UDK 669.14.018:519.68 ISSN 1580-2949
Original scientific article/Izvirni znanstveni članek MTAEC9, 44(3)147(2010)

COMPARATIVE MODELING OF WIRE ELECTRICAL DISCHARGE MACHINING (WEDM) PROCESS USING BACK PROPAGATION (BPN) AND GENERAL REGRESSION NEURAL NETWORKS (GRNN)

PRIMERJALNO MODELIRANJE ELEKTROEROZIJSKE ŽIČNE OBDELAVE (WEDM) Z UPORABO POVRATNOSTI (BPN) IN SPOŠNE NEVRONSKE REGRESIJSKE MREŽE (GRNN)

Onur Guven[1], Ugur Esme[2*], Iren Ersoz Kaya[3], Yigit Kazancoglu[4], M. Kemal Kulekci[5], Cem Boga[6]

[1]Mersin University, Engineering Faculty, Department of Mechanical Engineering, 33400, Mersin/Turkey
[2,3,5]Mersin University, Tarsus Technical Education Faculty Department of Mechanical Education, 33480, Tarsus-Mersin/Turkey
[4]Izmir University of Economics, Department of Business Administration, 35330, Balcova-Izmir/Turkey
[6]Cukurova University, Vocational School of Karaisali, 01770, Karaisali-Adana/Turkey
uguresme@gmail.com

Prejem rokopisa – received: 2010-01-22; sprejem za objavo – accepted for publication: 2010-03-02

The use of two neural networks techniques to model wire electrical discharge machining process (WEDM) is explored in this paper. Both the back-propagation (BPN) and General Regression Neural Networks (GRNN) are used to determine and compare the WEDM parameters with the features of the surface roughness. A comparison between the back-propagation and general regression neural networks in the modeling of the WEDM process is given. It is shown that both the back-propagation and general regression neural networks can model the WEDM process with reasonable accuracy. However, back propagation neural network has better learning ability for the wire electrical discharge machining process than the general regression neural network. Also, the back-propagation network has better generalization ability for the wire electrical discharge machining process than does the general regression neural network.

Keywords: WEDM, neural network, modeling, BPN, GRNN

Raziskana je uporaba dveh nevronskih mrež za modeliranje elekroerozijske žične obdelave (WEDM). Obe metodi: povratnostna (BPN) in splošna regresijska nevronska mreža (GRNN), sta uporabljeni za določitev in primerjavo VEDM-procesa. Dokazano je, da sta obe metodi primerni za modeliranje WEDM s sprejemljivo natančnostjo. Vendar pa ima povratnostna nevronska mreža boljšo sposobnost učenja in boljšo sposobnost posplošenja procesa kot splošna regresijska nevronska mreža.

Ključne besede: WEDM, nevronska mreža, modeliranje, BPN, GRNN

1 INTRODUCTION

Manufacturing industry is becoming ever more time-conscious with regard to the global economy, and the need for rapid prototyping and small production batches is increasing. These trends have placed a premium on the use of new and advanced technologies for quickly turning raw materials into usable goods; with no time being required for tooling.[1] Wire electrical discharge machining (WEDM) technology has been found to be one of the most recent developed advanced non-traditional methods used in industry for material processing with the distinct advantages of no thermal distortion, high machining versatility, high flexibility, rapid machining and high accuracy of complex parts.[2] The degree of accuracy of workpiece dimensions obtainable and the fine surface finishes make WEDM particularly valuable for applications involving manufacture of stamping dies, extrusion dies and prototype parts. Without WEDM the fabrication of precision workpieces requires many hours of manual grinding and polishing.[3–6]

The most important performance measures in WEDM are cutting speed, workpiece surface roughness and cutting width. Discharge current, discharge capacitance, pulse duration, pulse frequency, wire speed, wire tension, average working voltage and dielectric flushing conditions are the machining parameters which affect the performance measures.[3–7]

Tosun et al.[3] determined the effect of machining parameters on the cutting width and material removal rate based on the Taguchi method. Tosun and Cogun[4] investigated experimentally the effect of cutting parameters on wire electrode wear. Tosun et al.[5] investigated the effect of the cutting parameters on size of erosion craters (diameter and depth) on wire electrode experimentally and theoretically. Cogun and Savsar[6] investigated the random behaviour of the time-lag durations of discharge pulses using a statistical model for different pulse durations, pulse pause durations, and discharge currents in EDM.

