

## Research Article

# A Fuzzy Neural Network Based on Non-Euclidean Distance Clustering for Quality Index Model in Slashing Process

Yuxian Zhang,<sup>1</sup> Song Li,<sup>1</sup> Xiaoyi Qian,<sup>1</sup> and Jianhui Wang<sup>2</sup>

<sup>1</sup>School of Electrical Engineering, Shenyang University of Technology, Shenyang 110870, China

<sup>2</sup>College of Information and Science Engineering, Northeastern University, Shenyang 110003, China

Correspondence should be addressed to Yuxian Zhang; [yuxian524524@163.com](mailto:yuxian524524@163.com)

Received 27 November 2014; Revised 9 April 2015; Accepted 12 April 2015

Academic Editor: Jyh-Hong Chou

Copyright © 2015 Yuxian Zhang 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.

The quality index model in slashing process is difficult to build by reason of the outliers and noise data from original data. To the above problem, a fuzzy neural network based on non-Euclidean distance clustering is proposed in which the input space is partitioned into many local regions by the fuzzy clustering based on non-Euclidean distance so that the computation complexity is decreased, and fuzzy rule number is determined by validity function based on both the separation and the compactness among clusterings. Then, the premise parameters and consequent parameters are trained by hybrid learning algorithm. The parameters identification is realized; meanwhile the convergence condition of consequent parameters is obtained by Lyapunov function. Finally, the proposed method is applied to build the quality index model in slashing process in which the experimental data come from the actual slashing process. The experiment results show that the proposed fuzzy neural network for quality index model has lower computation complexity and faster convergence time, comparing with GP-FNN, BPNN, and RBFNN.

## 1. Introduction

The slashing process is a very important procedure in textile manufacturing processes. It can improve the ability of warp to resist the loading of weaving and improve the textile ability, so as to ensure the smooth going of weaving [1, 2]. Size add-on, size moisture regain, and elongation are very important quality indexes in slashing process in which size add-on (the amount of size added) is the key quality index which affects directly product quality in slashing process [3]. Since the slashing process employs complex physical, chemical, and thermal changes, it is difficult to build precise first-principle model that can explain why the desired quality appears in products. Hence, it is a significant task to establish an accurate quality index model and obtain true quality index.

Another option to solve this difficult situation is to use artificial intelligence technology. Nowadays various new technologies such as artificial neural networks and fuzzy system have been accepted as a potentially useful tool for modeling complex nonlinear systems [4, 5]. As a solution, artificial

intelligence technology has gained increasing popularity in various industries, like the chemical industry, bioprocess, steel industry, and so forth [6–8]. In the textile industry, most of companies have also built integrated databases to store empirical data from production process. And most popular quality index models were built on the basis of the linear statistical model. However, the applicability of these statistical models to the complicated nonlinear processes such as slashing process is limited.

Recently, some researchers have proposed new technology to solve the above problem. In [9] a neural network model for predicting initial load-extension behavior is presented in which a single hidden layer feed-forward neural network based on a back-propagation algorithm with four input neurons and one output neuron is developed to predict initial modulus in the warp and weft directions. In [10], the predictability of the warp breakage rate from a sizing yarn quality index using a feed-forward back-propagation network is investigated. An eight-quality index (size add-on, abrasion resistance, abrasion resistance irregularity, hairiness beyond

3 mm, breaking strength, strength irregularity, elongation, and elongation irregularity) and warp breakage rates are rated in controlled conditions. A good correlation between predicted and actual warp breakage rates indicates that warp breakage rates can be predicted by artificial neural networks. In [11], the performance of multilayer perceptron (MLP) and multivariate linear regression (MLR) models for predicting the hairiness of worsted-spun wool yarns from various top, yarn, and processing parameters is evaluated. In [12], a predictive model using empirical data is built by a neural network in order to describe complicated nonlinear relationship between operating parameters and quality index in slashing process. However, there are some shortages in the above methods, such as ambiguous physical interpretation for weight and number of layers and neurons in artificial neural networks.

Since Zadeh proposed the fuzzy logic theorem to describe complicated systems, it has become very popular and has been successfully used in various problems [13, 14]. More recently, fuzzy logic has been highly recommended for modeling to solve the inherent imprecision and vagueness characteristics in nonlinear system [15, 16]. On the other hand, a neural network has the ability to learn from input-output pairs, self-organize its structure, and adapt to it in an interactive manner. The integration of fuzzy inference system and neural network will make full use of their advantages and avoid disadvantages [17, 18]. The fuzzy neural network (FNN) is a multilayer feed-forward network which uses neural network learning algorithms and fuzzy reasoning to map an input space to an output space. With the ability to combine the verbal power of a fuzzy system with the numeric power of a neural system adaptive network, fuzzy neural network has been shown to be powerful in modeling numerous processes [19]. Fuzzy neural network has the advantage of allowing the extraction of fuzzy rules from numerical data or expert knowledge and adaptively constructs a rule base [20].

In this paper, a fuzzy neural network based on non-Euclidean distance for quality index model in slashing process is proposed in which the input space is partitioned into many local regions by the fuzzy  $c$ -means clustering based on non-Euclidean distance, and number of fuzzy rules and membership function are determined so that computation complexity is decreased. Then, the premise and consequent parameters of fuzzy neural network are trained by hybrid learning algorithm, and the convergence condition of consequent parameters is analyzed by Lyapunov function. The experiment shows that the proposed fuzzy neural network modeling method can restrain influence on noise data and decrease computation complexity, and the established size add-on model for slashing quality index has a faster convergence speed and higher accuracy.

The rest of the paper is organized as follows: the structure of fuzzy neural network is introduced in Section 2; the learning algorithm is proposed for fuzzy neural network in Section 3 in which structure identification and parameter estimation are introduced, respectively; a size add-on model is implemented for predicting slashing quality index in Section 4. Finally, the work is concluded in Section 5.

