Indian Journal of Fibre & Textile Research
Vol. 40, September 2015, pp. 267-272

Predicting the intermingled yarn number of nips and nips stability with neural network models

İlkan Özkan[1,a], Yusuf Kuvvetli[2], Pinar Duru Baykal[1] & Cenk Şahin[2]
[1]Department of Textile Engineering, [2]Department of Industrial Engineering Faculty of Engineering & Architecture, Çukurova University, 01330 Saricam-Adana, Turkey

Received 31 January 2014; revised received and accepted 10 June 2014

This study aims at predicting the effects of selected process parameters on nips stability and number of nips by using different artificial intelligence methods. Partially oriented polyester yarn with 283 dtex linear density and different numbers of filaments are intermingled with different speed and pressure levels. The feed forward neural network with multi-hidden layers (ML-FFNN) and general regression neural networks (GRNN) have been selected as artificial intelligence methods. The number of filaments, intermingling speed and pressure values are used as input variables on the artificial neural networks. The effects of number of hidden layers on the ML-FFNN and number of nodes in the hidden layer are investigated. Based on comparative results, the ML-FFNN is found to give better performance (at most 6%) than by GRNN in terms of prediction accuracy on train and test data sets. It can be concluded from this study that the neural networks has great ability to predict intermingling process parameters.

Keywords: Feed forward neural network, General regression neural network, Multi hidden layer, Intermingling, Nips stability, Number of nips

1 Introduction

Increasing economic constraints on the textile industry brought forward alternative and less expensive methods to conventional techniques. Intermingling is an alternative to sizing, twisting in texturing, drawing, spinning, and knotting in splicing, and also in blending as a new process. Therefore, the intermingling process, provided that the interest problems are overcome, appears to be promising for the future of textile industry[1]. Intermingled yarns number of nips and nips stability are important properties in terms of yarn structure. Speed and pressure levels are most effective parameters in intermingling processes. In the first instance, accurate prediction of these parameters is important for the production of yarn with the desired characteristics. For this purpose, there are a lot of different techniques available.

Artificial intelligence methods have various application areas in textile industry and there are different studies reported, about their application, in literature. Some of these studies are about classification for cotton yarn quality[2], prediction of physical and mechanical properties of yarn[3-6], characterization and evaluation of yarn surface appearance[7], optimizing the yarn spinning process[8], prediction of NEP rotor spinning yarn[9], fabric dying application[10-13], evaluating the apparent quality of knitted fabrics[14], and predicting thermal resistance of cotton fabrics[15]. There is a study about predicting the intermingled yarn strength and elongation properties by a single layer feed forward neural network model[16]. Most of studies, based on artificial neural networks, focus on determining and predicting the process parameters and product properties. Thus, the artificial neural networks can be considered as an alternative predicting method, besides the statistical and other mathematical models. However, it has been found that the study on the use of artificial intelligent-methods for predicting intermingled yarn properties is scanty. Hence, in the present study intermingling yarn number of nips and nips stability values are predicted using general regression neural networks and multi-layer feed forward neural network models. Speed, number of filaments and pressure values are selected as the input parameters of the prediction models.

Artificial neural networks (ANNs) are based on inspiration of physical nervous system. The computer and human brain have some similarities such as significantly complex, nonlinear and parallel computing[17]. Generalized regression neural networks (GRNN) models are one of the probabilistic neural networks model, proposed first time on Specht's

[a]Corresponding author.
E-mail: iozkan@cu.edu.tr

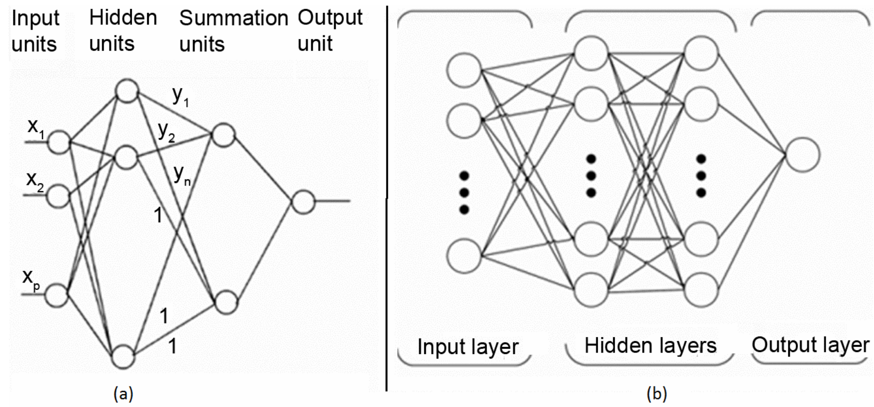

Fig. 1—(a) Map of a GRNN, and (b) Map of a multi-layer feed forward neural network

