (10.626)
f4	3000 (10)	**0.024 (0.011)**	0.0675 (0.029)	0.0661 (0.030)	0.05454 (0.025)	0.0469 (0.0256)	0.065 (0.2373)	0.057 (0.026)	0.057 (0.023)
	4000 (20)	0.021 (0.010)	0.0288 (0.023)	0.0272 (0.025)	0.0293 (0.027)	0.0239 (0.017)	0.027 (0.025)	**0.011 (0.005)**	0.055 (0.085)
	5000 (30)	0.021 (0.011)	0.0162 (0.013)	0.0175 (0.018)	0.0191 (0.015)	0.0169 (0.0149)	0.018 (0.051)	**0.01 (0.0035)**	0.116 (0.193)

It can be seen from Table III that MGPSO-IW has consistently found the optimal for f1, f2, and f3 with all the dimensions, which exceeds all the other PSO methods. For f4, MGPSO-IW provides the best result with 10 dimensions; with 20 dimensions, MGPSO-IW is inferior to HPSO-TVAC but exceeds the other PSO methods; with 30 dimensions, MGPSO-IW is inferior to all the other methods but the results are near the optimum. Consequently, the comparisons have shown the feasibility and effectiveness of MGPSO-IW which can provide high quality solution for both simple unimodal functions and multimodal functions.

IV THE PSO-SVM HYBRID SYSTEM

A. The Fitness Definition

To implement our proposed approach, this research used the RBF kernel function as defined by (6) for the SVM classifier because the RBF kernel function can analyze higher dimensional data and requires that only two parameters, C and γ be defined [8][22][23], . When the RBF kernel is selected, the parameters (C and γ) used as parameters must be optimized using the proposed PSO–SVM system.

Classification accuracy is the criteria used to design a fitness function. Thus, for the particle with high classification accuracy produces a high fitness value. The particle with high fitness value has high probability to effect the other particles' positions of the next iteration, so it should be appropriately defined.

For the fitness definition, the classification accuracy (acc) or hit rate denoting the percentage of correctly classified examples is evaluated by (14) in [8]. The numbers of correctly and incorrectly classified examples are indicated by *cc* and *uc*, respectively.

$$F = cc/(cc+uc) \qquad (14)$$

In the road icing the fitness definition has to be modified, because the road icing phenomenon only occurs in few days in a road icing season. For example, in the examples of Xianning from 1961 to 2006, there are 6592 days in road icing reasons, and there are only 597 road icing days (9.06%). And for Wuhan and Shiyan, the percents are 22.56% (from 1961 to 2006) and 26.28% (from 1963 to 2006), respectively. Consequently, an arbitrary prediction "there are not any road icing days in the road icing seasons" have the prediction accuracies 90.94%, 77.44% and 73.72% for Xianning, Wuhan and Shiyan, respectively. It can be seen that the predictions have high accuracies, but they provide no useful information for the reduction of the threat of the ice-related accidents.

With the days with road icing denoted by 1 (positive), and the days without road icing denoted by 0 (negative), this paper employs another fitness definition given as follows.

$$F = N\_R / (N\_R + N\_M + N\_W) \qquad (15)$$

where N_R is the number of the positive examples classified correctly, which encourages the solution that can predict more road icing days correctly. N_M is the number of positive examples classified incorrectly to negative, which punishes the prediction of missing road icing days. And N_W is the number of negative examples classified incorrectly to positive, which is used to punish excessive positive prediction.

By comparisons with (14), the employed fitness definition focuses more on the prediction of positive examples. With (15) the prediction accuracies of the above arbitrary prediction are zero for all the cities, and for a perfect prediction, the fitness will be 100% which is more reasonable.

B. The Frame of the Proposed PSO-SVM system

With the above fitness definition, MGPSO-IW is employed to train the parameters C and γ for SVM. Before the experimental procedure the historical data for training are split into two groups by cross-validation where N*P (0<P<1) examples are randomly selected from N examples for the training group, and the left N-NP examples are used for the validation group. The training group is used to build the SVM model. The validation group is used to determine the proper training iteration avoid overtraining. Moreover, a test group is employed to evaluate the model's classification accuracy which is independent to the training group and the validation group. The detailed experimental procedure for the training and testing is given as follows [8].

