Journal of Manufacturing Engineering, March 2011, Vol. 6, Issue 1, pp 24-29

PREDICTION OF FLANK WEAR IN DRILLING USING DIFFERENTIAL EVOLUTION TRAINED NEURAL NETWORKS

***Pandu R Vundavilli[1], Jagdeesh Anne[2] and Subba Rao C V[3]**

[1]Mechanical Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andra Pradesh-521180, India.
[2]Mechanical Engineering, Rungta College of Engineering & Technology, Bhilai, Chattishgarh -490024, India.
[3]Mechanical Engineering, QIS College of Engineering & Technology, Ongole, Andra Pradesh-523272, India.

ABSTRACT

The rising demand for enhanced performance of manufacturing system has led to new challenges for the development of complex tool condition monitoring techniques. Estimation of tool life generally requires considerable time and relatively expensive. In the present paper, Differential Evolution trained Neural Network (DE-NN) has been developed to predict the flank wear in drilling operation. In DE-NN, the flank wear prediction problem of a drilling operation has been modeled using an NN, whose weights and bias values are optimized offline, using DE. The performance of the developed approach has been compared in terms of their prediction accuracy.

Keywords: *Flank Wear, Drilling, Differential Evolution, Neural Network.*

1. Introduction

In the recent past, condition monitoring based maintenance philosophy is emerging to be the key component in lowering operating costs and increasing machine availability. It is important to note that the prediction of tool wear plays a very significant role to realize a fully automated manufacturing system. Among all the manufacturing operations, drilling is one of the major machining operation and widely used in the manufacturing industries. During drilling operation, flank wear can be considered as one of the important tool failure criterion and is used to replace the tool. Few experimental investigations were made to identify the wear of a drill bit. Brinksmeier [1] relied an eddy current sensor to measure the in-process torque, which is sensitive to tool wear and fracture. Oh et al. [2] estimated drilling torque using the spindle motor RMS current and controlled the torque through a PID controller by manipulating the feed-rate. Moreover, Zhang et al. [3] developed a model to predict the flank wear, after considering the influence of temperature, adhesion and abrasion of the flank wear. However, their model did not include the dynamics of the problem. In [4], it had been proved that the dynamic components have much higher impact on drill wear compared to those of the static forces. EI-Wardeny et al. [5] performed the condition monitoring of drill utilizing vibration signal. They presented a study using the kurtosis of the time domain and area under the power spectrum to monitor various types of drill wear.

Few attempts were made to model the drill wear using statistical regression analysis. The monitoring of tool wear based on current signals of spindle motor and feed motor was modeled using regression analysis [6]. Moreover, Chowdary and Raju [7] proposed a regression model to measure the flank wear and corner wear of drill bit in operation. During regression analysis, as the models are developed independently, the interdependency of the output responses might be lost. Hence, it is necessary to think of an alternative, which will consider all input parameters and responses as an integral system. Few researchers have used soft computing-based tools, such as Neural Networks (NN), Fuzzy Logic (FL), Genetic Algorithms (GA), Differential Evolution (DE) and their various combinations to model input-output relationships of various manufacturing systems.

A back propagation neural network (BPNN) had been proposed in [8] to predict the tool life in drilling, after considering the cutting parameters, namely cutting speed, feed and drill diameter. It is also important to note that radial basis function network (RBFN) was used in [9] to predict the flank wear, and compared the result with experimentally obtained data. O. Yumak et al. [10] developed FL and NN-FL systems to predict tool wear conditions in drilling. Later on Panda et al. [11] compared the results of BPNN and RBFN in predicting the drill flank wear and observed that the performance of former was better than the later. It is important to note that back propagation neural network utilizes gradient based algorithms to update the weights of the NN. Therefore, the chance of the solution to struck in the local minima is more. To avoid this

**Corresponding Author - E- mail: panduvundavilli@gmail.com*

problem and to obtain near optimal solution, the weight of the NN can be updating using the global optimizers, like GA [12], DE and etc. In [13], differential evolution trained neural network had been developed to predict the drill wear monitoring. However, they did not consider the feed and radial vibrations, which are a critical measure of drill wear.

