Fibers and Polymers 2008, Vol.9, No.2, 210-216

Modelling of Ring Yarn Unevenness by Soft Computing Approach

Abhijit Majumdar*, Mihai Ciocoiu[1], and Mirela Blaga[1]

Government College of Engineering and Textile Technology, Berhampore 742101, India
[1]*Faculty of Textile and Leather Engineering, Technical University of Iasi, Iasi 70050, Romania*
(Received July 16, 2007; Revised January 6, 2008; Accepted January 9, 2008)

Abstract: This paper demonstrates the application of two soft computing approaches namely artificial neural network (ANN) and neural-fuzzy system to forecast the unevenness of ring spun yarns. The cotton fiber properties measured by advanced fiber information system (AFIS) and yarn count have been used as inputs. The prediction accuracy of the ANN and neural-fuzzy models was compared with that of linear regression model. It was found that the prediction performance was very good for all the three models although ANN and neural-fuzzy models seem to have some edge over the linear regression model. The linguistic rules developed by the neural-fuzzy system unearth the role of input variables on the yarn unevenness.

Keywords: Artificial neural network, Fiber length, Fuzzy logic, Membership function, Regression, Short fiber content, Yarn unevenness

Introduction

Modelling of yarn properties is one of the most fascinating areas of research in the domain of textile engineering. Researchers have made relentless effort to formulate mathematical, statistical, empirical and intelligent models to predict various yarn properties. Most of these models focus on the prediction of yarn strength. Yarn unevenness, characterised by mass irregularity, is an important attribute of spun yarn quality as it influences the tensile properties of yarns as well as the appearance of fabrics. Unfortunately, there are only a few reported researches, which deal with modelling of yarn unevenness.

Martinedale [1] divided the entire unevenness variance into two components, one for random distribution of fibers and the other for the variation of fiber fineness. The second component was further explored and developed by Picard [2]. Cox and Townsend [3] have suggested that the irregularity of a yarn can be comprehensively characterized by the $V(L)$ and $B(L)$ curves which represent the dependence of nomalised variance of cross sectional areas of yarn on test length. Breny [4] demonstrated that the $V(L)$ curve can be determined from the fiber length distribution function, mean fiber diameter, dispersion of fiber diameter and yarn count and formulated some simplified expressions for $V(L)$ considering all the fiber are having equal length. Anderson and Foster [5] proposed an empirical equation to predict the unevenness of spun yarns from the unevenness, hank and number of ends of roving and draft. The equation was subsequently modified by Ratnam, Seshan and Govindarajulu [6]. They reported that the square of the fineness/maturity coefficient divided by the 50 % span length is highly correlated with spun yarn unevenness. Garde [7] found that the yarn unevenness is more intimately related with the number of thick and thin places than the number of neps. Zeidman *et al.* [8] categorised the factors influencing yarn unevenness in three groups: determinants of the number of fibers per cross-section, determinants of the local mean fiber fineness and determinants of the local mean orientation of fibers. Hunter [9] and Chasmawala *et al.* [10] developed statistical regression equations to predict the unevenness of spun yarns.

In recent years, there has been growing use of soft computing tools like artificial neural network (ANN), fuzzy logic and genetic algorithm to predict various yarn properties and to optimise process or material parameters. Rajamanickam *et al.* [11], *Guha et al.* [12] and Majumdar and Majumdar [13] have demonstrated that the prediction performance of ANN models is much better than that of classical mechanistic or regression models. Zhu and Ethridge [14] predicted the unevenness of spun yarns from advanced fiber information system (AFIS) parameters using ANN and obtained good prediction results. In a similar research, Guha [15] predicted the unevenness of ring and rotor spun yarns using ANN and found that the mean error of prediction was always less than 4 % for three different datasets. However, the major drawback of ANN model is that it acts like a 'black box' without revealing any physical information about the mechanics of the process. Fuzzy modelling technique can help ANN in a great way by discovering the linguistic rules relating the inputs and outputs. Majumdar *et al.* [16] employed the hybrid neural-fuzzy method to predict the tenacity of ring spun yarns. However, there has been no published literature, which focuses on the modelling of yarn unevenness using neural-fuzzy system.

In this research, an attempt has been made to predict the unevenness of carded ring spun yarns from the cotton fiber properties, measured by AFIS, using two soft computing approaches namely ANN and neural-fuzzy algorithm. The prediction accuracy of these models was compared with that of classical linear regression model.