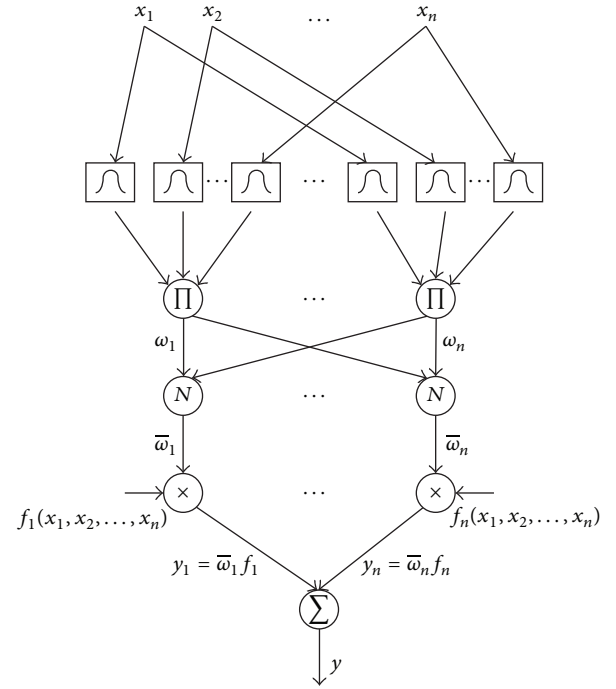


FIGURE 1: Structure of fuzzy neural networks.

## 2. Structure of Fuzzy Neural Networks

The fuzzy neural networks are composed of premise part and consequent part. The structure of fuzzy neural networks can be described as a multilayer neural network shown in Figure 1.

The first layer executes a fuzzification process, the second layer executes the fuzzy AND of the premise part of the fuzzy rules, the third layer normalizes the membership functions (MFs), the fourth layer executes the consequent part of the fuzzy rules, and finally the last layer computes the output of fuzzy system by summing up the outputs of layer four. And the node function of the layers is described as follows.

*Layer 1.* In the first layer, the node function is the fuzzy membership associated with the corresponding input

$$O_{1,i} = \mu_{A_i}(x_j), \quad (1)$$

where  $x_j$  is the inputs to node  $i$  and  $A_i$  are the linguistic labels characterized by appropriate membership functions  $\mu_{A_i}$ . Due to smoothness and concise notation, the Gaussian membership function is used:

$$\mu_{A_i}(x_j) = \exp\left(-\frac{\|x_j - c_i\|^2}{2\delta_i^2}\right), \quad (2)$$

where  $\{c_i, \delta_i\}$  are the premise parameters of fuzzy rules that change the shapes of the membership function.

*Layer 2.* In the second layer, the AND operator is applied to obtain one output that represents the result of the premise for

that rule, that is, firing strength. Firing strength means the degrees to which the premise part of a fuzzy rule is satisfied and it shapes output function for the rule. Hence the outputs  $O_{2,i}$  of this layer are the products of the corresponding degrees from Layer 1:

$$O_{2,i} = \omega_i = \prod_{j=1}^k \mu_{A_j}(x_j). \quad (3)$$

*Layer 3.* In the third layer, the aim is to calculate the ratio of the  $i$ th rule firing strength to the sum of all rules firing strength. Consequently,  $\omega_i$  is taken as the normalized firing strength:

$$O_{3,i} = \bar{\omega}_i = \frac{\omega_i}{\sum_{i=1}^k \omega_i}. \quad (4)$$

*Layer 4.* The node function of the fourth layer computes the contribution of each  $i$ th rule towards the total output and the function defined as

$$O_{4,i} = \bar{\omega}_i f_i = \bar{\omega}_i (a_0 + a_1 x_1 + \cdots + a_k x_k), \quad (5)$$

where  $\bar{\omega}_i$  is the  $i$ th node output from the previous layer.

For  $\{a_0, a_1, \dots, a_k\}$ , they are the coefficients of this linear combination that are referred to as consequent parameters.

*Layer 5.* In the fifth layer, the node computes the overall output by summing all the input signals. Accordingly, the defuzzification process transforms each rules fuzzy result into an output in this layer:

$$O_5 = \sum_{i=1}^k \bar{\omega}_i f_i = \frac{\sum_{i=1}^k \omega_i f_i}{\sum_{i=1}^k \omega_i}. \quad (6)$$

The fuzzy neural network consists of totally five layers to implement different node functions to learn and tune parameters using a hybrid learning mode. So our goal is to train adaptive networks to be able to approximate unknown functions given by training data and find the precise value of the above parameters.

### 3. Learning Algorithm for Fuzzy Neural Networks

The learning algorithm for the fuzzy neural networks employs two parts: structure identification and parameter estimation. In structure identification part, input space is partitioned into many local regions in order to reduce fuzzy rules and compact network structure. In parameter estimation part, the parameters of fuzzy neural networks are determined and the convergence is analyzed.

*3.1. Structure Identification.* In structure identification part, in order to obtain number of fuzzy rules, the fuzzy neural network carries out input space partitioning that splits the input space into many local regions. The fuzzy  $c$ -means clustering algorithm is usually used to partition input space in which a number of fuzzy rules are determined. However, there

are deficiencies in the traditional fuzzy  $c$ -means clustering algorithm [21–23]: on the one hand, the noise and outliers data in data set affect the clustering result; on the other hand, improper clustering number cannot expose truly clustering structure. Hence, some improvements are proposed in this paper:

- (i) For the noise and outliers data, a fuzzy  $c$ -means clustering based on non-Euclidean distance is proposed for partitioning input space and improving clustering robustness.
- (ii) For the optimal clustering number, the validity function based on improved partition coefficient is built to obtain optimal clustering number in which the separation and compactness are considered.

