ELSEVIER

Journal of Materials Processing Technology 117 (2001) 111–124

Journal of Materials Processing Technology

www.elsevier.com/locate/jmatprotec

Comparisons of neural network models on material removal rate in electrical discharge machining

Kuo-Ming Tsai, Pei-Jen Wang*

Department of Power Mechanical Engineering, National Tsing Hua University 101, Sec. II, Kuang Fu Rd., Hsinchu 30013, Taiwan, ROC

Received 3 April 2001

Abstract

Predictions for work removal based on physical models have been reported in electrical discharge machining (EDM) in recent years. However, when the change of polarity has been considered, few models have succeeded in giving consistent predictions. In this study, comparison of modeling the material removal rate of the work for various materials considering the change of polarity among six different neural networks together with a neuro-fuzzy network have been illustrated. The six neural networks are namely, the logistic sigmoid multi-layered perceptron (LOGMLP), the hyperbolic tangent sigmoid multi-layered perceptron (TANMLP), the fast error back-propagation hyperbolic tangent multi-layered perceptron (error TANMLP), the radial basis function networks (RBFNs), the adaptive TANMLP, and the adaptive RBFN. Also, the neuro-fuzzy network is the adaptive-network-based fuzzy inference system (ANFIS). Trained by the same experimental data selected with the method of design of experiment (DOE), the parameters of the above models have been optimized for further analysis. Based on the conclusions from the comparisons at checking the error among the network models, the best is the ANFIS with Bell-shape membership functions. Also, it can be concluded that the further experimental results have shown the accurate predictions based on the ANFIS model. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Neural network model; Material removal rate; Electrical discharge machining

1. Introduction

Electrical discharge machining is a non-traditional machining process for metal removing based on the nature that no tool force is generated during machining. The removal of metals in the EDM process is associated with the erosive effects occurring under a series of successive electrical sparks generating between the tool and the workpiece with electric potential submerged in a dielectric liquid environment. The process is widely used for manufacturing tools, dies, and other difficult-to-cut parts. Although, the EDM process has been accepted as the standard machining process in the tools, dies, and molds industry, the process is still treated as the so-called "know-how" process. Therefore, the tuning of EDM process variables to obtain energy efficiency and part accuracy has been empirical and difficult. Today, even though up-to-date computer technology has been applied, the EDM process is one of the expertise-demanding processes in the industry. From the literature, the exact mechanism of metal erosion during sparking is still debatable even based on the well-known physical laws. However, complex thermal conduction behavior may be widely accepted as the principal mechanism of metal erosion. This is why the models for correlating the process variables and material removal rate (MRR) are hard to be established accurately.

In the past decade, neural networks have been shown to be highly flexible modeling tools with capabilities of learning the mathematical mapping between input variables and output features for non-linear systems [1,2]. The superior performance of neural networks for modeling machining processes have been published elsewhere [3–17]. In the literature, multi-layer perceptrons based on the back-propagation (BP) technique have been employed for monitoring or modeling selected processes. For example, Rangwala and Dornfeld [3], Masory [4], Tansel et al. [5], and Tarng et al. [6,7] used BP or adaptive resonance theory (ART2) on the neural networks for monitoring tool wear and breakage in the turning or drilling process. On the other hand, Tansel et al. [8], Tarng et al. [9], and Lee et al. [10] also used BP or ART2-A on the neural networks for detecting and suppressing tool chatter in the turning or drilling process. Cariapa et al. [11], Tarng et al. [12,14], Liao and Lin [13], and Lee et al. [15] applied various neural networks for modeling and predicting the machining processes. Both Kao and Tarng [16] and Liu and Tarng [17] employed feed-forward neural networks with

* Corresponding author. Tel.: +886-3-571-9034; fax: +886-3-572-2840.
E-mail address: pjwang@pme.nthu.edu.tw (P.-J. Wang).

0924-0136/01/$ – see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0924-0136(01)01146-3

hyperbolic tangent functions and abductive networks for the on-line recognition of pulse types in the EDM process. Based on their results, discharge pulses were identified and then employed for controlling the EDM process. In the above literature, the effects of the change of polarity between the electrode and the work materials were all neglected.

The objective of this paper is to establish a better process model based on neural networks by comparing the predictions from different models under the effects of change of polarity between the electrode and the work materials in the EDM process. Initially, pertinent process variables affecting the MRR, namely the polarity of the electrode, the discharge time, the peak current, and the materials of both the tool and the workpiece, were screened by making use of the Taguchi method on design of experiments [18]. The DOE experimental data were later used for training the various process models. Finally, more experimental verification on the established process models was conducted, and comparisons among the models, including a statistical process model, were analyzed.

2. Neural networks

In past decades, numerous studies have been reported on the development of neural networks based on different architectures [1,2,19–21]. In general, neural networks are characterized by their architecture, activation functions, and learning algorithms or rules [2]. Each type of neural networks would have its own input–output characteristics, and therefore it could be applied only on some specific processes. Based on the recent developments in fuzzy theory, a fuzzy inference system can map a single given input to multi-outputs in a non-linear domain. Typically, the fuzzy inference system consists of membership functions, fuzzy logic operators, and prescribed if–then rules published elsewhere in the literature. In 1993, Jang et al. [22] introduced the adaptive-network-based fuzzy inference system (ANFIS) used in this paper. It is a very efficient system for solving ill-defined equations involving the automatic elicitation of knowledge expressed only by the if–then rules. Based on the authors experiences, ANFIS is only applicable for the cases of less than seven inputs and one output.

In this study, six neural networks and a neuro-fuzzy network are employed for modeling the MRR in the EDM process. All the networks are defined as follows.

1. Logistic sigmoid multi-layered perceptron (LOGMLP).
2. Hyperbolic tangent sigmoid multi-layered perceptron (TANMLP).
3. Radial basis function network (RBFN).
4. Fast error back-propagation multi-layered network with hyperbolic tangent functions (error TANMLP).
5. TANMLP with adaptive learning rate (adaptive TANMLP).
6. Radial basis function network with adaptive learning rate (adaptive RBFN).
7. Adaptive-network-based fuzzy inference system.

2.1. Architectures

Basically, neural networks are classified based on their architecture for simplicity. The architecture of the multi-layered perceptron networks and the RBFNs are shown in Figs. 1 and 2, respectively. In the figures, it is noted that the

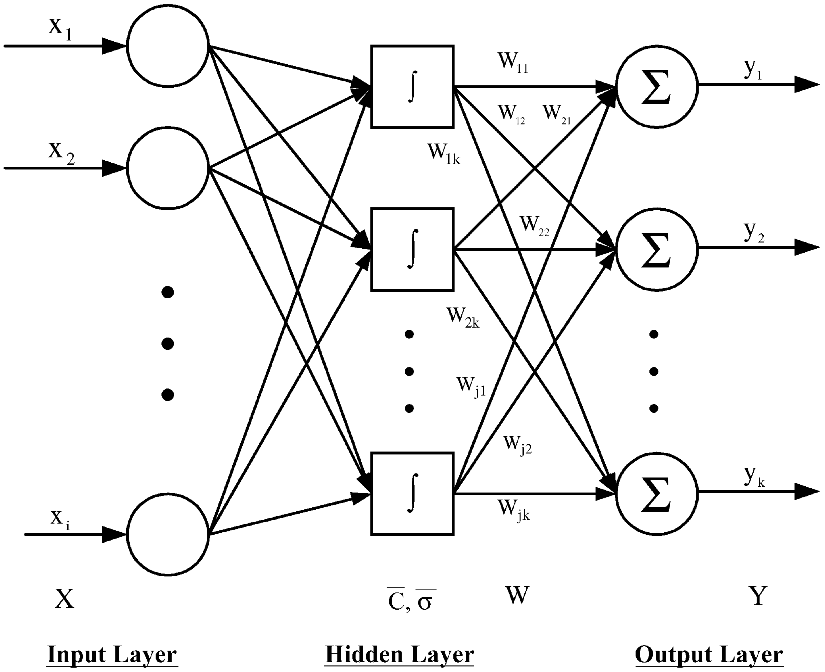


Fig. 1. Architecture of the multi-layered perceptron networks.

