Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, 2006 (pp97-103)

An IPSO-Based Integrated Neural Classifier for Steam Turbine Vibration Fault Diagnosis

H. S. SU, W. J. DANG
School of Information and Electrical Engineering
Lanzhou Jiaotong University
Lanzhou Gansu 730070
P.R. CHINA

Abstract: - To compensate the flaws of single neural network such as low classification precision and long training time as well as weak generalized ability etc, the paper proposes a novel ensemble neural classifier for stream turbine vibration fault diagnosis with improved PSO-based. The method fully utilizes the advantages of PSO, such as fast seeking speed and easily realizing mode etc, the integrated time of the whole network therefore becomes very short. Meanwhile, more neural networks are used to implement fault diagnosis concurrently, and their results are integrated with improved PSO-based. The studies indicate that the proposed method has higher precision of the classification and seekink speed, and is an ideal pattern classifier. In the end, a simulation experiment in stream turbine vibration fault diagnosis shows the method is extremely effective.

Key-Words: -PSO; Ensemble neural classifier; Entropy; Steam turbine; Vibration; Fault diagnosis

1 Introduction

In recent years, with the fast advancement of artificial neural networks(ANN), ANN has been in an increasingly broad application in diverse fields such as artificial intelligence(AI), patter recognition(PR), expert systems(ES), image processing and process control as well as many other related fields. Due to its prominent characteristics such as self-learning and non-linearity and parallel distributed information storage, ANN has already been considered very suitable to facility fault diagnosis[1], and has been successful applications in power systems[2],[3], mechanical engineering[4],[5], chemical engineering [6],[7] and control systems[8],[9], etc. Clearly, to achieve parallel algorithm and nonlinear mapping ability from input space to output space are the most excellent properties of ANN.

However, In process of applications, single ANN still exposes many problems such as incredible results, inscrutable behavior and weak generalized ability, etc, which restricts its farther applications. Hence, to overcome the flaws and achieve the optimal results, to train more neural networks and integrate their results are considered a realistic selection[10]. It can dramatically improve the generalized ability of neural networks and the correctness of classification results. Moreover, the difficulties in choosing the model of neural network are also better tackled. Based on it, the paper proposes an integrated neural network classifier for stream turbine vibtartion fault diagnosis based on improved particle swarm optimization (IPSO). The method fully applies the prominent excellences of PSO such as fast seeking speed and easy realization mode etc, the integrated time of the whole network therefore become very short. The method assigns diverse weights to the posterior probabilistic estimate of each network according to their respective properties, the identification rate for that is improved by weighted average means. Meanwhile, the training speed of the networks is also improved, further.

The following sections would include basic PSO algorithm and IPSO; the model of compositive neural network as well as algorithms; practical example and analysis; conclusions, etc.

2 PSO Algorithm

PSO is an evolutionary computation technique developed by Kennedy and Eberhart in 1953 inspired by social behavior of bird flocking and fishing schooling[11]. In the past decades, PSO algorithm has already achieved enormous progress. Many improved PSO algorithms derive from it, such as inertial weights method[12], constriction factors method [13], breeding and subpopulations method studied in [14], neighboring operator method in [15], adaptive PSO algorithm [16], adaptive mutation PSO [17], Discrete PSO method [18], concurrent PSO method [19], Hybrid PSO [20] and immune PSO algorithm [21] as well as gradient acceleration PSO method [22],etc. Presently, PSO has already been an important evolution tool, which has a broad application in diverse fields including neural network training, function optimization and fuzzy system control, etc.

2.1 Basic PSO Algorithm

PSO is a global random optimization algorithm, whose thinking originates from intelligent behavior of the colony. Specifically speaking, it mainly emulates the characteristics of the migration and gathering of the birds during seeking the food. The algorithm generates swarm intelligence to optimize the seeking aim by the cooperation and competition among the particles. PSO algorithm not only retains global scout strategy with swarm-based, the applied operation model called as displacement-speed is comparatively simple and programming is easily realized, but also it holds the unique optimization properties such as fast operation speed and relatively simple structure. PSO is a high efficient parallel seeking algorithm, and has prominent behavior in tackling non-linear optimization problems. A basic PSO algorithm is presented below[11].

