



Textile Fibre Classification Using Artificial Neural Network

Partha Pratim Deb*

Department of Computer
Science and Engineering

Techno India Agartala , Maheshkhola, Tripura(W),INDIA
an Engineering College affiliated to Tripura University

Dinabandhu Bhandari

Department of
Information Technology
Heritage Institute of Technology,
West Bengal, India

Abstract— Recognition of textile fiber has a major influence on the success of the textile industry. Resilience, dry tenacity, Wet tenacity moisture regain (absorbency), extension are the most important properties that are used in classifying fibers. Attempt has been made in developing a neural network (ANN) based methodology in determining the category of a fiber. The proposed methodology has been demonstrated on numerous samples collected from the laboratory and satisfactory performance has been observed.

Keywords— Artificial-Neural-Networks, resilience, dry-tenacity, Wet-tenacity, moisture regain (absorbency), extension, Category.

I. INTRODUCTION

Fiber recognition is a very sophisticated task that requires special skill. Generally, number of features is taken into consideration in categorizing a fiber. The fiber classification tree has shown in figure 1.

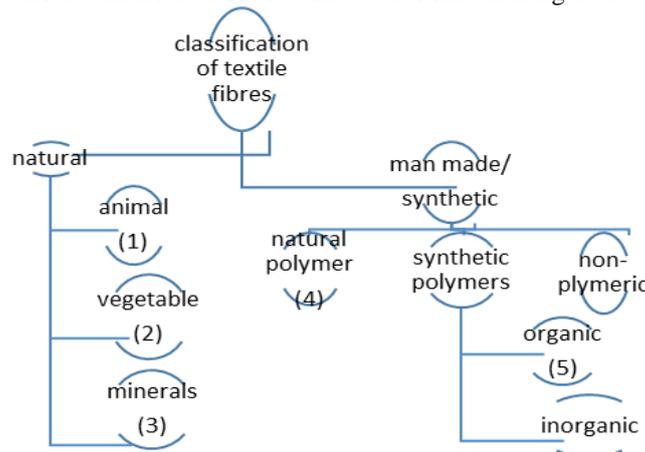


Figure 1 : classification of textile fibers

Some of the important properties or characteristics [1][7][8][9] considered in this purpose are given below.

- Dry tenacity
- Wet tenacity
- Dry elongation
- Wet elongation
- Dry modulus
- Wet modulus
- Loop tenacity
- Loop elongation
- Knot-tenacity
- Heat-absorbency
- Extension
- Resilience
- Moisture regain

A lot of maturity and skill may be required on the part of the engineer to categorize the fibers. The engineers use their experience manually to recognize a fiber. One can easily appreciate the complexity of the task is huge keeping in mind the number of parameters they need to consider. Moreover, the decision may be erroneous as significant amount of manual judgment is involved.

Neural networks are used to model complex relationships between inputs and outputs or to find patterns in data and also for future data prediction. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system that changes its structure during a learning phase. Neural network consists of an input layer, a hidden layer (or hidden layers), and an output layer, as shown in Fig 2.

A network with more than two layers can approximate any function, if the hidden layer is large enough. This has been proved using the Stone-Weierstrass theorem [19].

The high level diagram of a neural network is as follows [12]

$$x^0 \xrightarrow{W^1, b^1} x^1 \xrightarrow{W^2, b^2} \dots \xrightarrow{W^L, b^L} x^L$$

Where

$x^l \in R^{n_l}$ for all $l = 0, \dots, L$ and W^l is an $n_l \times n_{l-1}$ matrix for all $l = 1, \dots, L$. There are $L+1$ layers of neurons, and L layers of synaptic weights. The primary task here is to find the weights W and biases b so that the actual output x^L becomes closer to the desired output d . The most popular approach to find the optimal set of weights of an ANN is backpropagation, which is being explained below.

The backprop algorithm consists of the following steps.

The input vector x^0 is transformed into the output vector x^L , by evaluating the equation [12]

$$x_i^l = f(u_i^l) = f\left(\sum_{j=1}^{n_{l-1}} W_{ij}^l x_j^{l-1} + b_i^l\right)$$

For $l = 1$ to L .

The difference between the desired output d and actual output x^L is computed.

$$\delta_i^L = f'(u_i^L)(d_i - x_i^L)$$

The error at the output units is then propagated backward through the entire network, by evaluating [12]

$$\delta_j^{l-1} = f'(u_j^{l-1}) \sum_{i=1}^{n_l} \delta_i^l W_{ij}^l$$

From $l = L$ to 1 .