In the present work, an attempt has been made to develop DE trained NN to predict drill flank wear in operation. Drill diameter, feed rate, spindle speed, thrust force, torque, feed and radial vibrations are considered as inputs and drill flank wear has been considered as output of the model (that is, NN). The weights and bias values of the NN are optimized offline, using DE to minimize the Mean Square Error (MSE) in prediction of the flank wear. The experimental data available in [11] was used to train the NN.

The rest of the manuscript is organized as follows: Section 2 introduces the experimental details of the manufacturing process to be modeled. The proposed approach (that is, differential evolution trained NN) is explained in Section 3. Results are discussed and presented in Section 4. Section 5 provides with the concluding remarks of the present study.

2. Experimental Details

The detection of flank wear during cutting is one of the most crucial considerations. It is important to note that the flank wear in drill depends upon drill diameter, spindle speed, feed rate along with some other derived parameters, such as thrust force, torque and vibrations [11]. It is to be noted that the input-output data used for modeling have been collected from the available literature [11]. The experimental setup used in the above literature is shown in Fig. 1. An inverted metallurgical microscope was used to measure the drill flank wear and the experimental data has been taken from the literature [11]. In the above work, the materials used for the drill and work piece are High speed steel and cast iron, respectively.

3. Differential Evolution Trained Neural Network (DE-NN)

The flank wear prediction problem of drilling has been modeled using artificial NN. The parameters (that is, weights, coefficient of transfer functions and bias values) of the NN are optimized using an evolutionary algorithm, DE. The schematic diagram showing the operation of DE-NN is shown in Fig. 2. As the evolutionary algorithms are found to be computationally expensive, the DE-based training of NN is carried out offline. In the proposed approach, coefficient of transfer functions, bias value and the connecting weights of the fully connected feed-forward NN are optimized. Thus, the optimal NN will be evolved by the DE-based training.

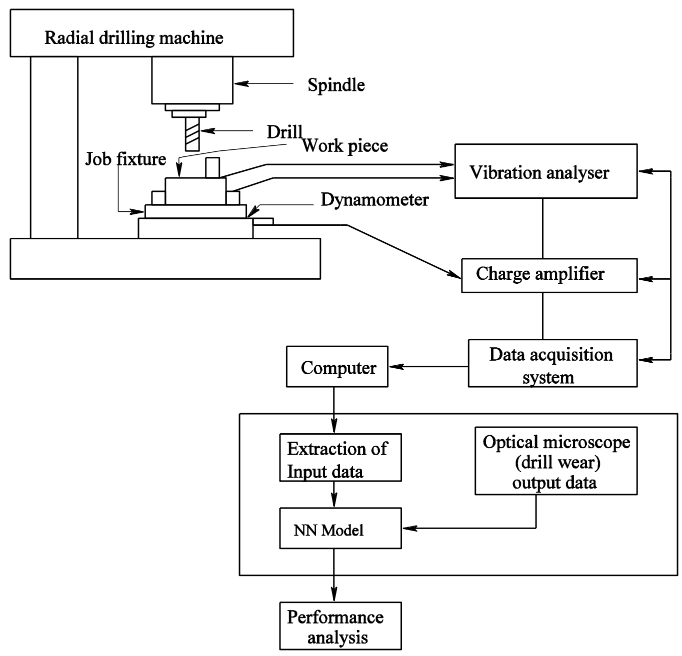


Fig. 1 Schematic Diagram showing the Experimental Setup [1]

In the present work, NN has been used to model the flank wear prediction of a radial drilling machine. Different parameters, such as drill diameter, spindle speed, feed rate, thrust force, torque, feed and radial vibrations that have influence on flank wear are considered as inputs and flank wear is treated as output.