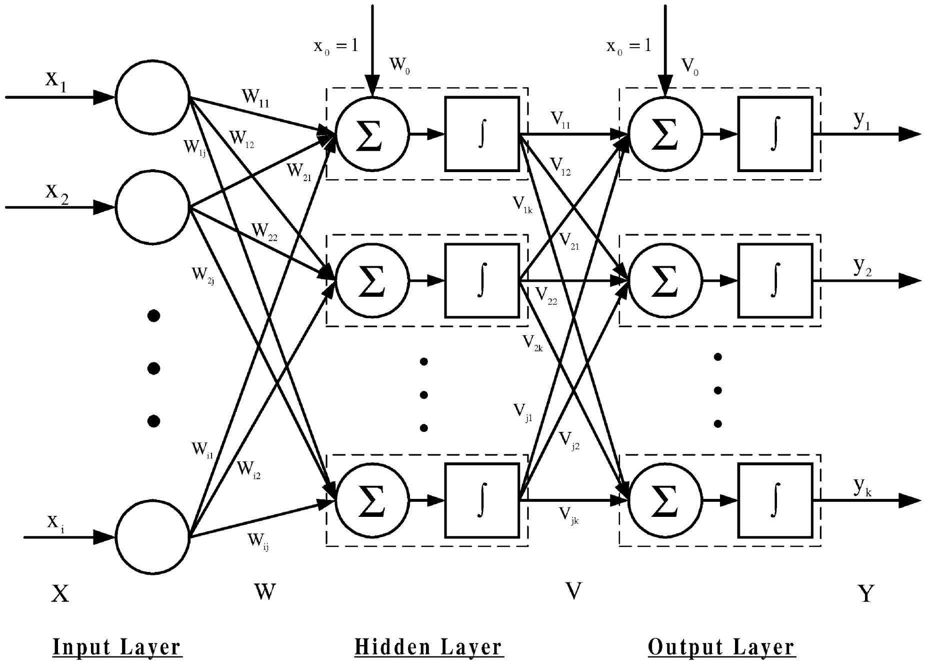


Fig. 2. Architecture of the RBFNs.

number of hidden layers is critical for the convergence rate during the training of parameters if the inputs and outputs have been given. In this study, only one hidden layer has been used in the multi-layered perceptron networks because the number of neurons has been assumed to be the most important parameter. As a result, the number of neurons has been determined by an optimization method. On the other hand, the architecture of the ANFIS has been based on a first-order Sugeno fuzzy inference model, as shown in Fig. 3. The advantages of the Sugeno structure have been reported as high computational efficiency, built-in optimal and adaptive techniques, and ensured continuity on the output surfaces [24]. A hybrid learning algorithm for identification of the near-optimal membership functions and other rule-based parameters has also been employed here to represent the desired input–output mappings.

2.2. Activation functions

In the neural networks described in the previous paragraphs, there are a large number of neurons in the hidden layers. In the networks, connections among the neuron are

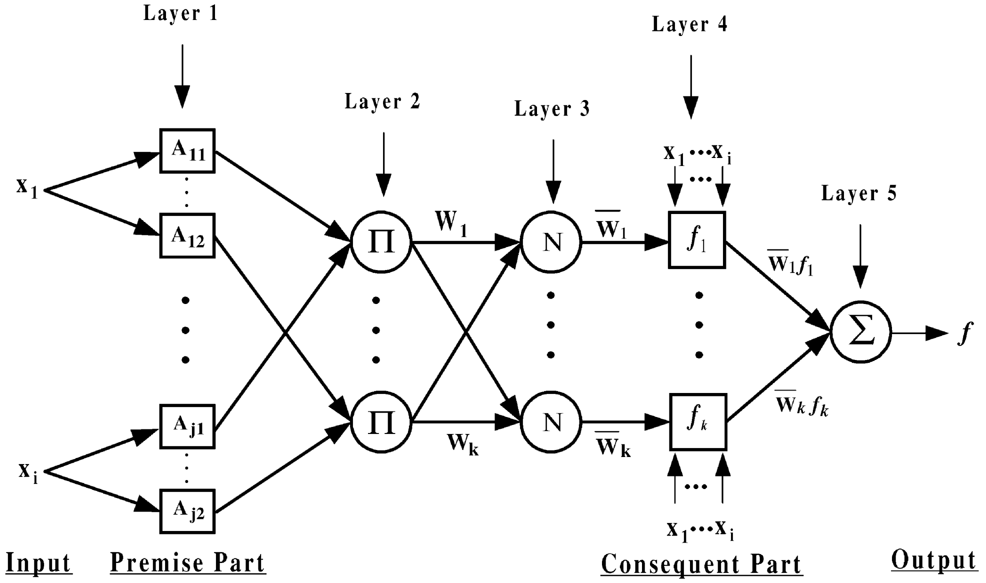


Fig. 3. Architecture of the ANFIS networks with multi-input Sugeno fuzzy model plus multi-rule (two membership functions at each input).

based on signal links with associate weightings. Each neuron is represented by its internal state, namely activation, which is functionally dependent on inputs. Sigmoid functions (S-shaped curves), such as logistic functions and hyper bolic tangent functions, are commonly adopted for the activation functions. Usually, a neuron transmits its activation signal to other neurons for information exchange. For comparison, several different activation functions have been employed in this study, being described in the following paragraph.

The activation function of LOGMLP is a continuous logistic function given as follows:

$$f(\text{net}) = \frac{1}{1+\exp(-\lambda\,\text{net})} \tag{1}$$

where $\lambda > 0$ is proportional to the gain which determines the steepest direction of the continuous function *f*(net) near net = 0. Similarly, the TANMLP and error TANMLP networks have continuous hyperbolic tangent functions defined as follows:

$$f(\text{net}) = \tanh\left(\frac{\lambda\,\text{net}}{2}\right) = \frac{1-\exp(-\lambda\,\text{net})}{1+\exp(-\lambda\,\text{net})} \tag{2}$$

In RBFN networks, the Gaussian distribution function is used as the activation function defined as follows:

$$f(\text{net}) = f(X; C_i; \sigma_i) = \mathrm{e}^{-(|X-C_i|^2/2\sigma_i^2)} \tag{3}$$

Forward pass:

Hidden layer:

$$net_j = w_0 + \sum_i w_{ij} x_i$$

$$H_j = f(net_j)$$

Output layer:

$$net_j = v_0 + \sum_j v_{jk} H_j$$

$$Y_k = f(net_j)$$

Error Function:

$$E = \sum_p E_p = \sum_p \frac{1}{2}(T_p - Y_p)^2$$

Back propagation:

$$\Delta V_j = -\eta \frac{\partial E}{\partial V_j} + \alpha V_{old}$$

$$V_{new} = V_{old} + \Delta V_j$$

where η is the learning rate, $\eta \in (0,1]$, and

α is the momentum coefficient, $\alpha \in [0,1]$.

$$\Delta W_i = -\eta \frac{\partial E}{\partial W_i} + \alpha W_{old}$$

$$W_{new} = W_{old} + \Delta W_i$$

Fig. 4. Network algorithm based on the Gradient descent method adopted in the LOGMLP, TANMLP, and adaptive TANMLP scheme.