*Corresponding author: abhitextile@rediffmail.com
Present address: Department of Textile Technology, Indian Institute of Technology, New Delhi 110016, India

Materials and Methods

Data Collection and Analysis

Cotton crop study results of 1997 and 1998 of International Textile Centre, Texas Tech University, USA, has been used in this research. Bales of two varieties of cotton namely Upland and Pima were tested for fiber properties (mean length, short fiber content, maturity, fineness, neps, seed coat neps and trash) using Zellweger Uster AFIS (N). Carded ring spun yarns of 16s, 22s and 30s were produced. Yarn unevenness was measured by Uster Tester III. The database was having 87 data sets of fiber and yarn properties for ring spun yarns. 72 data sets were randomly chosen for the training or model development. The remaining 15 datasets were used to validate the models. The summary statistics of fiber properties and yarn count is shown in Table 1.

Artificial Neural Network (ANN)

Artificial neural network is a potent data-modelling tool that is able to capture and represent any kind of input-output relationships. A typical multi-layer neural network is shown in Figure 1.

Here, one or more hidden layers can be sandwiched between the input and output layers. The number of hidden layers and the number of neurons per layer vary depending on the complexity of the problem. Each neuron receives a signal from the neurons of the previous layer and these signals are multiplied by separate synaptic weights (W). The weighted inputs are then summed up and passed through a transfer function (usually a sigmoid), which converts the output to a fixed range of values. The output of the transfer function is then transmitted to the neurons of next layer. Mathematically, these operations could be represented as shown below.

Table 1. Summary statistics of fiber properties and yarn count

Fiber/Yarn properties	Minimum	Maximum	Mean
Mean length (inch)	0.81	1.05	0.92
Short fiber content (%)	5.60	18.40	10.17
Maturity ratio	0.81	0.95	0.87
Fineness (millitex)	150	187	167.17
Neps (count/g)	180	545	316.40
Seed coat neps (count/g)	12	54	22.47
Trash (count/g)	30	1062	235.46
Yarn count (Ne)	30.8	15.8	23.92

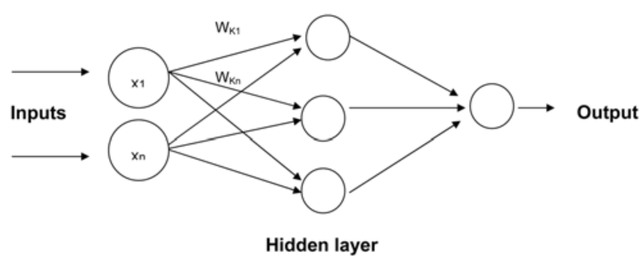

Figure 1. Structure of artificial neural network.

$$Z = \sum_{i=1}^{n} W_{ij} \cdot X_i \quad (1)$$

$$f(Z) = \frac{1}{(1+e^{-Z})} \quad (2)$$

where Z is the sum of weighted inputs; X_i is the ith input; W_{ij} is the weight connecting the ith input with the jth node and $f(Z)$ is the output of transfer function.

This process is continued and finally the output is produced at the output node. Predicted output is then compared with the desired output and an error signal is generated. The error signal is then minimised in iterative steps by adjusting the synaptic weights using a suitable training algorithm. Back-propagation algorithm developed by Rumelhart *et al.* [17] is the most popular among the existing neural network algorithms. The details of this algorithm can be found in many standard textbooks [18,19]. According to this algorithm, the error function (E) is defined as the sum of errors due to each pattern as shown in equation (3).

$$E = \sum_{p=1}^{m} E_p \quad (3)$$

where, E_p is the error associated with the pth pattern and m is total number of training patterns. The expression of E_p is shown in equation (4).

$$E_p = \frac{1}{2}\sum_{k=1}^{s} (T_k - Out_k)^2 \quad (4)$$

where T_k and Out_k denote the target output and predicted output respectively, at output node k and s is the total number of output nodes.

The corrections necessary in the synaptic weight between output and hidden layer are carried out by a delta rule as shown below.

$$\Delta W_{jk} = -\eta[\partial E/\partial W_{jk}] = \eta[(T_k - Out_k)Out_k(1 - Out_k)]out_j = \eta\delta_k out_j \quad (5)$$

where W_{jk} is the weight connecting the neuron j of hidden layer and neuron k of the output layer, ΔW_{jk}, the correction applied to W_{jk} at a particular iteration, η is a constant known as learning rate and out_j is the output of neuron j.

