Josphat Igadwa Mwasiagi
School of Engineering, Moi University, PO Box 3900 (30100), Eldoret, Kenya

The Prediction of Yarn Elongation of Kenyan Ring-Spun Yarn using Extreme Learning Machines (ELM)

Napovedovanje raztezka preje za kenijsko prstansko prejo z uporabo ekstremnega strojnega učenja (ELM)

Original Scientific Article/*Izvirni znanstveni članek*
Received/*Prispelo* 12-2016 • Accepted/*Sprejeto* 03-2017

Abstract

The optimization of the manufacture of cotton yarns involves several processes, while the prediction of yarn quality parameters forms an important area of investigation. This research work concentrated on the prediction of cotton yarn elongation. Cotton lint and yarn samples were collected in textile factories in Kenya. The collected samples were tested under standard testing conditions. Cotton lint parameters, machine parameters and yarn elongation were used to design yarn elongation prediction models. The elongation prediction models used three network training algorithms, including backpropagation (BP), an extreme learning machine (ELM), and a hybrid of differential evolution (DE) and an ELM referred to as DE-ELM. The prediction models recorded a mean squared error (mse) value of 0.001 using 11, 43 and 2 neurons in the hidden layer for the BP, ELM and DE-ELM models respectively. The ELM models exhibited faster training speeds than the BP algorithms, but required more neurons in the hidden layer than other models. The DE-ELM hybrid algorithm was faster than the BP algorithm, but slower than the ELM algorithm.
Keywords: cotton yarn, elongation, backpropagation, extreme learning machines, prediction

Izvleček

Optimizacija izdelave bombažne preje vključuje različne procese, zato je napovedovanje kakovostnih parametrov preje pomembno področje raziskav. V naši raziskavi smo se osredotočili na napovedovanje raztezka bombažne preje. Vzorce bombažnih vlaken in prej smo zbrali v tovarnah v Keniji. Zbrane vzorce smo testirali pod standardnimi pogoji testiranja. Parametre bombažnih vlaken, strojne parametre in raztezek preje smo uporabili za izdelavo modelov napovedovanja raztezka preje z uporabo treh mrežnih učnih algoritmov – algoritma vzvratnega razširjanja (BP), algoritma ekstremnega strojnega učenja (ELM) in hibridnega algoritma diferencialne evolucije (DE) v kombinaciji z EML, poimenovanega DE-ELM. Modeli za napovedovanje raztezka preje so zabeležili vrednost srednje kvadratne napake (mse) 0,001 pri uporabi 11, 43 in 2 nevronov v skritem nivoju za BP, ELM oziroma DE-ELM. V primerjavi z algoritmi BP so modeli ELM dosegli največje hitrosti, vendar so potrebovali največ nevronov. Algoritem hibridnega DE-ELM je bil hitrejši od algoritma BP, vendar počasnejši od algoritma ELM.
Ključne besede: bombažna preja, raztezek, vzvratno razširjanje, ekstremno strojno učenje, napovedovanje

1 Introduction

Textile industry is one of the important industries, especially for developing countries like Kenya, due to its labour-intensive nature. The industry is vast and produces a variety of products that include fibres, yarns, fabrics and garments. The manufacture of cotton ring-spun yarn involves the assembling together of a group of fibres and then passing them through a chain of processes that include opening, cleaning, drafting and twisting to bind the fibres together, so that they form a continuous strand. The raw material

Corresponding Author/*Korespondenčni avtor:*
Prof DEng Josphat Igadwa Mwasiagi
Telephone: +254725868329
E-mail:igadwa@gmail.com

DOI: 10.14502/Tekstilec2017.60.65-72

(fibre) used in cotton spinning is normally characterised by variations that may be attributed to the variety or type, and to cotton growing conditions. The task of the spinner is to ensure that fibre selection and spinning processes produce yarn of acceptable quality at the lowest cost.This may entail the use of optimisation techniques for maximum productivity and profitability.