Step 1. *Date preparation*: Training, validation, and test groups are represented as Tr, Va, and Te, respectively.

Step2. *Particle initialization and PSO parameter setting*: Generate initial particles comprised of C and γ, Set the PSO parameters including number of iteration, velocity boundary, population. Set iteration = 0, and perform the training process from step 3-7.

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

Step 4. *SVM model training*:

(a) Build a SVM model based on the training group and each particle (C and γ).

(b) Evaluate the classification accuracy on the validation group Va using the model above with (15) to generate the fitness of the particle.

Step 5. Update the local he the personal best according to the fitness evaluation results.

Step6. Update the position of each particle using the operations in Fig.3.

Step7. If stopping criteria (maximum iterations predefined) are met, go to the next step, otherwise go to step 3.

Step 8. With the stopping training iteration determined in the previous step, based the best particle found so far build the SVM model. And then measure the testing accuracy on the Te.

Step 9. End the training and testing procedure.

V. Experimental Results

The PSO-SVM system is performed to predict road icing in the three cities viz., Shiyan, Wuhan, and Xianning. For each city, three training data sets are employed viz., 1980 to 2006, 1994 to 2006, and 2000 to 2006, and P = 30%. The data from 2007 to 2008 of each city are employed as the test group for each city, respectively. The population is set to 30, the searching ranges for C and γ are as follows: $C\in$[100, 280], $\gamma\in$ [0.001, 5], the maximum iteration is 10. The other parameters are the same with those used for Table I. For each city with each training group, 10 runs are performed, and the best results are given in Table IV, where Period is the period of the training data sets, Positive cases is the number of positive cases (road icing days) in the test groups, FIT is the fitness of the model for test groups generated by (15).

TABLE IV.
The Simulation Results of the PSO-SVM System

City	Period	Positive Cases	FIT	N_R	N_M	N_W
Shiyan	1980-2006	69	89.33%	67	2	6
	1994-2006		84.21%	64	5	7
	2000-2006		83.10%	59	10	2
Wuhan	1980-2006	49	80.36%	45	4	7
	1994-2006		81.36%	48	1	10
	2000-2006		76.67%	46	3	11
Xianning	1980-2006	48	92.00%	46	2	2
	1994-2006		90.00%	45	3	2
	2000-2006		93.88%	46	2	1

It can be seen that PSO-SVM system performed well for all the cities, with which almost all the road icing days have been predicted successfully and the fitness of the runs are more than 80% except Wuhan in 2000-2006. PSO-SVM provided the best predictions for Xianning, with which with all the periods the fitness are more than 90%. And it can be seen that the system has predicted almost all the road icing days, and only predicted few non-road icing to road icing days in error. For Wuhan, the results are the worst where although it predicted almost all the road icing days, it predicted many non-road icing to road icing days, wrongly. For Shiyan, with the period 2000 to 2006, the system has missed 10 road icing days which need be improved.

It is obviously that the period 1980 to 2006 there are more examples than those in the other periods, while for Wuhan and Xianning, it failed to provide the best results. One of the reasons is that the configuration of the cities are keeping on changing, and the total length, distributing, and technologies, etc have changed. Consequently, the examples long time ago may provide less useful information than those in recent years.

VI. CONCLUSION

The PSO-SVM predication system is employed for automatic prediction of road icing conditions in three cities of Hubei Province, viz., Wuhan, Shiyan and Xianning. To improve the classification accuracy for road icing prediction, a modified particle swarm optimization is employed to simultaneously optimize the SVM kernel function parameter and the penalty parameter. With the data from 1980 to 2006, using the proposed approach, the road icing models for the three cities in Hubei province are created, which have been used for the prediction from 2007 to 2008. The results have shown the forecasting ability and reference values of the approach.

ACKNOWLEDGMENT

The author will thank the support of the project of outstanding young and middle-aged scientific and technological innovation team of Colleges and universities in Hubei Provincial No.T200904, and the Key project of Hubei University of Education.