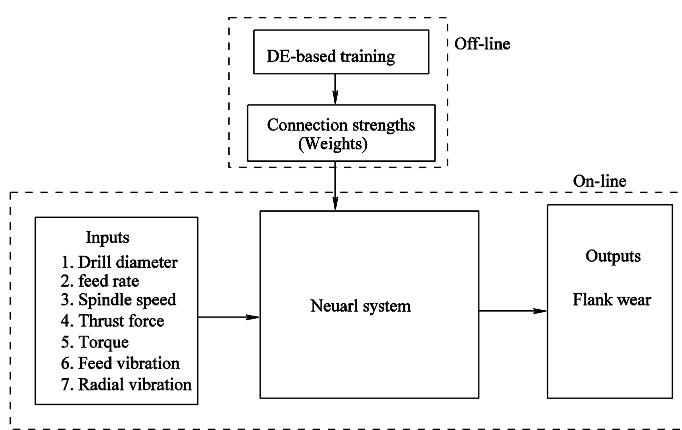


Fig. 2 Schematic View Showing the Working Principle of a DE-NN System

Fig. 3 shows a three layered feed-forward NN considered in this study. The input layer consists of seven neurons corresponding to the seven parameters, which influence the drill flank wear. The output layer consists of one neuron representing the response to be predicted from the network (that is, flank wear). In order to bring the various input parameters on to the same scale, the data used for training and testing of the network are normalized in the range of 0.1 – 0.9 [11] using equation (1).

$$X_{norm} = 0.1 + 0.8\left(\frac{X - X_{\min}}{X_{\max} - X_{\min}}\right) \qquad (1)$$

where X_{norm} is the normal value of a variable, X indicate the value before normalization, X_{min} and X_{max} are the minimum and maximum values of the variable, respectively. It is important to note that the performance of the network is greatly influenced by the topology of the network. Therefore, the number of neurons in the hidden layer and the transfer functions used in different layers of the network are to be carefully determined. The information related to the architecture of NN, such as the connecting weights [V] and [W], coefficients of transfer functions (ct_1, ct_2, and ct_3) and bias value (b_1) are coded in the DE-string, whereas the NN will compute the expected output. Let us assume that the hidden layer of the NN consists of M neurons. Then one particular population of DE used to represent such a network is as follows:

$$\underbrace{0.8546}_{V_{1,1}}...\underbrace{0.9421}_{V_{7,M}}\underbrace{0.1154}_{W_{1,1}}...\underbrace{0.3524}_{W_{M,1}}\underbrace{0.5541}_{ct_1}\underbrace{0.3452}_{ct_2}\underbrace{0.1254}_{ct_3}\underbrace{0.6543}_{b_1}$$

Differential evolution is a novel minimization method developed by Storn and Prince [14]. It starts with an initial population generated at random. The dimension of each population (also called as vector) depends on the number of parameters involved in the optimization. DE starts with a population of fixed number (NP) of D-dimensional vectors and the number is constant through out the training process of NN. In this case, each vector consists of weights of the NN (Vij and Wjk), coefficient of transfer functions (ct1, ct2 and ct3) and bias values (b_1), which need to be optimized. The parameters are initialized with random number between [0, 1]. The initial solutions of vectors are generated using the following formula:

$$X_{D,G} = X_{D,G}^{\min} + (X_{D,G}^{\max} - X_{D,G}^{\min}) \times r, \qquad (2)$$

Where X is the variable, X^{max} the upper bound, X^{min} the lower bound of the variable, and r is the uniformly distributed random number in the range [0, 1]. The population is successively improved by the mutation; crossover and selection operators (refer to Fig. 4).