Set a swarm comprising n particles in D-dimensional space, the i^{th} particle may be expressed as a D-dimensional vector $\boldsymbol{x}_i=(x_{i1},x_{i2},\ldots,x_{iD})$, $i=1,2,\ldots,n$, namely, the position of the i^{th} particle in D-dimensional space is $\boldsymbol{x}_i$, and each such position is named as a potential solution. The adaptability function value of $\boldsymbol{x}_i$ is calculated by substituting it into the aim function $f(\boldsymbol{x}_i)$, then, according to the value size, $\boldsymbol{x}_i$ can be weighed to be the good or the bad. The flight speed of the i^{th} particle is also a D-dimension vector, and written as $\boldsymbol{v}_i=(v_{i1},v_{i2},\ldots,v_{iD})$. Set until now, the optimal position sought by the i^{th} particle is $\boldsymbol{p}_i=(p_{i1},p_{i2},\ldots,p_{iD})$, and the optimal position sought by the overall particle swarm is $\boldsymbol{p}_g=(p_{g1},p_{g2},\ldots,p_{gD})$. Then the position and speed of the particle i can be evolved according to the following equation (**1**) and (**2**) .

$$v_{id}^{k+1} = \chi[w \times v_{id}^{k} + c_1 r_1 (p_{id}^{k} - x_{id}^{k}) + c_2 r_2 (p_{gd}^{k} - x_{gd}^{k})] \quad \textbf{(1)}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad \textbf{(2)}$$

In the above equation, χ expresses constriction factor, $\chi= 2/|2-\varphi-(\varphi^2-4\varphi)^{1/2}|$, $\varphi=c_1+c_2$, and $\varphi>4$. Also, w is inertia weight, which indicates an influence yielded by the present speed of the particle on its next generation. A suitable w can make the particle hold balanced exploration and exploitation abilities. To avoid the particle away from seeking space, every dimensional speed of the particle should be constricted in range of $-v_{max,d}$ to $+v_{max,d}$. If $v_{max,d}$ is too large, the particle will fly out the optimal solution, conversely, and it would be easy to fall into a local optimization. The parameters c_1 and c_2 are non-negative learning factors, the values of which usually are limited in range of 1 to 2, if the values are too small, the particle would be far away from the aim area, inversely, if too large, the particle can suddenly or possibly fly over aim area. $r_1$and r_2 are random variables with a scope of 0 to 1.

However, a PSO-based algorithm has many flaws, for instance, easily getting into local minimum, slow convergence speed and low diagnosis precision, etc. All these deficiencies must be improved.

2.2 Improved PSO Algorithm

To compensate for the above flaws of PSO described in former sections, diverse improved PSO algorithms have already been reported recently in [12-22]. Therefore, if we can effectively apply these algorithms, then the identification rate of the PSO algorithm must be well improved. Below we make some improvements from three facets.

1)Consideration of information entropy in PSO algorithm. Information entropy is introduced into PSO to aim to accelerate or slower the convergence speed of the particles. The evolutionary goal contained in the particles is enhanced by applying the information entropy to control seeking process, and the optimal solutions therefore can also be rapidly found [23].

Assume that $f(x_{ji})$ is the adaptable functional value of the i^{th} particle at the j^{th} generation, then, the ratio between the adaptive value of the i^{th} particle at the j^{th} generation and the sum of the adaptive values of the overall particle swarm at the j^{th} generation is expressed by

$$p(x_{ji}) = \frac{f(x_{ji})}{\sum_{i=1}^{n} f(x_{ji})}, \quad i=1,2,\ldots,n. \quad \textbf{(3)}$$

In (**3**), n is the sum of the particles. Then information entropy is constructed by

$$H(x_j) = \sum_{i=1}^{n} p(x_{ji}) \log_n p(x_{ji}) \quad \textbf{(4)}$$

$$\text{st.} \ \sum_{i=1}^{n} p(x_{ji}) = 1, \quad p(x_{ji}) \in (0,1). \quad \textbf{(5)}$$