study. GRNN estimates to continuous variable and converges to regression surface that can be linear or nonlinear. As GRNN models do not need an iteratively calculation algorithm, lots of different applications such as prediction and modelling can be built by using GRNN[18].

GRNN models are beneficial especially on continuous variables prediction such as time series and regression. This model estimates expected value of dependent variable by using set of independent variables. To find the expected value of dependent variable, the integration over a probability density function (pdf) is necessary. A nonparametric estimator known as Kernel estimation or Parzen windows is used for finding the pdf[19].

The network map of the GRNN model is given in Fig. 1(a) [20]. There are four layers on GRNN, such as input layer, hidden layer, summation layer and output layer. Independent variables of the regression are modeled as inputs of the input layer. Hidden layer provides the training pattern. There are two summation neurons, same as hidden neuron values and weighted by y values. The output is calculated by the last layer.

Multi-layer feed forward neural networks (ML-FFNN), one of the methods of ANN models, use a back-propagation learning algorithm to train the network. Even though ML-FFNN has some disadvantages, such as slow converging ability for some approximation functions and increasing training time with increasing number of weights, it has some abilities such as learning, nonlinearity, input-output mapping and robustness[21].

The map of ML-FFNN with two hidden layers is illustrated in Fig. 1(b). According to Fig. 2, ML-FFNN has three or more layers, namely input layer, hidden layer, output layer. Input or independent variables of system represent as input units or nodes in the input layer. The second or more layers are called hidden layers. There are different numbers of nodes in the hidden layers for providing nonlinearity ability. The last layer is called output layer. The outputs of the system are represented by output units or node in this layer.

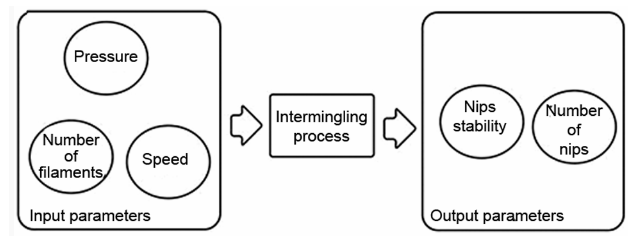

Fig. 2—Input and output parameters of the prediction system

2 Materials and Methods

2.1 Materials

Partially oriented polyester yarns (POY) with various numbers of filaments (34, 68, 47 and 100) were intermingled. All these filaments have round cross-section and theoretical number of all POY filaments is 283 dtex.

2.2 Methods

POY bobbins were intermingled with Hemaks HMX 114 model intermingling machine. Intermingling was performed using three different take up speed (150, 300, 450 m/min) and pressure (3, 5, 6 bar) levels. TEMCO Y profile LD 22 air-jet was used in this process. Number of nips and nips stability of intermingled yarns are tested using Itemat Lab TSI test device. Each bobbin was tested 10 times and the mean values are shown in Table 1. Thus, 360 different test instances are used for both ML-FFNN models and GRNN models.

Table 1—Number of nips and nips stability test results of intermingled yarns

POY bobbins	150 m/min			300 m/min			450 m/min		
	3 bar	5 bar	6 bar	3 bar	5 bar	6 bar	3 bar	5 bar	6 bar
Number of nips per meter									
POY 1 (283F34)	40.10	53.10	55.60	46.20	61.80	72.50	55.00	60.30	61.00
POY 2 (283F47)	63.00	79.30	73.20	65.60	77.60	74.20	59.80	66.70	63.40
POY 3 (283F68)	58.70	76.50	65.10	66.70	80.00	76.70	47.00	70.10	66.80
POY 4 (283F100)	80.50	82.30	84.40	55.20	50.80	72.20	67.40	77.80	73.20
Nips stability, %									
POY 1 (283F34)	62.00	80.50	84.30	62.70	78.50	85.70	93.90	97.50	96.90
POY 2 (283F47)	91.40	96.90	97.20	95.00	93.90	94.80	94.80	96.60	81.80
POY 3 (283F68)	82.00	95.90	91.20	81.40	93.20	87.50	65.30	97.70	96.90
POY 4 (283F100)	89.30	94.60	96.80	71.60	76.80	96.60	87.50	98.20	97.10

The test results are used as input data for the neural network models (ML-FFNN and GRNN). As shown in Fig. 2, the structure of prediction system has two main units, namely input and output parameters. The methodology of implementing ML-FFNN and GRNN is composed at five steps as given follows:

(i) Creating training, test and validation data sets – the data sets are composed of randomly divided main data set.