The weight change of the hidden layer to the input layer is calculated as shown in equation (6).

$$\Delta W_{ij} = \eta f'(net_j)X_i\sum_{k}\delta_k W_{jk} \quad (6)$$

ΔW_{ij}, the correction applied to the weight connecting input neuron i and hidden neuron j, X_i is the input received by the neuron i and $f'(net_j)$ is the derivative of net_j with respect to X_i.

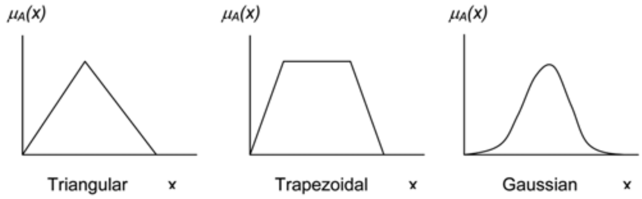

Figure 2. Different shapes of membership function graphs.

Fuzzy Set Theory and Fuzzy Logic

A fuzzy set is an extension of a classical crisp set. A fuzzy set contains elements with only partial membership ranging from 0 to 1 to define uncertainty for classes that do not have clearly defined boundaries [20,21]. If X is the universe of discourse and its elements are denoted by x, then a fuzzy set A in X is defined as a set of ordered pairs as $A = \{x, \mu_A(x)|x \in X\}$ where $m_A(x)$ is the membership function of x in A.

Once the fuzzy sets are chosen, a membership function for each set is created. A membership function is a typical curve that converts the input from 0 to 1, indicating the belongingness of the input to a fuzzy set. This step is known as "fuzzification". Membership function can have various forms, such as triangle, trapezoid, sigmoid and Gaussian as shown in Figure 2. The linguistic terms are then used to establish fuzzy rules. Fuzzy rules provide quantitative reasoning that relates input fuzzy sets with output fuzzy sets. A fuzzy rule base consists of a number of fuzzy if-then rules. For example, in the case of two-input and single-output fuzzy system, it could be expressed as follows:

If x is A_i and y is B_i then z is C_i

where x, y and z are variables representing two inputs and one output; A_i, B_i and C_i, the linguistic values of x, y and z respectively. The output of each rule is also a fuzzy set. Output fuzzy sets are then aggregated into a single fuzzy set. This step is known as "aggregation". Finally, the resulting set is resolved to a single output number by "defuzzification".

Neural-Fuzzy System and ANFIS Architecture

Neural-fuzzy system combines the fuzzification technique of fuzzy logic with the learning capability of ANN. Therefore, it possesses the merits of both the approaches and can fit the training data more accurately. Neural network technique aid the fuzzy modelling procedure to learn the information about the data set and compute the membership function parameters that best allow the associated fuzzy inference system (FIS) to track the given input-output data. ANFIS (adaptive network based fuzzy inference system) is a class of adaptive network that is functionally equivalent to FIS. Using a given input-output data, ANFIS constructs a FIS whose membership function parameters are tuned (adjusted) using either a back-propagation algorithm or a hybrid learning algorithm (a combination of back-propagation and least squares method).

Figure 3 illustrates the architecture of ANFIS having five layers assuming that there are two inputs x and y and only one output z. A common rule set with two fuzzy if-then rules is shown below.

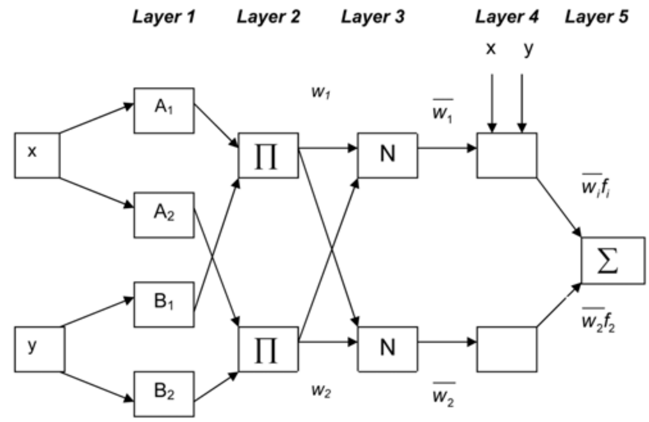

Figure 3. ANFIS architecture.