REFERENCES

[1] Jonas. Norrman, "Slipperiness on roads-an expert system classification," *Meteorological Applications*, vol. 7, pp. 27-36, Dec 2006.

[2] Alexander. Mahura, Claus. Petersen, and Bent. Hansen. Sass, "Road Icing Conditions in Denmark," *Ministry of climate and energy, Copenhagen*, Scientific Report 08-03, pp.1-25, 2008.

[3] Konstantin. Korotenko, "An Automated System for Prediction of Icing on the Road," *Lecture Notes In Computer Science*, vol. 2331, pp. 1193-1200, 2002.

[4] William P. Mahoney and William L. Myers, "Predicting Weather and Road Conditions: An Integrated Decision Support Tool for Winter Road Maintenance Operations," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1824, pp. 98-105, Jan 2003.

[5] Liu Mei, Yin Dongping, Wang Qinglou, and Gao Ping, "Weather Standard and Forecast Study on Winter Icing on the Road in Nanjing," *Scientia Meteorologica Sinica*, vol. 27, pp.685-690, 2007.

[6] Xu Xiu-hong, Jiao ling-ling, Wang Qing-yu, and Hu xiao-jing, "Study on Relationship between the Developmental Tendency of Ice Roads and Its Temperature during Recent Ten Years in Heilongjiang Province," *Journal of Anhui Agricultural Sciences*, vol. 37(5), pp.2123-2125 , 2009.

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

[8] Cheng-Lung Huang, and Jian-Fan Dun, "A Distributed PSO-SVM Hybrid System with Feature Selection and Paramter Optimization," *Applied Soft Computing*, vol. 8, pp.1381-1391, 2008.

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

[10] C. W. Jiang, and C. J. Lin, "A Simple decomposition method for support vector machine," *Machine Learning*, vol.46(1-3), pp.219-314, 2002.

[11] J. Kennedy, and R.C. Eberhart, "Particle swarm optimization," in: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp.1942-1948, 1995.

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

[13] A. Ratnaweera, S. Halgamuge, "Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients," *IEEE Transaction on Evolutionary Computation*, vol. 8, pp.240-255, 2004.

[14] Andrews P. S, "An Investigation into Mutation Operators for Particle Swarm Optimization," In: Proceedings of CEC2006, Piscataway, NJ: IEEE Press,pp. 1044-1051, 2006.

[15] Jiang Chuanwen and Etorre Bompard, "A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimisation," *Mathematics and Computers in Simulation*, vol.68(1), pp.57-65, 2005.

[16] Luitel. B and Venayagamoorthy. G.K. "Differential evolution particle swarm optimization for digital filter design," *IEEE congress on Evolutionary Computation*, 2008, pp. 3954-3961.

[17] Shu-Kai S. Fan, Yun-Chia Liang, and Erwie Zahara. "A genetic algorithm and a particle swarm optimizer hybridized with Nelder-Mead simplex search," *Computers & Industrial Engineering*, vol.50(4), pp.401-425, 2006.

[18] P. Y. Yin, "Genetic particle swarm optimization for polygonal approximation of digital curves," *Pattern Recognition and Image Analysis*, vol. 16(2), pp.223-233, 2006.

[19] K. V. "An introduction to differential evolution," *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. McGraw-Hill, London, pp. 79–108, 1999.

[20] Peng. Chen, Chunhua. Zhao, Jian Li, and Zhiming. Liu, "Solving the Economic Dispatch in Power System via a Modified Genetic Particle Swarm Optimization," *Proc. 2009 International Joint Conference on Computational Sciences and optimization*, pp. 201-204, IEEE Press, 2009.

[21] Khaled Mohamed Rasheed, *GADO: AGenetic algorithm for Continuous Design Optimization*, Ph.D. dissertation. The State University of New Jersey; 1998.

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

[23] H.T. Lin, C.J. Lin, *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, 2003.

LI. Jian earned a Ph.D. in System Analysis and Integration from Huazhong University of Science and Technology, China, in 2008. His is a lecture in the Department of Computer Engineering, the Hubei University of Education at Wuhan, China. His research interests including evolutionary algorithm, automatic and pattern recognition.