

Research Article

A Novel Classification Approach through Integration of Rough Sets and Back-Propagation Neural Network

Lei Si,¹ Xin-hua Liu,^{1,2} Chao Tan,¹ and Zhong-bin Wang¹

¹ School of Mechatronic Engineering, China University of Mining & Technology, Xuzhou 221116, China

² Xuyi Mine Equipment and Materials R&D Center, China University of Mining & Technology, Huai'an 211700, China

Correspondence should be addressed to Zhong-bin Wang; wangzbpaper@126.com

Received 13 September 2013; Revised 12 March 2014; Accepted 6 April 2014; Published 24 April 2014

Academic Editor: Olabisi Falowo

Copyright © 2014 Lei Si et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Classification is an important theme in data mining. Rough sets and neural networks are the most common techniques applied in data mining problems. In order to extract useful knowledge and classify ambiguous patterns effectively, this paper presented a hybrid algorithm based on the integration of rough sets and BP neural network to construct a novel classification system. The attribution values were discretized through PSO algorithm firstly to establish a decision table. The attribution reduction algorithm and rules extraction method based on rough sets were proposed, and the flowchart of proposed approach was designed. Finally, a prototype system was developed and some simulation examples were carried out. Simulation results indicated that the proposed approach was feasible and accurate and was outperforming others.

1. Introduction

Across a wide variety of fields, data are being accumulated at a dramatic pace, especially in the age of internet [1, 2]. There is much useful information hidden in the accumulated voluminous data, but it is very hard for us to obtain it. Thus, a new generation of computational tool is needed to assist humans in extracting knowledge and classifying the rapidly growing digital data; otherwise, these data are useless for us.

Neural networks are applied in several engineering fields such as classification problems and pattern recognition. Artificial neural network in its most general form attempts to produce systems that work in a similar way to biological nervous systems. It has the ability to simulate human thinking and owns powerful function and incomparable superiority in terms of establishing nonlinear and experiential knowledge simulation models [3–6]. Back-propagation neural network (BP neural network for short) is the core of feed-forward networks and embodies the essence of artificial neural network. However, conventional BP algorithm has slow convergence rate, weak fault tolerance, and nonunique results. Although many improved strategies have been proposed, such as additional momentum method, adaptive learning method,

elastic BP method, and conjugate gradient method [7–9], the problems above have still not been solved completely, especially when BP neural networks are applied in multidimensional and uncertain fields. Therefore, newer techniques must be coupled with neural networks to create more efficient and complex intelligent systems [10].

In an effort to model vagueness, rough sets theory was proposed by Pawlak in 1982 [11, 12]. Its outstanding feature is not to need the specific relation description about some characteristics or attributes, but to determine the approximate region of existing problems and find out the inherent rules through the indiscernibility relations and indiscernibility classes. This theory has been successfully applied in many fields such as machine learning, data mining, data analysis, and expert systems [13, 14]. In the paper, rough sets theory and BP neural network are presented as an integrated method because they can discover patterns in ambiguous and imperfect data and provide tools for data analysis. The decision table is established reasonably through discretizing attribution values. The features of decision table are analyzed using rough sets and subsequently a classification model based on these features is built through integration of rough sets and BP neural network. Actually, this model uses the

decision rules which are extracted by the algorithm based on value reduction for classification problems.

The paper is structured as follows. Some related works are outlined based on literature in Section 2. The basics of rough sets theory and BP neural network are presented in Section 3. The framework and key algorithms are proposed and the flowchart of the proposed approach is designed in Section 4. A simulation example and some comparisons are put forward to validate the proposed approach in Section 5. Our conclusions are summarized in Section 6.

2. Literature Review

In recent years, most of classifier architectures are mainly constructed based on artificial intelligence (AI) algorithms with the unceasing development and improvement of AI theory. In this section, we try to summarize some recent literatures which are relevant to the construction methods of classification system. Hassan et al. constructed a classifier based on neural networks and applied rough sets theory in attributes reduction to preprocess the training set of neural networks [15]. Berardi et al. proposed a principled approach to build and evaluate neural network classification models for the implementation of decision support system (DSS) and verified the optimization speed and classified samples accuracy [16]. In [17, 18], the architecture of neural network with fuzzy input was proposed to effectively improve the classification performance of neural network. Hu et al. presented a self-organized feature mapping neural network to reasonably determine the input parameters in pavement structure design [19]. In [20, 21], wavelet neural networks were applied in the construction of classification system and better prediction results were obtained. In [22, 23], a classifier in which improved particle warm algorithm was used to optimize the parameters of wavelet neural network was established and the application in nonlinear identification problems demonstrated its strong generalization capability. Sengur et al. described the usage of wavelet packet neural networks for texture classification problem and provided a wavelet packet feature extractor and a multilayer perceptron classifier [24]. Hassan presented a novel classifier architecture based on rough sets and dynamic scaling in connection with wavelet neural networks and the effectiveness was verified by the experiments [25].

Although many approaches for the establishment of classification systems have been presented in above literatures, they have some common disadvantages summarized as follows. On one hand, the classification effect of signal neural network cannot be guaranteed when the networks are applied in multidimensional and uncertain fields. On the other hand, the combination method of rough sets and neural networks can only deal with classification and recognition problems that possess discretized dataset and cannot process the problems that contain continuous dataset.

In this paper, a novel classification system based on the integration of rough sets and BP neural network is proposed. The attribution values are discretized through PSO and the decision table is constructed. The proposed model

takes account of attributes reduction by discarding redundant attributes and a rule set is generated from the decision table. A simulation example and comparisons with other methods are carried out and the proposed approach is proved feasible and outperforming others.

3. Basic Theory

3.1. Rough Sets Theory. Fundamental to rough sets theory is the idea of an information system $IS = (U, C)$, which is essentially a finite data table consisting of different columns labeled by attributes and rows labeled by objects of interest, and entries of the table are attribute's values. U is nonempty finite set of objects called a universe and C is a nonempty finite, where each $c \in C$ is called condition attribute. In $IS = (U, C)$, every $B \subseteq C$ generates an indiscernibility relation $Ind(B)$ on U which can be defined as follows:

$$Ind(B) = \{(x, y) \in U \times U : b(x) = b(y), \forall b \in B\}. \quad (1)$$

$U/Ind(B)$ is a partition of U by B . For all $x \in U$, the equivalence class of x in relation $U/Ind(B)$ can be defined as follows:

$$[x]_{Ind(B)} = \{x \in U : b(y) = b(x), \forall b \in B\}. \quad (2)$$

According to I , two crisp sets $\underline{B}X$ and $\overline{B}X$ called the lower and the upper approximation of the set of objects X can be defined as follows:

$$\begin{aligned} \underline{B}X &= \{x \in U : [x]_{Ind(B)} \subseteq X\}, \\ \overline{B}X &= \{x \in U : [x]_{Ind(B)} \cap X \neq \Phi\}, \end{aligned} \quad (3)$$

where $[x]_{Ind(B)}$ denotes the set of all equivalence classes of $Ind(B)$. $\underline{B}X$ is the set of all objects from U which can be certainly classified as elements of X employing the set of attributes B . $\overline{B}X$ is the set of objects of U which can be possibly classified as elements of X using the set of attributes B [25].