The synaptic weights and biases are updated using the results of the forward and backward passes,

$$\Delta W_{ij}^l = \eta \delta_j^l x_i^{l-1}$$

$$\Delta b_i^l = \eta \delta_i^l$$

These are evaluated for $l = 1$ to L . The order of evaluation doesn't matter.

Later I'll show that this is gradient descent on a cost function, but first let's see an application of backpropagation.

Let's define the cost function

$$E(W, b) = \frac{1}{2} \sum_{i=1}^{n_L} (d_i - x_i^L)^2$$

Where x^L a function of W and b is arises through the equations of the forward pass. This cost function measures the squared error between the desired and actual output vectors.

We're going to prove that backprop is gradient descent on this cost function. In other words, the backprop weight updates are equivalent to [12]

$$\Delta W_{ij}^l = -\eta \frac{\partial E}{\partial W_{ij}^l}$$

$$\Delta b_i^l = -\eta \frac{\partial E}{\partial b_i^l}$$

In general form the equation for a Tan-sigmoid function would appear as:

$$s(x) = \tanh(cx) = \frac{e^{cx} - e^{-cx}}{e^{cx} + e^{-cx}}$$

Where $c > 0$, $c = \text{const}$, is a positive scaling constant.

[13]The method of reducing the training time is the use of momentum factor because it enhances the training process.

The influence of the momentum on weight change is shown in the equation

$$[\Delta W]^{n+1} = -\eta \frac{\partial E}{\partial W} + \alpha [\Delta W]^n$$

$-\eta \frac{\partial E}{\partial W}$ = weight change without momentum,

$\alpha [\Delta W]^n$ = Momentum term

II. PROPERTIES OF THE ALGORITHM

The backprop algorithm has a number of interesting features

A. The forward and backward passes use the same weights, but in the opposite direction

$$x_j^{l-1} = \rightarrow x_i^l$$

$$\delta_j^{l-1} = \leftarrow \delta_i^l$$

B. The update for a synapse depends on variables at the neurons to which it is attached. In other words, the learning rules are local, once the forward and backward passes are complete.

In this investigation, an automatic method is proposed exploiting the effectiveness of Artificial Neural Networks (ANN) to categorize fibers. ANN model is developed using NN tool in MATLAB 7.7. In this classification strategy I have used 5 important physical properties from all of the physical properties to classify the fibers in 5 different categories.

III. SELECTION OF PARAMETERS

In solving the problem of fibre classification, the most important task is to analyze the fibres and select relevant parameters to be used in ANN. For this reason out of these parameters 5 important parameters have considered for the classification. They are

- resilience
- dry-tenacity
- Wet-tenacity
- moisture regain (absorbency)
- Extension

Resilience means the ability of a fiber to return to its original state after being disturbed. **Dry-tenacity** is the ability of fibers to overcome great strength or force in dry condition. **Wet-tenacity** is the ability of fibers to overcome great strength or force in wet condition. In another words tenacity is the greatest longitudinal stress of fibers can bear without tearing asunder, usually expressed with reference to a unit area of the cross section of the fiber in the both dry and wet conditions. **Moisture-regain** is the percentage of moisture in a fiber brought into equilibrium with a standard atmosphere after partial drying, calculated as a percentage of the moisture-free weight. **Extension** means the amount, degree, or range to which a fiber extends or can extend.

The measurement of the parameters have shown in the table 1.

Table 1: The measurement of the parameters

resilience	$\frac{\sigma_y^2}{2E}$ where σ_y is the yield strength and E is the Young's modulus
Dry-tenacity	centiNewton/tex (cN/tex)-> x 0.1132 Grams-force/Denier (G/D)
Wet-tenacity	centiNewton/tex (cN/tex)-> x 0.1132 Grams-force/Denier (G/D)
Moisture-regain	Moisture absorbency in percentage moisture regain. e.g. = 0.7, Absorbance units = 70%.
flexibility	Young's modulus=Tensile stress /tensile strain= $\frac{FL_0}{A_0 \Delta L}$ F is the force exerted on a fiber under tension; A0 is the original cross-sectional area of a fiber through which the force is applied; ΔL is the amount by which the length of the fiber changes; L0 is the original length of the fiber.

The classification values are Animal fibers (1), vegetable fibers (2), minerals fibers (3), Natural polymer fibers (4), Organic synthetic polymer fibers (5).