Prediction models for cotton yarn properties can be designed by using statistical, mathematical and artificial neural network (ANN) models. Ever since Cheng and Adams [1] reported the use of ANN models to predict yarn quality properties, the use of ANN in yarn quality property prediction models has grown in leaps and bounds, to the point where such models are used in the yarn spinning industry [2]. The need to improve the prediction of yarn quality properties thus cannot be overemphasised. This research paper compared the use of an extreme learning machine (ELM) with conventional backpropagation (BP) prediction models.

1.1 Yarn prediction models

The prediction of yarn quality properties using ANN can be accomplished using a single hidden layer feedforward network, whose network parameters include input to hidden layer weights ($\mathbf{W}^1$), hidden layer biases ($\mathbf{b}^1$), a hidden layer transfer function (f^1), hidden layer to output layer weights ($\mathbf{W}^2$), output layer biases ($\mathbf{b}^2$) and an output layer transfer function (f^2) as shown in Figure1. One of the most commonly used techniques to train a feedforward network is the backpropagation algorithm, where the weights and biases are iteratively updated until the set target error is attained. Feedforward networks have been used by several researchers to predict yarn quality properties [1–3]. According to Huang et al, [4–5] the weights and biases of a single hidden layer network can be randomly selected and then processed through the hidden layer transfer function (f^1). Eliminating the output layer function (f^2) in a single hidden layer network can render a single hidden layer feedforward network a linear system. The hidden layer to output layer weights ($\mathbf{W}^2$) can thus be analytically determined using a generalised inverse operation. Such modified networks were given the name extreme learning machines (ELM).

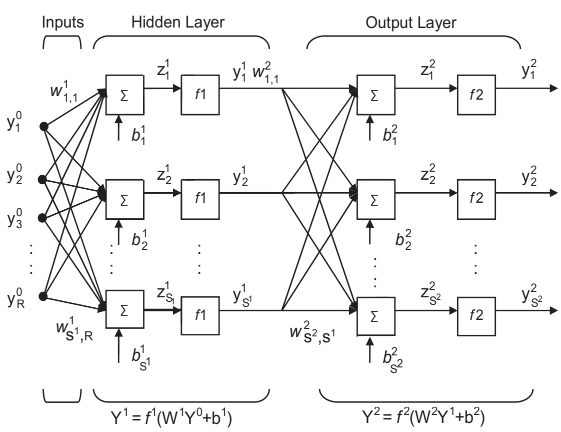

Figure 1: Single hidden layer feedforward network

Since an ELM chooses the input weights and hidden layer biases randomly, much of the training time traditionally spent in iteratively updating these parameters is saved. However, because the output weights are computed based on the prefixed input weights and hidden layer biases, there is a possibility that a set of non-optimal or unnecessary input weights and hidden layer biases could be selected. Research by Zhu et al [6] suggested that the problems associated with the ELM algorithms can be minimised by using the DE algorithm for the selection of initial weights and biases. This idea was implemented by combining the differential evolution (DE) and ELM algorithms to form a hybrid training algorithm. The hybrid algorithm, thereafter referred to as DE-ELM, works as follows: the DE algorithm selects the initial weights by using mutation, crossover and selection processes to search for the most suitable weights and biases. The selected weights and biases are sent to the ELM algorithm and used to train the yarn quality prediction model. It can thus be stated here that the difference between the operation of the ELM and the DE-ELM algorithms lies in the fact that the initial weights and biases of the ELM algorithm are selected randomly and then used for training, while in the DE-ELM algorithm the initial weights are also randomly selected, but are first put through the DE process (mutation, crossover and selection) before being used for network training.

2 Materials and methods

2.1 Materials

The aim of this research work was to design a yarn quality prediction model with special emphasis on Kenyan cotton lint and ring-spun yarn. Cotton lint

Table 1: Details of cotton lint and yarn samples

Cotton lint	Mill code	Yarn linear density[tex]	No. ofcops	Spindle speed [revolutions/min]
Voi AR	B	19.68	24	11,000
Voi AR	B	29.525	24	10,000
WT AR	A	19.68	24	12,000
Kitui AR	A	19.525	24	12,000
Kitui AR	A	24.604	24	11,000
Kitui AR	C	24.604	24	8,000