Rule 1: if x is A_1 and y is B_1 then $f_1 = p_1x + q_1y + r_1$

Rule 2: if x is A_2 and y is B_2 than $f_2 = p_2x + q_2y + r_2$

Layer 1: Every node in this layer is an adaptive node with a node function as shown below.

$$O_{1,i} = \mu_{Ai}(x) \text{ for } i = 1, 2 \text{ or } O_{1,i} = \mu_{B_{i-2}}(y) \text{ for } i = 3, 4$$

where x and y are the input to node i and A_i and B_i is a linguistic level such as long or short associated with this node. O_{1i} is the membership grade of a fuzzy set A (=$A1$, $A2$).

Layer 2: Every node in this layer is a fixed node levelled Π, whose output is the product of all the incoming signals.

$$O_{2,i} = \mu_{Ai}(x)\mu_{Bi}(y) = w_i\,,\ i = 1, 2$$

Each node output represents the firing strength of a rule. In general, any other T-norm operators that performs fuzzy AND can be used as the node function in this layer.

Layer 3: Every node in this layer is a fixed node labelled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rules' firing strengths. The output of this layer is called normalised firing strength.

$$O_{3,i} = \frac{w_i}{w_1 + w_2} = \overline{w_i} \qquad (7)$$

Layer 4: Every node in this layer is an adaptive node with the node function as shown below.

$$O_{4,i} = \overline{w_i f_i} = \overline{w_i}(p_1x + q_1y + r_1) \qquad (8)$$

where w_i is the normalised firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is a fixed node levelled Σ, which computes the overall output as the summation of all incoming signals.

$$\text{Overall output} = O_{5,i} = \sum \overline{w_i} f_i = \frac{\sum w_i f_i}{\sum w_i} \quad (9)$$

Optimisation of ANN and ANFIS Parameters

The ANN models having only one hidden layer were used in this investigation. The number of input nodes was eight as there were eight input parameters (mean length, short fiber content, maturity, fineness, neps, seed coat neps, trash and yarn count). Only one output node was kept for predicting yarn unevenness. The transfer function in the hidden and output layers was log-sigmoid as shown in equation (2). It was found that 10 nodes in the hidden layer were giving the best prediction performance in the test set data after 1000 iteration.

For neural-fuzzy modelling, selecting the proper combination of inputs and determining the number of membership functions for each input are very important since they determine the number of rules to be trained. If α is the number of membership function for each input and β is the number of inputs, then there are α^{β} rules to be trained. As there were only 72 data sets for training, the number of rules was kept at such a level that they could be adequately trained using the available data. An input selection scheme was employed to elicit the best combination of inputs. It was found that the combination of mean length, short fiber content and yarn count (Ne) were optimum for the neural-fuzzy modelling. Triangular, trapezoidal, sigmoid and Gaussian type membership functions were tried and it was found that the triangular form with two membership functions for each input is giving the best prediction accuracy. For neural-fuzzy modelling, the Fuzzy Logic Toolbox of MATLAB software (version 6.1) was used.

Results and Discussion

Linear Regression Model

The complete liner regression model was developed using all the eight inputs parameters. The equation is shown below.

$$\begin{aligned} Unevenness = 28.495 - 11.949ML + 0.379SFC - 7.244Maturity \\ - 0.012FF - 0.003Neps + 0.025Seed\ neps \\ - 0.002Trash + 0.321Yarn\ count(Ne) \end{aligned} \quad [R^2=0.848]$$

where *ML* is the mean length of cotton fiber in inch, *SFC* is short fiber content (%) and *FF* is fiber fineness in millitex.

From the R^2 value, it can be said that the equation is able to explain 85 % of the total variability of yarn unevenness. It was found that the mean length, short fiber content and yarn count (Ne) are the only three input variables, which are statistically significant at 99 % level. This supports the findings of input selection scheme of ANFIS model. The contribution of maturity, fiber fineness, neps, and seed coat neps was found to be statistically insignificant at 95 % level.

To remove the insignificant inputs from the regression model, forward stepwise regression equation was also developed which is shown below.

$$\begin{aligned} Unevenness = 17.578 - 9.886ML + 0.422SFC \\ - 0.002Trash + 0.317Yarn\ count(\text{Ne}) \end{aligned} \quad [R^2=0.837]$$

It is observed that this equation can explain about 84 % variability of yarn unevenness, which is almost same with that of complete regression equation. For a given yarn count, fiber fineness determines the number of fibers in yarn cross-section and thereby fiber fineness is expected to be a significant parameter influencing yarn unevenness. However, correlation analysis of cotton fiber parameters showed that fiber fineness is significantly correlated with short fiber content (correlation coefficient R=–0.638) and trash content (correlation coefficient R=0.652). The inter-correlation among cotton properties might have made the contribution of fiber fineness insignificant.