*3.1.1. Structure Identification Using Fuzzy Clustering Based on Non-Euclidean Distance.* In fuzzy  $c$ -means clustering algorithm, the objective function is

$$J = \sum_{i=1}^k \sum_{j=1}^n (\mu_{ij})^m \|x_j - c_i\|^2, \quad (7)$$

where  $k$  is the number of cluster centers,  $n$  is the number of data points,  $c_i$  is the  $i$ th cluster center,  $x_j$  is the  $j$ th data,  $m$  is the weight factor, and  $\mu_{ij}$  is the membership functions of fuzzy sets in the interval  $[0, 1]$ .

The Euclidean norm is used as a metric in fuzzy  $c$ -means clustering. However, the parameter estimate resulting from a clustering objective function based on this Euclidean metric may not be robust in a noisy environment because every data point is given equal weight even though a data point may be far away from other data points.

An ideal clustering method should be robust and able to tolerate these noisy environments. If we can give a small weight to those noisy points or outliers and a large weight to those compact points in the data set, then the weighted sum of  $x_j$  will be more robust.

In this paper, we proposed a new fuzzy  $c$ -means clustering algorithm in which a new non-Euclidean metric alternates the traditional Euclidean distance metric.

We replace the Euclidean norm with a new metric in fuzzy  $c$ -means clustering. The new metric is

$$\rho(x, y) = 1 - \exp(-\alpha \|x - y\|^2), \quad (8)$$

where  $\alpha$  is a coefficient of data dispersity and  $\|x - y\|$  is a Euclidean distance.

The data dispersity describes disperse degree of data point in data set. In a data set with the same number of data points, the disperse data points occupy larger space, and the compact data points have smaller space relatively. The data dispersity is defined as

$$\alpha = \frac{1}{n^{1/\beta}} \prod_{i=1}^p h_i, \quad (9)$$

where  $n$  is the number of data points,  $h_i = [1 - \exp(-L_i)]$ ,  $L_i$ ,  $i = 1, \dots, p$ , is the length of each dimension,  $p$  is the dimension of data set, and  $\beta$  is a positive integer.

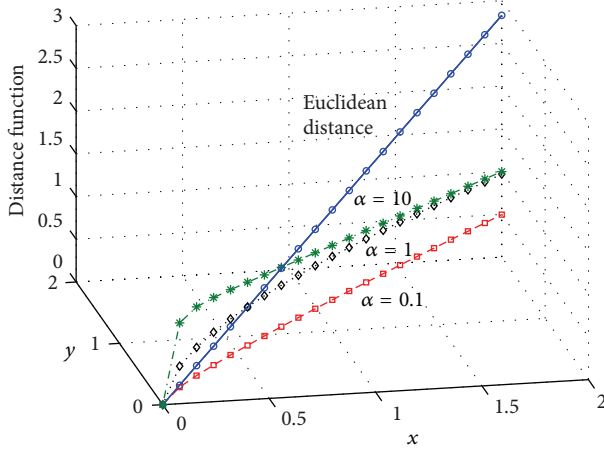


FIGURE 2: Contrast curve between Euclidean distance and non-Euclidean distance.

Note that the new metric  $\rho(x, y)$  is a monotone increasing function of  $\|x - y\|$ . And the new metric  $\rho(x, y)$  satisfies the following three conditions:

- (i)  $\rho(x, y) > 0 \quad \forall x \neq y, \rho(x, x) = 0$ ,
- (ii)  $\rho(x, y) = \rho(y, x)$ ,
- (iii)  $\rho(x, y) \leq \rho(x, z) + \rho(z, y) \quad \forall z$ .

Using the new metric, we propose a new fuzzy  $c$ -means clustering objective function as

$$J = \sum_{i=1}^k \sum_{j=1}^n (\mu_{ij})^m \left\{ 1 - \exp(-\alpha \|x_j - c_i\|^2) \right\}, \quad (10)$$

where  $k$  is the number of cluster centers,  $n$  is the number of data points,  $c_i$  is the  $i$ th cluster center,  $x_j$  is the  $j$ th element of dataset, and  $\alpha$  is a coefficient of data disperse level.

We plot the distance metric on both the Euclidean norm and the proposed non-Euclidean distance metric (8) with different  $\alpha$ , shown in Figure 2. We see that the new metric is bounded and monotone increasing with distance measure zero as  $\|x_j - c_i\| = 0$  and distance measure one as  $\|x_j - c_i\|$  tends to infinite. If  $\|x_j - c_i\|$  is larger than a level (i.e.,  $x_j$  keeps away from  $c_i$ ) then the distance measure will be close to its maximum value and give a small weight to  $x_j$ . However, the distance measure with the Euclidean norm presents a straight line. According to the analysis in Figure 2, the Euclidean norm is sensitive to noise or outliers point data and the proposed non-Euclidean norm is more robust than the common used Euclidean norm.

**3.1.2. Validity Function for Optimal Clustering Number.** In fuzzy  $c$ -means clustering, Bezdek proposed two cluster validity indexes, the partition coefficient (PC) and partition entropy (PE) [24, 25]. Although these validity indexes are widely cited in the literature, the major drawback is that they use only the fuzzy membership degrees for each cluster without considering the data structure of the clusters [26, 27]. To overcome this disadvantage, we present a novel validity

function based on improved partition coefficient for the optimal clustering number. It evaluates fuzzy  $c$ -partitions by exploiting the concepts of compactness and separation.

The proposed validity function is combined on the basis of partition coefficient method and introduces the exponential function to suppress the effect of noise or outliers point data and therefore takes into account the geometrical properties of the input data and improves the accuracy of clustering partition.