Decision table $DT = (U, C \cup D)$ is a special form of an information system and the major feature is the distinguished attribute set D , where $C \cap D = \Phi$ is called the decision attribute. Generally speaking, there is a certain degree of dependency between condition attribute and decision attribute, which can be defined as follows:

$$\gamma_C(D) = \frac{POS_C(D)}{|U|} = \frac{|\bigcup_{X \in U/D} \underline{C}X|}{|U|}, \quad (4)$$

where $POS_C(D)$ is referred to the C -positive region of D .

Due to the relevance between condition attribute and decision attribute, not all condition attributes are necessary for decision attribute so as to introduce the attribute reduction which is a smaller set of attributes that can classify objects with the same discriminating capability as that of the original set. As well known, the reduct is not the only one [26, 27].

When we want to determine the reduct of decision table, the significance degree of attributes is commonly used and can be defined as follows:

$$sig(\alpha, B; D) = \gamma_{\{\alpha \cup B\}}(D) - \gamma_B(D), \quad (5)$$

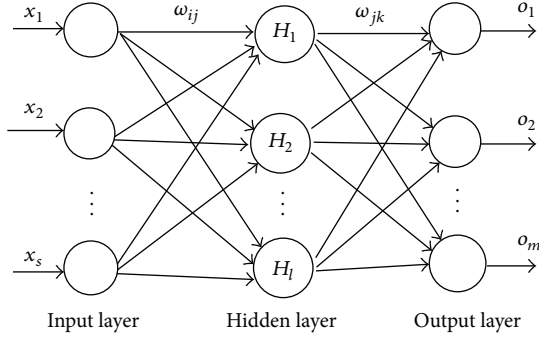


FIGURE 1: The topology structure of BP-NN.

where $\text{sig}(\alpha, B; D)$ denotes the significance degree of attribute α to attribute B , relative to decision attribute D .

For all $R \in C$, if $\text{POS}_{\{C-R\}}(D) = \text{POS}_C(D)$, then R in C is superfluous for D . Otherwise, R in C is indispensable for D . The set composed of all attributes that are indispensable for D is called the core of C relative to D , namely, relative core $\text{CORE}_D(C)$. This relative core cannot be removed from the system without a loss in the knowledge that can be derived from it.

When determined, the relative core of decision table must be firstly judged whether it is the reduct of C relative to D , namely, relative reduct. The judgment rules are shown as follows.

If the nonempty subset P of attribute set C satisfies the conditions: (1) $\text{POS}_P(D) = \text{POS}_C(D)$; (2) $\forall R \subseteq P$, $\text{POS}_R(D) \neq \text{POS}_C(D)$, then P is called a reduct of C relative to D , denoted as $\text{RED}_D(C)$.

In addition, decision rules can be perceived as data patterns which represent the relationship between the attributes values of a classification system [11, 12]. The form of the decision rules can be shown as IF η THEN δ , where η is the conditional part of a rule set and it is a conjunction of selectors; δ is the decision part of attribute d : ($d \in D$) and it usually describes the class of decision attribution.

3.2. BP Neural Network. BP neural network (BP-NN) which is surely a back-propagating neural network belongs to the class of networks whose learning is supervised and the learning rules are provided by the training set to describe the network behavior. The topology structure of BP-NN can be shown in Figure 1.

In Figure 1, input vector $X = \{x_1, x_2, \dots, x_s\}$ is furnished by the condition attribution values of reduced decision table and output vector $O = \{o_1, o_2, \dots, o_m\}$ is the prediction class of decision attribution. The output of hidden layer can be calculated as follows:

$$H_j = f\left(\sum_{i=1}^s \omega_{ij} x_i - a_j\right), \quad j = 1, 2, \dots, l, \quad (6)$$

where ω_{ij} is the connection weight between input and hidden layers; l is the number of hidden layer nodes; a_j is the threshold of hidden layer node j ; f is the activation function

of hidden layer and can be chosen as linear function or sigmoid function $f(x) = 1/(1 + e^{-x})$.

The output of output layer can be calculated as follows:

$$o_k = \sum_{j=1}^l H_j \omega_{jk} - b_k, \quad k = 1, 2, \dots, m, \quad (7)$$

where ω_{jk} is the connection weight between hidden and output layers; m is the number of output layer nodes; b_k is the threshold of output layer node k .

The training of network parameters ω_{ij} , ω_{jk} , a_j , and b_k would depend on the error value which can be calculated by the following equation:

$$e_k = Y_k - o_k, \quad k = 1, 2, \dots, m, \quad (8)$$

where o_k and Y_k are current and desired output values of the network, respectively. The weight values and thresholds of the network can be updated as follows:

$$\begin{aligned} \omega_{ij}(t+1) &= \omega_{ij}(t) + \zeta(t) H_j (1 - H_j) x_i \sum_{k=1}^m e_k \omega_{jk}, \\ \omega_{jk}(t+1) &= \omega_{jk}(t) + \zeta(t) H_j e_k, \end{aligned} \quad (9)$$

$$a_j(t+1) = a_j(t) + \zeta(t) H_j (1 - H_j) x_i \sum_{k=1}^m e_k \omega_{jk},$$

$$b_k(t+1) = b_k(t) + e_k,$$

where t is the current iteration times; $\zeta(t)$ is the learning rate and the range is the interval of $[0, 1]$.

The learning rate has great influence on the generalization effect of neural network. Larger learning rate would significantly modify the parameters of weight values and thresholds and increase the learning speed. But overlarge learning rate would produce larger fluctuations in the learning process, and excessively small learning rate would generate slower network convergence and stabilize the weight values and thresholds difficultly. Therefore, this paper presents the variable learning rate algorithm to solve the above problems, which can be described as follows:

$$\zeta(t) = \zeta_{\max} - \frac{t(\zeta_{\max} - \zeta_{\min})}{t_{\max}}, \quad (10)$$

where ζ_{\max} and ζ_{\min} are the maximum and minimum of learning rate; t_{\max} is the number of maximum iterations.