Esme et al.[7] modeled the surface roughness in WEDM process using design of experiments and neural networks. Scott et al.[8] have developed formulas for the solution of a multi-objective optimization problem to select the best parameter settings on a WEDM machine.

They used a factorial design model to predict the measures of performances as a function of a variety of machining parameters. Wang and Rajurkar[9] have developed a WEDM frequency monitoring system to detect on-line the thermal load on the wire to prevent the wire from rupture. Spur and Schoenbeck[10] have investigated a finite element model and they have explained the impact of a discharge on the anode as a heat source on a semi-infinite solid whose size and intensity are time-dependent in WEDM. Tarng et al.[11] developed a neural network system to determine settings of pulse duration, pulse interval, peak current, open circuit voltage, servo reference voltage, electric capacitance and wire speed for the estimation of cutting speed and surface finish. Spedding and Wang[12] presented parametric combination by using artificial neural networks and they also characterized the roughness and waviness of workpiece surface and the cutting speed. Liao et al.[13] performed an experimental study to determine the variation of the machining parameters on the MRR, gap width and surface roughness. They have determined the level of importance of the machining parameters on the metal removal rate (MRR). Lok and Lee[14] compared the machining performance in terms of MRR and surface finish by the processing of two advanced ceramics under different cutting conditions using WEDM. Ramakrishnan and Karunamoorthy[15] developed an artificial neural network with Taguchi parameter design. Tsai et al.[16] relationships between the heterogeneous second phase and the machinability evaluation of the ferritic SG cast irons in the WEDM process. Sarkar et al.[17] studied on the features of trim cutting operation of wire electrical discharge machining of γ-titanium aluminide. Caydas et al.[18] developed an adaptive neuro-fuzzy inference system (ANFIS) for modeling the surface roughness in WEDM process.

As indicated in the previous studies, most of the research works are focused on the effect of machining parameters, discharge energy, theory and experimental verification crater formation on the wire electrode. The present study focused on the comparative modeling and prediction of surface roughness to compare the techniques of back propagation network (BPN) and general regression neural network (GRNN).

2 EXPERIMENTAL DETAILS

As shown in **Figure 1**, the experimental studies were performed on an *Acutex WEDM* machine tool. Different settings of pulse duration (t), open circuit voltage (V), wire speed (S) and dielectric flushing pressure (p) were used in the experiments. Table feed rate (8.2 mm/min), pulse interval time (18 µs), and wire tension (1800 g) are kept constant during the experiments[7].

AISI 4340 steel plate was used as a workpiece material with (150 × 150 × 10) mm dimensions. CuZn37 *Suncut* brass wire with 0.25 mm diameter and tensile strength of 900 N/mm^2 was used in the experiments. Workpiece average surface roughness (R_a) measurements were made by using *Phynix TR–100* portable surface roughness tester. Cut-off length (λ) and traversing length (l) were set as 0.3 mm and 5 mm, respectively. Pulse duration, open circuit voltage, wire speed and dielectric flushing pressure were selected as input parameters and surface roughness (R_a) was selected as an output parameter[7].

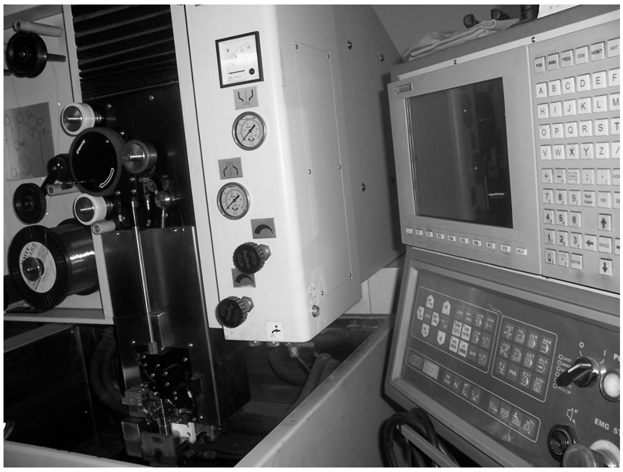

Figure 1: Acutex WEDM used in the experiments[7]
Slika 1: Acutex WEMM, uporabljen za preizkuse[7]

Four measurements were made and their average was taken as R_a value for a machined work surface. After collecting the experimental results both techniques namely back propagation neural network (BPN) and general regression neural network (GRNN) techniques were carried out to predict surface roughness (R_a).