Prediction Performance of Various Models

The prediction power of complete regression, ANN and neural-fuzzy models were evaluated separately in the 72 training data as well as in the 15 unseen testing data. Statistical parameters such as correlation coefficient and mean absolute error % were used to compare the prediction performance of the three models. The summary of results for the training data sets is shown in Table 2. It is observed that the correlation coefficients between actual and predicted value is around 0.92 for all the three models. However, mean absolute error of prediction was maximum for regression model (3.783 %), closely followed by ANN model (3.528 %) and neural-fuzzy model (3.496 %).

The summary and details of prediction results in the testing data sets are shown in Tables 3 and 4 respectively. From Table 3, it is observed that the correlation coefficients between the actual and predicted values were very high (higher than 0.95) for all the three models. However, similar to that of training data, the mean absolute error of prediction was maximum for liner regression model (3.073 %), followed by ANN model (2.794 %) and neural-fuzzy model (2.367 %). Although ANN and neural-fuzzy models show lower prediction

Table 2. Summary of prediction results in training data

Statistical parameter	Prediction models		
	Regression	ANN	Neural-fuzzy
Correlation coefficient (R)	0.921	0.925	0.918
Mean absolute error%	3.783	3.528	3.496

Table 3. Summary of prediction results in testing data

Statistical parameter	Prediction models		
	Regression	ANN	Neural-fuzzy
Correlation coefficient (R)	0.960	0.959	0.970
Mean absolute error%	3.073	2.794	2.367
Samples with more than 5 % error	4	2	1

Table 4. Detailed prediction results in testing data

Test sample no.	Actual	Linear regression		ANN		Neural-fuzzy	
		Predicted	Error%	Predicted	Error%	Predicted	Error%
1	23.32	23.96	2.744	24.32	4.288	24.7	5.918
2	18.01	16.96	5.830	17.51	2.776	17.2	4.498
3	17.82	16.68	6.397	17.39	2.413	17.1	4.040
4	23.34	23.28	0.257	23.50	0.686	23.3	0.171
5	19.92	20.32	2.008	20.25	1.657	20	0.402
6	22.34	22.82	2.149	23.21	3.894	22.5	0.716
7	20.41	20.81	1.960	20.15	1.274	20.3	0.539
8	18.68	18.99	1.660	18.52	0.857	18.6	0.428
9	20.72	21.16	2.124	20.99	1.303	21.1	1.834
10	22.49	23.11	2.757	23.44	4.224	23.2	3.157
11	22.97	22.37	2.612	22.82	0.653	22.4	2.481
12	19.22	20.31	5.671	19.96	3.850	19.8	3.018
13	20.49	19.25	6.052	18.95	7.516	19.9	2.879
14	16.61	16.26	2.107	17.49	5.298	17.2	3.552
15	19.77	20.12	1.770	19.53	1.214	19.4	1.872

error than that of regression model in training and testing data, the differences were not statistically significant at 95 % level. Table 4 shows the prediction results for 15 individual testing samples. It is observed that for regression model, four test samples were having more than 5 % prediction error. In contrast, ANN and neural-fuzzy models were having only two and one test samples in each case, where the prediction error was higher than 5 %.

Linguistic Rules of ANFIS

The main advantage of ANFIS over ANN is that the former can discover linguistic rules relating input and output variables. Figure 4 shows the eight ANFIS rules which are linguistically relating three input variables (mean length, short fiber content and yarn count) with the output variable (yarn unevenness). The membership function of each input variable has two levels, namely low and high. The output variable (yarn unevenness) has eight levels of membership function. From rules 1 and 2, it can be inferred that if yarn is made finer keeping the mean fiber length and short fiber content constant then yarn unevenness will increase. Similarly, comparing rule 1 and 3, it can be inferred that increase in short fiber

