JAIDM
Journal of AI and Data Mining
Vol. 2, No .1, 2014, 73-78.

Yarn tenacity modeling using artificial neural networks and development of a decision support system based on genetic algorithms

M. Dashti[1], V. Derhami[2*], E. Ekhtiyari[1]

1. Textile Engineering Department, Yazd University
2. Electrical and Computer Engineering Department, Yazd University

Received 25 April 2013; accepted 16 September 2013

**Corresponding author: vderhami@yazd.ac.ir (V. Derhami).*

Abstract

Yarn tenacity is one of the most important properties in yarn production. This paper focuses on modeling of the yarn tenacity as well as optimally determining the amounts of effective inputs to produce the desired yarn tenacity. The artificial neural network is used as a suitable structure for tenacity modeling of cotton yarn with 30 Number English. The empirical data was initially collected for cotton yarns. Then, the structure of the neural network was determined and its parameters were adjusted by the back propagation method. The efficiency and accuracy of the neural model was measured based on the error value and coefficient determination. The obtained experimental results show that the neural model could predicate the tenacity with less than 3.5% error. Afterwards, utilizing genetic algorithms, a new method is proposed for optimal determination of input values in the yarn production to reach the desired tenacity. We conducted several experiments for different ranges with various production cost functions. The proposed approach could find the best input values to reach the desired tenacity considering the production costs.

Keywords: *Artificial neural network, Genetic algorithm, Yarn tenacity, Modeling, Cotton yarn.*

1. Introduction

The quality and features of yarn determine possibility of using it in production of different fabrics. In this regard, tenacity is of special importance [1]. In fact, the yarn tenacity affects every next step in the processes of using it. This research was performed based on a request from Nakhchin and Nakheaftab factories, which are two distinguished textile production factories located in Yazd, Iran. This research aims at determining the best input values to produce 100% cotton yarn for a desired tenacity.

We are concerned with two constraints: first, although the effective parameters in the yarn tenacity are almost known but it is not clearly determined how these parameters affect the final yarn tenancy. In other words, there is no accurate mathematical model for this purpose. Second, optimal determination of input values to reach a desirable tenacity has not been investigated yet. In fact, nonlinearity and complexity of the relation giving the yarn tenacity in terms of the effective parameters, have led the textile engineers to determine the values of input materials only by "trial and error" and their former experiences. Investigations show that although there is many research findings focused on modeling of tenacity but only a few of them proposed a practical approach to optimal determination of values for effective inputs. Some usual methods for the yarn tenacity modeling are mechanical models, mathematical models [2], statistical (regression) methods, fuzzy modeling [3], and artificial neural network models [7]. These primary methods (mechanical, mathematical and statistical) require highly experienced personnel as well as numerous of test steps, therefore they could not give accurate models with reasonable computational costs [8].

Artificial Neural Network (ANN) which is inspired from evolution of biological neurons of brain is a powerful method for modeling of complex phenomena. Some of its characteristics such as the ability of learning and generalization, robustness against disturbances, and information parallel processing have made ANN superior to other modeling approaches. Nowadays, ANNs are widely used for solving many engineering problems in modeling, controlling, and patterning recognition [5], [6].

Already in an ANN based yarn tenacity prediction research, the five parameters: spun fibers upper half mean length, package hardness, fineness, proportions of fiber length uniformity, and maturity of fibers content were used as neural network input parameters [7]. The accuracy of this neural model was 12%. In another study, 14 fiber properties have been used as neural network inputs to predict yarn tenacity [8]. These inputs are values of impurity, number of each package impurity (amount of trash), upper half mean length, strength and length increase to the extent on which the fiber is torn (elongation of break), fiber fineness, brightness, yellowness, fiber maturity content, standard fiber fineness (norm), length uniformity, and micronaire. The accuracy of the later neural model was 8%.

Two other papers focused on fiber properties measured by High Volume Instrument (HVI) and included upper half mean length, length uniformity, short fiber content, strength, maturity ratio, fineness, grayness, and yellowness used as neural network input [9],[10].

In this research, we use neural networks in modeling of yarn tenacity of 100% cotton with 30 Ne, where a new approach, which is based on genetic algorithms, is used for optimal determination of values of input materials. The main advantages and innovations on this research are:

1- Proposing an accurate neural model for predicting yarn tenacity of 100% cotton yarn. Although there are some research works concerning neural modeling, but our model is a real case study with different inputs and conditions; hence, we use a different structure of ANN.

2- Proposing a new idea to find the optimal values of inputs to reach desired yarn tenacity by using genetic algorithms. To the best of our knowledge, this research is the first research of its kind.