3 ARTIFICIAL NEURAL NETWORKS (ANN)

It is well known that modeling the relationships between the input and output variables for non-linear, coupled, multi-variable systems is very difficult. In recent years, neural networks have demonstrated great

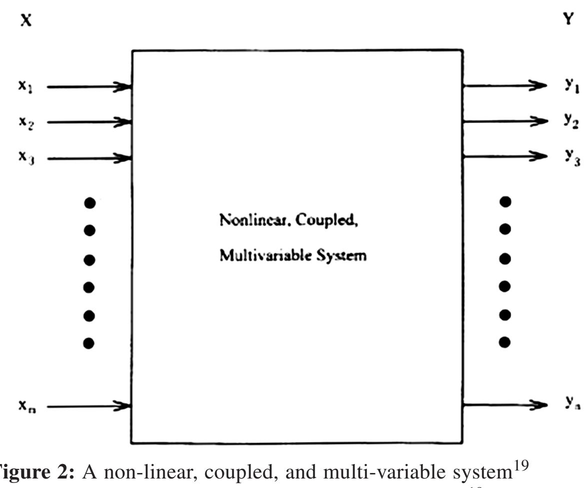

Figure 2: A non-linear, coupled, and multi-variable system[19]
Slika 2: Nelinearen, povezan in multivariabilni sistem[19]

potential in the modeling of the input–output relationships of complicated systems.[19,20] Consider that $X = \{x_1, x_2\ldots, x_m\}$ is the input vector of the system where m is the number of input variables and $Y = \{y_1, y_2\ldots, y_n\}$ is the corresponding output vector of the system where n is the number of output variables[19] as shown in **Figure 2**. In this section, the use of back-propagation and general regression networks to construct the relationships between the input vector X and output vector Y of the system will be explored.

3.1 Back-Propagation Networks (BPN)

The back-propagation network is composed of many interconnected neurons that are often grouped into input, hidden and output layers. The neurons of the input layer are used to receive the input vector X of the system and the neurons of the output layer are used to generate the corresponding output vector Y of the system. The back-propagation network used in this study is shown in **Figure 3**. For each neuron a summation function for all the weighted inputs are calculated as:

$$\mathrm{net}_j^k = \sum_j w_\mu^k o_i^{k-1} \tag{1}$$

where net^k_j is the summation function for all the inputs of the j-th neuron in the k-th layer, w^k_{ji} is the weight from the i-th neuron to the j-th neuron and o_i^{k-1} is the output of the i-th neuron in the (k–1)-th layer.

Setting 5-hidden layers resulted in lowest error between predicted and experimental results. Therefore, in the present work, 4-inputs, 5-hidden layer, 1 output layer (4 : 5 : 1 model) back propagation neural network has been used. The used BPN algorithm is shown in **Figure 3**.

As shown in Eq. (1), the neuron evaluates the inputs and determines the strength of each one through its weighting factor, i.e. the larger the weight between two neurons, the stronger is the influence of the connection.[19] The result of the summation function can be treated as an input to an activation function from which the output of the neuron is determined. The output of the neuron is then transmitted along the weighted outgoing connections to serve as an input to subsequent neurons. To modify the connection weights properly, a supervised learning algorithm involving two phases is employed.[21] The first is the forward phase which occurs when an input vector X is presented and propagated forward through the network to compute an output for each neuron.[19,20] Hence, an error between the desired output y_j and actual output o_j of the neural network is computed.[19] The summation of the square of the error E can be expressed as:

$$E = \frac{1}{2}\sum_{j-1}^{n}\left(y_j - o_j\right)^2 \tag{2}$$

The second is the backward phase which is an iterative error reduction performed in a backward direction. To minimize the error between the desired and actual outputs of the neural network as rapidly as possible, the gradient descent method, adding a momentum term,[21] is used. The new incremental change of weight $\Delta w^k_{ji}(n + 1)$ can be expressed as:

$$\Delta w_{ji}^k(n+1) = -\eta \frac{\partial E}{\partial w_{ji}^k} + \alpha \Delta w_{ji}^k(n) \tag{3}$$

where η is the learning rate, α is the momentum coefficient and n is the index of iteration. Through this learning process, the network memorizes the relationships between input vector X and output vector Y of the system through the connection weights.[19–21]