In (**4**), $H(x_j)$ expresses information entropy value of the overall particles. Based on $H(x_j)$, the optimal level may be weighed at present generation. According to the principle of maximum entropy (POME) proposed by Jaynes in [24], the adaptive values of all particles should infinitely approach to $1/n$, at the moment, $H(x_j)$ attains the maximum, and the particles become gradually stable. If $H(x_j)$ approaches to zero, the adaptive values of the particles at the j^{th} generation are more decentralized and uneven, the algorithm therefore is more unstable, and conversely, if $H(x_j)$ approaches to one, then the distribution of the adaptive values of the particles at present generation are more even, and the algorithm is more stable.

2) Improvement of inertia weight w

We can know that w can be described by

$$w = w_{max} - \frac{w_{max} - w_{min}}{G} \times g . \quad \textbf{(6)}$$

In the above equation, G expresses the gross iterative time, g represents the current iterative time. According to (**6**), the weight factor w has a linear degression from w_{max} to w_{min} with the evolution of iterative time.

According to the description in [25], the seeking ability of PSO is determined by w and v_{max}. On one hand, if the values are larger, it then is advantageous to global scout, and the convergence speed is enough fast, however, it isn't easy to find more precise solutions. On another hand, if the value is smaller, it would be is advantageous to local scout, and more precise solutions can be found, but the convergence speed is relatively slower. We know that equation (**6**) is a linear function whose slope is $(w_{\max}-w_{\min})/G$. Thus, if we apply diverse $w_{\max}$ and $w_{\min}$ according to the convergence conditions of the particles, then the convergence speed of the particles can be quicken up or slowed down freely. At the initial time of the algorithm, $p(x_{ji})$ of the particles have larger diversities, and $H(x_j)$ approaches to 0, at the moment, the convergence speeds of the particles should accelerate, that is, the slope of w should be added. At the middle time, the $p(x_{ji})$ of the particles approach one another, $H(x_j)$ approaches to one, the slope of w should be decreased to slow down the convergence speeds to get into precise scout stage. As $H(x_j)$ approaches to one and the algorithm almost stagnates, the slope of w is added again and a part of the particles are initialized so as to depart from local optimal solutions. This is described by

if $0<H(x_j)<k1$, then $w_{max}=1.8$ and $w_{min}=0.1$;
if $k1<H(x_j)<k2$, then $w_{max}=0.9$ and $w_{min}=0.2$; (7)
if $H(x_j)>k2$, then $w_{max}=1.4$ and $w_{min}=0.4$.

3) To let the particles not get into local optimization, as optimal information of the swarm stagnates, the part of the particles are initialized, and the speeds and locations of the particles are updated again to keep their activation. Hence, some measures should be taken to avoid the situation, for instance, if one particle is almost immobile in sequential several steps, the speed of the particle then is updated, or if its flying speed exceeds the limitation and its adaptive value hasn't gain improvement in sequential several steps, the flying speed of the particle is initialized once again [26]. Based on the improved PSO algorithm, below we will present its application in integrated neural networks in detail.

3 Ensemble Neural Classifier

IPSO is applied to optimize neural networks, whose main aim is to optimize the weight values of neural networks to reduce the seeking time. Fig.1 shows the principle diagram of ensemble neural networks based on improved PSO.