(ii) Building neural network model – the structure and parameters of neural network are selected in this step.

(iii) Training the neural network model – in this step, built model is trained with the training data set until the stopping criterion is reached.

(iv) Calculating the performance measures – performance of the model for the train, test and validation data sets is calculated.

(v) Evaluating the prediction performance – prediction accuracy for both train and test data set is evaluated.

All of the neural network models are implemented by using Matlab software package. The model performance could be affected by number of nodes on each hidden layer. For this reason, the ML-FFNN models are implemented by different number of nodes in the hidden layer and the best one is selected. The best ML-FFNN and GRNN models are compared by their train and test data set in terms of prediction accuracies.

Eighty per cent of the data set is selected for training data set and remain instances for testing data set and same data sets are used to compare methods objectively. Validation provides to generalize the model and stop training to avoid over fitting. Therefore, five percentage of the train data set is selected randomly as validation data set for the ML-FFNN models.

3 Results and Discussion

3.1 Setting up to ML-FFNN Model

Neural network model can behave differently under different conditions, such as number of hidden layer, number of nodes in the hidden layer, changing training function, etc. Due to the fact that parameter selection can cause bad performance of prediction, different parameter values should be tried to find the best prediction performance. Therefore, seventy (ranging from 1 to 70) number of nodes in the hidden layerare tested for the ML-FFNN, which has one layer (Fig. 3). The best performance of R value is obtained for 35 nodes in the hidden layer for predicting stability of nips. Same results are found for predicting number of nips when the number of nodes in the hidden layer is set to 33. The ML-FFNN which has 35 nodes in the hidden layer provides better mean R value than 33 nodes for both prediction of number of nips and nips stability. Therefore, the number of nodes in the hidden layer is set to 35.

Five different numbers of hidden layer are also analyzed (Fig. 4). The best performance of R value is reached with four hidden layers for predicting stability of nips. When the number of hidden layer is two, similar results are found for predicting the number of nips. The ML-FFNN model is built for both of the number of hidden layers 2 and 4 to select best model.

The other parameters determined are maximum number of epochs is set to1000; Levenberg-Marquardt is used as training function; and the performance of training is analyzed by mean squared error (MSE). Mentioned parameters are used for both ML-FFNN models.

3.2 Type of ML-FFNN Model

3.2.1 Two Hidden Layers (ML2-FFNN)

Effect of different numbers of hidden layer has been discussed before. In this section two hidden layer