The structure of this paper is as follows. In the second section, yarn tenacity and the parameters which affect yarn tenacity are investigated and discussed. The third section deals with introducing neural networks; in the fourth section, modeling of yarn tenacity of 100% cotton with 30 NE (the most popular yarn) using neural network is presented. In the fifth section, a method to find optimal values of inputs to reach a desirable tenacity is proposed. Finally, in the last section, summary and conclusions are provided.

2. Effective parameters on yarn tenacity

The resistance of yarn against tensile forces is called yarn tenacity. It is the minimum force which is needed to tear out that yarn [1]. Several factors are involved in yarn tenancy, and the most important ones are the properties of the fibers used to produce yarn (raw materials) such as: upper half mean length, length uniformity, short fiber content, fibers strength, maturity ratio, yellowness, linear density, and fiber length increase (which is measured by HVI).

Here, the input variables have been chosen with respect to the related research [7,8,10]. The production process was set fixed for the whole time. This means, five adjustable parameters for textile machines such as spin tube, breaker speed, rotor speed, were fixed. Moreover, yarn twist and yarn counts were set constant as well. In this way, in our model, the seven mentioned parameters associated with fiber property were considered as effective parameters on yarn tenacity.

3. Artificial neural networks

Artificial neural network is a structure inspired from the human brain. It is very useful in modeling complex functions. A neural network consists of an interconnected group of artificial neurons where an artificial neuron is a mathematical function representing an abstraction of biological neurons [4].

Figure 1 shows the structure of a neuron. Vector $x=[x_1,...,x_n]^T$ is input and the scalar y is the output of the neuron.

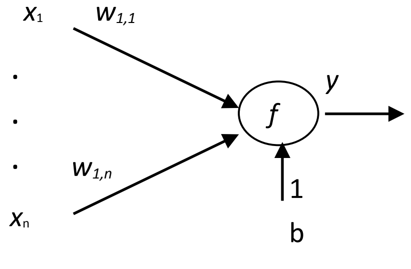

Figure 1. Model of a multi-input neuron

The influence of x_i on y is determined by $w_{1,i}$. Another input is bias parameter that its corresponding weight is 1.

The output of the neuron is computed by:

$$y = f(z) = f(Wx + b) \tag{1}$$

$$z = \sum_{i=1}^{n} x_i w_{1,i} + b = Wx + b \tag{2}$$

$$W = [w_{1,1}, \ldots, w_{1,n}] \tag{3}$$

The activation function *f* could be linear or either nonlinear. Here, the designer selects a suitable activation function with respect to the problem features. Table 1 shows some widely used activation functions.

Table 1. Some activation functions

Function	**Definition**	**Range**
Linear	$f(z) = z$	(- ∞,+∞)
Logistic	$f(z) = \dfrac{1}{1+e^{-az}}$, a is slope parameter	(0,+1)
Hyperbolic	$f(z) = \dfrac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	(-1,+1)
Threshold	$f(z) = \begin{cases} 0 & if \quad z < 0 \\ 1 & if \quad z \geq 0 \end{cases}$	{0,+1}

In comparison with single layer networks, multilayer neural networks have more capabilities. Double layers feed forward neural networks (with sigmoid functions in first layer and that of linear in second layer) can estimate any continuous function with arbitrary precision [4].

In this research, we employed a feed forward neural network with two layers, where the first layer is known as the hidden layer. Figure 2 shows the corresponding network structure. This structure is presented as: *n:nh:o* where *n* is the number of inputs, *nh* is the number of hidden layer neurons, and *o* is the number of output layer neurons. The output of network is computed by:

$$y = f_o\left(\sum_{i=1}^{nh} o_i w_i^2 + b\right) \tag{4}$$

$$o_i = f_i\left(\sum_{j=1}^{n} x_j w_{i,j}^1 + b_i\right) \quad i = 1, \ldots, nh \tag{5}$$

where, y is the final output, f_o is the activation function of output neuron, w_i^2 is the weight of link between *i-th* output neuron in the hidden layer and the final output, b is the bias of output neuron, o_i is the output of *i-t*h neuron in hidden layer, $w_{i,j}^1$ is the weight of link between *j-th* input and *i-th* neuron in hidden layer, b_i is the bias of *i-th* neuron in hidden layer, and f_i is the activation function in *i-th* neuron in hidden layer.

