Soft Computing Techniques for the Prediction of Hybrid Composites

C. Kavitha

Abstract: Soft computing techniques such as Artificial Neural Networks and Fuzzy Logic are widely used in applications of manufacturing technology. Surface roughness plays a vital role in quality of the product using machining parameters. Soft computing techniques are applied to predict the surface roughness in an economical manner. In this paper, prediction of surface roughness is evaluated using ANFIS [Adaptive Neuro-Fuzzy Inference System] methodology for the cutting parameters of end-milling process for machining the halloysite nanotubes (HNTs) with aluminium reinforced epoxy hybrid composite material. Experimental datas are used to analyse the relationship between the input parameter such as depth of cut (d), cutting speed (S), feed-rate (f) and output parameters as surface roughness. Datas are classified into training and testing with different types of membership functions. The observed results accurately predict the output which was not used in training and it is almost very close to the actual output obtained in the experimental work. Moreover it was found that gbellmf is helpful for better prediction with minimum error.

Keywords : ANFIS, Halloysite nanotubes, Surface roughness, depth of cut, cutting speed and feed rate.

I. INTRODUCTION

Surface roughness plays a vital role in manufacturing field. It influences the mechanical properties, cost and quality of a product. Research activities are involved to decrease the surface roughness by varying many types of hybrid composition of materials. There are many statistical tools to find the influence parameter, optimizing the parameter and tools to predict the output without doing many experimental datas.

I.El-Sonbaty et.al obtained Polymer composite products by primary manufacturing process and secondary manufacturing processing. Secondary process involves drilling and saw cutting. Drilling is employed to make bolted or riveted assemblies and defects like delamination, crack, undesired hole surface roughness related to tool wear encountered. Sheet molding compound composite was drilled under different cutting speed, feed and drill point angles. R.C.L.Dutra et.al investigate the influence of parameters such as cutting speed, feed, drill size, fiber volume fraction on the thrust force, torque and surface roughness in drilling processes of fiber-reinforced composite materials. They found that Drill diameter combined with feed has a significant effect on surface roughness. Moon II Kim et.al proposed Hybrid composites containing Polypropylene fiber and mercapto-modified polypropylene blend fibers display higher impact strength than plain carbon fiber composites but the performance is lower than plain Polypropylene fiber and mercapto-modified polypropylene epoxy composite. Minggang dong et.al found that Zirconium oxide-impregnated halloysite nanotubes were added to epoxy resins to obtain epoxy composites with improved mechanical, thermal and flexural properties. Alper Uysal et.al improve the prediction of surface roughness in end milling process by ANFIS and leave-one-out cross-validation (LOO-CV) approach. In this paper, cutting parameters of end-milling process for machining the halloysite nanotubes (HNTs) with aluminium reinforced epoxy hybrid composite material is modeled using adaptive neuro fuzzy inference system.

II. METHODOLOGY - ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) METHOD

ANFIS methodology was proposed by Jang in 1993. Geoffrey Boothroyd et.al proposed ANFIS as a hybrid combination of two methods such as Adaptive Neural Network (ANN) and Takagi-Sugeno Fuzzy Inference system. Fuzzy logic is implemented in Neural network to construct an input-output mapping for the associated parameters through input, output membership functions based on human knowledge in the form of Fuzzy Inference System (FIS) followed by IF-THEN rules to determine the optimal distribution of membership functions. Neural networks are used to train and test the input-output data by back-propagation, gradient descent, least squares method by reducing the sum of the squared errors to predicate the output. Architecture of the ANFIS structure and schematic structure of ANFIS network in each layer is represented in Fig. 1 and Fig. 2.

Fig.1 Architecture of ANFIS structure
Fig. 2 Schematic structure of ANFIS network

P.M. Pradhan et al investigated as ANFIS consist of five network layers used to create fuzzy inference system. Every layer has different function such as Fuzzy layer, Product layer, Normalized layer, Defuzzify layer and Output layer. Layer consists of two different types of nodes such as squares and circles. Square nodes are adaptive nodes where the factors could be changed and the circles are fixed nodes where the factors are fixed whose parameters change during the training process. Muhammad Rizal et.al used Input nodes to represent the training values, output nodes represent predicted values and in hidden layers there are nodes functioning as membership functions and rules. The present layers’ inputs are derived from the nodes in the previous layers. The rule base of ANFIS contains fuzzy IF–THEN rules of the Sugeno type. For a first-order Sugeno fuzzy inference system, the two rules may be stated as:

Rule 1: IF x is A1 AND y is B1, THEN f is f1(x, y)
Rule 2: IF x is A2 AND y is B2, THEN f is f2(x, y),
where x and y are the inputs of ANFIS, A1 and B1 are the fuzzy sets, and f1(x, y) = p1 x + q1 y + r1, p1, q1, r1 are design parameters that are determined during the training process, f1 is a first order polynomial and represents the outputs of the first order Sugeno fuzzy inference system.