IV. NEURAL NETWORK BASED FIBER CLASSIFICATION

As Artificial neural networks (ANN) [11] are developed for pattern recognition by using the computational complexity between input and output, we here want to build an approach for thousands of fiber recognition by their physical properties. Here out of these 13 parameters, we select only 5 parameters as suggested by the handloom and textile

department of Tripura govt. Using these 5 parameters we design a neural network using 8 hidden neurons represent good training result as shown in table 4. These 5 parameters found to be very useful for recognizing the no. of fibers in the training phase as well as in testing phase.

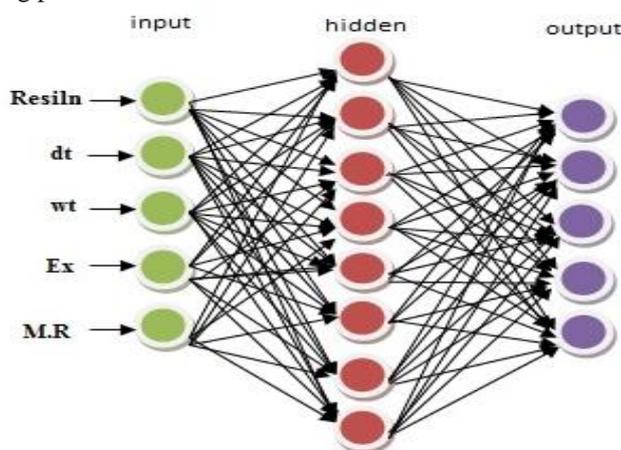


Figure 2: three layered multi-layer perceptron (5-8-5)

Ideally a system designer wants the same range of values for each input feature in order to minimize bias within the neural network for one feature over another. Data normalization can also speed up training time by starting the training process for each feature within same scale[20]. It is especially useful for modeling applications where the inputs are generally on widely different scales. The normalized data is determined by

Min-max normalization and is expressed as

$$X = (p - \text{minp}) / (\text{maxp} - \text{minp})$$

Where

X = normalized value

Minp = minimum in input

Maxp = maximum in input

The fiber test data is divided into 2 parts using 70:30 mode of distribution. A total of 67 fibers which is obtained from different parts of chitter district with wide range of WL from laboratory tests. Among 45 fibers data is used for testing and remaining 22 fibers data is used for training. The typical normalized data used for training phase is presented in Table 2 and in Table 3 presents the typical normalized data used for testing phase. The symbols used for the input parameters are reslin (resilience), dt (dry-tenacity), wt (wet tenacity), ex (extension), M.R (moisture regain), output,

45 fibers test data was used for training the neuron, typical training data is presented in Table-1 [4][5][8][9][10]. Remaining 22 fibers test data was used for testing the network model developed for prediction of Category of fibers. Typical testing data is presented in Table-2 [4][5][8][9][10]. The feed forward back propagation training network models have been coded into a MATLAB program using neural network toolbox. The MATLAB software enables training with different convergence criteria, tolerance level, activation functions and number of epochs. The neural network models studied in this investigation uses transfer function = 'TANSIG' as activation function. A constant value of learning rate equals to 0.2 was assigned for all the models. The network training/learning halts automatically once the mean square error value converges to a tolerances value of 0.05 or the Number epochs become equal to 3000 whichever is earlier. After this the network model is ready for prediction of desired output.

TABLE 2. TRAINING DATA

Resiln	dt(cn /tex)	wt(cn /tex)	Ex(%)	M.R(%)	o/p (actual)	o/p (target)
0.96	1	0.95	0.36	0.2	0.7	1
1	0.98	1	0.42	0.18	1.2	1
0.63	0.83	0.77	0.85	0.55	1.7	2
0.67	0.66	0.54	0.81	0.54	2.2	2
1	0.78	0.41	1	1	2.12	2
0.52	1	0.88	0.55	0.91	3.05	3
0.52	0.9	0.86	0.6	0.91	2.78	3
0.13	0.96	0.54	0.76	0.26	4.2	4
0.13	0.68	0.61	0.69	0.25	3.7	4
0.52	0.51	0.61	0.97	0.28	4.06	4
0.52	0.49	0.72	0.97	0.28	4.03	4
0.68	0.7	0.59	0.88	0.99	4.67	5
0.68	0.81	0.74	1	1	4.82	5