Forward pass:

Hidden layer:

$$net_j = w_0 + \sum_i w_{ij} x_i$$

$$H_j = f(net_j)$$

Output layer:

$$net_k = v_0 + \sum_j v_{jk} H_j$$

$$Y_k = f(net_k)$$

Error Function:

$$E = \sum_p E_p = \sum_p \frac{1}{2}(T_p - Y_p)^2$$

$$\lambda = \exp(-\mu / E^2)$$

Back propagation:

$$E_o(\lambda) = \lambda E_p + (1-\lambda)\tanh(\beta E_p)$$

$$V_{new} = V_{old} + \eta E_o(\lambda) H_j$$

where λ is a weight factor that define a variety of alternative

objective functions between the two extremes

corresponding to $\lambda = 0$ and $\lambda = 1$, $\lambda \in [0,1]$,

and η is the learning rate, $\eta \in (0,1]$.

$$E_j(\lambda) = (1 - H_j^2) \sum E_o(\lambda) W_{ij}$$

$$W_{new} = W_{old} + \eta E_j(\lambda) X$$

Fig. 5. Network algorithm based on the fast error back-propagation adopted in the error TANMLP scheme.

where C_i is the center of the Gaussian distribution and σ_i the standard deviation of the Gaussian distribution. In the ANFIS, one membership function has been the Gaussian, but the other has been the Bell-shape function defined as follows:

$$f(\text{net}) = f(X; a_i, b_i, c_i) = \frac{1}{1 + |(X - c_i)/a_i|^{2b_i}} \tag{4}$$

where the $\{a_i, b_i, c_i\}$ is a parameter set. Because these parameters determine the x coordinates of the two corners underlying the Bell-shape function, the parameter b_i is usually positive. (If b_i is negative, the shape of this function looks like an upside-down bell.)

2.3. Algorithms

During the training of the neural networks, it is critical to choose the appropriate algorithms because the efficiency and the convergence are primarily the key issue at this stage. This means the algorithms are the selection rules for the weights in order to accomplish the desired mapping between the inputs and the outputs. Based on the least-squares approach, the quadratic error function between the actual outputs and the network outputs is expressed by

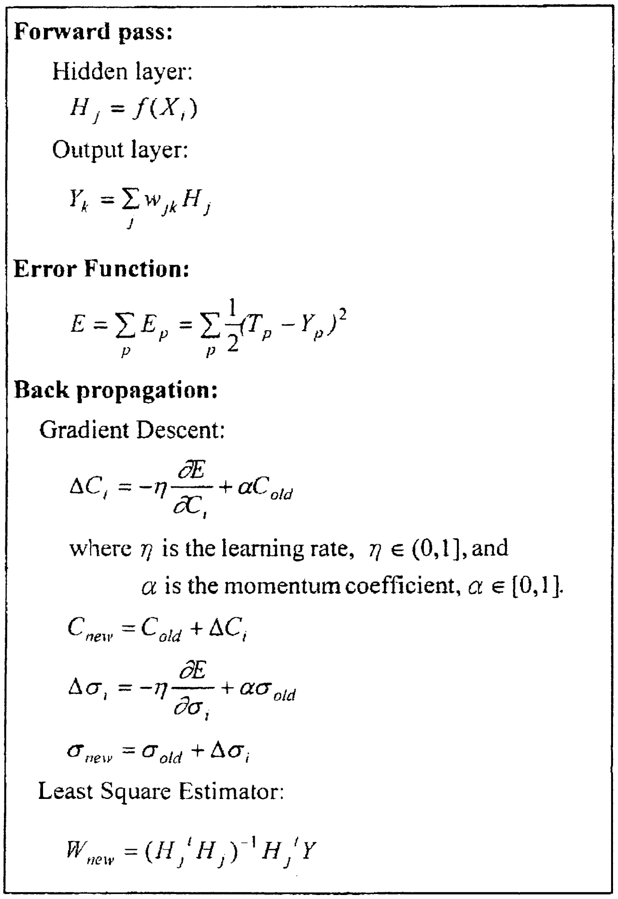
$$E = \sum_p E_p = \sum_p \frac{1}{2}(T_p - Y_p)^2 \tag{5}$$

where T_p is the target values and Y_p the outputs of the neural networks.

In this study, only supervised network algorithms, such as the delta learning rule (i.e. gradient descent) with momentum, the fast error back-propagation learning rule with momentum, and a hybrid of the delta learning and the least-squares estimator were employed the three network algorithms are shown in Figs. 4–6, respectively. However, the ANFIS networks have used a special hybrid learning algorithm to update their parameters. Considering the convergence conditions, both the least-squares method and the back-propagation gradient descent method have been employed for linear and non-linear parameters, respectively. In all the algorithms, an error measure for final check, which is a normalized root-mean-square of error (RMSE), is defined as follows:

$$\text{RMSE} = \frac{1}{\text{length}(\mathbf{T}_p - \mathbf{Y}_p)} \sqrt{\sum_p \frac{1}{2}(\mathbf{T}_p - \mathbf{Y}_p)^2} \tag{6}$$

where $\mathbf{T}_p$ is the target vector (i.e. experimental values) and $\mathbf{Y}_p$ is the predicted vector (training values). As for the adaptive algorithms, the values of the learning coefficients have to be adequately increased when the RMSE of the current epoch is smaller than the RMSE of the previous epoch. Otherwise, the values have to be adequately decreased when the RMSE of the current epoch is larger than the RMSE of the previous epoch.



Forward pass:

Hidden layer:

$H_j = f(X_i)$

Output layer:

$Y_k = \sum_j w_{jk} H_j$

Error Function:

$E = \sum_p E_p = \sum_p \frac{1}{2}(T_p - Y_p)^2$

Back propagation:

Gradient Descent:

$\Delta C_i = -\eta \frac{\partial E}{\partial C_i} + \alpha C_{old}$

where η is the learning rate, $\eta \in (0,1]$, and α is the momentum coefficient, $\alpha \in [0,1]$.

$C_{new} = C_{old} + \Delta C_i$

$\Delta \sigma_i = -\eta \frac{\partial E}{\partial \sigma_i} + \alpha \sigma_{old}$

$\sigma_{new} = \sigma_{old} + \Delta \sigma_i$

Least Square Estimator:

$W_{new} = (H_j{}' H_j)^{-1} H_j{}' Y$

Fig. 6. Network algorithm based on the combination of the Gradient descent method and least-squares estimator adopted in the RBFN and adaptive RBFN networks.

3. Experimental verification

A schematic drawing of the experimental apparatus and a photograph of the EDM machine attached with a personal computer are shown in Fig. 7. All the experiments were conducted on a Model Mold Maker III CNC EDM machine, made by Sodick in Japan. The EDM machine was attached with a MARK XI pulse charge generator and a computer-controller to produce rectangular-shaped current pulses during discharging. Throughout the experiments, the dielectric fluid was SPE oil produced by Castrol. In particular, for better control of the dielectric environment, the fluid was kept in a stainless steel container during the experiments. The MRR data were later measured by making use of an A200S-D electronic micro-weight balance, made by Sartorius in Germany.