4. The Proposed Approach

This section tries to present a new approach aiming at connecting rough sets with BP neural network to establish a classification system. The section has five main parts and can be elaborated through the following subsections.

4.1. The System Construction Process. The construction process of classification system can be shown as follows.

- (1) Preprocessing historical dataset to obtain the sample data is the application precondition of rough sets and BP neural network. In the process of preprocessing, the historical dataset would be discretized through PSO algorithm to construct the decision table.
- (2) Simplify the input of neural network including input dimension and training set number through the attribute reduction based on significance degree algorithm. Deleting the same row in the decision table can simplify the training samples and eliminating the superfluous column (condition attribute) can simplify the network input dimension number. Thus, reduce the decision table using some reduct or sum of several calculated reducts; that is, remove from the table attributes not belonging to the chosen reducts.
- (3) BP neural network integrated with rough sets is used to build the network over the reduced set of data. Decision rules are extracted from the reduced decision table as the basis of network structure through a value reduction algorithm of rough sets.
- (4) Perform network learning for parameters ω_{ij} , ω_{jk} , a_j and b_k . Do this step until there is no significant change in the output of network and the system construction process can be shown as Figure 2.

4.2. The Discretization Algorithm for Decision Table. Due to the fact that rough sets theory cannot directly process continuous values of attributes, the decision table composed of continuous values must be discretized firstly. Thus, the positions of breakpoints must be selected appropriately to discretize the attributes values, so as to obtain fewer breakpoints, larger dependency degree between attributes, and less redundant information.

In this paper, the basic particle swarm optimization algorithm (PSO) described in [28] is presented to optimize the positions of breakpoints. In PSO, a swarm of particles are initialized randomly in the solution space and each particle can be updated in a certain rule to explore the optimal solution P_{best} after several iterations. Particles are updated through tracking two "extremums" in each iteration. One is the optimal solution P_{best} found by the particle itself and another is the optimal solution G_{best} found by the particle swarm. The specific iteration formulas can be expressed as follows:

$$\begin{aligned} v_i^{n+1} &= \omega_n v_i^n + c_1 r (P_{\text{best}} - x_i^n) + c_2 r (G_{\text{best}} - x_i^n), \\ x_i^{n+1} &= x_i^n + v_i^{n+1}, \end{aligned} \quad (11)$$

where $i = 1, 2, \dots, M$; M is the number of particles; n is the current iteration times; c_1 and c_2 are the acceleration factors; r is the random number between 0 and 1; x_i^n is the current position of particle i ; v_i^n is the current update speed of particle i ; ω_n is the current inertia weight and can be updated by the following equation:

$$\omega_n = \frac{(\omega_{\max} - \omega_{\min})(N_{\max} - n)}{N_{\max}} + \omega_{\min}, \quad (12)$$

where ω_{\max} and ω_{\min} are the maximum and minimum of inertia weight; N_{\max} is the number of maximum iterations.

The main discretization steps for decision table through the PSO can be described as follows.

- (1) Normalize the attribute values and initialize the breakpoints number h , the maximum iterations N_{\max} , the minimum dependency degree between condition attribute and decision attribute γ_{\min} , the particles number M , the initial position x_i^0 , and initial velocity v_i^0 which are $h \times \mu$ matrixes (μ is the number of attributes).
- (2) Discretize the attribute values with the position of particle i and calculate the dependency degree γ_i as the fitness value of each particle to initialize P_{best} and G_{best} .
- (3) Update x_i , v_i ($v_i \leq v_{\max}$), and ω_n . Discretize the attribute values and calculate the fitness value of each particle so as to update P_{best} and G_{best} .
- (4) If $n > N_{\max}$ or $\gamma_i \geq \gamma_{\min}$, then the iteration is ended and G_{best} or x_i is output. Otherwise, go to step (3).
- (5) According to the needs, set different h and return to step (1) to search the optimal particles. Then, compare the optimal particles and select the better one as the basis of discretization for decision table. The more detailed discretization process is elaborated in Section 4.5.

4.3. Rough Sets for the Attribution Reduction. One of the fundamental steps in proposed method is reduction of pattern dimensionality through feature extraction and feature selection [10]. Rough sets theory provides tools for expressing inexact dependencies within data [29]. Features may be irrelevant (having no effect on the processing performance) and attribution reduction can neglect the redundant information to enable the classification of more objects with a high accuracy [30].

In this paper, the algorithm based on attribution significance degree is applied in the attribution reduction of decision table. The specific reduction steps can be described as follows.

- (1) Construct the decision table $DT = (U, C \cup D)$ and calculate the relative core $\text{CORE}_D(C)$ of original decision table. Initialize reduct = $\text{CORE}_D(C)$ and the redundant attribute set redundant = $C - \text{CORE}_D(C) = \{\alpha_1, \alpha_2, \dots, \alpha_\tau\}$ (τ is the number of attributes in redundant).
- (2) If $\text{POS}_{\text{reduct}}(D) = \text{POS}_C(D)$ and, for all $R \subseteq \text{reduct}$, $\text{POS}_R(D) \neq \text{POS}_C(D)$, then the set reduct is a relative reduct of decision table.
- (3) Otherwise, calculate the significance degree of each attribute in redundant, respectively, through the following equation:

$$\text{sig}(\alpha_j, \text{reduct}; D) = \gamma_{\{\alpha_j \cup \text{reduct}\}}(D) - \gamma_{\text{reduct}}(D), \quad (13)$$

$j = 1, 2, \dots, \tau.$

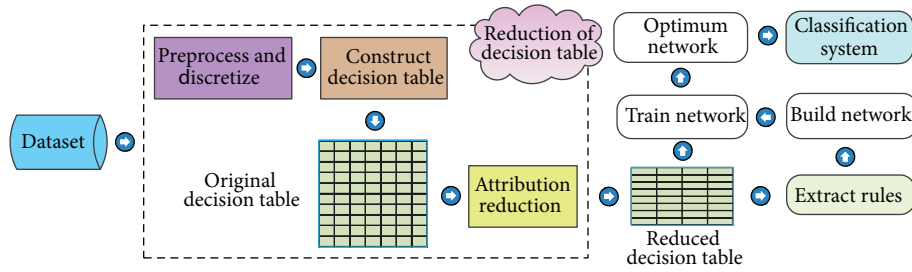


FIGURE 2: The system construction process.

The attribute in redundant with the maximum significance degree is marked as α_{\max} . Update the reduct and redundant through the following equations:

$$\begin{aligned} \text{reduct} &= \text{reduct} \cup \{\alpha_{\max}\}, \\ \text{redundant} &= \text{redundant} - \{\alpha_{\max}\}. \end{aligned} \quad (14)$$

- (4) Go to step (2) until $\text{POS}_{\text{reduct}}(D)$ satisfies the conditions; then output $\text{RED}_D(C) = \text{reduct}$.