The proposed validity function is

$$V_{uv} = \frac{1}{n} \sum_{i=1}^k \sum_{j=1}^n u_{ij}^m + \frac{1}{n} \sum_{i=1}^k \sum_{j=1}^n u_{ij}^m \exp\left(-\frac{\|x_j - c_i\|^2}{\varepsilon}\right), \quad (11)$$

where  $\varepsilon = (1/n) \sum_{j=1}^n \|x_j - \bar{x}\|^2$  and  $\bar{x} = (1/n) \sum_{j=1}^n x_j$ .

The first item in validity function  $V_{uv}$  is partition coefficient which describes intradata compactness, and the second item describes the product between interdata separation and the intradata compactness. In the validity function  $V_{uv}$ , the bigger value indicates better cluster partition. It is worth noting that the second item in validity function introduces an exponential function  $\exp(-\|x_j - c_i\|^2/\varepsilon)$  to weaken the influence on noise or outliers data.

Now we have the exact procedure of structure identification as follows.

**Step 1.** Initialize clustering number  $c_{\min} = 2$ ,  $c_{\max} = \sqrt{n}$ , give weight factor  $m$ , and set iterative error  $\varepsilon$  and maximum iterations  $T$ .

**Step 2.** Choose initial cluster centers  $c^t = \{c_1^t, c_2^t, \dots, c_k^t\}$ .

**Step 3.** According to (9), compute data disperse level  $\alpha$ .

**Step 4.** Compute the new metric  $\rho(x, y)$  and clustering objective function  $J$  and update new cluster centers  $c^{t+1} = \{c_1^{t+1}, c_2^{t+1}, \dots, c_k^{t+1}\}$ .

**Step 5.** Compute the validity function  $V_{uv}$ .

**Step 6.** If  $c = c_{\max}$ , then the clustering procedure is terminated; otherwise go to Step 2 until achieving the maximum iterations  $T$ .

**Step 7.** Choose the optimal clustering number according to the maximum value of validity function  $V_{uv}$  and then output corresponding clustering centers.

After obtaining cluster centers, the number of fuzzy rules is determined in accordance with the number of cluster centers, and the shape of membership function in domain is confirmed.

### 3.2. Parameters Estimation

**3.2.1. Hybrid Learning Algorithm.** The fuzzy neural network applies a hybrid learning algorithm, the gradient descent, and the least-squares method, to update parameters. We employ the gradient descent to tune premise nonlinear

parameters  $\{c_i, \delta_i\}$ , while the least-squares method is used to identify consequent parameters  $\{a_0, a_1, \dots, a_k\}$ . The learning procedure has two steps: In the first step, the least-squares method is used to identify the consequent parameters, while the premise parameters are assumed to be fixed for the current cycle through the training set. Then, the error signals propagate backwards. Gradient descent method is used to update the premise parameters, through minimizing the overall quadratic cost function, while the consequent parameters remain fixed.

When the premise parameters are fixed, the consequent parameters are expressed as

$$\begin{aligned} O_5 = Y &= \sum_{i=1}^k \omega_i f_i = \sum_{i=1}^k \omega_i (a_0 + a_1 x_1 + \dots + a_k x_k) \\ &= AX, \end{aligned} \quad (12)$$

where  $A = [\omega_i \ \omega_i x_i]$ ,  $X = [a_0 \ a_i]^T$ ,  $i = 1, \dots, k$ , and  $Y$  is the output of fuzzy neural networks.

The least-squares method is used to identify the consequent parameters:

$$X = (A^T A)^{-1} A^T Y. \quad (13)$$

Then the gradient descent method is used to update the premise parameters  $\{c_i, \delta_i\}$ , through minimizing the overall quadratic cost function, while the consequent parameters remain fixed. The premise parameters are updated as follows:

$$c_i = c_i - \eta \left( \frac{\partial e}{\partial c_i} \right), \quad (14)$$

$$\delta_i = \delta_i - \eta \left( \frac{\partial e}{\partial \delta_i} \right), \quad (15)$$

where  $e$  is the error between the desired output and the actual output and  $\eta$  is the learning rate,  $0 < \eta < 1$ .

The procedure of hybrid learning is as follows.

*Step 1.* Set error and max training times  $T$ .

*Step 2.* Fix the premise parameters  $\{c_i, \delta_i\}$ , using the least-squares method to identify the consequent parameters  $\{a_0, a_1, \dots, a_k\}$ .

*Step 3.* Fix the consequent parameters, using the gradient descent method for training the premise parameters  $\{c_i, \delta_i\}$ :

$$\begin{aligned} c_i &= c_i - \eta \left( \frac{\partial E}{\partial c_i} \right), \\ \delta_i &= \delta_i - \eta \left( \frac{\partial E}{\partial \delta_i} \right). \end{aligned} \quad (16)$$

*Step 4.* Repeat Steps 2 and 3, until the error is satisfied or the max training times  $T$  are reached.

**3.2.2. Convergence Analysis.** To ensure the convergence of consequent parameters, we employ a discrete Lyapunov function. Lyapunov function is defined as

$$V(k) = \frac{1}{2} e^2(k), \quad (17)$$

where  $e(k) = \hat{Y}(k) - Y(k)$  is error between the desired output and the actual output.

Then the deviation  $\Delta V(k)$  is

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) = \frac{1}{2} [e^2(k+1) - e^2(k)] \\ &= \Delta e(k) \left[ e(k) + \frac{1}{2} \Delta e(k) \right]. \end{aligned} \quad (18)$$

The deviation  $\Delta e(k)$  is

$$\begin{aligned} \Delta e(k) &= e(k+1) - e(k) = \left[ \frac{\partial e(k)}{\partial W} \right]^T \Delta W \\ &= \left[ \frac{\partial e(k)}{\partial c} \quad \frac{\partial e(k)}{\partial \delta} \right] \begin{bmatrix} \Delta c \\ \Delta \delta \end{bmatrix}, \end{aligned} \quad (19)$$

where  $[\partial e(k)/\partial W]^T = [\partial e(k)/\partial c \ \partial e(k)/\partial \delta]^T$  and  $\Delta W = [\Delta c \ \Delta \delta]$ .