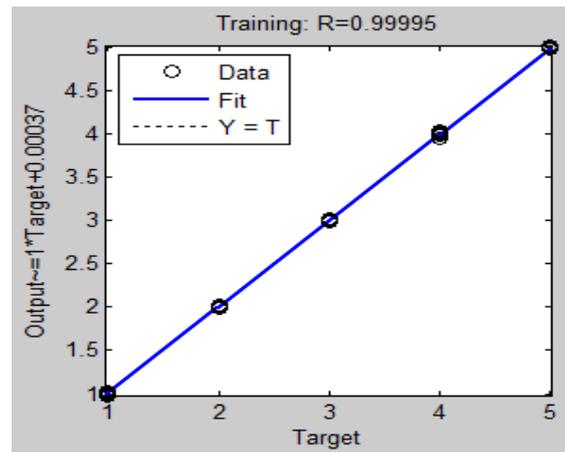


FIGURE 3: PLOT REGRESSION FOR TRAINING (ACTUAL VS. TARGET)

TABLE 3. TESTING DATA

Resiln	dt(cn) /tex)	wt(cn) /tex)	Ex(%)	M.R.(%)	o/p (actual)	o/p (target)
0.96	1	0.95	0.36	0.2	0.7	1
1	0.98	1	0.42	0.18	1.2	1
0.63	0.83	0.77	0.85	0.55	1.7	2
0.67	0.66	0.54	0.81	0.54	2.2	2
1	0.78	0.41	1	1	2.12	2
0.52	1	0.88	0.55	0.91	3.05	3
0.52	0.9	0.86	0.6	0.91	2.78	3
0.13	0.96	0.54	0.76	0.26	4.2	4
0.13	0.68	0.61	0.69	0.25	3.7	4
0.52	0.51	0.61	0.97	0.28	4.06	4
0.52	0.49	0.72	0.97	0.28	4.03	4
0.68	0.7	0.59	0.88	0.99	4.67	5
0.68	0.81	0.74	1	1	4.82	5

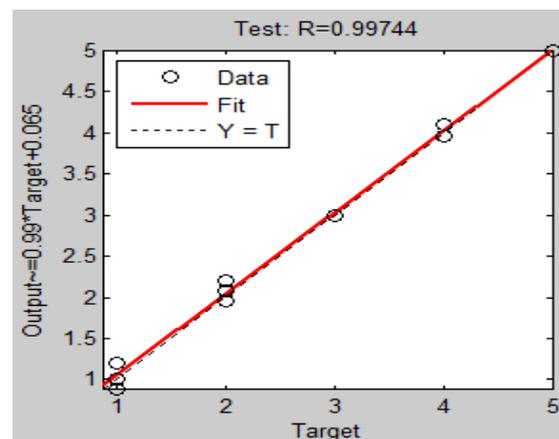


FIGURE 4: PLOT REGRESSION FOR TESTING (ACTUAL VS. TARGET)

V. RESULT AND SUCCESS RATE

The no. of hidden neurons used and the performance and prediction percentage has shown in table 4. Here we have started evaluating the no of hidden neurons which will give the minimum error rate during training phase. And after testing the no of hidden neurons (5, 7 and 8) will come to the conclusion that using 8 hidden neurons will give the minimum error rate.

We first train the network using Plot regression for training data has shown in figure 3. Plot regression (targets (T), outputs (Y)) plots the linear regression of targets relative to outputs. For a perfect fit, the data should fall along a 45 degree line, where the network outputs are equal to the targets. For this problem, the fit is reasonably good for all data sets, with R values in each case of 0.93 or above. The blue line shows the best fit. And the classes 1, 2,3,4,5 are perfectly classified along the plot regression.

The plot regression for validation is shown in figure 5. The validation set is generally used to compare the different architectures or parameters of the neural network. For example, while the training set is used to train the network, the validation set would be used to determine how many layers/nodes to have. The purpose of this validation set is to determine when to stop the training before the NN starts over fitting. The green line shows the best fit for validation.

In this training we observed that the mean square error for training the samples is 0.00175. Using this training performance we experiment it on the testing data and plot regression for testing data has shown in figure 4. As like as the blue line shows the best fit for training, the red line shows the best fit for testing. As we observe the mean square error from the plot regression for the testing data is 0.00194 which is the almost the minimum error rate occurred during testing phase.

The plot regression for all training, validation and testing is shown in figure 6. The black line shows the fit.