In this study, three different pure metals were employed for the experimentation. While copper was used as the tool (the upper electrode), aluminum and iron were used as the workpiece (the lower electrode). In all the experiments, the pertinent process parameters and their levels for each set of the experiments are listed in Table 1. Also, the physical characteristics together with the mechanical dimensions of

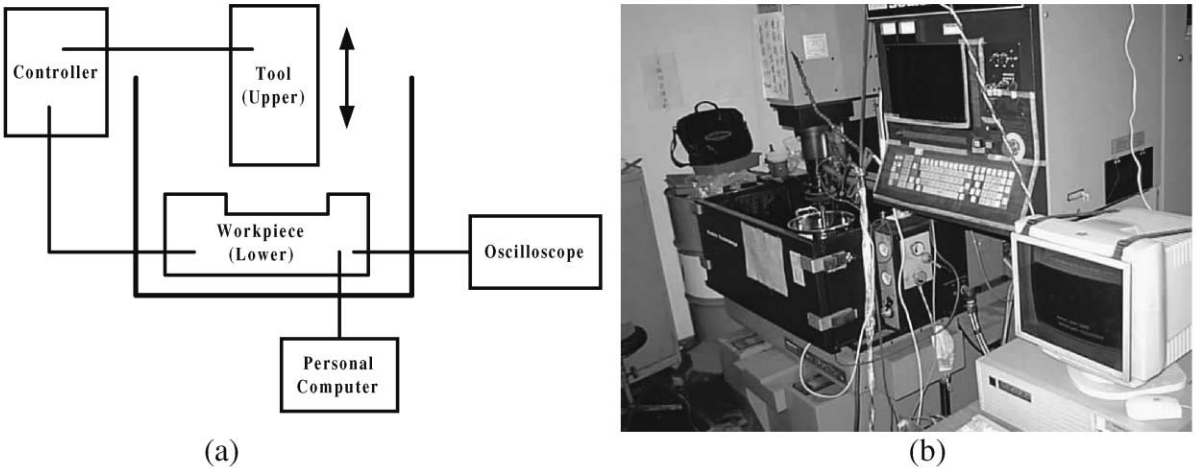


Fig. 7. The experimental apparatus: (a) the schematic drawing; (b) photograph of the machine and the personal computer.

Table 1
Pertinent process parameters and values for the experiments

Symbol	Factor	Level				
		1	2	3	4	5
PL	Polarity of upper electrode	−	+			
ON	Discharge time (μs)	20	30	60	100	
I_p	Main power peak current (A)	12	22.5	30	39	48
An	Tool material (upper electrode)	Cu				
Ca	Work material (lower electrode)	Al	Fe			
OFF	Quiescent time (μs)	60				
V	Main power voltage (V)	90				
SV	Servo standard voltage	2				

Table 2
Physical characteristics and mechanical dimensions of the tool and the work

Materials	Composition (%)	Density (kg/m^3)	Machined roughness R_{max} (μm)	Dimensions (mm^2)
Tool				
Cu	>99.95	8896.6	4.02	ϕ9.5 × 50
Workpiece				
Fe	>99.9	7870	1.08	ϕ20 × 12
Al	>99.5	2699	2.76	ϕ20 × 12

the tool and the workpieces are tabulated in Table 2. In order to produce adequate data for model training, 80 sets of experimental conditions were scheduled on the machine. Each set of the conditions was measured by running 40 consecutive tests both in the case of copper–aluminum and copper–iron combinations.

Known for its capabilities in establishing the neural networks models, MATLAB with associate toolboxes, copyrighted by MathWork in the USA, was used for coding the algorithms. Also, with the help of Pentium III processors on a personal computer, the programs could be executed and finished in a few minutes.

4. Results and discussion

Before applying the neural networks for modeling the EDM process, first there was the need to decide the architecture and the topology of the networks; for example, the number of hidden layers and the number of neurons in each layer in the networks. Based on the previous experience from the work on the semi-empirical model [24], five inputs and one output in the networks would suffice for the interests in this study. Therefore, the number of neurons in the input and output layer should be set to five and one, respectively. Also, the back-propagation architecture with one hidden layer is enough for the majority of the applications, because it can form arbitrary mapping between a set of given inputs and outputs [2]. In this study, one hidden layer for the neural networks was used. For determining the optimal number of neurons in the hidden layer, a procedure was used to optimize the number of neurons in the hidden layer for various neural networks based on the results from 5000 epochs as shown in Fig. 8. By comparing the results, the number of hidden neurons were found to be 25 for the LOGMLP, 12 for the TANMLP, 12 for the RBFN, 30 for the error TANMLP, 50 for the adaptive TANMLP, and 12 for the adaptive RBFN. It is noted that the RBFN and the adaptive RBFN networks have less neuron than the others. Also, the final results for the two networks versus the various numbers of hidden neurons are shown and compared in Tables 3 and 4. In these two tables, the prediction, the

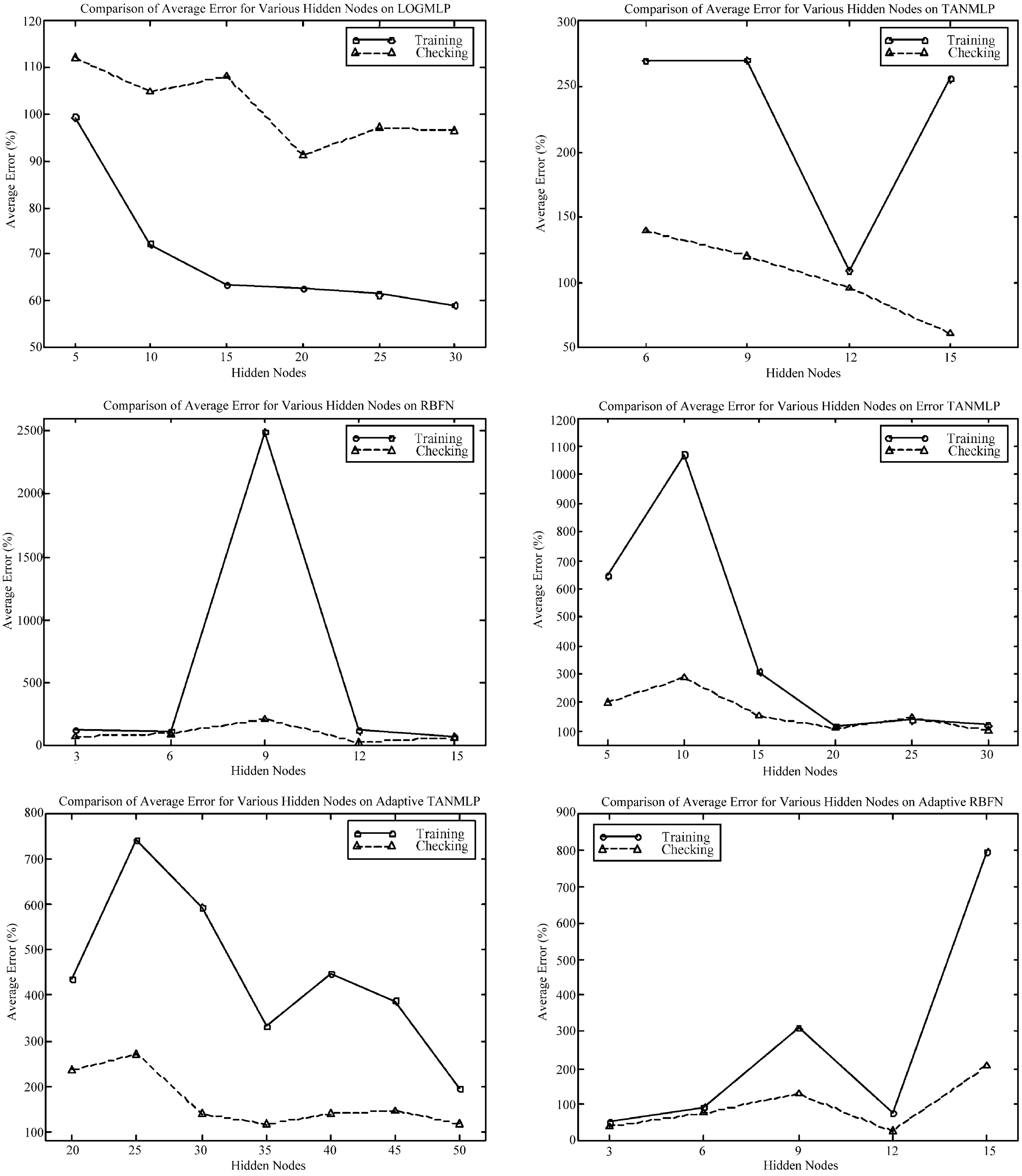


Fig. 8. Comparison of the average error among various network models with various numbers of hidden nodes.