And  $\Delta W = -\eta e(k)(\partial e(k)/\partial W) = \eta e(k)(\partial Y(k)/\partial W)$ .

Taking  $\Delta W$  into  $\Delta V(k)$ , we can derive

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) = \frac{1}{2} [e^2(k+1) - e^2(k)] \\ &= \Delta e(k) \left[ e(k) + \frac{1}{2} \Delta e(k) \right] = \left[ \frac{\partial e(k)}{\partial W} \right]^T \eta e(k) \\ &\quad \cdot \frac{\partial Y(k)}{\partial W} \cdot \left\{ e(k) + \frac{1}{2} \left[ \frac{\partial e(k)}{\partial W} \right]^T \eta e(k) \frac{\partial Y(k)}{\partial W} \right\} \\ &= - \left[ \frac{\partial Y(k)}{\partial W} \right]^T \eta e(k) \frac{\partial Y(k)}{\partial W} \\ &\quad \cdot \left\{ e(k) - \frac{1}{2} \left[ \frac{\partial Y(k)}{\partial W} \right]^T \eta e(k) \frac{\partial Y(k)}{\partial W} \right\} = -e^2(k) \\ &\quad \cdot \sum \left\{ \frac{1}{2} \eta \left( \frac{\partial Y(k)}{\partial W} \right)^2 \cdot \left[ 2 - \eta \left( \frac{\partial Y(k)}{\partial W} \right)^2 \right] \right\}. \end{aligned} \quad (20)$$

According to Lyapunov stability theorem, if  $\Delta V(k) < 0$ , then the system is stable. When the equation  $2 - \eta(\partial Y(k)/\partial W)^2 > 0$  is satisfied, then  $\Delta V(k) < 0$ . Hence, we can obtain convergence condition for the consequent parameters; that is,

$$\eta < \frac{2}{(\partial Y(k)/\partial W)^2}. \quad (21)$$

Let  $P_{\max} = [\max_k |\partial Y(k)/\partial c| \ \max_k |\partial Y(k)/\partial \delta|]^T$ ; then the convergence condition for consequent parameters in fuzzy neural network satisfies

$$0 < \eta < \frac{2}{(P_{\max})^2}. \quad (22)$$



#### 4. Experiments and Result Analysis

In this section, we firstly identify a nonlinear system for verifying the effectiveness of the proposed fuzzy neural network. Then, the fuzzy neural network is applied for modeling quality index in slashing process, and the accuracy and convergence time are compared with fuzzy neural network with grip partition, BP neural network, and RBF neural network.

**4.1. A Nonlinear System.** For a nonlinear system

$$y(k) = \frac{y(k-1)}{1 + y(k-1)^2} + \mu(k)^3, \quad (23)$$

where

$$\mu(k) = \begin{cases} \text{rands}(1, 1) & k \leq 100 \\ \sin\left(\frac{\pi k}{45}\right) & k > 100. \end{cases} \quad (24)$$

In this subsection, 290 samples' data are used to identify the nonlinear system, in which 200 samples' data are training data and 90 samples' data are testing data. In structure identification part, the fuzzy clustering proposed in Section 3.1 is used to partition input space. The cluster validity function with different clustering number is shown in Figure 3. And we obtain that the optimum clustering number is three, and then three fuzzy rules are generated.

In parameter identification part, the premise parameter and consequent parameter are trained by hybrid learning algorithm in which the premise parameter is trained by gradient descent and the consequent parameter is tuned by least-squares method. On the basis of convergence condition, the learning rate  $\eta$  is 0.015. In the training phase, the convergence time is 2.543 s.

After the training phase, 90 samples' data are used to verify the performance of model. Figure 4 shows the comparison between model output and the actual output in which maximum relative error is 1.1725, maximum absolute error is 0.0401, and root-mean square error is 0.0137. Figure 5 shows the relative error of model. Hence, the model is valid and has a high accuracy.

**4.2. Quality Index Model in Slashing Process.** In this subsection, the proposed fuzzy neural network based on non-Euclidean distance clustering is applied in size add-on model for predicting slashing quality index.

The size add-on is the most important quality index in slashing process. In actual slashing process, many operating parameters affect add-on in slashing process, such as size concentration, size box temperature, speed, nip pressure, raw material and structure of warp, and surface characteristics of squeeze rollers. Among these operating parameters, the major operating variables are size concentration, size temperature, low nip pressure, high nip pressure, and sizing speed. Hence, these major operating variables are selected as input of fuzzy neural networks and size add-on is the output.

In this experiment, the original data from experimental prototype should be preprocessed in order that the wrong or

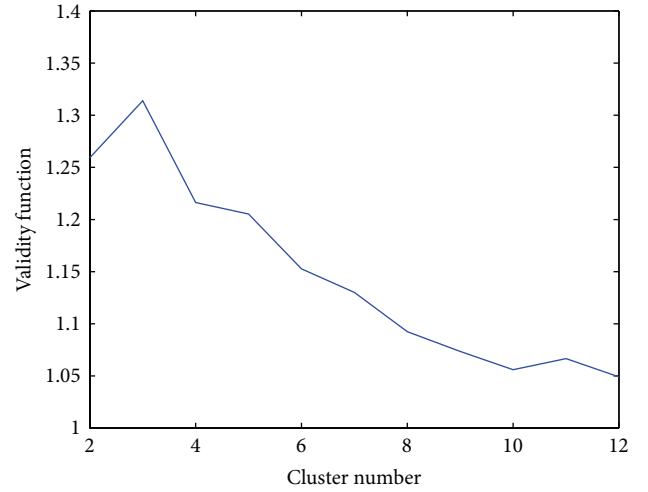


FIGURE 3: Contrast curves for difference clustering number.