Table 2: Inputs for yarn elongation prediction models

No.	Input Type	Input factor	Input value		
			Minimum	Maximum	Mean
1	HVI fibre property	Length [mm]	24.77	33.45	29.30
2	HVI fibre property	Length uniformity [%]	78.30	87.30	83.44
3	HVI fibre property	Micronaire	3.27	5.89	3.90
4	HVI fibre property	Maturity ratio	0.82	0.93	0.86
5	HVI fibre property	Spinning consistency index (SCI)	108	187	157
6	HVI fibre property	Short fibre index (SFI)	5.90	9.70	7.39
7	HVI fibre property	Strength [cN/tex]	21.09	35.81	29.10
8	HVI fibre property	Elongation [%]	3.98	8.84	6.13
9	HVI fibre property	Trash Grade	1	4	2
10	HVI fibre property	Yellowness (+b)	8.90	12.30	10.69
11	Yarn parameter	Yarn count [tex]	19.41	31.56	23.59
12	Yarn parameter	Twist [m^{-1}]	701.18	936.61	848.425
13	Machine parameter	Spindle speed [revolutions/min]	8,000	12,000	10,851
14	Machine parameter	Ring diameter [mm]	42	50	43

and ring-spun yarn samples were collected from textile mills in Kenya, with care being taken to ensure that the selected factories were similar in terms of machinery, work culture, technology, quality and maintenance policies. This was done with the aim of minimising sample variances that could arise due to inter-factory differences. The details of the cotton lint and yarn samples collected are presented in Table1. The data used in this research work were compiled after the collection and testing of the samples. The collected data consisted of 144 samples each having 14 factors that were deemed as input factors, as shown in Table 2.The 14 factors included cotton fibre properties, machine parametersand yarn quality properties. The output of the prediction model was yarn elongation.

2.2 Methods

Three types of prediction models were designed in this research work. As is the practice in network training, the input data were pre-processed. Data pre-processing included the data normalisation process, where the inputs were scaled to fall within a set limit [7]. As mentioned earlier, a total of 144 input data were used. The data were subdivided into three sets: training, validation and testing sets in a ratio of 4:1:1 respectively. This was done randomly. All the networks used a single hidden layer feedforward network with one output (yarn elongation). The BP-trained prediction model was designed using three layers (e.g. input, hidden and output layers), as discussed by Mwasiagi et al [3]. The BP algorithm referred to as the Levenberg-Marquardt algorithm (LM) [8] was used to train the elongation prediction model, while performance was assessed using a mean squared error (mse) and correlation coefficient (R-value), as explained by Ham and Kostanic et al [7] and applied by Mwasiagi et al [3]. The performance of the BP algorithm was monitored as the number of neurons in the hidden layer was varied from 2 until the network-set mse level of 0.001 was attained. The training error, time and R-value were recorded.

The second training algorithm used to train the elongation prediction models was the ELM algorithm. This was done in similar manner as the BP algorithm, until the set mse level was attained. The ELM model was improved by using the DE-ELM hybrid algorithm, which used the DE algorithm for the selection of the initial weights and biases. The network was trained thereafter using the ELM model.The performance of the DE-ELM yarn quality property prediction models was monitored as the number of generations was varied from 1 to 10 in increments of 1. For comparison purposes, the number of neurons attained by the BP algorithm to achieve the set mse level of 0.001 (which was 11) was varied from 11 to 2 in increments of 1.

3 Results and discussion

3.1 Prediction of yarn elongation using BP algorithms

Using the BP algorithms, the elongation prediction model was trained starting from 2 neurons in the hidden layer. The number of neurons was increased in steps increments of 1 until the set target error (0.001) was attained. The results of the elongation prediction model, as presented in Table3, showed that the yarn prediction model attained the set target error when the number of neurons in the hidden layer reached 11. That process took 1.58 seconds. The fully trained yarn elongation model with 11 neurons in the hidden layer was exposed to the testing data and the R-value of the BP yarn elongation model was 0.894 (Figure 2). Table 4 presents the predicted and measured values of yarn elongation.