Gradient descent with momentum, implemented by `traindm`, allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low pass filter, momentum allows the network to ignore small features in the error surface. Without momentum a network can get stuck in a shallow local minimum. With momentum a network can slide through such a minimum. Gradient descent with momentum depends on two training parameters. The parameter `lr` indicates the learning rate, similar to the simple gradient descent. The parameter `mc` is the momentum constant that defines the amount of momentum. `mc` is set between 0 (no momentum) and values close to 1 (lots of momentum). We need to be able to compute the gradient G of the loss function with respect to each weight of the network. It tells us how a small change in that weight will affect the overall error. Figure 7 shows the gradient value of the network.

Figure 8 shows the performance analysis. This figure doesn't indicate any major problems with the training. The validation and test curves are very similar. If the test curve had increased significantly before the validation curve increased, then it is possible that some over fitting might have occurred. Here the best validation performance is 0.015058 at epoch 11.

The mean square error rate for training and testing data has shown in table 5.

Table 5: mean square error rate for training and testing data

Ann data	Mean square error
Training data	0.00175
Testing data	0.00194

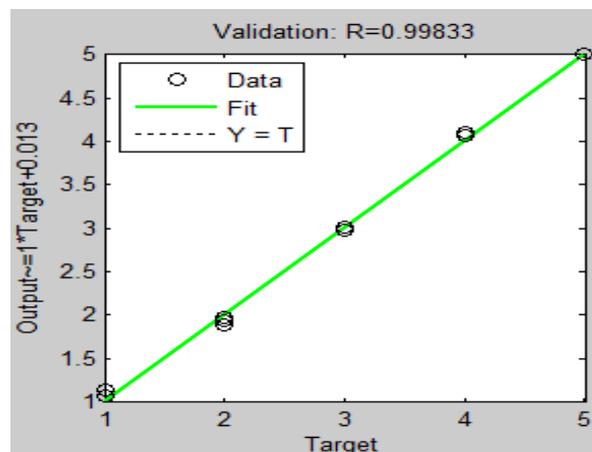


Figure 5: plot regression for validation

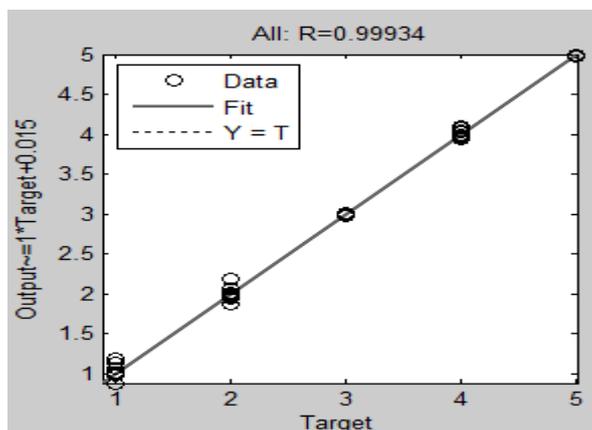


Figure 6: plot regression for training, validation and testing regression.

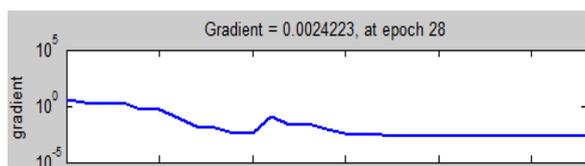


Figure 7: gradient value

Table 4: performance evaluation

Ann network	performance	prediction
5-5-1	0.734	55%
5-7-1	0.832	72%
5-8-1	0.868	95%

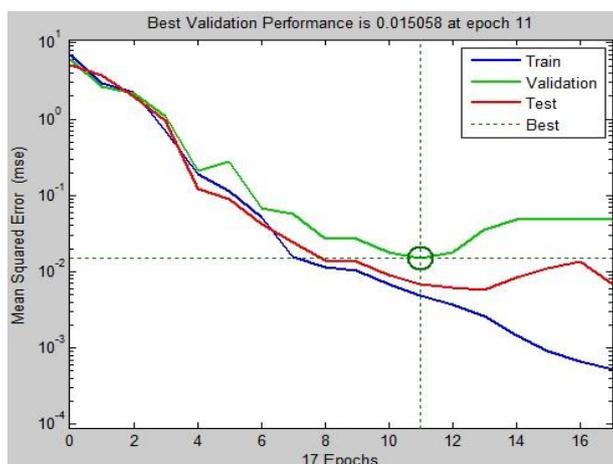


Figure 8: performance validation

VI. SCOPE AND FUTURE WORK

The textile fiber classification with other neural network applications like radial basis neural network, recurrent neural network can also be useful for classifying the fibers which will decrease the overhead calculation time for several thousand fibers. The area and color prediction can also be done by collecting more data sets collected from different places in the world which needs huge amount of searching of different fibers from different corners of the world.