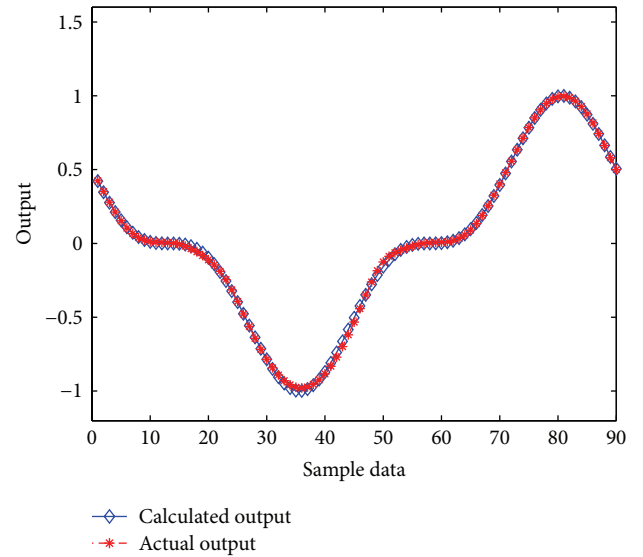


FIGURE 4: Contrast curve for model output.

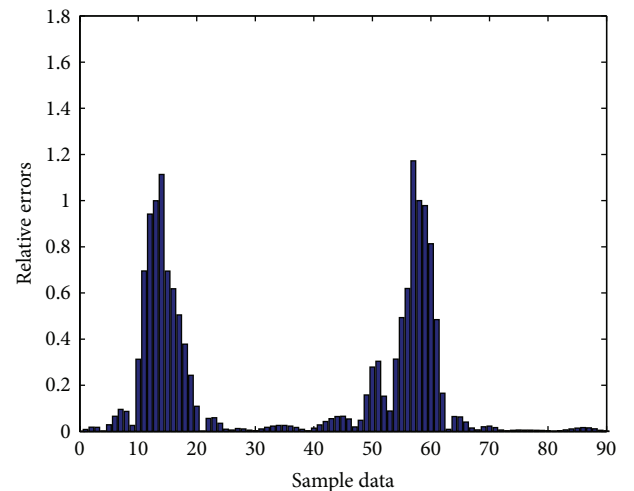


FIGURE 5: Relative error of model.

TABLE 1: Every operating variable variation.

Operating variable	Minimum	Maximum	Change rate
Size concentration (%)	9	11	22.22%
Temperature (°C)	89	93	4.49%
Speed (m/min)	32	54	68.75%
Low nip press (kN)	11	12	9.09%
High nip press (kN)	13	21	61.54%

uncorrelated information in original data is removed. In this paper, we adopt the following methods to deal with wrong or uncorrelated data:

- (i) According to technological requirements, the sample data on operating variables must be limited. For example, the nip pressure should be limited when it exceeds threshold value.
- (ii) The wrong data will be rejected. For example, when the temperature of size box is negative, it should be rejected.

After data preprocessing procedure, 240 samples' data are used to build size add-on model. However, the original data are dispersed and irregular because of complex physics, chemistry, and thermodynamic process in slashing process. The variation range of size concentration, size temperature, low nip pressure, high nip pressure, and sizing speed is shown in Table 1. In Table 1, the change rate describes fluctuation of operating variables:

$$\text{Change rate (\%)} = \frac{OV_{\max} - OV_{\min}}{OV_{\min}} \times 100\%, \quad (25)$$

where  $OV_{\max}$  is the maximum value of operating variable and  $OV_{\min}$  is the minimum value of operating variable.

From Table 1, the fluctuation of operating variables is observed in which the variation of high nip pressure and speed is particularly obvious.

On the basis of the above analysis, the proposed fuzzy neural network is used to build size add-on model. The fuzzy  $c$ -means clustering based on non-Euclidean distance is used to partition input space and the optimum clustering number can be determined by the validity function based on improved partition coefficient. Table 2 lists the clustering centers and Figure 6 shows the validity function corresponding to different clustering numbers.

After partitioning input space, 13 clustering centers are obtained. The obtained fuzzy rule is as follows:

If (Concentration is in 1 cluster  $i$ ) and (Temperature is in 2 cluster  $i$ ) and (Speed is in 3 cluster  $i$ ) and (Low press is in 4 cluster  $i$ ) and (High press is in 5 cluster  $i$ ) then (Add-on is out 1 cluster  $i$ ),  $i = 1, 2, \dots, 13$ .

Next, the hybrid learning algorithm is adopted in training phase. The gradient descent is used to train premise parameter of fuzzy neural networks. And the least-squares method is used to obtain consequent parameters. On the basis of convergence condition, the learning rate selects

TABLE 2: The clustering centers.