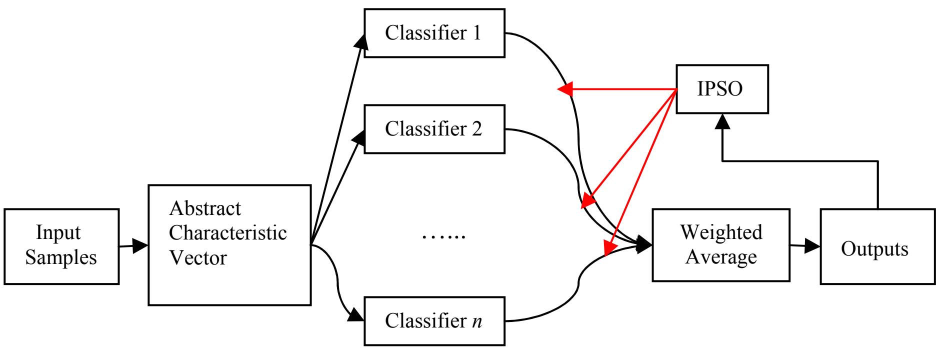

Fig.1 The principle diagram of neural networks ensemble with IPSO-based

Set classification objects have c patterns, which is expressed as $\omega=(\omega_1, \omega_2,\ldots, \omega_c)$. Assume that M neural classifiers are used to integrate diagnosis results with IPSO-based as shown in Fig.1. Set input samples vector is $\boldsymbol{X}$, the true posterior probability distribution is $q(\boldsymbol{X})=[p$

$(\omega_1|\boldsymbol{X}), p(\omega_2|\boldsymbol{X}),\ldots, p(\omega_c|\boldsymbol{X})]$. The estimation of the posterior probability of the i^{th} neural classifier may be described by

$$q_i(\boldsymbol{X})=[\hat{p}_i(\omega_1|\boldsymbol{X}),\hat{p}_i(\omega_2|\boldsymbol{X}),\ldots,\hat{p}_i(\omega_c|\boldsymbol{X})] \quad \textbf{(8)}$$

Due to insufficient samples and incomplete training and so on, the error $\varepsilon_{ij}(\boldsymbol{X})$ between the true posterior probability $p_i(\omega_j|\boldsymbol{X})$ and the estimated posterior probability $\hat{p}_i(\omega_j|X)$ is generated, and expressed as $\varepsilon_{ij}(\boldsymbol{X})= p_i(\omega_j|\boldsymbol{X})- \hat{p}_i(\omega_j|X)$, the error rate is for that higher than one of Bayesian classer. Hence, the correctness of the posterior probability must be improved.

To resolve this problem, we use the weighted average sum of each posterior probability to reduce the classification error. Considering the differences among diverse network classifiers and their relativities, diverse weight values are endowed to diverse networks. Set the weight value of each classifier is w_i, i=1,2,…,M, we then have

$$\hat{q}(\boldsymbol{X})=\sum_{i=1}^{M} w_i q_i(\boldsymbol{X}) \quad \textbf{(9)}$$

The principle to select the weights to let error $e=\frac{1}{2}\sum \|q(X)-\hat{q}(X)\|_2$ lower than the minimum ε, this means

$$\frac{dE\{|q(\boldsymbol{X})-\hat{q}(\boldsymbol{X})|^2\}}{dw_i}=0, i=1,2,\ldots,M. \quad \textbf{(10)}$$

where

$$E\{|q(\boldsymbol{X})-\hat{q}(\boldsymbol{X})|^2\}= =\sum_{j-1}^{c} E\{(p(\omega_j|\boldsymbol{X})-\sum_{i=1}^{M} w_i\,\hat{p}_i(\omega_j|\boldsymbol{X}))^2\}. \quad \textbf{(11)}$$

and so,

$$\sum_{i=1}^{M} w_i \sum_{j=1}^{c} E\{(\hat{p}_i(\omega_j|\boldsymbol{X})\hat{p}_k(\omega_j|\boldsymbol{X})\}=\sum_{j=1}^{c} E\{p(\omega_j|\boldsymbol{X})\hat{p}_k(\omega_j|\boldsymbol{X})\} \quad k=1,2,\ldots,M. \quad \textbf{(12)}$$

order

$$i_{ik}=\sum_{j=1}^{c} E\{\hat{p}_i(\omega_j|\boldsymbol{X})\hat{p}_k(\omega_j|\boldsymbol{X});$$
$$o_k=\sum_{j=1}^{c} E\{p(\omega_j|\boldsymbol{X})\hat{p}_k(\omega_j|\boldsymbol{X})\}. \quad \textbf{(13)}$$

$$W=\begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_M \end{bmatrix}, I=\begin{bmatrix} i_{11} & i_{12} & \cdots & i_{1M} \\ i_{21} & i_{22} & \cdots & i_{2M} \\ \cdots & \cdots & \cdots & \cdots \\ i_{M1} & i_{M2} & \cdots & i_{MM} \end{bmatrix}, O=\begin{bmatrix} o_1 \\ o_2 \\ \cdots \\ o_M \end{bmatrix}.$$