Therefore, the irrelevant features are neglected and the DT is reduced. Thus, a reduced decision table will be constructed and can be regarded as the training set to optimize the structure of RS-BPNN classifier.

4.4. Rough Sets for the Rules Extraction. The knowledge in the trained network is encoded in its connection weights and it is distributed numerically throughout the whole structure of neural network. For the knowledge to be useful in the context of a classification system, it has to be articulated in a symbolic form usually in the form of IF-THEN rules. Since neural network is a “black box” for users, the rules which are implicit in the connection weights are difficult to understand, so extracting rules from neural network is extremely tough because of the nonlinear and complicated nature of data transformation conducted in the multiple hidden layers [31]. In this paper, the decision rules are indirectly extracted from the reduced decision table using the value reduction algorithm of rough sets [32]. This algorithm can be elaborated in the following steps.

Step 1. Checking the condition attributes of reduced decision table column by column. If bringing conflicting objects after deleting one column, then preserving the original attribute value of these conflicting objects; If not bringing conflicts, but containing duplicate objects, then marking this attribute value of duplicate objects as “*”; marking this attribute value as “?” for the other objects.

Step 2. Deleting the possible duplicate objects and checking every object that contains the mark “?”. If the decisions can be determined only by the unmarked attribute values, then “?” should be modified as “*”; or else as the original attribute value. If all condition attributes of a particular object are marked, then the attribute values marked with “?” should be modified as the original attribute value.

Step 3. Deleting the objects whose condition attributes are marked as “*” and the possible duplicate objects.

Step 4. If only one condition attribute value is different between any two objects and this attribute value of one object is marked as “*”, and for this object if the decision can be determined by the unmarked attribute values, then deleting another object, or else deleting this object.

Each object represents a classification rule in the decision table which is value-reduced through the above steps and the number of attributes which are not marked as “*” of every object makes up the condition number of this rule. Moreover, this rule set can provide the rational basis for the structure of BP neural network.

4.5. The Flowchart of the Proposed Approach. According to the above description about the classifier based on integration algorithm of rough sets and BP neural network, the proposed approach is an iterative algorithm and can be coded easily on a computer, and the flowchart can be summarized as shown in Figure 3.

5. Simulation Example

A classifier named RS-BPNN (short for rough sets with BP neural network) has been set up and implemented by VC 6.0. In this section, an engineering application of shearer running status classification in a coal mine was put forward as a simulation example to verify the feasibility and effectiveness of the proposed approach. The classification capabilities of different types of neural network models were compared and the proposed approach was proved outperforming others.

5.1. The Construction of Decision Table. Due to the poor working conditions of coal mining, there are many parameters relating to the running status of shearer, mainly including cutting motor current, cutting motor temperature, traction motor current, traction motor temperature, traction speed, tilt angle of shearer body, tilt angle of rocker, and vibration frequency of rocker transmission gearbox, marked as $C_1, C_2, C_3, C_4, C_5, C_6, C_7,$ and $C_8,$ respectively. According to the information database acquired from the 11070 coal face in the number 13 Mine of Pingdingshan Coal Industry Group Co., 200 groups datasets of shearer running status were rearranged as shown in Table 1. This “running status” referred

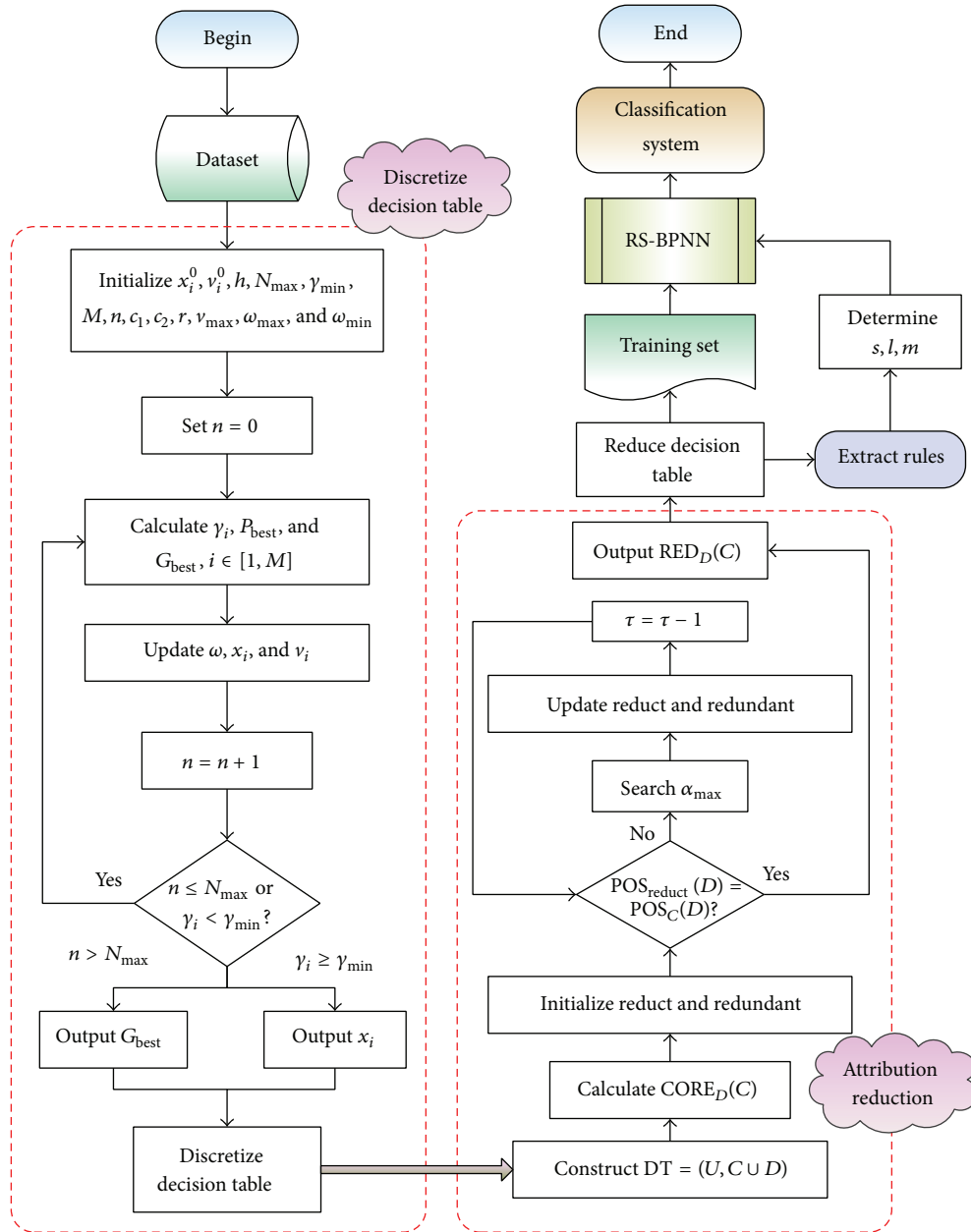


FIGURE 3: The flowchart of the proposed approach.

to the cutting status of shearer rocker and could be expressed by the ratio of actual cutting load to rated load.