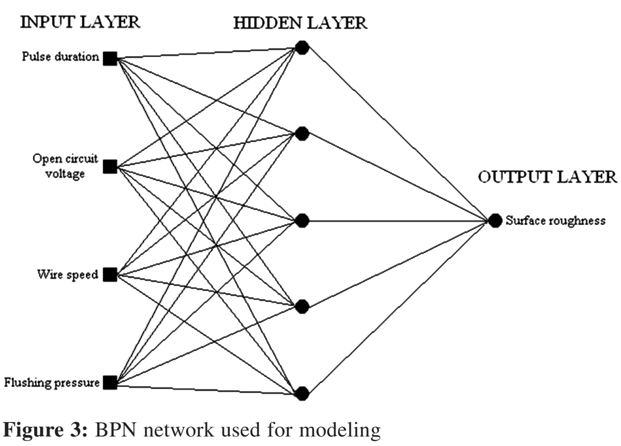

Figure 3: BPN network used for modeling
Slika 3: BPN mreža, uporabljena za modeliranje

4 GENERAL REGRESSION NEURAL NETWORKS (GRNN)

The General Regression Neural Networks (GRNN) introduced by Donald Specht in 1990 is a memory-based feed forward neural network based on the approximate estimation of the probability density function from observed samples using Parzen-window estimation.[22] It approximates any arbitrary function between input and output vectors. This approach removes the necessity to specify a functional form of estimation.

The method utilizes a probabilistic model between an independent random vector X (input) and a dependent scalar random variable Y (output). Let x and y be the particular measured values of X and Y, respectively, and $\hat{g}(x, y)$ is the joint continuous probability density function of X and Y. A good choice for a non-parametric estimate of the probability density function g is the Parzen window estimator as proposed by Parzen and performed for multidimensional cases by Cacoullos.[22–26] Given a sample of n real D dimensional x_i vectors and corresponding scalar y_i values, the estimate of joint probability density in GRNN is given by;

$$\hat{g}(x,y) = \frac{1}{(2\pi)^{(d+1)/2}\sigma^{(d+1)}}\frac{1}{n}\sum_{i=1}^{n}\left[\exp\left(\frac{(x-x_i)^T(x-x_i)}{2\sigma^2}\right)\exp\left(\frac{(y-y_i)^2}{2\sigma^2}\right)\right]$$

(4)

where σ is the window width of a sample probability, called the smoothing factor of the kernel[24]. The expected value of Y given x (the regression of Y on x) is given by;

$$E[Y/x]=\frac{\int_{-\infty}^{+\infty} Y\cdot g(x,Y)\mathrm{d}Y}{\int_{-\infty}^{+\infty} g(x,Y)\mathrm{d}Y} \tag{5}$$

Using Eq. (4), Eq. (5) becomes;

$$\hat{y}(x)=E[Y/x]=\frac{\sum_{i=1}^{n}\left[y_i\exp(d_i)\right]}{\sum_{i=1}^{n}\exp(d_i)} \tag{6}$$

where d_i is the distance between the input vector and the i^{th} training vector, and is given by;

$$d_i^2=-\frac{(x-x_i)^T(x-x_i)}{2\sigma^2} \tag{7}$$

The estimate $\hat{y}(x)$ is thus a weighted average of all the observed y_i values where each weight is exponentially proportional to its Euclidean distance from x.

As shown in **Figure 4**, the structure of the GRNN consists of 4 layers; the input layer, the hidden (pattern) layer, the summation layer and the output layer. As a preprocessing step, all input variables of the training data are scaled. Then, they are copied as the weights into the pattern units.

As a preprocessing step, all input variables of the training data are scaled. Then, they are copied as the weights into the pattern units. The summation layer has two units that can be denoted as the numerator and the denominator of Eq. (6). The output layer gives the estimate of the expected value of $\hat{y}(x)$. If y and $\hat{y}$ are the vector variables, the results above are generalized by adding with one summation unit for each component of y in the output layer.