Table 3: Results of the BP elongation prediction model

Neurons	$(mse)_{tr}$	Time [s]	Iteration
2	0.064	1.282	15
3	0.039	1.297	18
4	0.0211	1.307	19
5	0.0137	1.32	18
6	0.0091	1.36	20
7	0.0073	1.39	22
8	0.0052	1.42	24
9	0.0032	1.45	24
10	0.00199	1.51	26
11	0.00089	1.58	25

Table 4: Predicted and measured values for the BP model

No.	Predicted value [%]	Measured value [%]
1	7.71	7.35
2	7.27	7.25
3	7.10	7.33
4	7.67	7.40
5	7.4	6.91
6	6.89	7.11
7	6.96	7.27
8	7.02	6.84
9	6.73	6.87
10	6.76	6.77
11	6.91	6.91
12	6.59	6.60
13	7.32	7.20
14	7.21	7.09
15	6.65	6.82
16	6.27	6.92
17	6.75	6.54
18	5.66	6.32
19	6.60	6.68
20	6.63	6.65
21	5.77	6.08
22	6.18	6.16
23	5.85	6.01
24	5.59	6.28

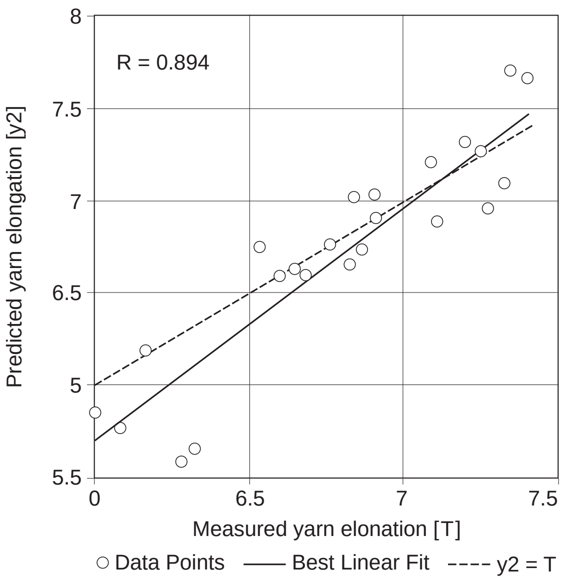

Figure 2: Predicted and measured values of the BP algorithm

3.2 Prediction of yarn elongation using the ELM algorithm

The experiments for the prediction of yarn elongation using the ELM algorithms were carried out starting with 2 neurons in increments of 1 until the set target error was attained. The results of the ELM elongation prediction model, as shown in Table 5, improved rapidly, especially when the number of neurons was varied from 2 to 12 in increments of 1.