Then the formula (**12**) may be written as

$$IW=O \quad \textbf{(14)}$$

Hence, we have

$$W=I^{-1}O \quad \textbf{(15)}$$

and

$$E\{|q(\boldsymbol{X})-\hat{q}(\boldsymbol{X})|^2\}=\sum_{j-1}^{c} E\{(p^2(\omega_j|\boldsymbol{X})\}-O^T I^{-1}O. \quad \textbf{(16)}$$

In practice, I and O may be estimated by statistic data. Set N training samples are $x_1, x_2,\ldots, x_N$, then

$$\hat{i}_{ik}=\frac{1}{CN}\sum_{n=1}^{N}\sum_{j=1}^{c} \hat{p}_i(\omega_j|x_n)\,\hat{p}_k(\omega_j|x_n) \quad \textbf{(17)}$$

$$\hat{o}_k=\frac{1}{CN}\sum_{n=1}^{N}\sum_{j=1}^{c} p(\omega_j|x_n)\,\hat{p}_k(\omega_j|x_n) \quad \textbf{(18)}$$

According to (**17**) and (**18**), we may work out ***W*** based on (**13**). While the complexity of the resolved problem is lower, the calculation of the reverse matrix is no difficult, however, and the complexity of the resolved problem is larger, the calculation of the reverse matrix is influenced by data noise coming from individual network easily, which results in a homologous fall in the generalized ability of integration neural networks. At the moment, we may pick up some good networks with high precision to integrate and delete some bad networks with low precision so as to reduce the complexity of the problem.

It can also be known well from that the essential of the algorithm is to seek the linear relations between the true posterior probability and several estimated posterior probabilities, which just is multi-element linear regression in statistics.

4 Application

Table 1 is some familiar fault sources and fault symptoms information in stream turbine vibration fault diagnosis in [27]. Ten kinds of typical faults in rotation machines are selected as outputs of neural networks, the score peak energy values in the range of nine frequency ranges are used to act as input characteristic vector, a sheet of two-dimensional decision table then is formed. It is necessary to say that frequency score energy values are gained by three-time wavelet package decomposition to faults wave-shape in the range of given frequency and must be probabilistically complete.

Table 1. Fault sources and symptoms table of stream turbine

Sample D	*a* 0.01-0.39f	*b* 0.40-0.49f	*c* 0.50f	*d* 0.51-0.99f	*e* f	*f* 2f	*g* (3/5)f	*h* odd-timef	*i* high-frequencyf
1	0.00	0.00	0.00	0.00	0.90	0.05	0.05	0.00	0.00
2	0.00	0.30	0.10	0.60	0.00	0.00	0.00	0.00	0.10
3	0.00	0.00	0.00	0.00	0.40	0.50	0.10	0.00	0.00
4	0.10	0.80	0.00	0.10	0.00	0.00	0.00	0.00	0.00
5	0.10	0.10	0.10	0.10	0.20	0.10	0.10	0.10	0.10
6	0.00	0.00	0.00	0.00	0.20	0.15	0.40	0.00	0.25
7	0.00	0.00	0.10	0.90	0.00	0.00	0.00	0.00	0.00
8	0.00	0.30	0.10	0.60	0.00	0.00	0.00	0.00	0.00
9	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
10	0.00	0.00	0.00	0.00	0.00	0.80	0.20	0.00	0.00

f: 50HZ, D1: unbalance, D2: vapour impulsion force even, D3: deflection centre, D4: oil film whorling, D5: rotator collision, D6: symbiosis looseness fault, D7: push force bearing fault, D8: gasping vibration, D9: bearing seat looseness fault, D10:unequal bearing rigidity.