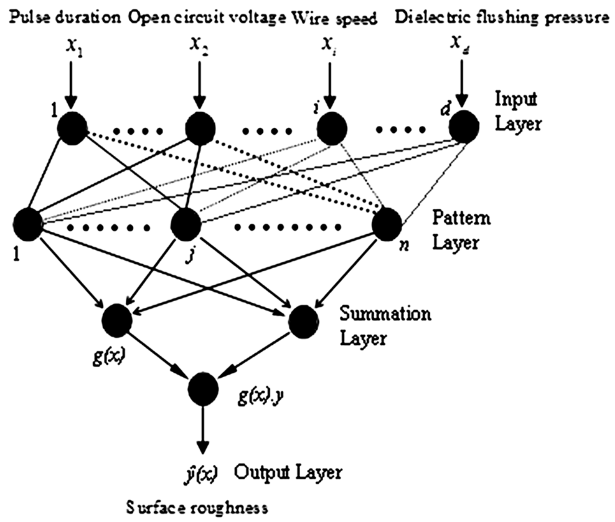

Figure 4: Constructed GRNN network
Slika 4: Zgrajena GRNN-mreža

The only adjustable parameter of the network is σ, the smoothing factor for the kernel function. It is critical to decide an optimum value for σ. The larger values of this factor cause the density to be smooth, and $\hat{y}(x)$ then converges to the sample mean of the observed y_i. On the other hand, when σ is chosen very small, the density is forced to have non-gaussian shapes. Then, the oscillatory points have a bad effect on the estimate. All values of y_i are taken into account where the points closer to x are given heavier weights, if the optimum value of σ is selected.[24–26] Therefore, in this study, σ was chosen as 0.57 due to the optimum value of success rate that was found after iterative calculation of σ values between 0.1 and 0.9.

The main advantage of GRNN according to other techniques is fast learning. It is a one-pass training algorithm, and does not require an iterative process. The training time is just the loading time of the training matrix. Also, it can handle both linear and non-linear data because the regression surface is instantly described everywhere, even just one sample is enough for this. Thus, other existing pattern nodes tolerate faulty samples. Another advantage is the fact that adding new samples to the training set does not require re-calibration of the model. As the sample size increases, the estimate surface converges to the optimal regression surface. Thus, it requires many training samples to span the variation in the data and all these to be stored for the future use. Solely, this causes a trouble of an increase in the amount of the computation to evaluate new points. In the course of time, highly improvements in the speed of the computer's processing power prevent this being a major problem. Furthermore, this also can be overcome by applying the various clustering techniques for grouping samples that each center is represented by this group of samples.[22–26] However, there is only one disadvantage that there is no intuitive method for choosing the optimal smoothing factor.

5 RESULTS AND DISCUSSIONS

In this study, twenty-eight set of data under different process condition was used for training and testing of the BPN and GRNN. Sixteen of them were used as a training purpose and the rest were used as testing purposes. **Table 1** shows the design matrix and training set used for BPN and GRNN analysis.

Testing the validation of BPN and GRNN results was made using the input parameters according to the design matrix given in **Table 2**.

These comparisons have been depicted in terms of percentage error in **Figure 5** for validation set of experiments. From **Table 2** it is evident that for our set of data the BPN result predicts the surface roughness nearer to the experimental values than the GRNN results. But,

Table 1: Neural network training set[7]
Tabela 1: Podatki za trening nevronske mreže[7]

WEDM input Parameters					Output
Exp. no	*t*/ns	*V*/V	*S*/(mm/min)	*p*/(kg/cm^2)	R_a/μm
1	200	300	12	16	2.12
2	200	60	4	16	1.13
3	900	60	4	6	2.14
4	200	60	12	16	1.24
5	200	300	12	6	2.32
6	200	300	4	16	1.98
7	900	60	12	16	2.15
8	900	300	12	6	3.85
9	200	300	4	6	2.10
10	900	300	4	16	3.24
11	900	60	12	6	2.26
12	900	300	12	16	3.65
13	900	60	4	16	2.01
14	200	60	4	6	1.18
15	900	300	4	6	3.55
16	200	60	12	6	1.24

Table 2: Test and comparison of BPN and GRNN results
Tabela 2: Preizkusi in primerjava BPN- in GRNN-rezultatov