Table 3
Final results of the MRR model based on the RBFN; number of hidden nodes is the main variable

	Hidden nodes				
	3	6	9	12	15
Epoch in optimal	5000	5000	443	5000	2182
RMSE	0.03164	0.02193	0.05316	0.01751	0.01967
Running time (s)	356	375	465	774	454
Average training error (%)	125.37	112.15	2492.92	117.96	70.53
R^2	0.9595	0.9805	−0.1912	0.9876	0.9843
Average checking error (%)	74.61	93.55	356.18	24.71	67.22

Table 4
The same as Table 3, except for the adaptive RBFN

	Hidden nodes				
	3	6	9	12	15
Epoch in optimal	5000	5000	4998	5000	4998
RMSE	0.02598	0.02514	0.03526	0.01967	0.01820
Running time (s)	362	372	404	816	435
Average training error (%)	52.31	93.72	310.59	75.84	796.94
R^2	0.9727	0.9744	0.9481	0.9843	0.6709
Average checking error (%)	38.82	77.65	129.72	25.93	208.2

training and the checking errors, are defined as follows:

$$\text{error in \%} = \left| \frac{\text{experimental results} - \text{predictions}}{\text{experimental results}} \right| \times 100 \tag{7}$$

It should be noted that the checking errors of the best cases are 24.71 and 25.93% with 12 hidden neurons for both the RBFN and the adaptive RBFN, respectively.

Based on the user's guide, there are eight different membership functions supported in the MATLAB fuzzy

Table 5
Number of intrinsic parameters for the ANFIS with various membership functions

Membership function	Two Bell MFs	Two Gaussian MFs	Three Bell MFs	Three Gaussian MFs
Number of inputs	5	5	5	5
Number of nodes	92	92	524	524
Number of linear parameters	192	192	1458	1458
Number of non-linear parameters	30	20	45	30
Total number of parameters	222	212	1503	1488
Number of training data pairs	80	80	80	80
Number of checking data pairs	10	10	10	10
Number of fuzzy rules	32	32	243	243

Table 6
Final results of ANFIS model with two Bell membership functions; the bold-face column indicates the best results

	Epochs							
	1	20	**40**	60	120	250	350	450
Final RMSE	0.02434	0.01662	**0.01190**	0.00801	0.00595	0.00527	0.00509	0.00497
Running time (s)	1.26	25	**49**	74	146	296	411	525
Average training error (%)	22.16	21.86	**24.13**	12.60	11.01	10.83	10.81	10.80
R^2	0.9760	0.9888	**0.9943**	0.9974	0.9986	0.9989	0.9990	0.9990
Average checking error (%)	72.81	36.39	**21.24**	48.23	48.26	43.02	41.76	41.11

logic toolbox [23]. In this study, two and three of both the Bell and the Gaussian membership functions were employed for ease of manipulation. The characteristics of these types of MF are shown in Table 5. Evidently, the computer run-time will increase with the increase in the number of membership function because the nodes, the parameters, and the if–then rules grow exponentially. From Tables 6–9, the final results of the ANFIS model versus various epochs

Table 7
The same as in Table 6, except with two Gaussian membership functions

	Epochs							
	1	11	**20**	40	120	250	350	450
Final RMSE	0.01999	0.01621	**0.01381**	0.01007	0.00850	0.00833	0.00820	0.00818
Running time (s)	1.26	13	**23**	45	137	287	425	546
Average training error (%)	18.89	17.31	**18.26**	45.99	71.86	71.17	69.73	68.83
R^2	0.9838	0.9894	**0.9923**	0.9960	0.9971	0.9972	0.9973	0.9973
Average checking error (%)	44.74	32.29	**26.33**	45.77	82.44	82.84	80.91	78.68

Table 8
The same as in Table 6, except with three Bell membership functions

	Epochs							
	1	20	40	80	120	**140**	160	250
Final RMSE	0.01193	0.00476	0.00300	0.00253	0.00185	**0.00155**	0.00130	0.00109
Running time (s)	119	2251	4485	9015	13524	**15692**	18041	28068
Average training error (%)	11.77	8.43	6.76	6.53	7.39	**7.88**	6.69	6.01
R^2	0.9942	0.9991	0.9996	0.9997	0.9999	**0.9999**	0.9999	0.9999
Average checking error (%)	89.91	50.04	46.05	29.97	28.98	**25.58**	30.97	30.92

Table 9
The same as in Table 6, except with three Gaussian membership functions

	Epochs							
	1	5	20	40	80	120	180	250
Final RMSE	**0.0084**	0.00668	0.00389	0.00311	0.00293	0.00285	0.00281	0.00279
Running time (s)	**185**	576	2281	4569	9141	13281	20274	27993
Average training error (%)	**9.92**	9.27	7.55	6.84	6.78	6.79	6.81	6.83
R^2	**0.9971**	0.9982	0.9994	0.9996	0.9997	0.9997	0.9997	0.9997
Average checking error (%)	**34.27**	35.61	46.26	47.52	42.50	40.92	39.98	39.87

Table 10
Final results of the MRR model based on various network models; Cu–Al and Fe are employed for the tool and work materials, respectively

	LOGMLP	TANMLP	RBFN	ERROR TANMLP	Adaptive TANMLP
Hidden nodes	25	12	12	30	50
Epochs	5000	5000	5000	5000	5000
Final RMSE	0.05229	0.04180	0.01751	0.05601	0.06651
Running time (s)	385	415	774	517	609
Average training error (%)	62.73	109.19	117.96	120.30	196.12
R^2	0.8893	0.9292	0.9876	0.8795	0.8209
Average checking error (%)	91.24	96.05	24.71	102.56	116.26
	Adaptive RBFN	ANFIS (two Bell MFs)	ANFIS (two Gaussian MFs)	ANFIS (three Bell MFs)	ANFIS (three Gaussian MFs)
Hidden nodes	12				
Epochs	5000	40	20	140	1
Final RMSE	0.01967	0.01190	0.01381	0.00155	0.0084
Running time (s)	816	50	23	15692	185
Average training error (%)	75.84	24.12	18.26	7.88	9.92
R^2	0.9843	0.9942	0.9923	0.9999	0.9971
Average checking error (%)	25.93	21.24	26.33	25.58	34.27

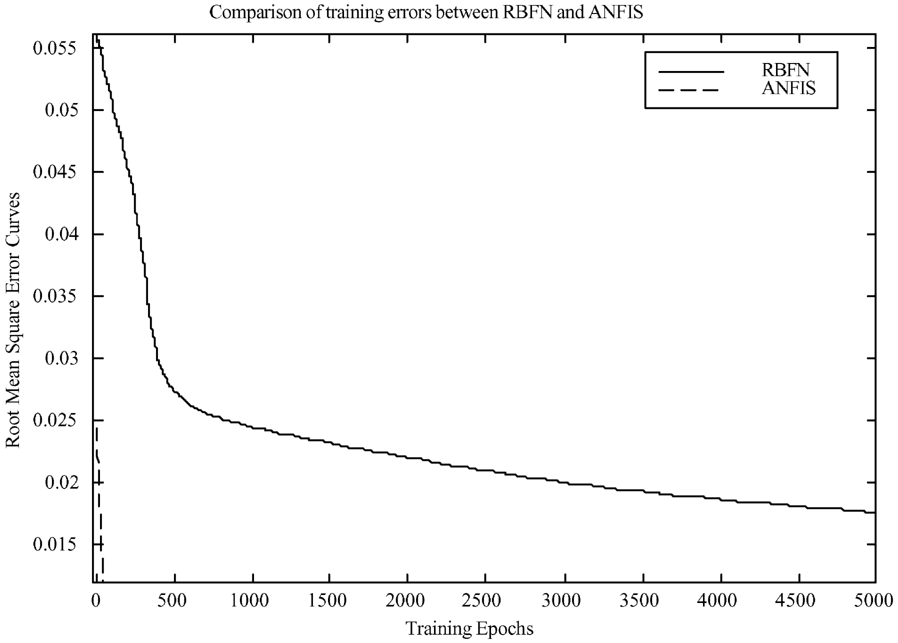


Fig. 9. Comparison of training errors between the RBFN and the ANFIS model: the ANFIS model is seen to be much better than the RBFN model.