Therefore, in decision table DT , the condition attribute set and decision attribute set could be expressed as $C = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8\}$ and $D = \{d\}$ (d denoted the shearer running status). The values of decision attribution were discretized based on the following assumptions: for the ratio of actual cutting load to rated load of shearer being greater than 1.6, $d = 4$; for the range of this ratio being in interval of (1.2, 1.6], $d = 3$; for the range of this ratio being in interval of (0.8, 1.2], $d = 2$; for this ratio being less than 0.8, $d = 1$. The discretized values of condition attributes could be determined through the algorithm described in Section 4.2.

The condition attributes to be discretized were $C_1 \sim C_8$, so $\mu = 8$. Other parameters of PSO were initialized as follows: $M = 100$, $N_{max} = 200$, $\gamma_{min} = 0.9$, $v_{max} = 0.6$, $c_1 = c_2 = 1.6$, r is a random number from $[0, 1]$, $\omega_{max} = 0.9$, and $\omega_{min} = 0.4$. As the breakpoints number h had greater influence on the dependency degree $\gamma_C(D)$ of discretized decision table, the comparison of two important parameters: dependency degree and simulation time, was analyzed when h was assigned various values. The comparison results were shown in Figure 4.

From Figure 4, it was observed that the dependency degree and simulation time both showed ascending tendency with the increase of breakpoints number h . The dependency

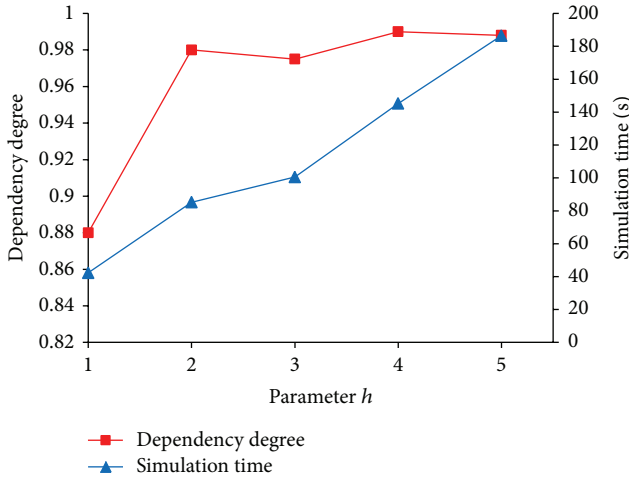


FIGURE 4: The change curves of dependency degree and simulation time.

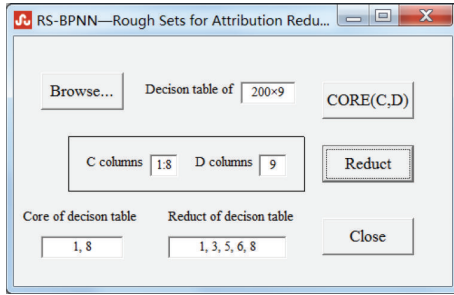


FIGURE 5: Attribution reduction.

degree was not significantly increased and the simulation time was obviously increased when $h \geq 2$. So, h was considered being equal to 2 and $\gamma_C(D) = 0.98$. The breakpoints of each attribute value were shown in Table 2. The breakpoints were applied to discretizing the parameter values in Table 1. If the value was in interval of $(0, Br_1]$, the discretized attribution value was labeled as 1; if the value was in interval of (Br_1, Br_2) , the discretized attribution value was labeled as 2; otherwise, it was labeled as 3. The discretized decision table was shown in Table 3.

5.2. Relative Reduct and Rules Extraction. Through the attribution reduction algorithm presented in Section 4.3, the function “CORE(C, D)” was invoked to obtain the relative core of decision table and $CORE_D(C) = \{C_1, C_8\}$, shown in Figure 5. As the minimum relative reduct of decision table, $RED_D(C) = \{C_1, C_3, C_5, C_6, C_8\}$ was determined by invoking function “Reduct.”

In the module of “Rough Set for Attribution Reduction,” a new decision table of 200×6 was established by the attributions $\{C_1, C_3, C_5, C_6, C_8\}$ and decision attribution $\{d\}$. The function “Refresh (DT)” was called to reject the repeated objects in this new decision table and a reduced decision table was provided finally, containing 79 objects, as shown in Figure 6. Subsequently, the decision rules were extracted

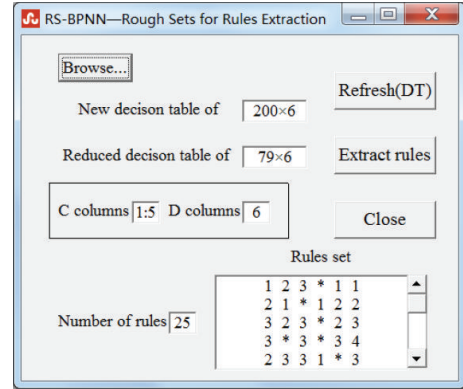


FIGURE 6: Rules extraction.

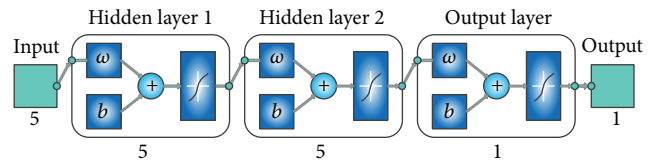


FIGURE 7: The structure of RS-BPNN classifier.

through function “Extract rules” which was based on the value reduction algorithm and 25 rules were obtained, as shown in Figure 6. The extracted rules could reveal the potential rules (knowledge) in dataset; for example, the first rule “123 * 11” in Figure 6 illustrated that if cutting motor current $C_1 \in (0, 48.56]$, traction motor current $C_3 \in (82.37, 95.42)$, traction speed $C_5 \in [3.24, +\infty)$, tilt angle of shearer body C_6 was arbitrary value, and vibration frequency of rocker transmission gearbox $C_8 \in (0, 651.28]$, then the running status class was 1, which was in accordance with engineering practice situation. Moreover, RS-BPNN classifier, in fact, was to diagnose the status of objects according to these potential rules (knowledge) and the structure of network could be constructed reasonably based on the number of rules so as to enhance the classification accuracy.