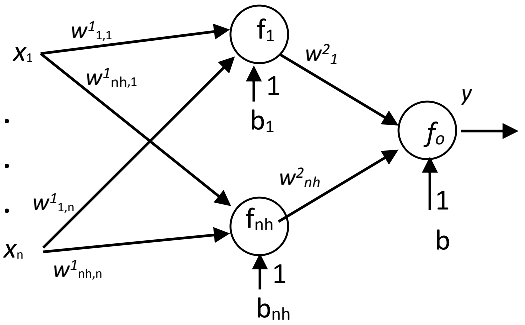

Figure 2. The feed forward neural network.

The weights $(w_{i,j}^1, w_i^2, b_i, b)$ are adjustable parameters that need to be tuned by training. In fact, the objective of network training is adjusting these parameters in such a way that the network generates desired output for different inputs.

4. Modeling

Since neural networks find and learn patterns in the training data, in the first step of modeling, we need some training data. The cotton yarn with 30 NE (Number English) is produced in tenacity range of 13 to 16. Some experiments have already been conducted on cotton fibers and the yarn 30 NE produced from them in order to collect required input data in the mentioned range. As stated in Section 2, seven properties (see Table 2) of cotton fiber are considered as effective parameters on yarn tenacity and they are selected as inputs of our model. The network output would be the yarn tenacity. The activation function for the neurons in the hidden layer is sigmoid and it is linear for the output neuron. Here, we use the obtained data from 33 cotton samples for ring carded spun yarns for input-output data.

The training data must cover the whole input range and need to have a suitable dispersion. Generally, if we use more training data, we would often be more accurate in prediction. In the primary experiments of this research, we used 100 data samples. The obtained result for this data was poor. Therefore, we had to obtain and use more data. Here, we encountered some practical constraints. For example, in measuring yarn tenacity for each sample, first we had to produce the yarn with those input materials. Finally, we totally prepared 990 experimental data for tuning our neural network.

During first experiments, we observed that the major problem in training our neural model was over fitting. In fact, we could see an error which was getting smaller and smaller in the course of training (for the training data) but the magnitude of the error for the test data was not acceptable. This phenomenon was due to the fact that during training phase, the network parameters are adjusted to reduce the error for the training data; hence, the network is fitted for the training data and this is why the property of generalization of the neural network degrades considerably. As a result, network's error in the output of the network was too high for any data other than the training data.

Providing a solution for the problem, we used some valid data, which are not used for training, but they are only used to stop the training properly to avoid over fitting. After each training epoch, network's error was calculated for valid data and if the training procedure was going in a way that the error was increasing the training would be stopped. Therefore, we divided our data into three parts: Training data, valid data, and test data. They included 800, 90, and 100 samples, respectively. The test data was used in the final to assess our neural model. The statistical features of the mentioned data shown in Table 2.

4.1. Neural network structure

Previous studies focused on modeling have shown that feed forward networks are suitable for modeling. We used a two layer feed forward neural network for modeling. It consists of seven inputs and one output. The number of neurons in hidden layer is different for these networks. Activation function for neurons in hidden layer is logistic and for the last layer is linear. These networks have been trained using training data regarding the valid data. The training method was the error back propagation algorithm. Results of the experiments on different structures are shown in Table 3.

Table 2. Statistical summary of data for fiber properties

Fibers Character-istics	Fiber Tenacity (CN/Tex)	Fiber Elong-ation	%50 Average Length	length uni-formity	Micro-naire	Reflection Degree	Yellow-ness
Maximum	34	6.9	1.2	83.2	5	80.4	11.4
Minimum	26.5	5.3	0.97	79.2	3.1	73	8
Average	29.05	6.27	1.06	81.57	4.23	77.2	9.34
Standard Deviation	1.41	0.46	0.05	1.05	0.45	2.28	0.72

Table 3. Results of training on different structures of neural networks

Network Structure	7:9:1	7:10:1	7:11:1	7:12:1	7:13:1	7:14:1
Error Percentage for Training Data	3.3%	3.1%	0.8%	3.5%	2%	1.1%
Error Percentage for Test Data	12%	3.5%	10.4%	10.4%	6.4%	11.6%
R2 for Test Data	0.91	0.95	0.90	0.87	0.87	0.83

The first row in the table shows structure of the neural network. The error rate for training data has been given in the second row. The third row shows the error percent for test data, and the determination coefficient [4] has been given in the last row.

As Table 3 shows, the best results have been obtained from the neural network model consisting of 10 neurons in the hidden layer. The training stopped after 106 epochs. The percentage error for this structure in the training phase is 3.1% and 3.5% for the test data. Comparing the similar results from [2, 5] this error is acceptable. Therefore, the ANN with structure of 7:10:1 is selected as a neural model to predict yarn tenacity.