corresponding to each of the cases are tabulated, with boldface columns indicating the best results. They are 40 epochs for two Bell MFs, 20 for the two Gaussian MFs, 140 for the three Bell MFs, and one for the three Gaussian MFs. These cases were later used for predictions on the MRR together with the two neural networks models (i.e. RBFN and adaptive RBFN) and the semi-empirical model [24] shown as follows:

$$\dot{V} = A_1 \left[\frac{\alpha^2}{H_\mathrm{v}^{1/2}}\right] \left(\frac{I_\mathrm{p}}{\sigma^{1/2}\rho^{1/2}\alpha^{3/2}}\right)^{a_1} \left(\frac{\tau_\mathrm{on} H_\mathrm{v}}{\alpha}\right)^{b_1} \times \left(\frac{E}{\rho\alpha^2 H_\mathrm{v}^{1/2}}\right)^{c_1} (J_\mathrm{a})^{d_1} \tag{8}$$

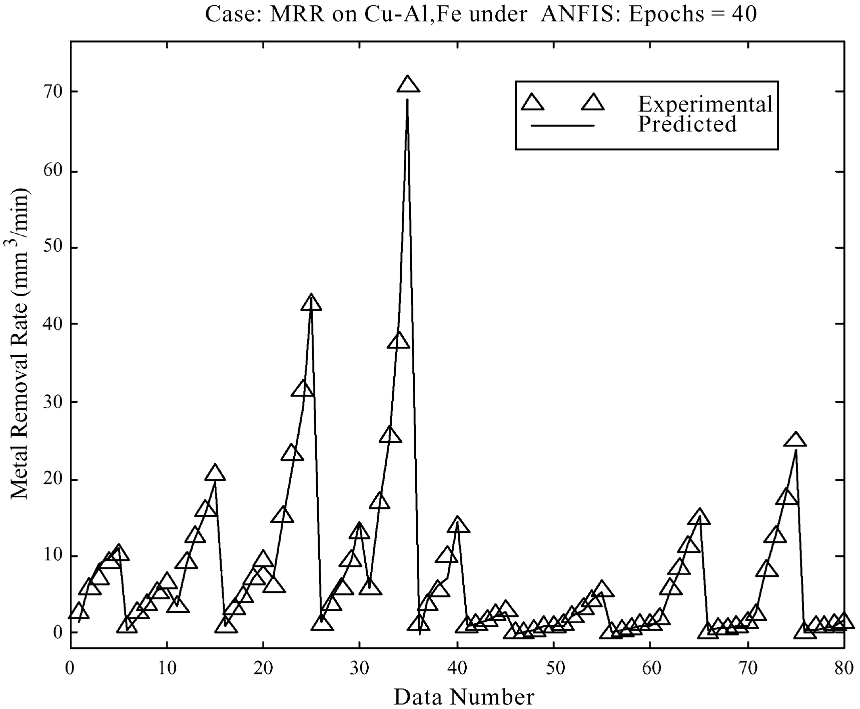


Fig. 10. Comparison between the measured and predicted MRR results based on the ANFIS model by making use of two Bell membership functions.

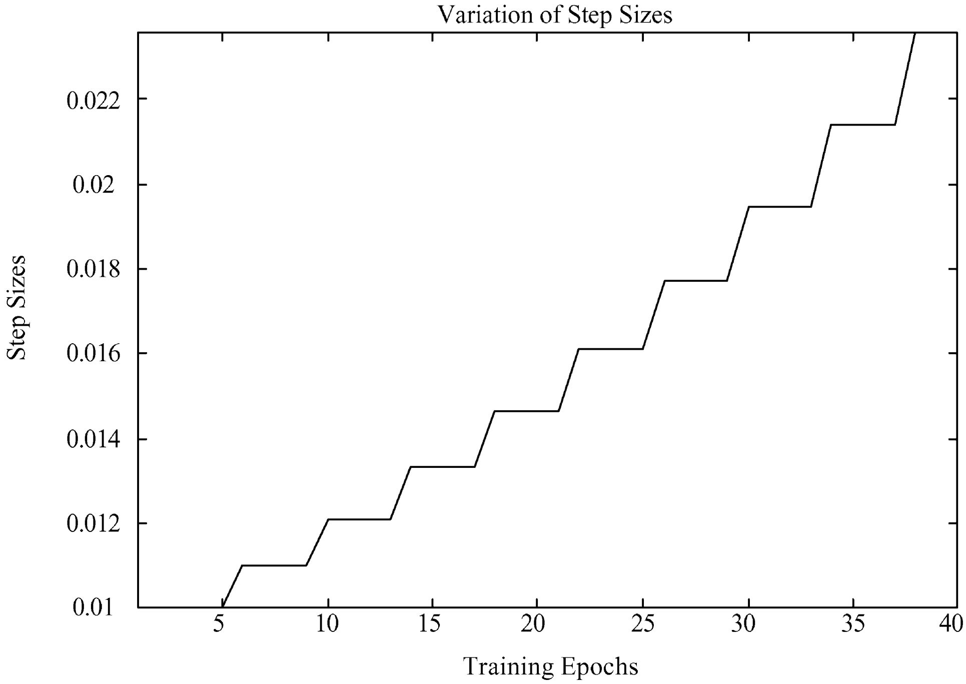


Fig. 11. Plots of step size for convergence versus training epochs for the ANFIS model.

where $\dot{V}$ is the MRR; A_1 a constant which is a function of the material; I_p the peak current; τ_{on} the discharge time; E the input energy to work; $\alpha = \kappa/\rho C_p$ the thermal diffusivity; κ the thermal conductivity; C_p the specific heat capacity; ρ the density; σ the electric conductivity; H_v the latent heat of evaporation; and $J_a = T_v C_p/H_v$ the Jacob number. This semi-empirical model has been established by employing dimensional analysis based upon pertinent process parameters screened by the design of experiments method.

Together with all the models, Table 10 shows the final results for MRR for comparison. It is noted that the best of

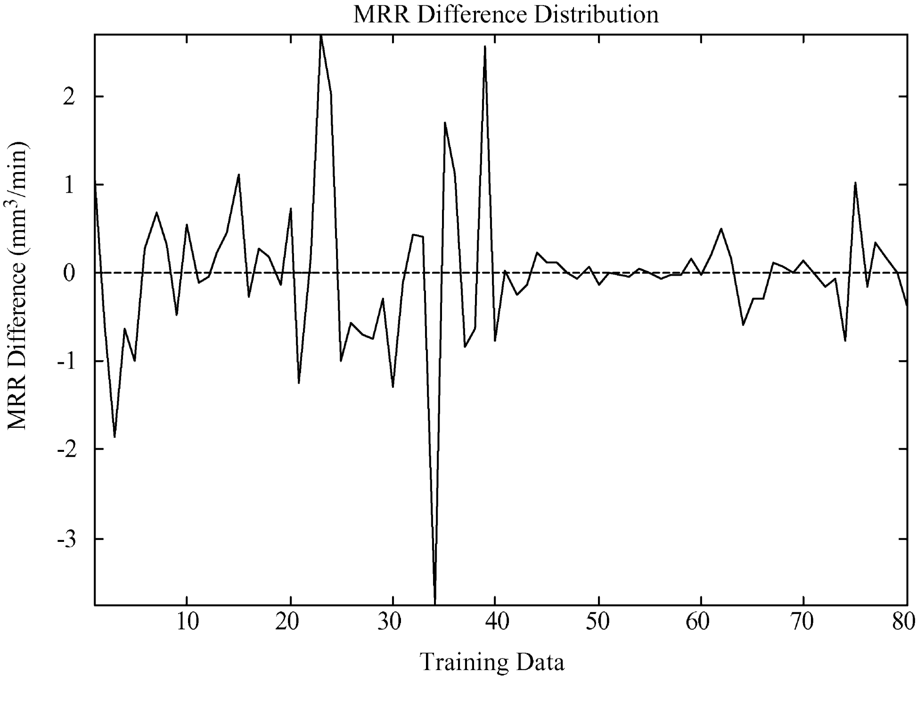


Fig. 12. Plots of MRR differences calculated based on Fig. 10 versus the number of training data after the training procedure.