5.3. The Structure and Testing of RS-BPNN Classifier. Because the dataset obtained 25 rules (knowledge), the RS-BPNN should be constructed with double hidden layers and the number of neurons was 5×5 , which must completely contain these rules (knowledge). The number of input layer nodes $s = 5$; the number of output layer nodes $m = 1$. Other parameters of network were determined as follows: $\zeta_{\max} = 0.9$; $\zeta_{\min} = 0.4$; $t_{\max} = 500$; the initial connection weights were assigned somewhat random values. The structure of RS-BPNN classifier was shown in Figure 7.

The reduced decision table, as the training set, was presented to the network several times. The testing set, composed of 40 samples extracted from the information database randomly, was not seen by the network during the training phase and it was only used for testing the generalization of neural network after it was trained, as shown in Table 4.

TABLE 1: The running status of shearer and its corresponding parameters.

Samples	C_1 (A)	C_2 ($^{\circ}$ C)	C_3 (A)	C_4 ($^{\circ}$ C)	C_5 (m/min)	C_6 ($^{\circ}$)	C_7 ($^{\circ}$)	C_8 (Hz)	Running status
1	45.12	78.74	65.46	88.16	3.51	2.52	45.12	628.85	0.78
2	48.19	75.22	70.78	84.28	3.42	3.62	53.22	683.46	0.79
3	50.25	80.19	85.14	91.25	2.88	5.36	40.22	705.76	0.82
4	45.26	82.43	62.45	84.15	3.12	2.85	49.36	642.14	0.76
5	64.28	77.16	90.45	91.41	2.56	2.26	58.55	823.29	1.22
6	46.22	85.24	68.55	80.46	3.26	2.86	58.46	711.24	0.83
7	72.56	85.12	102.42	102.36	2.76	10.26	38.41	881.25	1.17
8	62.33	74.24	92.46	84.46	3.56	8.26	60.27	821.46	1.12
9	44.23	89.19	63.15	85.78	3.82	3.22	46.25	578.26	0.75
10	89.23	81.24	120.36	86.14	2.01	1.05	50.36	1035.48	1.63
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
198	49.22	75.24	71.22	80.25	3.46	4.26	45.26	542.16	0.72
199	68.65	87.16	118.64	90.46	4.66	4.36	49.35	845.29	1.36
200	50.22	78.15	90.46	95.44	3.22	7.22	44.21	682.64	1.11

TABLE 2: The corresponding breakpoints of each attribute value at $h = 2$.

Breakpoints	C_1 (A)	C_2 ($^{\circ}$ C)	C_3 (A)	C_4 ($^{\circ}$ C)	C_5 (m/min)	C_6 ($^{\circ}$)	C_7 ($^{\circ}$)	C_8 (Hz)
Br_1	48.56	81.49	82.37	84.79	2.41	4.16	47.35	651.28
Br_2	65.25	85.78	95.42	90.25	3.24	7.38	54.25	847.46

TABLE 3: The discretized decision table.

U	C								D
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	
1	1	1	1	2	3	1	1	1	1
2	1	1	1	1	3	1	2	2	1
3	2	1	2	3	2	2	1	2	2
4	1	2	1	1	2	1	2	1	1
5	3	1	2	3	2	1	3	2	3
6	1	2	1	1	3	1	3	2	2
7	3	2	3	3	2	3	1	3	2
8	2	1	2	1	3	3	3	2	2
9	1	3	1	2	3	1	1	1	2
10	3	1	3	2	1	1	2	3	4
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
198	2	1	1	1	3	2	1	1	1
199	3	3	3	3	3	2	2	2	2
200	2	1	2	3	2	2	1	2	2

After the training phase, the RS-BPNN classifier was obtained. In order to test the performance of this classifier expediently, a testing interface was designed as shown in Figure 8. The testing set was imported with the ‘‘Import samples’’ button and was processed with the functions of ‘‘Discretize’’ and ‘‘Reduce attribution.’’ The prediction results of RS-BPNN classifier were output through the function of ‘‘Classify’’ in Figure 8. The contrast of prediction class and actual class was shown brightly in Figure 9.

When the classification accuracy of RS-BPNN was computed in Figure 8, the output of classifier was impossibly equal

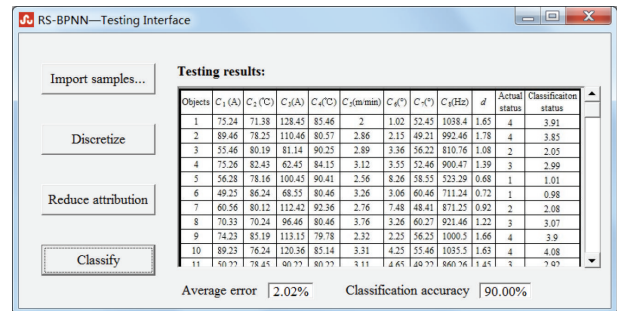


FIGURE 8: The testing interface of RS-BPNN classifier.

to desired output. Therefore, it allowed the output to differ from the desired value. If the difference between classifier output and required value of decision was less than some present value, it was regarded as a correct one. This tolerance margin was decided during simulation evaluation and finally it was 5% of the desired value. Seen from Figures 8 and 9, the classification accuracy was 90.00% and the average classification error was 2.02%. The present model showed higher accuracy and lower error for the classification of shearer running status. The testing results showed that the proposed classifier was proved to be satisfactory and could be used in engineering application in the future.

5.4. Discussion. To evaluate the classification capabilities of different types of neural network models (WNN short for wavelet neural network, RS-WNN short for rough sets coupling with wavelet neural network, BP-NN short for back-propagation neural network, and RS-BPNN short for rough

TABLE 4: The testing samples for the classifier.

Samples	C_1 (A)	C_2 ($^{\circ}$ C)	C_3 (A)	C_4 ($^{\circ}$ C)	C_5 (m/min)	C_6 ($^{\circ}$)	C_7 ($^{\circ}$)	C_8 (Hz)	d
1	75.24	71.38	128.45	85.46	2.00	1.02	52.45	1038.44	1.65
2	89.46	78.25	110.46	80.57	2.86	2.15	49.21	992.46	1.78
3	55.46	80.19	81.14	90.25	2.89	3.36	56.22	810.76	1.08
4	75.26	82.43	62.45	84.15	3.12	3.55	52.46	900.47	1.39
5	56.28	78.16	100.45	90.41	2.56	8.26	58.55	523.29	0.68
6	49.25	86.24	68.55	80.46	3.26	3.06	60.46	711.24	0.72
7	60.56	80.12	112.42	92.36	2.76	7.48	48.41	871.25	0.92
8	70.33	70.24	96.46	80.46	3.76	3.26	60.27	921.46	1.22
9	74.23	85.19	113.15	79.78	2.32	2.25	56.25	1000.46	1.66
10	89.23	76.24	120.36	85.14	3.31	4.25	55.46	1035.48	1.63
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
38	58.46	74.24	90.22	80.25	2.46	3.26	55.26	849.16	1.35
39	65.02	87.16	118.64	90.46	5.06	9.24	48.26	1025.29	1.75
40	50.22	85.15	81.25	89.44	3.34	3.46	44.25	672.64	0.72

TABLE 5: Classification performance of the four models of neural networks.