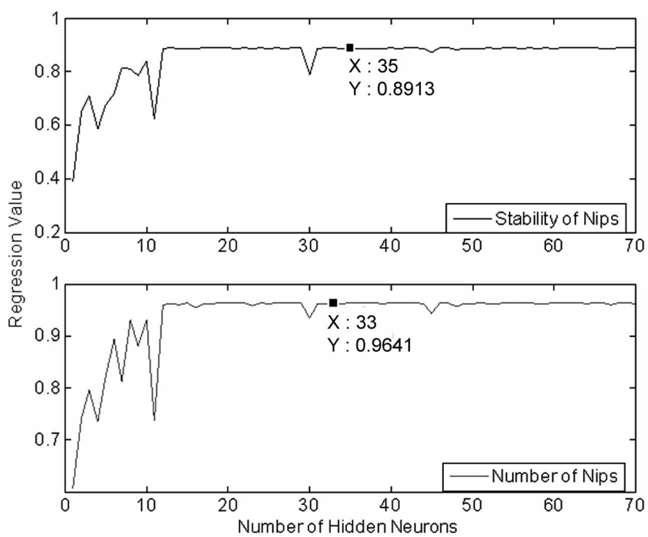

Fig. 3—Performance evaluation for different number of hidden neurons

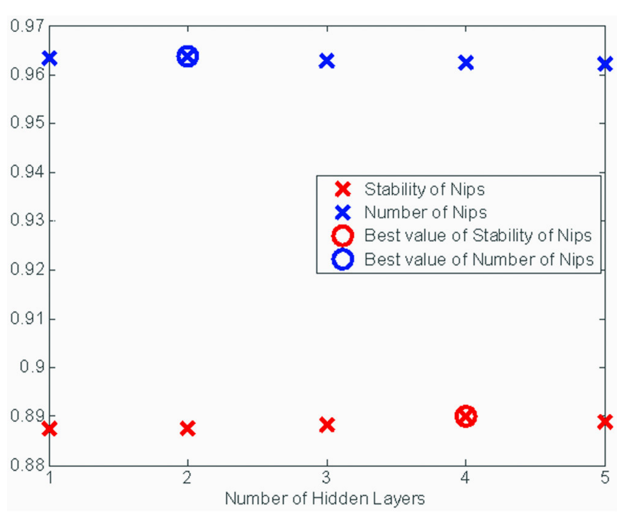

Fig. 4—Performance evaluation for different number of nodes in the hidden layer

models are implemented. The model is applied and the training performance of the ML2-FFNN is reached in epoch 4 with the best value of MSE as 10.5935 to predict the number of nips and nips stability.

Training, validation, test and overall performances of the ML2-FFNN are shown in Fig. 5. According the results, overall R and R^2 value are calculated as 0.96266 and 0.92671 respectively. The ML2-FNN model predicts successfully for all instances. The