Jang J used to Characteristics of each layer in ANFIS described as below:

**Layer 1:** Fuzzy layer has adaptive nodes with node function which converts the inputs into linguistic labels to calculate membership values of the data. The output of these layer nodes is membership value of input in following structure:

\[ O_{ij} = \mu_{A_i}(x_j) \]

for i = 1, 2

\[ O_{ij} = \mu_{B_{i-3}}(x_2) \]

for i = 3, 4

where \( x_j \) and \( x_2 \) are the input to node i, \( A_i \) and \( B_{i-3} \) are fuzzy sets which contains linguistic terms such as small, large etc. \( \mu_{A_i}(x_j) \) and \( \mu_{B_{i-3}}(x_2) \) are the membership functions.

Membership functions are different types such as triangular, trapezoidal, generalized bell shaped, gaussian curve, gaussian combination, \( \Pi \)-shaped, difference between two sigmoidal and product of two sigmoidal membership functions are used to represent the linguistic terms.

**Layer 2:** Product layer has fixed nodes, it is marked by a circle and it is labeled as \( \Pi \). The output of each node function has to be multiplied by input signals from the previous layer. The nodes of this layer are called rule nodes and it computes the firing strength of the associated rule.

\[ O_{2j} = w_i = \mu_{A_i}(x_j) \mu_{B_{i-3}}(x_2) \]  

for i=1,2

**Layer 3:** Normalized layer has fixed nodes, marked by a circle and labeled by N. The \( i^{th} \) node of this layer calculates the normalized firing strength as ratio of \( i^{th} \) node firing strength to the sum of all rules firing strengths. Output of the \( i^{th} \) node of this layer can be represented as

\[ O_{3i} = \frac{w_i}{w_1 + w_2} \]  

for \( i = 1, 2 \)

**Layer 4:** Defuzzify layer has adaptive nodes, marked by a square and labeled by D. This layer computes the product of normalized firing strength and first order polynomial [sugeno model]. Takagi_sugeno type output of this layer can be represented as

\[ O_{4j} = w_i f_i = \frac{w_i}{w_1 + w_2} \left( p_1 x_1 + q_1 x_2 + r_1 \right) \]

where the evaluation of right hand side polynomials perform consequent parameters as \( \left\{ p_1, q_1, r_1 \right\} \), \( w_i \) is the normalized weighting factor of the \( i^{th} \) rule, \( f_i \) is the output of the \( i^{th} \) rule of this node.

**Layer 5:** Output layer has fixed node marked by a circle and labeled by \( \sum \). This layer computes the overall output as the summation of all incoming signals

\[ O_{5j} = \sum w_i f_i = \sum w_i \]

Hybrid learning algorithm is used to increase the convergence rate and it is used to update the premise parameters by integrating the least square and gradient descent method. To optimize the consequent parameters least square method is employed and to optimize the premise parameters gradient descent method is used Ilhan Asilturk et.al. Total output of ANFIS is determined by consequent parameters \( p, r, q \) and is given by the following equation

\[ O_{5j} = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \]

i.e., \[ O_{5j} = \frac{w_1}{w_1 + w_2} \left( p_1 x_1 + q_1 x_2 + r_1 \right) + \frac{w_2}{w_1 + w_2} \left( p_2 x_1 + q_2 x_2 + r_2 \right) \]

**III. RESULTS AND DISCUSSION - PREDICTION OF SURFACE ROUGHNESS BY ANFIS:**

ANFIS methodology is used to predict surface roughness for the cutting parameters of end-milling process for machining the halloysite nanotubes (HNTs) with aluminium reinforced epoxy hybrid composite material. In this study,
Data has been extracted from J.S.Pang et al. and shown in Table 1. Inputs are machining parameters such as depth of cut \((d)\) (mm), cutting speed \((S)\) (rpm), feed rate \((f)\) (mm/min) and the outputs are the surface roughness \((Ra)\) (µm) of the machined composite surface.