According to Table 1, input characteristic vector is ***X***={*a*,*b*,*c*,*d*,*e*,*f*,*g*,*h*,*i*}, and output vector is ***Y***={*D*}. According to discrete method of Kohonen network [28], a discretized decision table is gained below.

Table 2. Discretized decision table

Sample D	*a* 0.01-0.39f	*b* 0.40-0.49f	*c* 0.50f	*d* 0.51-0.99f	*e* f	*f* 2f	*g* (3/5)f	*h* odd-timef	*i* high-frequencyf
1	0	0	0	0	1	0	0	0	0
2	0	1	1	1	0	0	0	0	1
3	0	0	0	0	1	1	0	0	0
4	1	1	0	1	0	0	1	0	0
5	1	1	1	1	1	1	0	1	1
6	0	0	0	0	1	1	1	0	1
7	0	0	1	1	0	0	0	0	0
8	0	1	1	1	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0
10	0	0	0	0	0	1	0	0	0

Known from the distinguishable matrix of rough set [29], {*b*,*e*,*f*, *i*} is core attribute set, we therefore can get some reduction attribute sets such as {*a*,*b*,*d*,*e*,*f*,*i*} or {*b*,*c*,*e*,*f*,*h*,*i*}, etc. Here the attribute set *X'*={*a*,*b*,*d*,*e*,*f*,*i*} is selected as conditional attributes and *D* serves for decision attribute, a simplified decision table is gained as shown Table 3. Seen from the Table 3, the original 9 attributes now become the six, the redundant conditional attributes are ignored, and the decision table is simplified greatly, the structure of neural network therefore is simplified.

If the condition attributes in Table 3 are used to serve for the inputs of multilayer perceptron, and the

Table 3. Simplified decision table

D	*a*	*b*	*d*	*e*	*f*	*i*
1	0	0	0	1	0	0
2	0	1	1	0	0	1
3	0	0	0	1	1	0
4	1	1	1	0	0	0
5	1	1	1	1	1	1
6	0	0	0	1	1	1
7	0	0	1	0	0	0
8	0	1	1	0	0	0
9	1	0	0	0	0	0
10	0	0	0	0	1	0

decision attribute is the outputs of the network classifier, and the samples in Table 3 are also used to train the perceptron. Then the trained network may be applied to implement fault diagnosis. Presently, to select the suitable structure of neural network still hasn't theoretical instructions, according to previous experience, the sum of the nodes in hidden layer shouldn't be smaller than the sum of the nodes in input layer. According to the proposed method in this paper and for convenient analysis, we select two neural networks to integrate for faults diagnosis of stream turbine with IPSO-based, and the structures of these two neural networks are respectively selected as 6-8-10 and 6-10-10. The overall diagnosis network is seen in Fig.2.

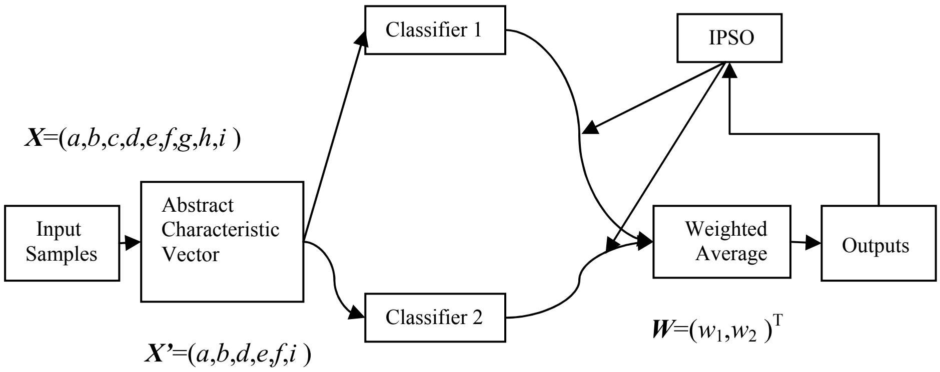

Fig. 2. The block diagram of ensemble diagnosis networks with IPSO-based

Set parameters $c1=c2=2.0$, $k1=0.5$, $k2=0.95$, $G=800$, adaptive function $f(x_{ji})=1/e$, error $\varepsilon\leq 0.002$, then a diagnosis example is given blow.