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$
1	11.24	90.00	47.58	12.02	16.96
2	9.77	93.03	48.09	11.90	15.11
3	10.25	88.92	52.42	11.96	19.03
4	11.46	89.96	42.99	10.87	19.97
5	9.67	92.01	41.42	11.32	14.99
6	10.76	90.06	31.98	11.00	18.02
7	9.40	91.74	53.49	10.94	17.82
8	10.24	98.12	45.01	10.94	20.96
9	9.72	93.06	45.00	10.99	14.92
10	10.00	89.99	43.45	11.12	16.86
11	9.66	91.99	33.58	11.11	12.92
12	8.60	90.00	42.97	11.15	17.54
13	9.85	92.26	33.80	10.93	13.07

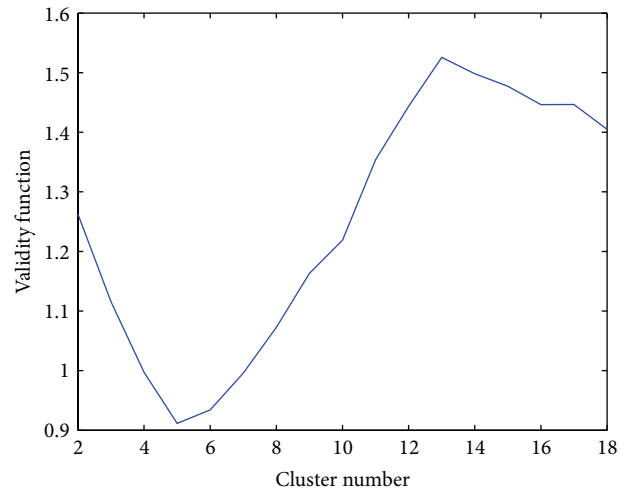


FIGURE 6: Variation of validity function corresponding to different clustering number.

$\eta = 0.01$ . By training, the training error decreased gradually, and training accuracy is achieved ultimately. Finally, the premise parameters and consequent parameters of fuzzy neural networks are determined.

In order to verify the performance of difference models, we use the 10-fold cross validation to compare the performance of difference models. The data set is divided into 10 subsets. Each time, one of the 10 subsets is used as the test set and the other 9 subsets are put together to form a training set. Then the average error across all 10 trials is computed. The proposed fuzzy neural network based on non-Euclidean distance clustering (nED-FNN) is compared with fuzzy neural network with grip partition (GP-FNN), BP neural network (BPNN), and RBF neural network (RBFNN). And the convergence time, max relative error (MRE), max absolute error (MAE), and root-mean square error (RMSE) are compared, respectively. The above models are running in MATLAB 2008a on computer with Intel i7-2637 M 1.7 GHz CPU and 4 G memory.

TABLE 3: Performance contrast with difference models.

	nED-FNN	GP-FNN		BPNN	RBFNN
		Membership = 2	Membership = 3		
Converge time	52.15 s	19.24 s	539.60 s	28.27 s	3.84 s
Max relative error (MRE)	0.0086	0.0459	0.0216	0.0229	0.0315
Max absolute error (MAE)	0.0904	0.5122	0.2346	0.2388	0.3569
Root-mean square error (RMSE)	0.0222	0.1886	0.0548	0.0779	0.1220

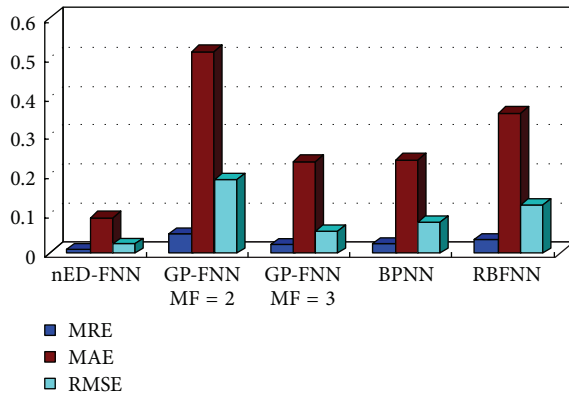


FIGURE 7: Errors contrast with different models.

In training phase, the training steps are 150 for each method. In nED-FNN, the membership function is Gaussian, convergence time is 52.15 s in training phase, and root-mean square error is 0.0222. In GP-FNN, the numbers of membership functions are 2 and 3, respectively, the input type of membership function is Gaussian, convergence times are 19.24 s and 539.60 s, respectively, in training phase, and root-mean square errors are 0.1886 and 0.0548. In BPNN, the nodes of hidden layer are 50, convergence time is 28.27 s in training phase, and root-mean square error is 0.0779. In RBFNN, the nodes of hidden layer are 50, convergence time is 3.84 s in training phase, and root-mean square error is 0.1220. The performance contrast with difference models is shown in Table 3.

The comparison analysis in Table 3 shows that the convergence speed is slow in GP-FNN method, and the convergence time is longer than 500 s when the number of membership functions is more than 2. In contrast with GP-FNN, BPNN, and RBFNN, the proposed nED-FNN is not only high accuracy but also low computational complexity and short convergence time among the above modeling methods. Figure 7 shows errors of different models.

After comparison analysis, 60 samples' data are used to build size add-on model. Figure 8 shows contrast curve between the output of size add-on model and actual samples data.

## 5. Conclusions

In this paper, we proposed a novel fuzzy neural network based on non-Euclidean distance clustering. A non-Euclidean distance clustering is used to partition input space in order to

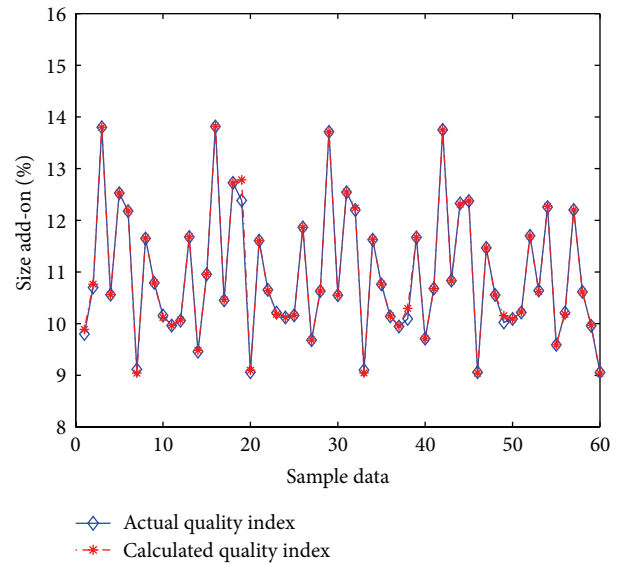


FIGURE 8: Contrast curve for size add-on model.

reduce influence on noise and outlier data and computational complexity. Then, the premise parameters and consequent parameters of fuzzy neural network are trained by hybrid learning algorithm, and the convergence condition of consequent parameters in fuzzy neural network is obtained by the discrete Lyapunov function. The proposed fuzzy neural network is used to build size add-on model in slashing process. The experiment results show that the proposed fuzzy neural network for size add-on has lower computation complexity and faster convergence time, comparing with GP-FNN, BPNN, and RBFNN.