The models	Training accuracy (%)	Testing accuracy (%)	Classification error (%)	Classification time (s)
WNN	90.00	82.50	3.56	23.22
RS-WNN	92.50	87.50	2.66	18.48
BP-NN	90.00	85.00	2.86	22.26
RS-BPNN	95.00	90.00	2.02	18.33

TABLE 6: Classification results of the four methods.

The methods	Classification accuracy (%)		
	Class I	Class II	Class III
WNN	93.33	86.67	80.00
RS-WNN	100	93.33	86.67
BP-NN	93.33	80.00	86.67
RS-BPNN	100	93.33	100

sets coupling with back-propagation neural network), their best architecture and effective training parameters should be found. In order to obtain the best model, both inputs and outputs should be normalized. The number of nodes in the hidden layer of WNN and RS-WNN was equal to that of wavelet base. If the number was too small, WNN/RS-WNN may not reflect the complex function relationship between input data and output value. On the contrary, a large number may create such a complex network that might lead to a very large output error caused by overfitting of the training sample set.

Therefore, a search mechanism was needed to find the optimal number of nodes in the hidden layer for WNN and RS-WNN models. In this study, various numbers of nodes in the hidden layer had been checked to find the best one. WNN and RS-WNN yielded the best results when the number of hidden nodes was 8.

In this subsection, WNN, RS-WNN, BP-NN, and RS-BPNN were provided to solve the problem of above simulation example. The configurations of simulation environment for four algorithms were uniform and in common with

above simulation example. Initially, for each network, the connection weights were assigned somewhat random values. In order to avoid the random error, the training set of input was presented to the networks 100 times and the average values were calculated. The training accuracy, testing accuracy, classification error, and classification time of four algorithms were shown in Table 5.

From the table, it was observed that the proposed model had a better classification capability and better performance than other models of neural networks in predicting the nonlinear, dynamic, and chaotic behaviors in the dataset, and the proposed model was proved outperforming others. Furthermore, the comparison results were suggesting that the proposed model represents a new good method for classification and decision making, and the new method can be treated as a promising tool for extracting rules from the dataset in industrial fields.

In order to illustrate the superiority of proposed method, Iris data set [23], as a benchmark data set from UCI database, was used to verify the classifiers based on above four types of neural networks and the compared results were shown in Table 6. The sample data had been reduced firstly based on attribution reduction algorithm, and then they were used to train neural networks so the networks coupling with rough sets (RS-WNN and RS-RBNN) could acquire better classification accuracy than signal networks (WNN and BPNN). The rules were extracted through value reduction algorithm to construct RS-BPNN structure, so the classification accuracy of classifier based on RS-BPNN was higher than that of RS-WNN classifier.

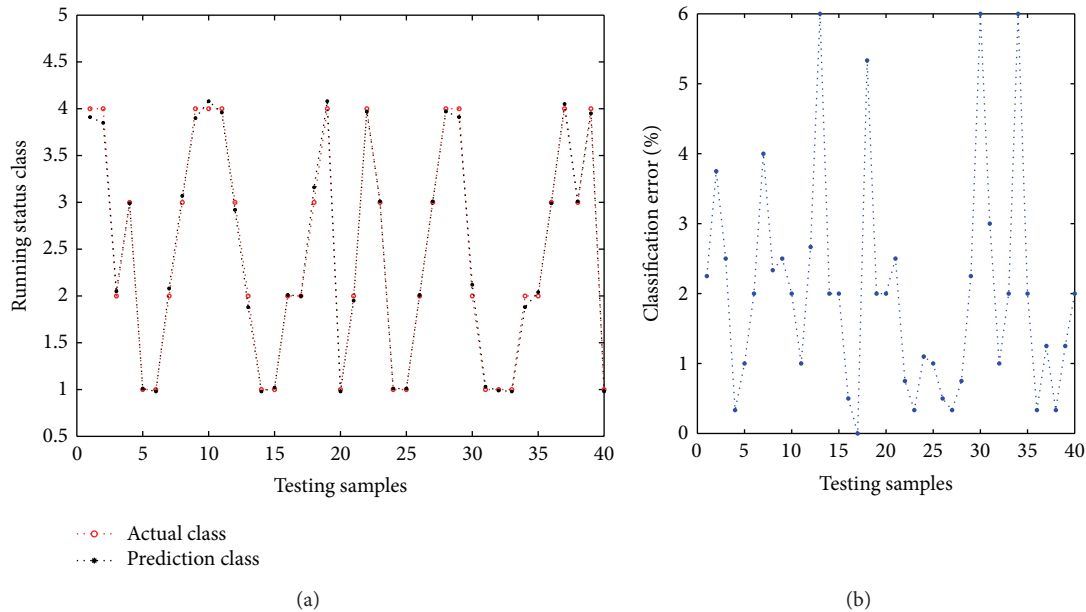


FIGURE 9: The contrast of prediction class and actual class.

6. Conclusions

This paper proposed a novel classifier model based on BP neural network and rough sets theory to be applied in nonlinear systems. The decision table was constructed and discretized reasonably through the PSO algorithm. The decision rules were extracted by the use of value reduction algorithm which provided the basis for the network structure. In order to verify the feasibility and superiority, the proposed approach was applied to a classification problem of an industrial example. The results of comparison simulations showed that the proposed approach could generate more accurate classification and cost less classification time than other neural network approaches and the rough sets based approach. The classification performance of proposed model demonstrated that this method could be extended to process other types of classification problems such as gear fault identification, lithology recognition, and coal-rock interface recognition.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The supports of National High Technology Research and Development Program of China (no. 2013AA06A411), National Natural Science Foundation of China (no. 51005231), Xuyi Mine Equipment and Materials R&D Center Innovation Fund Project (no. CXJJ201306), and Jiangsu Province National Engineering Research Center Cultivation Base for Coal Intelligent Mining Equipment in carrying out this research are gratefully acknowledged.