these model, the ANFIS with two Bell membership functions. Compared to the RBFN model, the ANFIS model has faster convergence and better performance, as shown in Fig. 9. For detailed illustration, the final parametric results of the ANFIS model are shown from Figs. 10–13. In these figures, the minimum average checking error is 21.24%. In Fig. 12, larger training differences are observed from data set 1–40 because the copper–aluminum combination has a higher MRR than the combinations.

As a further step for verification, experiments were scheduled with process parameters set to the border on the training process window. The comparison among all the models is shown in Table 11 for illustration. In particular, the semi-empirical, the RBFN, the adaptive RBFN, and the ANFIS with two Bell MFs were plotted on the same scale, as shown in Fig. 14. It is noted that the ANFIS model is the still the best, with 16.33% checking error. That means that the predictions of MRR in the EDM process by making

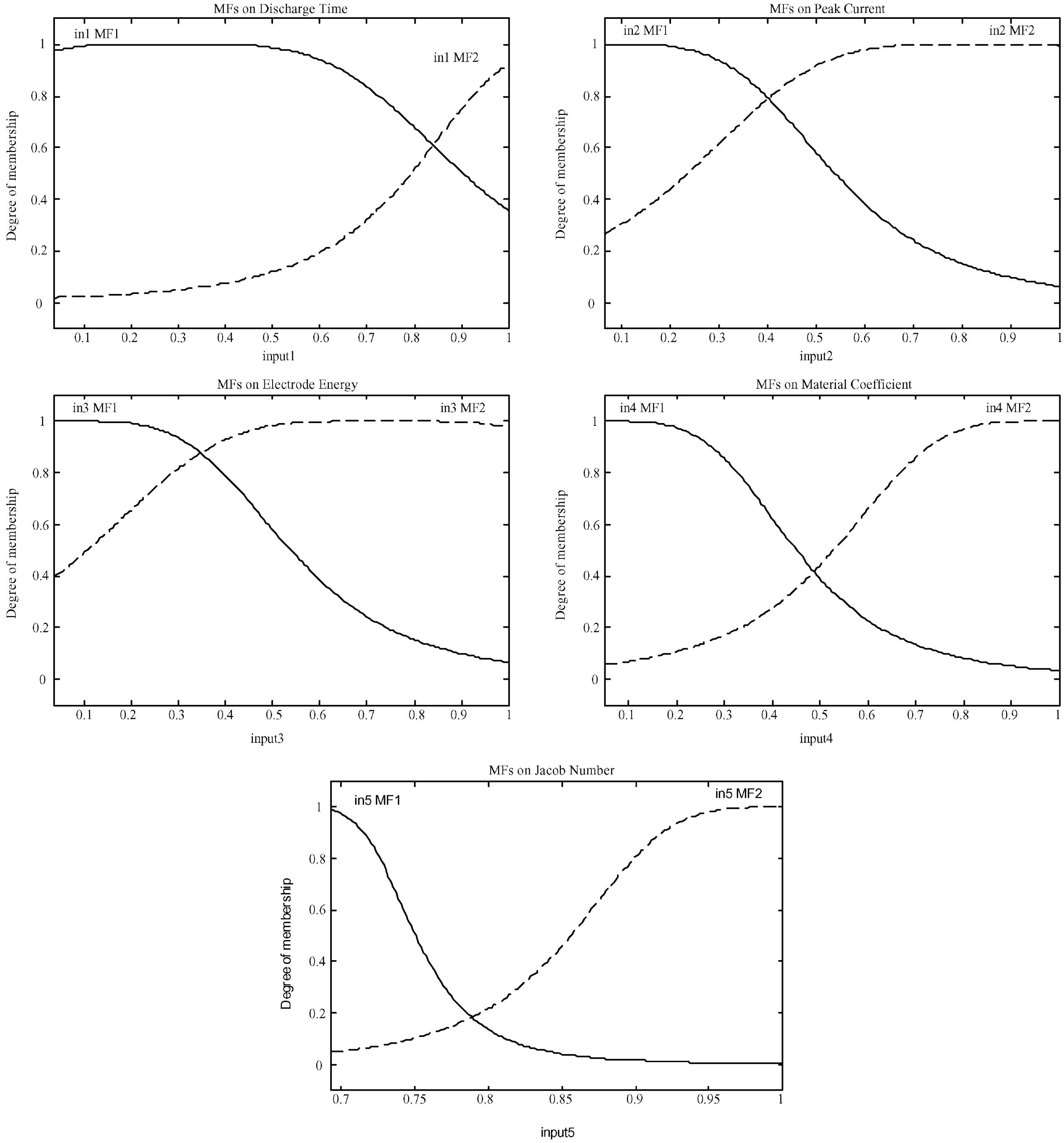


Fig. 13. Final values of the membership functions for the ANFIS inputs after the training procedure.

Table 11
Experimental verification based on the different process parameters indicated in the note used for the training procedure with three different network models

Case	Materials	Experiment	Semi-empirical model	Error (%)	Adaptive RBFN	Error (%)	ANFIS (two Bell MFs)	Error (%)	ANFIS (three Bell MFs)	Error (%)
1	Cu(+):Al(−)[a]	17.0111	21.3676	25.61	16.2475	4.49	13.8762	18.43	20.0963	18.14
2	Cu(−):Al(+)[a]	3.203	4.7611	48.65	4.0318	25.87	2.9381	8.27	3.6503	13.97
3	Cu(+):Al(−)[b]	36.797	34.8231	5.36	28.8421	21.62	23.8915	35.07	29.1847	20.69
4	Cu(−):Al(+)[b]	3.7048	7.7593	109.44	6.1454	65.88	4.1042	10.78	6.2911	69.81
5	Cu(−):Fe(+)[c]	0.7497	1.3008	73.51	0.9610	28.18	0.5216	30.43	0.5732	23.55
6	Cu(+):Fe(−)[a]	8.399	7.0195	16.42	7.7245	8.03	9.3593	11.43	10.3924	23.73
7	Cu(−):Fe(+)[a]	0.5108	2.7309	434.62	0.4142	18.92	0.4837	5.30	0.6689	30.95
8	Cu(+):Fe(−)[b]	17.9365	11.4399	36.22	12.2517	31.69	14.0474	21.68	15.7809	12.02
9	Cu(−):Fe(+)[b]	0.864	4.4505	415.11	0.7413	14.20	0.8160	5.55	0.9167	6.10
Average error (%)				129.44		24.79		16.33		24.33

[a] τ_{ON}: 120 s; I_{p}: 22.5 A; τ_{OFF}: 60 s; $V = 120$ V; SV = 2.