5. Optimal input values for desired tenacity

After finding the suitable neural model, we turn our attention to the determination of input parameters to reach the desired tenacity using Genetic Algorithms (GA). This is, in fact, a multi-goal problem. From one side, the production cost should be minimized and from other side the tenacity of produced yarn should be equal to or higher than the desired tenacity. If the cost reduces but the tenacity is less than the desired value, it would not be acceptable. From the other side, if the tenacity improves but the cost goes too high, it would not be acceptable too. In order to overcome this dilemma, we first convert it into an optimization problem and then propose an approach to solve it by GA. In Section 2, we introduced the seven variables which are effective in yarn tenacity. The cost production function $C(x)$ would be:

$$C(x) = \sum_{i=1}^{7} v_i x_i \tag{6}$$

Where, v_i is the assigned weight for the *i-th* variable x_i. Weights show how effective each parameter is in the cost function. In other words, more expensive variables are expressed by higher weights.

The evaluation criterion in GA is the fitness function. Here, the algorithm looks for the responses in such a way that the fitness function decreases to a minimum value. The space of the problem is defined by all combinations of the values of variables x_i where they fall in the ranges mentioned in Table 2. Our objective is to reach the desired tenacity while the cost function (Eq. 6) is minimized. The most important challenge in using GA is defining the suitable fitness function. Here, the fitness function is defined as:

$$Fit.F = K * f(e) + C(x) \tag{7}$$

$$e = T_{desired} - T_{actual}(x)$$

$$T_{actual}(x) \approx Output\ of\ NeuralModel\ of\ Tenacity \tag{8}$$

$$f(e) = \begin{cases} e & e > 0 \\ 0 & Otherwise. \end{cases} \tag{9}$$

In the above fitness formula, the first term is the error function. The error is equal to the difference between the desired tenacity and the actual tenacity ($T_{actual}(x)$). The actual tenacity is the tenacity of produced yarn with x_i's (the amounts of input materials). We use the neural model obtained in Section 4, to predict the actual tenacity for each x and finally to compute *f(e)* in Eq. (9). In the first relation, *K* serves as a weight parameter.

Each intermediate solution for a problem in GA is called a chromosome. A chromosome consists of genes, corresponding to a series of values given to the problem variables (in our case x_i's). The number of genes is equal to the number of variables; therefore in our case, each chromosome will have seven genes. Using values of genes in chromosome as the neural model input, we can get the tenacity of the yarn as well as the production cost using Eq.6. In this way, the fitness function can be computed for each chromosome.

If the predicted tenacity (output of neural model) is less than the desired tenacity, then the first term in fitness function would be positive and its amount is error proportion. In a situation where the predicated tenacity is equal or greater than desired tenacity, the first term in fitness function would be zero. The second term stands for the production cost function. As the cost increases, the fitness function will increase, too.

Regarding the amount of *K* in Eq. 7, since the main objective is to reach the desired tenacity, *K* has to be determined in such a way that GA finds answers having first term equal to zero. Since the maximum of cost function is 1000, we set *K* to 1000 as well. This value for *K* lets one thousandth of error from the desired tenacity be equal to one unit in the second term (cost function). This will guide our GA model to find the solutions with values in first term equal to zero.

In general, a GA is inspired from evaluation theory. It looks for a chromosome that minimizes the fitness function. The algorithm begins with a random population (some chromosomes), then it uses present population to generate new population based on the following steps in each iteration [11]:

1. The value of fitness function is computed for each chromosome in the current population.
2. The algorithm selects some chromosomes based on their fitness values. These chromosomes are called parents and used to generate next generation. Some well known approaches for selection of parents are: Roulette-wheel selection, rank selection, and elitist selection.
3. The algorithm generates children with applying crossover operation on selected parents.
4. Mutation on a child is changing one or more of its properties randomly. Mutational children are produced in this step.
5. The obtained children are added to the population.

The algorithm continues until it finds a child who fits the desired criteria.

6. Experimental results

To assess the proposed approach, we performed a number of experiments with different desired tenacities and different weights (v_i*'s*) as cost function. The objective was finding the input parameters for yarn production such that the obtained tenacity becomes greater than the desired tenacity and as a result, the production cost is minimized. In the experiments, we used one-point crossover, and mutation rate of 0.1; based on the primary results, the population size was set to 40.

The final experimental results are shown in Table 4. The first column is the desired tenacity, the second column is the obtained tenacity of the produced yarn, the third column is the final value of the fitness function at the termination of the search process in GA, and the last column is the value of *V* for each experiment.