REFERENCES

- [1] An overview of sewing threads mechanical properties on seam quality by Sumit Mandal, Nandita Abraham, Pearl Academy of Fashion, New Delhi.
- [2] FABRIC FORMING SYSTEMS By Peter Schwartz, Trever Rhodes and Mansour Mohamed
- [3] TEXTILE IDENTIFICATION, CONSERVATION, AND PRESERVATION By Rosalie Rosso King.
- [4] Characterization and Factors Effecting Fiber Properties Roger M. Rowell^{1,2}, James S. Han¹ and Jeffrey S. Rowell².
- [5] Atchison, J.E., Data on Non-Wood Plant Fibers XVII, Pulp and Paper Manufacture Volume 1, Properties of Fibrous Raw Materials and their Preparation for Pulping, M. J. Kocurek, Series Editor, The Joint Textbook Committee of the Paper Industry, 157, 1983b.
- [6] Bhargava, R.L.V., Bamboo, Pulp and Paper Manufacture Volume 3, Secondary Fibers and Non-wood Pulping, M.J. Kocurek, Series Editor, The Joint Textbook Committee of the Paper Industry, 71, 1987..
- [7] Physical Properties of Abaca (Musa textiles nee) Fiber Reinforced High Impact Polystyrene (HIPS) Composites Agung Efriyo Hadi¹, 4*, Mohd. Sapuan Salit¹, Megat Mohd. Hamdan Megat Ahmad², Khairul Zaman Hj Mohd. Dahlan³ and Mustofa Usman⁴.
- [8] Physical Properties of Textile Fibers Web Address: <http://www.researchandmarkets.com/reports/617165/OfficeCode:OCSCB1966>, Published by Alirrehman.
- [9] Propertise-of-Natural-Fibre published by Muhammad Riyaz UI Islam.
- [10] PREDICTION OF COMPRESSION INDEX OF SOILS USING ARTIFICIAL NEURAL NETWORKS (ANNs) V. PHANI KUMAR and CH. SUDHA RANI, Postgraduate Student, Department of Civil Engineering, S.V.U.College of Engineering, Tirupathi. 517502 Associate Professor, Department of Civil Engineering, S.V.U.College of Engineering, Tirupathi. 517502.
- [11] Multilayer perceptrons and backpropagation learning Sebastian Seung 9.641 Lecture 4: September 17, 2002.

- [12] Rajasekaran S and G A Vijayalakshmi Pai, Neural Networks, Fuzzy Logic and Genetic Algorithms-Synthesis and Applications, Prentice-Hall of India, 2003 (ISBN 81-203-2186-3).
- [13] Sigmoid Activation Functions, Course Authors: Dr. John Ringwood, School of Electronic Engineering, DCU email: ringwood@eeng.dcu.ie
- [14] Chapter entitled 'Grading and classification of cotton fibres' in the book 'The Sugar Industry and Cotton Crops', Nova Science Publishers, NY. (in press).
- [15] Abhijit Majumdar, Prabal Kr. Majumdar & Bijan Sarkar , Application of Adaptive Neuro-Fuzzy System for the Prediction of Cotton Yarn Strength from HVI fibre properties, Journal of the Textile Institute, 2005, No 1, Pages 55-60.
- [16] Campbell, Flake C. (2008). Elements of Metallurgy and Engineering Alloys. ASM International. p. 206. ISBN 9780871708670.
- [17] Askeland, Donald R.; Phulé, Pradeep P. (2006). The science and engineering of materials (5th ed.). Cengage Learning. p. 198. ISBN 978-0-534-55396-8.
- [18] Bishop, Errett (1961), "A generalization of the Stone–Weierstrass theorem", *Pacific Journal of Mathematics* **11** (3): 777–783.
- [19] Min Max Normalization Based Data Perturbation Method for Privacy Protection YOGENDRA KUMAR JAIN Head of the Department e-mail: ykjain_p@yahoo.co.in SANTOSH KUMAR BHANDARE Research Scholar Vidisha (M. P.) 464001 India e-mail: santosh.mits@gmail.com