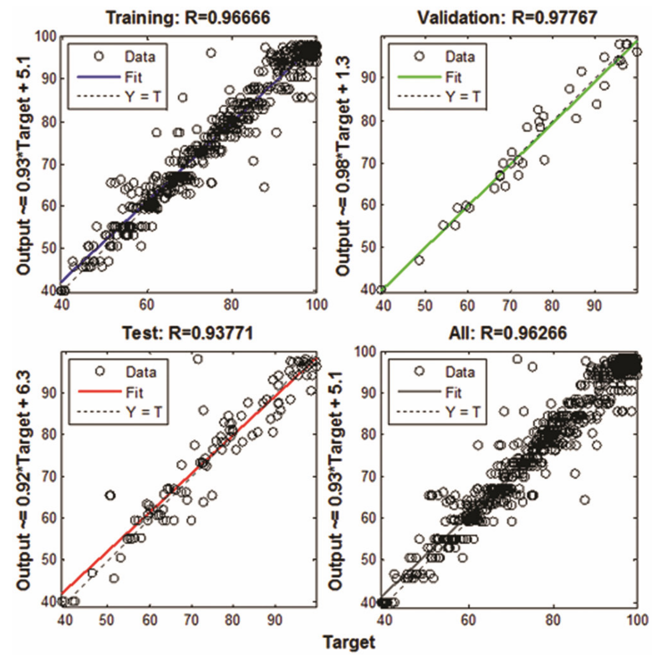

Fig. 5—ML2-FNN fitting graph

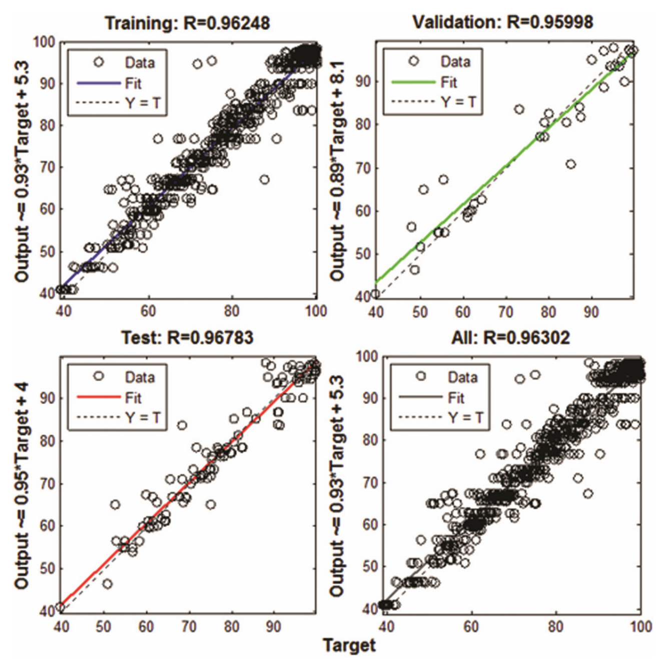

Fig. 6—ML4-FNN fitting graph

Table 2—Comparison of different ML-FFNN models

Model	Train data results				Test data results			
	Nips stability		Number of nips		Nips stability		Number of nips	
	R	R^2	R	R^2	R	R^2	R	R^2
ML2-FFNN	0.9014	0.8125	0.9679	0.93683	0.8157	0.6654	0.9357	0.8755
ML4-FFNN	0.887	0.7868	0.964	0.9293	0.9107	0.8294	0.948	0.8987
% Difference	1.44	2.57	0.39	0.753	-9.5	-16.4	-1.23	-2.32

Table 3—Comparison of ML-FFNN and GRNN models

Method	Train data results				Test data results			
	Nips stability		Number of nips		Nips stability		Number of nips	
	R	R^2	R	R^2	R	R^2	R	R^2
FFNN Model (ML4-FFNN)	0.887	0.7868	0.964	0.9293	0.9107	0.8294	0.948	0.8987
GRNN Model	0.8858	0.7846	0.9460	0.8949	0.8770	0.7691	0.9459	0.8947
% Difference	0.12	0.22	1.8	3.44	3.37	6.03	0.21	0.4

results demonstrate that the ML2-FFNN model explains 92.7% of the mean variation of both nips stability and number of nips.

3.2.2 Four Hidden Layers (ML4-FFNN)

In this section, four hidden layer model is implemented. To predict the number of nips and nips stability, the best value of MSE is found in epoch 3 with a value of 27.035.

Training, validation, test and overall performances of the ML4-FFNN are shown in Fig. 6. According to the results, overall R and R^2 value are calculated as 0.96302 and 0.92741 respectively. The ML4-FNN model predicts successfully for all instances. The results demonstrate that the ML4-FFNN model explains 92.74% of the mean variation of both nips stability and number of nips.

3.2.3 Comparison of ML-FNNN Models

In order to compare performance of different ML-FFNN models, training and test data results with both predictions of nips stability and number of nips are given in Table 2. The ML4-FFNN model has more accurate prediction than ML2-FFNN model, especially on test data set. The differences among models' R values increase about 16 per cent on prediction of nips stability. According to training datasets, the ML2-FFNN model has much better (2.57%) performance. However, the ML4-FFNN model has much accurate performance on test dataset (16.4%). Consequently, ML4-FFNN model is useful to predict nips stability and number of nips values.

3.3 GRNN Model

GRNN also use the same training and test data sets as used with ML-FFNN models. Training and test results of the GRNN are shown in Fig. 7. According to the