Example: At one time fault pattern $\boldsymbol{X}$={0.39,0.07,0,0.06,0,0.13,0,0,0.35} is sensed, after discretized, $\boldsymbol{X}$={1,0,0,0,0,1,0,0,1}. After reduction by rough set, $\boldsymbol{X'}$={1,0,0,0,1,1}. Then $\boldsymbol{X'}$ is sent to input neurons of classifier 1 and classifier 2. The gained weight vector is $W=[0.462, 0.538]^T$ after applying the algorithm with IPSO-based. The output result of classifier 1 is $\boldsymbol{Y}_1$={0, 0, 0.21, 0.95, 0, 0.9, 0, 0, 0.23, 0} and the output result of classifier 2 is $\boldsymbol{Y}_2$={0, 0, 0.17, 0.99, 0, 0.92, 0, 0, 0.21, 0}, the final output result is $Y=w_1\times Y_1+ w_2\times Y_2$={0, 0,0.19, 0.97, 0, 0.91,0,0,0.22,0}. Since fault patterns $D4$ and $D6$ have larger outputs and larger than 0.5, that is, 0.97 and 0.91, we therefore can conclude that the faults are symbiosis looseness fault or oil film whorling. In the end, spot practical checking validates the correctness of the diagnosis results.

This is totally similiar to the result in [27], but here the applied method is differencent. In the above text we only analyse the ensemble of two neural classifiers, and their types are also homogeneous, actually, we also apply more neural classifiers to integrate, however, their types but may be heterogeneous.

8 Conclusion

The proposed method in this paper can effectively integrate the diagnosis results of more neural networks, and the correctness of fault diagnosis therefore is improved dramatically. Meanwhile, Due to applying the IPSO-based algorithm, the integration precision of the networks is improved greatly, and the time of system ensemble is also reduced well. In addition, while the structure of the networks is complex, those network classifiers with low diagnosis precision may be eliminated, thus the complexity of the algorithm may for that be reduce. Both simulation and experiment indicate the proposed method is extremely effective.

References:

[1]H. J. Yu, *Intelligent Diagnosis Based on Neural Networks*, Metallurgy Industry Publication House, Beijing ,2000.

[2]Y. Zhang, An Artificial New Network Approach to Transformer Fault Diagnosis, *IEEE Trans. on Power Delivery*, Vol.11, 1996, pp.1836-1841.

[3]Hongsheng Su and Qunzhang Li, Cooperative multi-immune-agent systems for substation fault diagnosis based on ontology, *WSEAS Transactions on Power Systems*, Vol.1, No.2, 2006, pp.446-453.

[4] H. Sunan, R. Wei, Use of Neural Fuzzy Networks with Mixed Genetic/Gradient Algorithm in Automated Vehicle Control, *IEEE Trans. on Industrial Electronics*, Vol.11, 1999, pp. 1090-1016

[5] J. Yang, Z. S. Fen, K. L. Huang, *Intelligent Fault Diagnosis Technology for Equipment*, National Defence Industry Press, Beijing,2004.

[6] J. C. Hoskins, K. M. Kaliyur, Himmelblau D. M.,Fault Diagnosis in Complex Chemical Plants Using Artificial Neural Networks, *AIChE Journal*, Vol.11,1991, pp.137-141.

[7] H. Q. Li, B. W. Wan, Application of Modular Wavelet Neural Networks in Industries Product Quality Control, *Control and Decision*, Vol.19, 2004,pp.195-298.

[8] K. S. Narendra, Mukhopadhyay S., Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks, *Neural Networks*, Vol.7,1994, pp.737-742.

[9] K. J. Hunt, D. Sbarbaro, R. Zbkowski, Gawthrop P. J., Neural Networks for Control Systems – A Survey, *Automatica*, Vol.28,1992, pp.1083-1112.