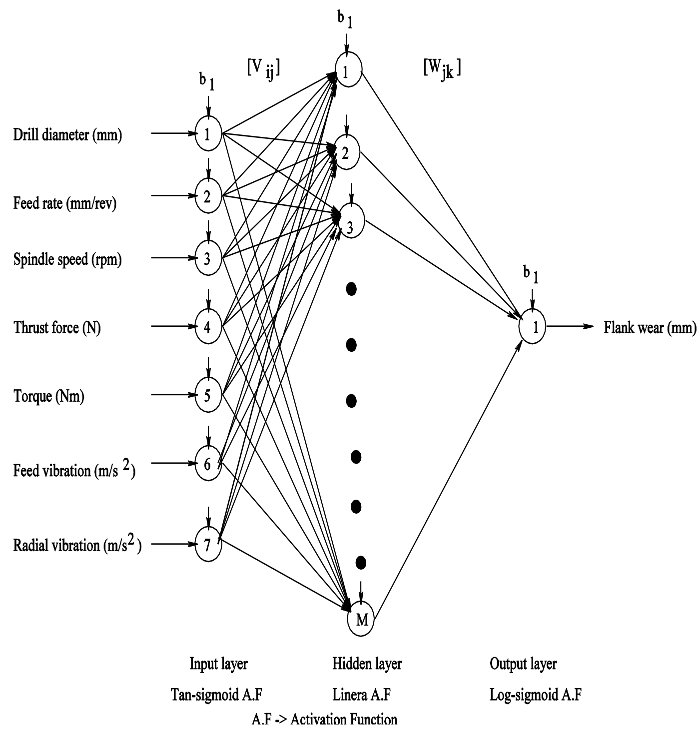


Fig. 3 Architecture of the Proposed Neural Network

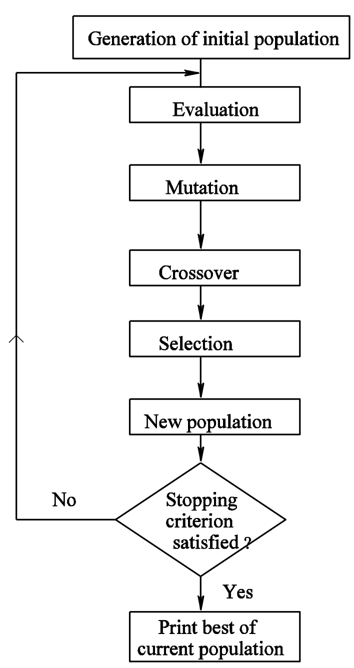


Fig. 4 Flow Chart Showing Working Principle of Differential Evolution

The new parameter vectors in the next generation are obtained after extracting the distance and direction information from the current vectors. Considering $X_{i,G}$ as the target vector in the G^{th} generation, a corresponding donor vector $V_{i,G+1}$ is obtained. In the present study, "DE/rand/1" mutation scheme has been employed to generate the donor vector or mutant vector. The expression for generating the donor vector for the said mutation scheme is as follows:

$$V_{i,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) \quad (3)$$

Where F is mutation constant or scaling factor in [0, 2], which controls the amplification of difference between two individuals, and *i, r1, r2, r3* are the index of individuals selected randomly and are distinct. The crossover operator has been introduced to increase the diversity of the mutant vectors. The trail vector $U_{ji,G+1}$ is developed from the elements of target vector, $X_{i,G}$ and the elements of the donor vector, $V_{i,G}$ as follows:

$$U_{ji,G+1} = \begin{cases} V_{ji,G+1}, if\ (ran_{j,i} \le CR) \\ X_{ji,G}, otherwise \end{cases} (j = 1, 2, ...n) \quad (4)$$

To determine the member for the next generation, the trail vector produced by the crossover operator has been compared with the target vector. If the trail vector produces a smaller objective functional value, it is passed to the next generation otherwise target vector is copied in to the next generation.

$$X_{i,G+1} = \begin{cases} U_{i,G+1}, if\ (f(U_{i,G+1}) < f(X_{i,G})) \\ X_{i,G}, otherwise \end{cases} (i = 1, 2, ...NP) \quad (5)$$

Out of 64 data sets (that is, taken from [11]), 54 data sets (training data set) are selected at random are used for training of the network, and the remaining 10 data sets shown in Table 1 are used for testing of the network. As a batch mode of training is adopted, the whole training set is passed through the NN represented by a GA-string. The Mean Square Error (MSE) in prediction is used as the fitness of the DA-vector. Thus, the fitness F of the DE-vector is calculated like the following:

$$F = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} (T_i - O_i)^2 \quad (6)$$

Where N represents the number of training scenarios, T_i and O_i represents the target and predicted outputs, respectively. Mutation, recombination and selection continue until some stopping criterion is met. Here, the criterion is the number of generations is equal to the predefined maximum value.