[b] τ_{ON}: 120 s; I_{p}: 30.0 A; τ_{OFF}: 60 s; $V = 120$ V; SV = 2.

[c] τ_{ON} : 60 s; I_{p}: 30.0 A; τ_{OFF}: 60 s; $V = 120$ V; SV = 2.

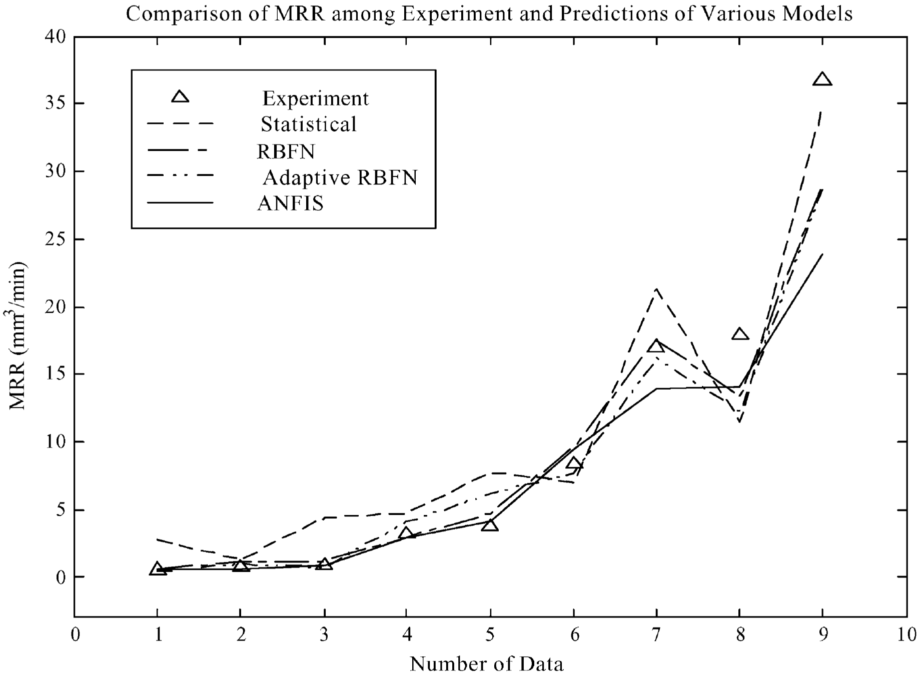


Fig. 14. Comparison of MRR among the measured values and predictions based on various network models: the ANFIS shows better predictions than the others.

use of the ANFIS model are in good agreement with the experimental results.

5. Conclusions

In this paper, six neural models and a neuro-fuzzy model for MRR in EDM process have been established and analyzed based on the pertinent machine process parameters. The networks, namely the LOGMLP, the TANMLP, the RBFN, the error TANMLP, the adaptive TANMLP, the adaptive RBFN, and the ANFIS have been trained and compared under the same experimental conditions for two different materials. According to the training results, the comparison has shown that the ANFIS model is more accurate than the other models. Also, further experimental verification have shown that the predictions on the MRR for extrapolative conditions have reached 16.33% error. As a result, the MRR in the EDM process including change of polarity can be predicted with reasonable accuracy.

Based on the previous experience with the semi-empirical model, it is possible to conclude that the EDM process can be correctly modeled based on the ANFIS approach even though the EDM process is well known for its complex and stochastic nature. The present study has explored a new process model, which includes all the pertinent process parameters, for reasonably predicting the MRR. In other words, the exact machining time can be better controlled on the shop floor in practical applications. Conclusively speaking, an effective modeling tool for the MRR in the EDM process has been established with promising potential applications in industry.

Acknowledgements

The authors are grateful for the financial support from the National Science Council, Grant No. NSC-34234D.

References

[1] J.A. Freeman, D.M. Skapura, Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Reading, MA, 1992.

[2] L. Fausett, Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[3] S. Rangwala, D. Dornfeld, Sensor integration using neural networks for intelligent tool condition monitoring, J. Eng. Ind. ASME 112 (1990) 219–228.

[4] O. Masory, Monitoring machining processes using multi-sensor reading fused by artificial neural network, J. Mater. Process. Technol. 28 (1991) 231–240.

[5] I.N. Tansel, C. Mekdeci, O. Rodriguez, B. Uragun, Monitoring drill conditions with wavelet based encoding and neural networks, Int. J. Mach. Tools Manuf. 33 (1993) 559–575.

[6] Y.S. Tarng, Y.W. Hseih, S.T. Hwang, An intelligent sensor for monitoring milling cutter breakage, Int. J. Adv. Manuf. Technol. 9 (1994) 141–146.

[7] Y.S. Tarng, Y.W. Hseih, S.T. Hwang, Sensing tool breakage in face milling with a neural network, Int. J. Mach. Tools Manuf. 34 (1994) 341–350.

[8] I.N. Tansel, A. Wagiman, A. Tziranis, Recognition of chatter with neural network, Int. J. Mach. Tools Manuf. 31 (1991) 539–552.

[9] Y.S. Tarng, T.C. Li, M.C. Chen, On-line drilling chatter recognition and avoidance using an ART2 — a neural network, Int. J. Mach. Tools Manuf. 34 (1994) 949–957.

[10] B.Y. Lee, Y.S. Tarng, S.C. Ma, Modeling of the process damping force in chatter vibration, Int. J. Mach. Tools Manuf. 35 (1995) 951–962.

[11] V. Cariapa, K.S. Akbay, R. Rudraraju, Application of neural networks for compliant tool polishing operations, J. Mater. Process. Technol. 28 (1991) 241–250.

[12] Y.S. Tarng, S.T. Hwang, Y.S. Wang, A neural network controller for constant turning force, Int. J. Mach. Tools Manuf. 34 (1994) 453–460.

[13] T.W. Liao, L.J. Chin, A neural network approach for grinding process: modeling and optimization, Int. J. Mach. Tools Manuf. 34 (1994) 919–937.

[14] Y.S. Tarng, S.C. Ma, L.K. Chung, Determination of optimal cutting parameters in wire electrical discharge machining, Int. J. Mach. Tools Manuf. 35 (1995) 1693–1701.

[15] B.Y. Lee, H.S. Liu, Y.S. Tarng, Modeling and optimization of drilling process, J. Mater. Process. Technol. 74 (1998) 149–157.

[16] J.Y. Kao, Y.S. Tarng, A neural network approach for the on-line monitoring of the electrical discharge machining process, J. Mater. Process. Technol. 69 (1997) 112–119.

[17] H.S. Liu, Y.S. Tarng, Monitoring of the electrical discharge machining process by adaptive networks, Int. J. Adv. Manuf. Technol. 13 (1997) 264–270.

[18] K.M. Tsai, P.J. Wang, Study on parameters optimization for electric discharge machining, in: Proceedings of the 14th National Conference on Mechanical Engineering, Vol. 5, Chinese Society of Mechanics, Toayuan, Taiwan, 1997, pp. 165–171.

[19] J.S.R. Jang, C.T. Sun, E. Mizutani, Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice-Hall, Englewood Cliffs, NJ, 1997.

[20] N.B. Karayiannis, A.N. Venetsanopoulos, Artificial Neural Networks: Learning Algorithms, Performance Evaluation, and Applications, Kluwer Academic Publishers, Dordrecht, 1993.

[21] J.M. Zurada, Introduction to Artificial Neural Systems, West Publishing Co., 1992.

[22] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybernet. 23 (1993) 665–685.

[23] N. Gulley, J.S.R. Jang, Fuzzy Logic Toolbox: For Use with MATLAB User's Guide, The MathWorks, Inc., USA, 1995.

[24] P.J. Wang, K.M. Tsai, Semi-empirical model on work removal and tool wear in electrical discharge machining, Int. J. Mater. Process. Technol. 114 (2001) 1–17.