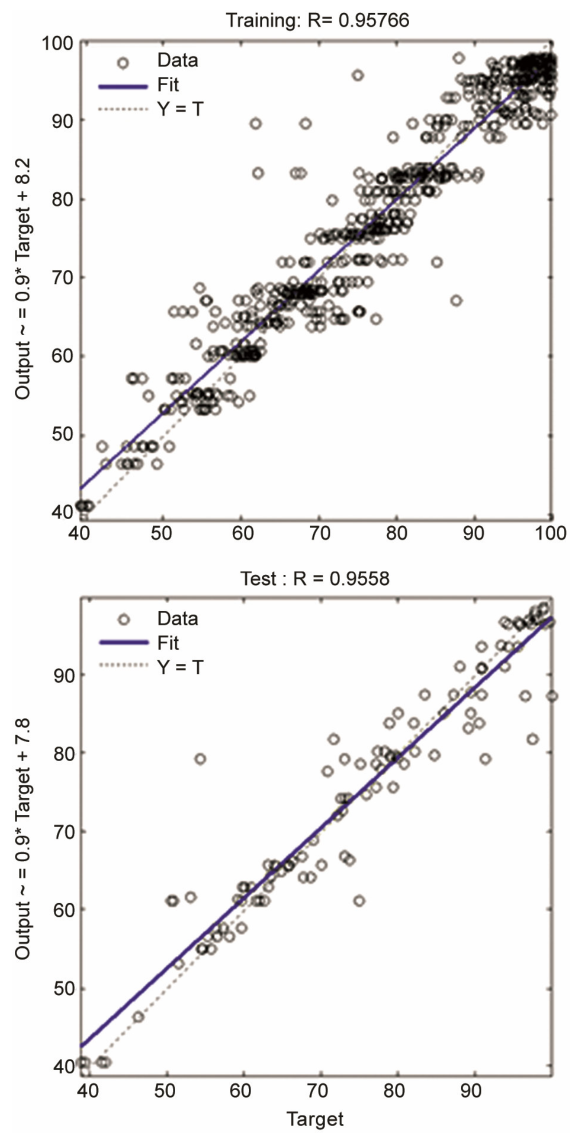

Fig. 7—GRNN fitting graph

training results, R and R^2 values are calculated as 0.95766 and 0.9171 respectively. The R and R^2 values are calculated as 0.9558 and 0.9136 respectively for the test results. The GRNN model predicts successfully for all instances. The results show that the GRNN model explains about 92% of the mean variation in both nips stability and number of nips.

3.4 Comparison between GRNN Model and ML-FNNN Model

In order to compare the performances of the best ML-FFNN and GRNN models, training and test data results with both predictions of nips stability and number of nips are analysed (Table 3). The results are seemed to be very close for both models. However, the ML-FFNN provides more accurate prediction fornumber of nips on training data set and nips stability on test data set. The differences among models' R values increase about 6 % on prediction of nips stability. Consequently, ML-FFNN model is the best model to predict nips stability and number of nips values.

4 Conclusion

The study shows that according to R and R^2 values, the ML4-FFNN model suppresses to GRNN and ML2-FFNN. It can be concluded that neural networks has great ability to predict intermingling process parameters. For the further studies, different input and output variables can be investigated, and different neural network methods may also combine each other to predict more accurately. Furthermore, the proposed prediction model (ML-FFNN) may be further refined with using new data set.

References

1 Demir A & Behery M H, *Synthetic Filament Yarn Texturing Technology* (Prentice-Hall, USA), 1997, 319.

2 Amin A E, *Knowledge-Based Systems,* 39 (2013) 124.

3 Faber K & Kowalski B R, *Chemometrics Intelligent Laboratory Systems,* 34 (1996) 293.

4 Demiryürek O & Koç E, *Fibers Polym,* 10(2) (2009) 237.

5 Khan Z, Lim A E, Wang L, Wang X & Beltran R, *Text Res J,* 79(8) (2009) 714.

6 Koç E & Demiryürek O, *Industria Text,* 62 (2011) 81.

7 Liang Z, Bingang X, Zheru C & Dagan F, *Expert Systems Applications,* 39 (2012) 4201.

8 Sette S, Boullart L & Langenhove L V, *Eng Appl Artificial Intelligence,* 9(6) (1996) 681.

9 Bo Z, *IERI Procedia,* 1 (2012) 248.

10 Senthilkumar M & Selvakumar N, *Dyes Pigm,* 68 (2006) 89.

11 Oğulata RT & Balci O, *J Text Inst*, 98(5) (2007) 409.

12 Balci O, Oğulata S N, Sahin C & Oğulata R T, *Fibers Polym,* 9(2) 2008 217.

13 Balcı O & Oğulata R T, *J Text Apparel*, 19(1) (2009) 61.

14 Semnani D & Vadood M, *Eng Appl Artificial Intelligence,* 23 (2009) 217.

15 Mitra A, Majumdar A, Majumdar P K & Bannerjee D, *Exp Thermal Fluid Sci,* 50 (2013) 172.

16 Özkan İ, Kuvvetli Y, Duru Baykal P & Erol R, *J Text Inst,* 105(11) (2014) 1203.

17 Haykin S, *Neural Networks a Comprehensive Foundation* (Prentice Hall International), 1991.

18 Specht Donald F, *IEEE Transact Neural Networks,* 2(6) (1991) 568.

19 Specht D F & Romsdahl H, Experience with Adaptive Probabilistic Neural Networks and Adaptive General Regression Neural Networks, *Proceedings*, *Neural Networks IEEE World Congress on Computational Intelligence,* (IEEE, USA), 2, 1994, 1203-1208.

20 Niwa T, *J Chem Info Computer Sci,* 43(1) (2003) 113.

21 Svozil D, Vladimir K & Jiȓí P, *Chemometrics Intelligent Laboratory Systems,* 39(1) (1997) 43.