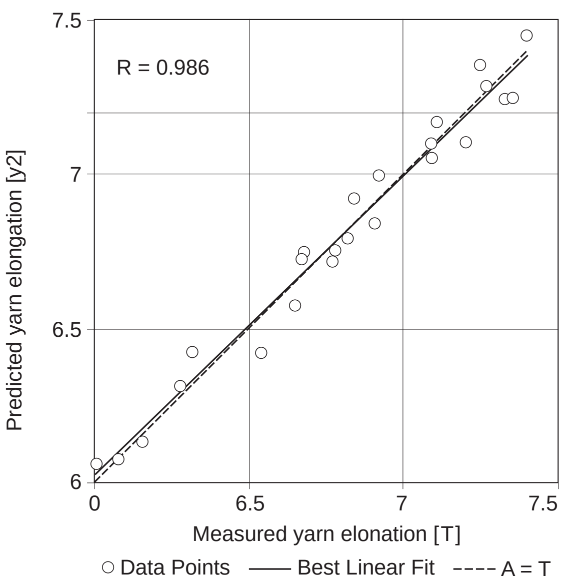

Figure 3: Predicted and measured values of the ELM model

Table 5: Results of the ELM elongation prediction model

No. of neurons	$(mse)_{tr}$	Time [s]
2	0.09554	0.01516
3	0.08128	0.0155
4	0.07258	0.0160
5	0.05929	0.0164
6	0.04840	0.0166
7	0.04000	0.0169
8	0.03168	0.0173
9	0.02111	0.0177
10	0.01646	0.0180
11	0.01000	0.0183
12	0.00941	0.0189
13	0.00810	0.0191
14	0.00774	0.0194
15	0.00706	0.0196
16	0.00624	0.0202
17	0.00518	0.0205
18	0.00436	0.0208
19	0.00372	0.0215
20	0.00281	0.0216
21	0.00260	0.0219
22	0.00230	0.0224
23	0.00221	0.0226
24	0.00203	0.0229
25	0.00194	0.0233
26	0.00197	0.0236
27	0.00185	0.0239
28	0.00185	0.0244
29	0.00185	0.0247
30	0.00182	0.0251
31	0.00176	0.0255
32	0.00168	0.0258
33	0.00166	0.0262
34	0.00160	0.0265
35	0.00157	0.0269
36	0.00152	0.0273
37	0.00150	0.0276
38	0.00116	0.0279
39	0.00112	0.0282
40	0.00112	0.0313
41	0.00111	0.0301
42	0.00102	0.0301
43	0.00097	0.0317

Table 6: Predicted and measured values for the ELM model

No.	Predicted value [%]	Measured value[%]
1	7.41	7.35
2	7.19	7.25
3	7.26	7.33
4	7.45	7.4
5	6.89	6.91
6	7.05	7.11
7	7.35	7.27
8	6.96	6.84
9	6.81	6.87
10	6.81	6.77
11	6.95	6.91
12	6.79	6.6
13	7.27	7.2
14	7.16	7.09
15	6.84	6.82
16	6.74	6.92
17	6.5	6.54
18	6.41	6.32
19	6.68	6.68
20	6.46	6.65
21	6.19	6.08
22	6.13	6.16
23	5.92	6.01
24	6.29	6.28

Thereafter, the change in the mse value was very small, requiring an increase of over 30 neurons to change from an mse value of 0.00941 to 0.00097, when the set target error was attained. The elongation prediction model needed 43 neurons in the hidden layer, and for that it took only 0.0317 seconds for training. This is much faster than the BP model, which needed 1.58 seconds to attain the set mse level of 0.001. When exposed to the testing data, the 43-neuron elongation prediction model had an R-value of 0.986 (Figure 3). Table 6 presents the predicted and measured values of the ELM elongation model.

3.3 Prediction ofyarn elongation using the DE-ELM algorithm

The ELM elongation prediction model needed 43 neurons in the hidden layer in order to achieve the set target error of 0.001. An improvement was made to the ELM algorithm using the DE-ELM model, which as previously mentioned is a hybrid of the DE and ELM algorithms. The training of the DE-ELM elongation model was carried out in a similar manner as that of the yarn elongation model discussed in section 3.1. The results of the training of the DE-ELM elongation model are presented in Table 7. The level at which the model attained the set target error are marked in Table 7, while the model is presented in detail in Table 8.

As is evident from the table above, the two-neuron model required five generations to attain the set target mean square error (mse) of 0.001. As the number