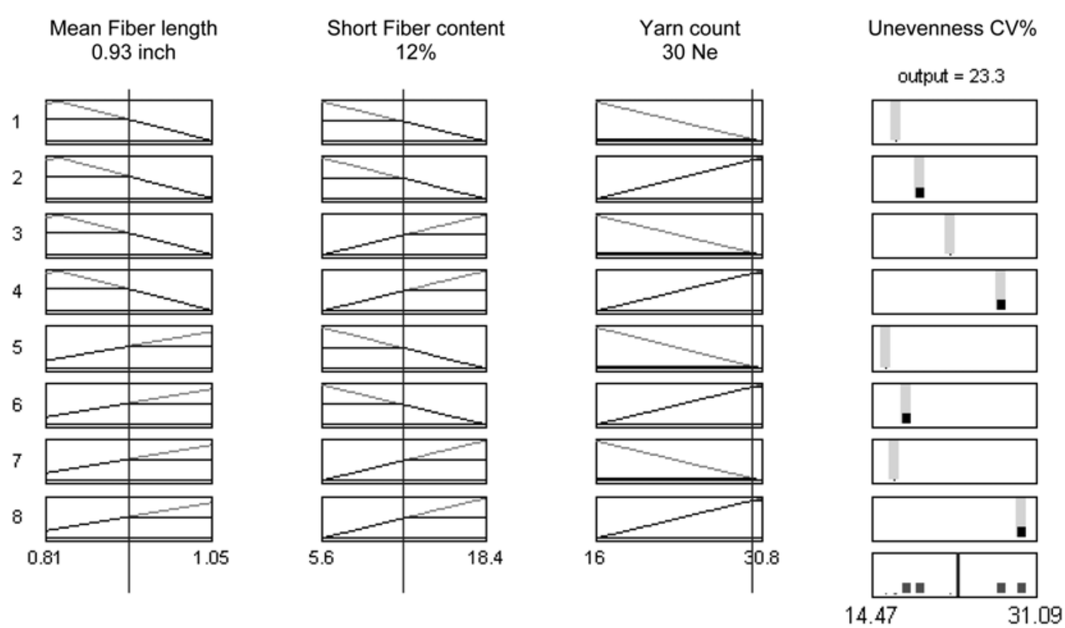

Figure 4. ANFIS rules.

content increases the yarn unevenness rather drastically. Rule 5 depicts that minimum yarn unevenness is expected when fiber length is high, short fiber content is low and yarn count is coarse. These linguistic rules are in agreement with the established norms of spinning technology. Figure 4 also shows that when mean fiber length is 0.93 inch, short fiber content is 12 % and yarn count is 30 Ne, four fuzzy rules (rules 2, 4, 6 and 8) becomes active with different strength or membership values and produces output (yarn unevenness). These four membership functions are first added and then defuzzified using the weighted average method to produce the unevenness CV% of 23.3, as shown in the lowest block of extreme right column of Figure 4.

Figure 5 depicts the effect of mean fiber length and short fiber content on yarn unevenness in accordance with the ANFIS rules keeping the third input variable (yarn count) constant at the mid level (23.4 Ne). It is observed that as the fiber length increases there is consistent reduction in yarn unevenness, although the effect is not very pronounced. It is also noted that the increase in short fiber content in cotton leads to drastic increase in yarn unevenness. During roller drafting, short fibers float in between the front and back roller nip and their velocity is really uncertain. Thus, the short fibers generate drafting waves and increase the unevenness of fiber strand.

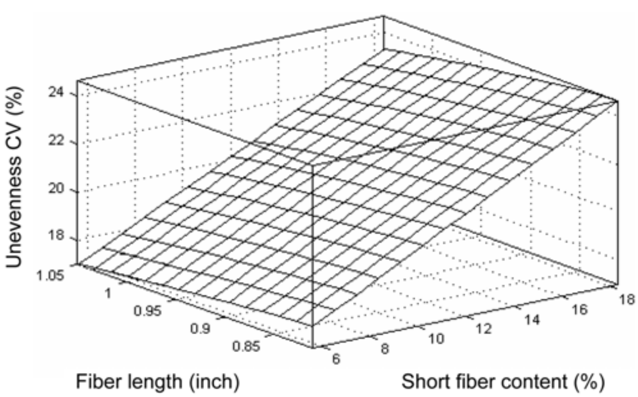

Figure 5. Effect of fiber length and short fiber content on yarn unevenness.