References

- [1] W. Zhu and F. Wang, "On three types of covering-based rough sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1131–1143, 2007.
- [2] Z. Huang, Z. Yu, and Y. Zheng, "Simple method for predicting drag characteristics of the wells turbine," *China Ocean Engineering*, vol. 20, no. 3, pp. 473–482, 2006.
- [3] S. G. Sklavos and A. K. Moschovakis, "Neural network simulations of the primate oculomotor system: IV. A distributed bilateral stochastic model of the neural integrator of the vertical saccadic system," *Biological Cybernetics*, vol. 86, no. 2, pp. 97–109, 2002.
- [4] I. G. Belkov, I. N. Malyshev, and S. M. Nikulin, "Creating a model of passive electronic components using a neural network approach," *Automation and Remote Control*, vol. 74, no. 2, pp. 275–281, 2013.
- [5] J. S. Chen, "Neural network-based modelling and thermally-induced spindle errors," *International Journal of Advanced Manufacturing Technology*, vol. 12, no. 4, pp. 303–308, 1996.
- [6] A. D. Cartwright, I. S. Curthoys, and D. P. D. Gilchrist, "Testable predictions from realistic neural network simulations of vestibular compensation: integrating the behavioural and physiological data," *Biological Cybernetics*, vol. 81, no. 1, pp. 73–87, 1999.
- [7] B. Zou, X. Y. Liao, Y. N. Zeng, and L. X. Huang, "An improved BP neural network based on evaluating and forecasting model of water quality in Second Songhua River of China," *Chinese Journal of Geochemistry*, vol. 25, no. 1, pp. 167–170, 2006.
- [8] Y. X. Shao, Q. Chen, and D. M. Zhang, "The application of improved BP neural network algorithm in lithology recognition," in *Advances in Computation and Intelligence*, vol. 5370 of *Lecture Notes in Computer Science*, pp. 342–349, 2008.
- [9] W. H. Zhou and S. Q. Xiong, "Optimization of BP neural network classifier using genetic algorithm," in *Intelligence Computation and Evolutionary Computation*, vol. 180 of *Advances in Intelligent Systems and Computing*, pp. 599–605, 2013.

- [10] W. Duch, R. Setiono, and J. M. Zurada, "Computational intelligence methods for rule-based data understanding," *Proceedings of the IEEE*, vol. 92, no. 5, pp. 771–805, 2004.
- [11] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.
- [12] Y. Wu and S. J. Wang, "Method for improving classification performance of neural network based on fuzzy input and network inversion," *Journal of Infrared and Millimeter Waves*, vol. 24, no. 1, pp. 15–18, 2005.
- [13] Z. Pawlak, "Some issues on rough sets," in *Transactions on Rough Sets I*, vol. 3100 of *Lecture Notes in Computer Science*, pp. 1–58, 2004.
- [14] Z. Pawlak and A. Skowron, "Rough sets: some extensions," *Information Sciences*, vol. 177, no. 1, pp. 28–40, 2007.
- [15] Y. Hassan, E. Tazaki, S. Egawa, and K. Suyama, "Decision making using hybrid rough sets and neural networks," *International Journal of Neural Systems*, vol. 12, no. 6, pp. 435–446, 2002.
- [16] V. L. Berardi, B. E. Patuwo, and M. Y. Hu, "A principled approach for building and evaluating neural network classification models," *Decision Support Systems*, vol. 38, no. 2, pp. 233–246, 2004.
- [17] H. Ishibuchi, K. Kwon, and H. Tanaka, "A learning algorithm of fuzzy neural networks with triangular fuzzy weights," *Fuzzy Sets and Systems*, vol. 71, no. 3, pp. 277–293, 1995.
- [18] Y. Hayashi, J. J. Buckley, and E. Czogala, "Fuzzy neural network with fuzzy signals and weights," *International Journal of Intelligent Systems*, vol. 8, no. 4, pp. 527–537, 1993.
- [19] C. C. Hu, D. Y. Wang, K. J. Wang, and J. Cable, "Seasonal roadway classification based on neural network," *Journal of South China University of Technology (Natural Science)*, vol. 37, no. 11, pp. 22–26, 2009.
- [20] A. Subasi, M. Yilmaz, and H. R. Ozcalik, "Classification of EMG signals using wavelet neural network," *Journal of Neuroscience Methods*, vol. 156, no. 1-2, pp. 360–367, 2006.
- [21] D. Sahoo and G. S. Dulikravich, "Evolutionary wavelet neural network for large scale function estimation in optimization," in *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pp. 658–668, Portsmouth, Va, USA, September 2006.
- [22] J. R. Zhang, J. Zhang, T. M. Lok, and M. R. Lyu, "A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1026–1037, 2007.
- [23] W. Zhang, Y. B. Shi, L. F. Zhou, and T. Lu, "Wavelet neural network classifier based on improved PSO algorithm," *Chinese Journal of Scientific Instrument*, vol. 31, no. 10, pp. 2203–2209, 2010.
- [24] A. Sengur, I. Turkoglu, and M. C. Ince, "Wavelet packet neural networks for texture classification," *Expert Systems with Applications*, vol. 32, no. 2, pp. 527–533, 2007.
- [25] Y. F. Hassan, "Rough sets for adapting wavelet neural networks as a new classifier system," *Applied Intelligence*, vol. 35, no. 2, pp. 260–268, 2011.
- [26] W. Ziarko, "The discovery, analysis and representation of data dependencies in databases," in *Knowledge Discovery in Databases*, pp. 213–228, 1993.
- [27] Y. Hassan and E. Tazaki, "Emergent rough set data analysis," in *Transactions on Rough Sets II: Rough Sets and Fuzzy Sets*, J. F. Peters, Ed., pp. 343–361, Springer, Berlin, Germany, 2005.
- [28] D. B. Chen and C. X. Zhao, "Particle swarm optimization with adaptive population size and its application," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 39–48, 2009.
- [29] K. Thangavel and A. Pethalakshmi, "Dimensionality reduction based on rough set theory: a review," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 1–12, 2009.
- [30] J. M. Wei, "Rough set based approach to selection of node," *International Journal of Computational Cognition*, vol. 1, no. 2, pp. 25–40, 2003.
- [31] E. Xu, L. S. Shao, Y. Z. Zhang, F. Yang, and H. Li, "Method of rule extraction based on rough set theory," *Chong Qing: Computer Science*, vol. 38, pp. 232–235, 2011.
- [32] L. Y. Chang, G. Y. Wang, and Y. Wu, "An approach for attribute reduction and rule generation based on rough set theory," *Bei Jing: Journal of Software*, vol. 10, pp. 1206–1211, 1999.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