Exp. No	WEDM input parameters					Modeling			
						Back Propagation Neural Network (BPN)		General Regression Neural Network (GRNN)	
	t/ns	*V*/V	*S*/(mm/min)	*p*/(kg/cm^2)	R_a/μm	*predicted*	*error*	*predicted*	*error*
1	300	80	4	6	1.30	1.26	3.08	1.38	-6.15
2	400	90	5	8	1.50	1.36	9.33	1.55	-3.33
3	500	150	6	10	2.08	2.02	2.88	1.96	5.77
4	700	250	10	14	3.18	3.48	-9.43	3.22	-1.26
5	350	60	12	16	1.29	1.24	3.88	1.43	-10.85
6	450	70	5	16	1.58	1.53	3.16	1.50	5.06
7	550	100	8	11	2.08	2.02	2.88	1.88	9.62
8	750	180	4	6	2.92	3.11	-6.51	2.72	6.85
9	850	200	10	8	3.27	3.47	-6.12	3.14	3.98
10	200	300	12	8	2.23	2.37	-6.28	2.31	-3.59
11	250	300	4	10	1.96	2.00	-2.04	2.14	-9.18
12	300	250	6	20	1.89	1.81	4.23	2.02	-6.88
						Average error: 4.99% CPU time = 2.3 min		Average error: 6.04% CPU time = 0.074 sec	

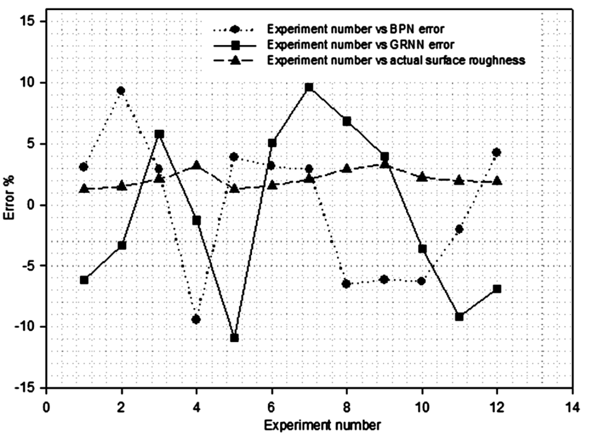

Figure 5: BPN and GRNN errors in prediction of the surface roughness
Slika 5: BPN- in GRNN-napake pri napovedi hrapavosti površine

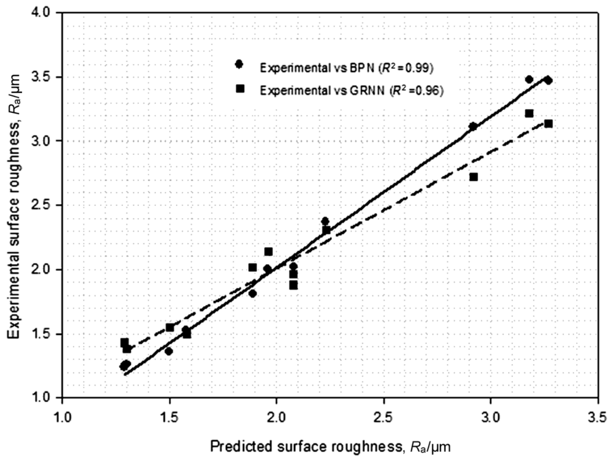

Figure 6: Comparison of predicted and experimental results
Slika 6: Primerjava napovedanih in eksperimentalnih rezultatov

GRNN is much faster than BPN with the CPU times of *0.074* s and *2.3* min respectively. In the prediction of surface roughness values the average errors for BPN and GRNN are calculated as *4.99 %* and *6.04 %* respectively.

The value of the multiple coefficient of R^2 is obtained as *0.99* for BPN and *0.96* for GRNN which means that the fitted line is very close to the experimental results. **Figure 6** represents the comparison of predicted (both BPN and GRNN) and actual results. Both BPN and GRNN results showed that the predicted values have been very close to experimental values.

6 CONCLUSIONS

The prediction of optimal machining conditions for the required surface finish and dimensional accuracy plays a very important role in the process planning of the wire erosion discharge machining process. This paper has described a neural network approach and comparison of Back Propagation Networks (BPN) and General Regression Neural Networks (GRNN) networks for the modeling of wire electrical discharge machining process using small set of data. Both the BPN and GRNN networks were used to construct the complicated relationships between the process parameters and the surface roughness. The experimental results has showed that the BPN network has better learning ability (with average error of *4.99 %* and multiple coefficient of R^2 of *99 %*) for the wire electrical discharge machining process than the GRNN (with average error of *6.04 %* and multiple coefficient of R^2 of *0.96*) network. Training of BPN network consumed more CPU time (elapsed time *2.3* min) than the GRNN (elapsed time *0.074* s). In addition to this, the back propagation network has better generalization ability for the wire electrical discharge machining process than the general regression neural network modeling.