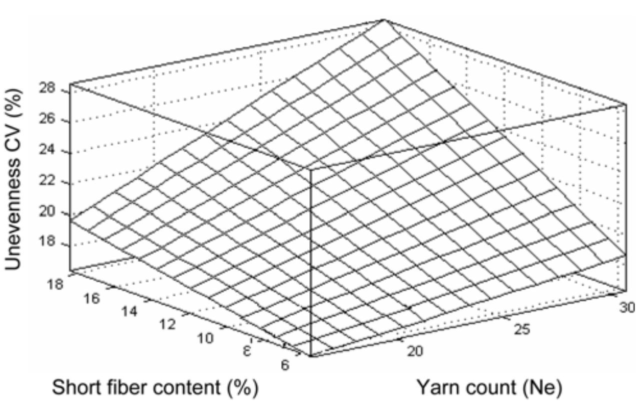

Figure 6. Effect of yarn count and short fiber content on yarn unevenness.

Figure 6 shows the impact of yarn count and short fiber content on yarn unevenness keeping fiber length constant at mid level (0.93 inch). In general, finer yarns exhibit higher unevenness, as expected. The effect of short fiber content on unevenness is very gradual for coarser yarns and radical for finer yarns. In case of finer yarns, there are fewer fibers in the yarn cross-section and thus the generation of drafting wave causes increase of irregularity by a greater extent as compared to coarser yarns. Therefore, short fiber content should be given more importance while selecting the cotton fibers for finer counts.

Conclusion

Yarn unevenness has been modelled using ANN and adaptive neural-fuzzy algorithm. The prediction accuracy of ANN and neural-fuzzy method was found to be slightly better and consistent than that of linear regression model. For neural-fuzzy modelling, only three input variables were chosen and eight fuzzy rules were trained. Training of more fuzzy rules by using more number of inputs, could yield much better prediction performance from the neural-fuzzy model. Fuzzy linguistic rules showed that higher mean fiber length and lower short fiber content reduces the yarn unevenness. The role of short fiber content was found to be more decisive than that of mean length in determining the yarn unevenness.

References

1. J. G. Martinedale, *J. Text. Inst.*, **36**, T35 (1945).
2. H. C. Picard, *J. Text. Inst.,* **42**, T503 (1951).
3. D. R. Cox and M. W. Townsend, *J. Text. Inst.*, **42**, P107 (1951).
4. H. Breny, *J. Text. Inst.*, **44**, P1 (1953).
5. S. L. Anderson and G. A. R. Foster, *J. Text. Inst.*, **46**, T551 (1955).
6. T. V. Ratnam, K. N. Seshan, and K. J. Govindarajulu, *J. Text. Inst.*, **65**, 61 (1974).
7. A. R. Garde and T. A. Subramanian, "Process Control in Spinning", ATIRA Publication, Ahmedabad, India, 1978.
8. M. I. Zeidman, M. W. Suh, and S. K. Batra, *Text. Res. J.,* **60**, 1 (1990).
9. L. Hunter, *Melliand Textilber,* E123 (1988).
10. R. J. Chasmawala, S. M. Hansen, and S. Jayaraman, *Text. Res. J.*, **60**, 61 (1990).
11. R. Rajamanickam, S. M. Hansen, and S. Jayaraman, *Text. Res. J.*, **67**, 39 (1997).
12. A. Guha, R. Chattopadhyay, and Jayadeva, *J. Text. Inst.*, **92**, 139 (2001).
13. P. K. Majumdar and A. Majumdar, *Text. Res. J.*, **74**, 652 (2004).

14. R. Zhu and M. D. Ethridge, *J. Text. Inst.*, **87**, 509 (1996).
15. A. Guha, Ph.D. Thesis, Indian Institute of Technology, Delhi, 2001.
16. A. Majumdar, P. K. Majumdar, and B. Sarkar, *J. Text. Inst.*, **96**, 55 (2005).
17. D. E. Rumelhart, G. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation, in Parallel Distributed Processing", Vol. 1, MIT Press, Cambridge, 1986.
18. S. Haykin, "Neural Networks: A Comprehensive Foundation", 2nd Ed., Pearson Education, Singapore, 2004.
19. J. M. Jurada, "Introduction to Artificial Neural Networks", West Publishing Company, NY, 1992.
20. L. A. Zadeh, *Information and Control*, **8**, 338 (1965).
21. H. J. Zimmerman, "Fuzzy Set Theory and Its Applications", 2nd Ed., Allied Publishers Limited, New Delhi, 1996.