Table 7: Variation of $(mse)_{tr}$ for the DE-ELM elongation model

G	Number of neurons in the hidden layer									
	11	10	9	8	7	6	5	4	3	2
1	0.00326	0.0033	0.004134	0.005446	0.006856	0.008317	0.015876	0.017983	0.024461	0.028023
2	0.00127	0.0019	0.00254	0.00149	0.001632	0.001731	0.003238	0.005141	0.008281	0.011881
3	0.00035	0.0002	0.000462	0.000595	0.000784	0.000912	0.00103	0.001303	0.002362	0.006336
4	$7.6x10^{-5}$	$2.6x10^{-5}$	$6.6x10^{-6}$	$3.0x10^{-5}$	$7.6x10^{-5}$	0.000112	0.000306	0.000396	0.000586	0.002411
5	$1.8x10^{-7}$	$3.2x10^{-8}$	$2.7x10^{-10}$	$7.3x10^{-10}$	$1.0x10^{-8}$	$7.1x10^{-5}$	$5.6x10^{-5}$	$7.8x10^{-6}$	$4.6x10^{-5}$	$5x10^{-5}$
6	$1.9x10^{-16}$	$1.9x10^{-13}$	$3.3x10^{-13}$	$2.3x10^{-15}$	$1.7x10^{-12}$	$8.3x10^{-7}$	$8.8x10^{-7}$	$3.2x10^{-10}$	$1.2x10^{-6}$	$2.25x10^{-6}$
7	$9.8x10^{-16}$	$7.9x10^{-23}$	$1.1x10^{-25}$	$1.2x10^{-22}$	$1.9x10^{-16}$	$1.4x10^{-13}$	$1.4x10^{-12}$	$5.1x10^{-16}$	$6.8x10^{-8}$	$4x10^{-8}$
8	$2.6x10^{-31}$	$1.5x10^{-31}$	$2.6x10^{-29}$	$8.4x10^{-26}$	$3.0x10^{-23}$	$2.2x10^{-21}$	$3.2x10^{-24}$	$3.7x10^{-19}$	$1.4x10^{-17}$	$6.4x10^{-11}$
9	$1.7x10^{-34}$	$2.1x10^{-33}$	$7.8x10^{-32}$	$1.7x10^{-30}$	$1.2x10^{-28}$	$3.6x10^{-27}$	$2.6x10^{-26}$	$1.2x10^{-26}$	$6.2x10^{-21}$	$7.9x10^{-19}$
10	0	$1.2x10^{-34}$	$2.1x10^{-33}$	$8.9x10^{-33}$	$1.9x10^{-31}$	$8.7x10^{-31}$	$6.3x10^{-30}$	$2.6x10^{-28}$	$3.3x10^{-25}$	$9.8x10^{-23}$

of neurons was reduced from 11 to 2 in increments of 1, the number of generations needed to attain the set target error increased from 3 to 5. The two-neuron, five-generation model was exposed to the testing data. The results are presented in Figure 4 and Table 9. The R-value for the DE-ELM elongation was 0.994.

Table 8: Results of the DE-ELM elongation prediction model

Neuron	G	$(mse)_{tr}$	R-value	Time [s]
11	3	0.00035	0.987	10.000
10	3	0.0004	0.986	0.8925
9	3	0.000462	0.986	0.875
8	3	0.000695	0.986	0.8906
7	3	0.000784	0.985	0.6719
6	3	0.000912	0.981	0.625
5	4	0.000306	0.981	0.9328
4	4	0.000396	0.98	0.6706
3	4	0.0006	0.978	0.6563
2	5	0.00005	0.977	0.6875

Figure 4 presents the measured and predicted values of the optimum elongation prediction model (DE-ELM) when exposed to the testing data. The predicted elongation values tracked the measured elongation values so closely that a success rate of 99.4% was achieved. This could be an indication of the model's very good generalisation property. It is evident from the graph in Figure 4 that a high proportion of the predicted values are either on or close to the best fit line. This could be an indication of the network's good generalisation.

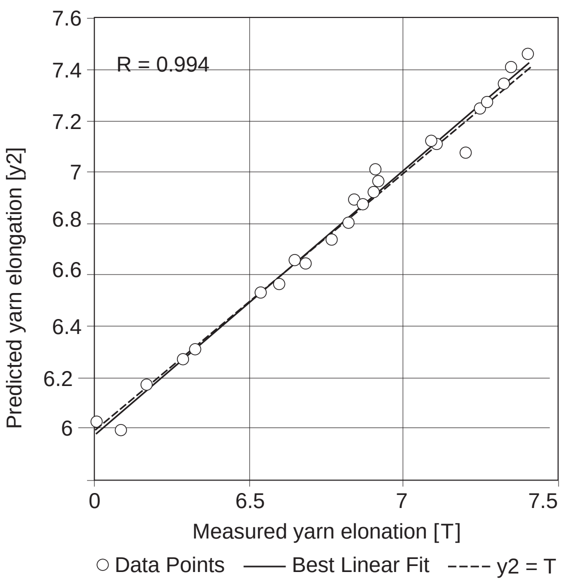

Figure4: Predicted and measured values for the DE-ELM model

Table 9: Predicted and measured values of yarn elongation

No.	Predicted value [%]	Measured value [%]
1	7.41	7.35
2	7.25	7.25
3	7.34	7.33
4	7.46	7.40
5	6.92	6.91
6	7.10	7.11
7	7.27	7.27
8	6.89	6.84
9	6.78	6.87
10	6.54	6.77
11	7.09	6.91
12	6.67	6.60
13	7.08	7.20
14	7.20	7.09
15	7.07	6.82
16	6.97	6.92
17	6.53	6.54
18	6.31	6.32
19	6.64	6.68
20	6.66	6.65
21	6.00	6.08
22	6.17	6.16
23	6.03	6.01
24	6.27	6.28