ANFIS methodology is solved by Neuro fuzzy designer in MATLAB toolbox, which applies fuzzy inference techniques to data modeling using a graphical user interface. Initially datas are trained in ANFIS using existing input / output training data set. Based on the training datas it constructs Sugeno-type fuzzy inference system (FIS) whose parameters are tuned to membership function. Membership functions associated with the input / output parameters changes through the learning process and the adjustment of these parameters is done by a gradient vector such as grid partition or sub clustering. Optimization method is used to adjust the parameters to reduce some error measure by sum of the squared difference between actual and desired outputs. Datas which are trained by FIS uses optimization method such as backpropagation alone or in combination with a least square method for membership function parameter estimation. The training process continues till the desired number of training steps (epochs) or the desired root mean squared error (RMSE) between the desired and the generated output is achieved. If the training process over then the testing data will be used in the same manner to test the generalization ability of tuned system and achieve the desire output.

Input data set for training / testing process are machining parameters such as depth of cut \((d)\) (mm), cutting speed \((S)\) (rpm) and feed rate \((f)\) (mm/min). Table 1 and Table 2 are describing parameters to reduce some error measure by sum of the squared difference between actual and desired outputs. Datas associated with the input / output parameters changes through the learning process and the adjustment of these parameters is done by a gradient vector such as grid partition or sub clustering. Optimization method is used to adjust the parameters to reduce some error measure by sum of the squared difference between actual and desired outputs. Datas which are trained by FIS uses optimization method such as backpropagation alone or in combination with a least square method for membership function parameter estimation. The training process continues till the desired number of training steps (epochs) or the desired root mean squared error (RMSE) between the desired and the generated output is achieved. If the training process over then the testing data will be used in the same manner to test the generalization ability of tuned system and achieve the desire output. 

<table>
<thead>
<tr>
<th>Record</th>
<th>Input 1 ((d))</th>
<th>Input 2 ((S))</th>
<th>Input 3 ((f))</th>
<th>Output ((Ra))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4</td>
<td>500</td>
<td>20</td>
<td>1.15</td>
</tr>
<tr>
<td>2</td>
<td>0.4</td>
<td>500</td>
<td>60</td>
<td>1.18</td>
</tr>
<tr>
<td>3</td>
<td>0.4</td>
<td>1000</td>
<td>20</td>
<td>0.96</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>1000</td>
<td>40</td>
<td>0.62</td>
</tr>
<tr>
<td>5</td>
<td>0.4</td>
<td>1500</td>
<td>40</td>
<td>0.36</td>
</tr>
<tr>
<td>6</td>
<td>0.4</td>
<td>1500</td>
<td>60</td>
<td>0.29</td>
</tr>
<tr>
<td>7</td>
<td>0.6</td>
<td>500</td>
<td>20</td>
<td>1.43</td>
</tr>
<tr>
<td>8</td>
<td>0.6</td>
<td>500</td>
<td>40</td>
<td>0.89</td>
</tr>
<tr>
<td>9</td>
<td>0.6</td>
<td>1000</td>
<td>40</td>
<td>0.86</td>
</tr>
<tr>
<td>10</td>
<td>0.6</td>
<td>1000</td>
<td>60</td>
<td>0.77</td>
</tr>
<tr>
<td>11</td>
<td>0.6</td>
<td>1500</td>
<td>20</td>
<td>0.37</td>
</tr>
<tr>
<td>12</td>
<td>0.6</td>
<td>1500</td>
<td>60</td>
<td>0.84</td>
</tr>
<tr>
<td>13</td>
<td>0.8</td>
<td>500</td>
<td>40</td>
<td>1.8</td>
</tr>
<tr>
<td>14</td>
<td>0.8</td>
<td>500</td>
<td>60</td>
<td>1.81</td>
</tr>
<tr>
<td>15</td>
<td>0.8</td>
<td>1000</td>
<td>20</td>
<td>1.15</td>
</tr>
<tr>
<td>16</td>
<td>0.8</td>
<td>1000</td>
<td>60</td>
<td>2.11</td>
</tr>
<tr>
<td>17</td>
<td>0.8</td>
<td>1500</td>
<td>40</td>
<td>1.25</td>
</tr>
<tr>
<td>18</td>
<td>0.8</td>
<td>1500</td>
<td>20</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Fig.3 ANFIS Structure based on grid partition
Table 2: Testing datasets for ANFIS model