## Conflict of Interests

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

## Acknowledgment

This research is supported by the Natural Science Foundation of China under Grant 61102124.

## References

- [1] S. Kim and G. J. Vachtsevanos, "An intelligent approach to integration and control of textile processes," *Information Sciences*, vol. 123, no. 3, pp. 181–199, 2000.



- [2] G. J. Vachtsevanos, J. L. Dorrity, A. Kumar, and S. Kim, "Advanced application of statistical and fuzzy control to textile processes," *IEEE Transactions on Industry Applications*, vol. 30, no. 3, pp. 510–516, 1994.
- [3] Y. X. Zhang, M. Liu, and J. H. Wang, "A hybrid model for size add-on in slashing processes," in *Proceeding of the Chinese Control and Decision Conference (CCDC '11)*, pp. 1791–1795, May 2011.
- [4] L.-X. Wang, "The WM method completed: a flexible fuzzy system approach to data mining," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 768–782, 2003.
- [5] H. S. Yu, J. Z. Peng, and Y. D. Tang, "Identification of nonlinear dynamic systems using Hammerstein-type neural network," *Mathematical Problems in Engineering*, vol. 2014, Article ID 959507, 9 pages, 2014.
- [6] P. P. Angelov and D. P. Filev, "An approach to online identification of Takagi-Sugeno fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 34, no. 1, pp. 484–498, 2004.
- [7] M.-Y. Chen, "A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering," *Information Sciences*, vol. 220, pp. 180–195, 2013.
- [8] S. H. Park, D. S. Kim, J. H. Kim, and M. G. Na, "Prediction of the reactor vessel water level using fuzzy neural networks in severe accident circumstances of npps," *Nuclear Engineering and Technology*, vol. 46, no. 3, pp. 373–380, 2014.
- [9] M. Hadizadeh, A. A. A. Jeddi, and M. A. Tehran, "The prediction of initial load-extension behavior of woven fabrics using artificial neural network," *Textile Research Journal*, vol. 79, no. 17, pp. 1599–1609, 2009.
- [10] G. F. Yao, J. S. Guo, and Y. Y. Zhou, "Predicting the warp breakage rate in weaving by neural network techniques," *Textile Research Journal*, vol. 75, no. 3, pp. 274–278, 2005.
- [11] Z. Khan, A. E. K. Lim, L. Wang, X. Wang, and R. Beltran, "An artificial neural network-based hairiness prediction model for worsted wool yarns," *Textile Research Journal*, vol. 79, no. 8, pp. 714–720, 2009.
- [12] Y. Zhang and M. Liu, "An operating parameters setting method using empirical data for slashing process," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 5, pp. 2013–2023, 2010.
- [13] W. Wang, F. Ismail, and F. Golnaraghi, "A neuro-fuzzy approach to gear system monitoring," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 5, pp. 710–723, 2004.
- [14] X.-H. Yuan, H.-X. Li, and X. Yang, "Fuzzy system and fuzzy inference modeling method based on fuzzy transformation," *Acta Electronica Sinica*, vol. 41, no. 4, pp. 675–680, 2013.
- [15] Z. Deng, Y. Jiang, K.-S. Choi, F.-L. Chung, and S. Wang, "Knowledge-leverage-based TSK fuzzy system modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1200–1212, 2013.
- [16] C.-F. Juang and C.-Y. Chen, "Data-driven interval type-2 neural fuzzy system with high learning accuracy and improved model interpretability," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1781–1795, 2013.
- [17] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [18] R. Gogoi and K. Kumar Sarma, "ANFIS based information extraction using k-means clustering for application in satellite images," *International Journal of Computer Applications*, vol. 50, no. 7, pp. 13–18, 2012.
- [19] R. H. Abiyev and O. Kaynak, "Type 2 fuzzy neural structure for identification and control of time-varying plants," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 12, pp. 4147–4159, 2010.
- [20] R. H. Abiyev, O. Kaynak, T. Alshamleh, and F. Mamedov, "A type-2 neuro-fuzzy system based on clustering and gradient techniques applied to system identification and channel equalization," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 1396–1406, 2011.
- [21] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
- [22] C.-C. Hung, S. Kulkarni, and B.-C. Kuo, "A new weighted fuzzy C-means clustering algorithm for remotely sensed image classification," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 543–553, 2011.
- [23] T. Chaira, "A novel intuitionistic fuzzy c means clustering algorithm and its application to medical images," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1711–1717, 2011.
- [24] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 370–379, 1995.
- [25] J. M. Chen, "The improved partition entropy coefficient," in *Multimedia and Signal Processing*, vol. 346 of *Communications in Computer and Information*, pp. 1–7, Springer, Berlin, Germany, 2012.
- [26] J.-L. Fan and C.-M. Wu, "Clustering validity function based on partition coefficient combined with total variation," *Acta Electronica Sinica*, vol. 29, no. 11, pp. 1561–1563, 2001.
- [27] D. Chen, X. Li, D.-W. Cui, and R. Fei, "Cluster validity function based on fuzzy degree," *Pattern Recognition and Artificial Intelligence*, vol. 21, no. 1, pp. 34–41, 2008.