7 REFERENCES

[1] J. Wang, , 19 (**2003**), 114

[2] U. Caydas, A. Hascalik, Journal of Materials Processing Technology, 202 **(2008)**, 574–582

[3] N. Tosun, C. Cogun, G. Tosun, Journal of Materials Processing Technology, 152 **(2004)**, 316–322

[4] N. Tosun, C. Cogun, H. Pihtili, Int. J. Adv. Manuf. Technol., 21 **(2003)**, 857–865

[5] N. Tosun, C. Cogun, A. Inan, Machining Science and Technology, 7 **(2003)**, 209–219

[6] C. Cogun, M. Savsar, International Journal of Machine Tools and Manufacture, 3 **(1990)**, 467–474

[7] U. Esme, A. Sagbas, F. Kahraman, Iranian Journal of Science & Technology, Transaction B, Engineering, 33 **(2009)**, 231–240

[8] D. Scott, S. Boyina, K. P. Rajurkar, Int. J. Prod. Res., 11 **(1991)**, 2189–2207

[9] W. M. Wang, K. P. Rajurkar, Monitoring sparking frequency and predicting wire breakage in WEDM, Sensors and Signal Processing for Manufacturing, ASME, Production Engineering Division (PED), New York, 55 **(1992)**, 49–64

[10] G. Spur, J. Schoenbeck, Anode erosion in wire-EDM-A theoretical model. CIRP Ann., 1 **(1993)**, 253–256

[11] Y. S. Tarng, S. C. Ma, L. K. Chung, International Journal of Machine Tools and Manufacture, 35 **(1995)**, 1693–1701

[12] T. A. Spedding, Z. Q. Wang, Precis. Eng., 20 **(1997)**, 5–15

[13] Y. S. Liao, J. T. Huang, H. C. Su, Journal of Materials Processing Technology, 71 **(1997)**, 487–493

[14] Y. K. Lok, T. C. Lee, Journal of Materials Processing Technology, 63 **(1997)**, 839–843

[15] R. Ramakrishnana, L. Karunamoorthyb, Journal of Materials Processing Technology, 207**(2008)**, 343–349

[16] T. C Tsai, J.T. Horng, N.M. Liu, C.C. Chou, K.T. Chiang, Materials and Design, 29 **(2008)**, 1762–1767

[17] S. Sarkar, S. Mitra, B. Bhattacharyya, Journal of Materials Processing Technology, 205**(2008)**, 376–387

[18] U. Caydas, A. Hascalik, S. Ekici, doi:10.1016/j.eswa.2008.07.019, **(2008)**

[19] S. C. Juang, Y. S. Tarng, H. R. Lii, Journal of Materials Processing Technology, 75 **(1998)**, 54–62

[20] J. A. Freeman, D. M. Skapura, Neural Networks: Algorithms, Application and Programming Techniques, Addison–Wesley, New York, **(1991)**

[21] J. McClelland, D. Rumelhart, Parallel Distributed Processing, MIT Press, Cambridge, MA, 1 **(1986)**

[22] D. Specht, A General Regression Neural Network. IEEE Trans. Neural Networks, 2 **(1991)**, 568–576

[23] T. Cacoullos, Estimation of a multivariate density. Annals of Inst. Stat. Math., 18 **(1996)**, 178–189

[24] D. Yeung, C. Parzen, Window Network Intrusion Detectors, Proceedings of the Sixteenth International Conference on Pattern Recognition, Quebec City, Canada, 4 **(2002)**, 385–388

[25] I. Ersoz Kaya, T. Ibrikci, A. Cakmak, Journal of Computational Intelligence in Bioinformatics, 1 **(2008)**, 1–11

[26] U. Esme, I. Ersoz, A. Ozbek, F. Kahraman, A. Sagbas, Cukurova University Journal of Engineering and Architecture, 23 **(2008)**, 57–66