<table>
<thead>
<tr>
<th>Record</th>
<th>Input 1 (d)</th>
<th>Input 2 (S)</th>
<th>Input 3 (f)</th>
<th>Output (Ra)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4</td>
<td>500</td>
<td>40</td>
<td>1.94</td>
</tr>
<tr>
<td>2</td>
<td>0.4</td>
<td>1000</td>
<td>60</td>
<td>0.77</td>
</tr>
<tr>
<td>3</td>
<td>0.4</td>
<td>1500</td>
<td>20</td>
<td>1.06</td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td>500</td>
<td>60</td>
<td>1.28</td>
</tr>
<tr>
<td>5</td>
<td>0.6</td>
<td>1000</td>
<td>20</td>
<td>1.1</td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td>1500</td>
<td>40</td>
<td>0.63</td>
</tr>
<tr>
<td>7</td>
<td>0.8</td>
<td>500</td>
<td>20</td>
<td>1.14</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>1000</td>
<td>40</td>
<td>1.43</td>
</tr>
<tr>
<td>9</td>
<td>0.8</td>
<td>1500</td>
<td>60</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Table 3: Comparing the result of different membership functions.

<table>
<thead>
<tr>
<th>NO</th>
<th>No. of Membership function</th>
<th>Function Type</th>
<th>Output Function</th>
<th>Error (RMSE)</th>
<th>Training Error</th>
<th>Testing Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3 3 3 Trimf</td>
<td>Constant</td>
<td>1.1547x10^{-6}</td>
<td>1.1686</td>
<td>0.000122</td>
<td>0.7779</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>6.3513x10^{-6}</td>
<td>1.1686</td>
<td>0.000122</td>
<td>0.7779</td>
</tr>
<tr>
<td>2</td>
<td>3 3 3 trapmf</td>
<td>Constant</td>
<td>1.1548x10^{-6}</td>
<td>1.1686</td>
<td>0.000122</td>
<td>0.7779</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>6.5701x10^{-6}</td>
<td>1.1686</td>
<td>0.000122</td>
<td>0.7779</td>
</tr>
<tr>
<td>3</td>
<td>3 3 3 gbellmf</td>
<td>Constant</td>
<td>1.3184x10^{-6}</td>
<td>0.88744</td>
<td>0.92096</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>0.00020792</td>
<td>0.76018</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3 3 3 Gaussmf</td>
<td>Constant</td>
<td>1.3189x10^{-6}</td>
<td>0.92096</td>
<td>0.92096</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>0.00018292</td>
<td>0.7779</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3 3 3 Gauss2mf</td>
<td>Constant</td>
<td>1.1554x10^{-6}</td>
<td>1.1675</td>
<td>0.00021836</td>
<td>1.1668</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>0.00021836</td>
<td>1.1668</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3 3 3 pimf</td>
<td>Constant</td>
<td>1.1548x10^{-6}</td>
<td>1.1686</td>
<td>0.0001229</td>
<td>1.1638</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>6.5701x10^{-6}</td>
<td>1.1686</td>
<td>0.0001229</td>
<td>1.1638</td>
</tr>
<tr>
<td>7</td>
<td>3 3 3 dsignf</td>
<td>Constant</td>
<td>1.1558x10^{-6}</td>
<td>1.1665</td>
<td>0.0001229</td>
<td>1.1638</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>0.0001983</td>
<td>1.1547</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3 3 3 psignf</td>
<td>Constant</td>
<td>1.1555x10^{-6}</td>
<td>1.167</td>
<td>0.0001983</td>
<td>1.1547</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

The ANFIS model has been attempted for predicting surface roughness of the cutting parameters of end-milling process for machining the halloysite nanotubes (HNTs) with aluminium reinforced epoxy hybrid composite material. Experimental values are compared with predicted values of ANFIS and it is found that it is accurate. ANFIS model is checked with different types of membership function with constant and linear type. In that ANFIS model with gbellmf membership function is selected as best based on the minimum RSME prediction error of about 1.3184x10^{-6}. It is found that the ANFIS method can be able to attain a better prediction model of the experimental values.

ACKNOWLEDGMENT

We thank the anonymous referees for their useful suggestions.

REFERENCES:


AUTHORS PROFILE

C. Kavitha working as Assistant Professor in Department of Mathematics, Sathyabama Institute of Science and Technology, Chennai. Field of specialization is Fuzzy optimization. Presented papers in conferences, Attended Seminars and FDP at various levels. Published many papers in National and International journals.