For efficiency improvement of our algorithm, all input values, x_i, were normalized to be in the range of [0, 1]

Table 4. The results of the experiments

$T_{desired}$	$T_{obtained}$	*Fit.F*	*V*
14	14.03	660.01	[4 7 3 5 1 2 6]
15	15.01	668.81	[4 7 3 5 1 2 6]
16	16.00	679.67	[4 7 3 5 1 2 6]
14	14.04	718.26	[1 6 3 7 5 2 4]
15	15.00	720.97	[1 6 3 7 5 2 4]
16	16.01	724.81	[1 6 3 7 5 2 4]

Table 4 shows that the proposed approach was capable to find values for the input parameters in a way that the obtained tenacity is satisfactory; the amount of first term of the fitness function is equal to zero and the values of the fitness function are equal to the production cost.

Meanwhile, the obtained values for inputs depend on the amounts of v's. For example, the values in the fourth row, ($T_{desired}$=16, *V*=[4 7 3 5 1 2 6]), is:

x=[26.754 5.977 0.876 70.510 2.703 64.818 7.216]

and for the last row with the same desired tenacity and *V*=[1 6 3 7 5 2 4] is:

x=[30.966 4.558 0.855 70.236 2.713 64.890 7.233]

Comparing the two results indicates that the second case, which is the weight of first input variable in the cost function, has been decreased (changed from 4 to 1), the value of this input has been increased. In opposite, the value of forth input variable has been reduced due to its weight increase.

6. Conclusion

In this research, a neural network model of yarn tenacity for 100% yarn cotton 30 NE using empirical data is presented. The output of neural model for test data confirmed the accuracy of the proposed model. Based on the obtained results in modeling section, feed forward neural network with 10 neurons in hidden layers was a suitable structure for modeling of tenacity. We also used GA for optimal determination of values of inputs in yarn manufacturing. The results of the experiment showed that GA with the defined fitness function can find the best values for inputs such that produced yarn with the obtained values of inputs satisfy the desired tenacity while the production cost becomes minimal. The proposed method can be used to find the best-input values for any kind of yarn production with a desired tenacity. For using the proposed method, a user determines the desired tenacity of yarn and then he/she assigns weights of input materials based on their prices. Afterwards, our proposed method presents the amounts of input materials for yarn production so that the produced yarn has the desired tenacity, and above all, the production cost has been minimized.

References

[1] Taheri, A., (1990). General Technology of Cotton Textile Industry," (in Persian) Aghabik.

[2] Majumdar, P. K., and Majumdar, A., (2004). Predicting the Breaking Elongation of Ring Spun Cotton Yarns Using Mathematical, Statistical, and Artificial Neural Network Models", Textile Research Journal, Vol.74, No.7, pp. 652-655.

[3] Shams Nateri, A. (2005). Using Neuro-Fuzzy For Prediction Ring Spun Yarn Strength From Cotton Fibers Properties", 3rd International Industrial Simulation Conference, June 9-11.

[4] M. Menhaj, (2002). Fundamental of Neural Networks," Amikabir University publication, (in Persian).

[5] Gharehaghaji, A.A., Palhang, M., and Shanbeh, M., (2006). Using Artificial Neural Network Algorithm to Predict Tensile Properties of Cotton-covered Nylon Core Yarns", Esteghlal, Vol. 24, No.2.

[6] Jackowska-Strumillo, L., Ackowski, T., Cyniak, D., Czekalski, J., (2004). Neural Model of the Spinning Process for Predicting Selected Properties of Flax/Cotton Yarn Blends", Abres & Textiles In Eastem Europe, Vol. 12, No. 4, pp. 17-21.

[7] Sette, S., Bouliart, L., Van Langenhove, L. Kiekens, P., (1997). Optimizing the Fiber to Yarn Production Process with a Combined Neural Network/Genetic Algorithm Approach", Textile Research Journal, Vol. 67, No. 2, pp. 84-92.

[8] Cheng, L., Adams, D. L., (1995). Yarn Strength Prediction Using Neural Networks, Part 1: Fiber Properties and Yarn Strength Relationship", Textile Research Journal, Vol. 65, No. 9, pp. 495-500.

[9] Majumdar, A., Majumdar, P. K. , Sarkar, B., (2004). Selecting Cotton Bales By Spinning Consistency Index And Micronaire Using Artificial Neural Networks", AUTEX Research Journal, Vol. 4, No. 1, pp.1-8.

[10] Pynckels, F., Sette, S., Van Langenhove, L., Kiekens, P., and Impe. K. (1995). Use of Neural Nets for Detemining the Spinnability of Fibres"; Journal of the Textile Institute; October Vol. 86, No. 395, pp. 425-437.

[11] Back, T. (1996). Evolutionary Algorithms in Theory and Practice", Oxford University Press, New York.