[10]Louisa Lam(ed.), Optimal Combinations of Pattern Classifiers, *Pattern Recognition Letters,* Vol.16,1995,pp.945-954.

[11] J. Kennedy, R. Eberhart, Particle Swarm Optimization, In *Proc. of IEEE International Conference on Neural Network*, Perth ,WA, Australia ,1995, pp.1942-1948

[12] R. Eberhart, Y. Shi, Comparing Inertia Weights and Constrictions Factors in Particle Swarm Optimization, In *Proc. of the 2000 International Conference on Evolutionary Computation*, San Diego, California, IEEE Service Centre, Piscataway, NJ,1995,pp. 84-88.

[13] L. P. Zhang, H. Yu, D. Z. Chen, S. X. Hu, Analysis and Improvement of Particle Swarm Optimization Algorithm, *Information and Control*, Vol.133,2004,pp.1-5.

[14] M. Lovbjerg, T. K. Rasmussen, T. Krink, Hybrid Particle Swarm Optimizer with Breeding and Subpopulations, In *Proc. of the Genetic and Evolutionary Computation Conference*, San Francisco, USA, 2001.

[15] P. N. Suganthan, Particle Swarm Optimizer with Neighborhood Operator. In *Proc. of the Congress on Evolutionary Computation*, Piscataway, NJ, IEEE Service Center,1999.

[16] Y. Shi, R. Eberhart,Fuzzy Adaptive Swarm Optimization. In *Proc. of the Congress on Evolutionary Computation*, Seoul, Korea, 2001.

[17] Z. S. Lu, Z. R. Hou, *Particle Swarm Optimization with Adaptive Mutation*, Acta Electronica Sinca, Vol.32, 2004,pp.416-421.

[18] J. Kennedy, R. Eberhart, A Discrete Binary Version of the Particle Swarm Algorithm. In *Proc. of IEEE International Conference on Computational Cybernetics and Simulation*, Orlando, USA,1997.

[19] F. Z. Zhang, D. Xue, Optimal Concurrent Design Based on Distributed Product Development Lifecycle Modeling. *Robotics and Computer Integrated Manufacturing*, Vol.42,2001,pp. 595-605.

[20] Q., Lu J. G. Yang, A Hybrid PSO Based Method for VQ Index Assignment. In *Proc. of 2005 Chinese Control and Decision Conference*, Harbin, China,2005,pp.903-905

[21] Y. Gao, S. L. Xie, Particle Swarm Optimization Algorithm with Immunity, *Computer Engineering and Application*, Vol.40,2004, pp.4-7.

[22] J. W. Wang, D. W. Wang, Particle Swarm Optimization Algorithm with Gradient Acceleration, *Control and Decision*, Vol.19, 2004,pp.1298-1304.

[23] Z. Q. Liu, X. S. Gu, Particle Swarm Optimization with Information Entropy, In *System Simulation Technology & Application*, Vol.7. USTC, Hefei, China,2005,pp.210-215.

[24] E. T. Jaynes, Information Theory and Statistical Mechanics. *Physical Review*, Vol.106,1957, pp.620-630.

[25] B. Ren, Z. P. Feng, Improved Genetic Algorithm and Particle Swarm Optimization as well as Comparison between Them, *Journal of Nanjing Normal University (Engineering and Technology Edition),* Vol.2,2002,pp.14-20.

[26] G. Chen, J. Yu, Enhanced Particle Swarm Optimization its Application in Soft-sensor, *Control and Decision*, Vol.20,2005, pp.377-381.

[27] H. S. Su, Q. Z. Li, Application Study of Composite Neural Networks in Fault Diagnosis Systems Based on GA Learning, *Computer Science*, Vol.32,2005,pp.255-256.

[28] K. Huang, S. Chen, Z. Zhou, W. Zhang, Multi-source Information Fusion Method Based on Rough Sets and Evidence Theory, *Information and Control*, Vol.33,2004, pp. 422-425

[29] H. L. Zeng, *Intelligent Calculation*, Chongqing University Press, Chongqing, China, 2004.