3.4 Comparison of the BP, ELM, and DE-ELM elongation models

Table 10 presents a comparison of the elongation prediction models obtained using the BP, ELM and DE-ELM. The ELM model needed 43 neurons in the hidden layer to reach the set target error of 0.001. This is one of the disadvantages of the ELM algorithm, i.e. that it requires more neurons than the BP prediction models.

Table 10: Comparison of elongation prediction models

Model	Input factors	No. of neurons	No. of generations	$(mse)_{tr}$	R-value	Iteration	Time [s]
BP	14	11	N/A	0.00089	0.894	25	1.58
ELM	14	43	N/A	0.00097	0.986	N/A	0.0317
DE-ELM	14	2	5	0.00005	0.994	N/A	0.6875

The ELM algorithm was however faster than the other models. The time required by the ELM model for training was over 80 times faster than that needed by the BP model. The DE-ELM model provides very good performance with a significantly reduced number of neurons in the hidden layer. Its training speed is, however, slower than that of the ELM model, but still much faster than that of the BP models. The DE-ELM elongation prediction model can therefore be considered the optimum model.

4 Conclusion

Yarn elongation prediction models using BP, ELM and DE-ELM models were designed and trained. The performance of the BP algorithms was compared to two non-BP algorithms, i.e. ELM and DE-ELM algorithms, during the prediction of yarn elongation. The model recorded an mse value of 0.001 using 11, 43 and 2 neurons in the hidden layer for the BP, ELM and DE-ELM models respectively. The ELM models exhibited the fastest training speeds relative to the BP algorithms, but needed more neurons in the hidden layer than the other models. The hybrid model (DE-ELM) was second in terms of speed after the ELM model. The performance of the DE-ELM model is thus far better than that of the BP model in terms of training time and better than the ELM model in terms of the number of neurons in the hidden layer.

References

1. CHENG Luo, ADAMS, David L. Yarn strength prediction using neural networks: Part I: Fibre properties and yarn strength relationship. *Textile Research Journal*, 1995, **65**(9), 495–500, doi: 10.1177/004051759506500901.
2. FURFERI, Rocco, GELLI, Maurizio. Yarn strength prediction: a practical model based on artificial neural networks. *Advances in Mechanical Engineering*, 2010, **8**, 1–10, ID 640103, doi: 10.1155/2010/640103.
3. MWASIAGI, Josphat Igadwa, WANG, Xin Hou, HUANG, Xiu Bao. Use of input selection techniques to improve the performance of an artificial neural network during the prediction of yarn quality properties. *Journal of Applied Polymer Science*, 2008, **108**(1), 320–327, doi: 10.1002/app.27586.
4. HUANG, Guang-Bin, Babri, HA. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Transactions on Neural Networks*, 1998, **9**(1), 224–229, doi: 10.1109/72.655045.
5. HUANG, Guang-Bin, CHEN, Lei, SIEW, Chee-Kheong. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 2006, **17**(4), 879–892, doi: 10.1109/tnn.2006.875977.
6. ZHU, Qin-Yu, QIN, AK., SUGANTHAN, PN., HUANG, Guang-Bin. Evolutionary extreme learning machine. *Pattern Recognition*, 2005, **38**(10), 1759–1763.
7. HAM, Fredric M., KOSTANIC, Ivica. *Principles of Neurocomputing for Science & Engineering*. 1st edition. Boston : McGraw-Hill Science/Engineering/Math, 2000, 642. ISBN-13: 978-0070259669.
8. HAGAN, MT., MENHAJ, MB. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 1994, **5**(6), 989–993, doi: 10.1109/72.329697.