Table: Input-output Data for Test Cases

S. No	**Drill dia** (mm)	**Spindle speed** (rpm)	**Feed rate** (mm/rev)	**Thrust Force** (N)	**Torque** (Nm)	**Feed vibration** (m/s^2)	**Radial vibration** (m/s^2)	**Flank wear** (mm)
1	9	500	0.13	1088.1	10.67	37.36	39.28	0.1
2	9	400	0.18	186.4	15.01	29.32	30.48	0.15
3	9	315	0.36	2778	27.82	17.63	18.4	0.12
4	10	500	0.18	1504.8	15.11	41.12	42.24	0.12
5	10	315	0.13	1627.3	15.8	33.27	35.52	0.11
6	10	250	0.18	1869.7	18.64	22.51	24.24	0.18
7	11	400	0.25	2538.9	25.42	38.23	39.28	0.15
8	11	315	0.25	2753.8	27.68	31.58	33.41	0.16
9	12	500	0.25	1856.3	23.51	54.62	56.24	0.1
10	12	315	0.25	2612.6	26.21	41.52	43.37	0.21

4. Results and Discussion

The performance of NN generally depends on various parameters, such as number of neurons in the hidden layer and type of transfer functions used in each layer of NN. The NN is found to have 7 neurons in its hidden layer. Moreover, the NN is seen to yield minimum MSE with the combination of Tan sigmoid, Linear and Log sigmoid transfer functions at input, hidden and output layers, respectively. The expressions for transfer functions used in different layers of NN are as given below:

input layer: $$y = \frac{e^{ct_1 X} - e^{-ct_1 X}}{e^{ct_1 X} + e^{-ct_1 X}} \quad (7)$$

hidden layer: $y = ct_2 X$ (8)

output layer: $$y = \frac{1}{1 + e^{-ct_3 X}} \quad (9)$$

After fixing the number of neurons in the hidden layer and type of transfer functions, the number of DE variables are found to be equal to 60 ((7×8) + 3 + 1). The total variables represent connecting weights ([V] and [W]), coefficients of transfer functions (ct_1, ct_2 and ct_3) and bias value (b_1). During training the connecting weights, coefficient of transfer functions and bias values are varied in the ranges of (-1.0, 1.0), (0.0, 1.0) and (0.0, 0.000001), respectively. As the performance of DE depends on its parameters, namely crossover, mutation and number of vectors, a detailed parametric study is conducted to determine the optimal parameters (refer to Fig. 5).

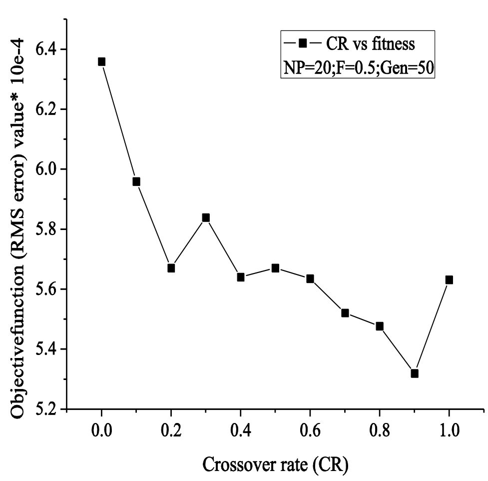
The optimal values of DE-parameters, such as crossover rate, mutation factor and number of generations are found to be equal to 0.9, 0.5 and 50, respectively.

The optimized values of connecting weights $[V_{ij}]$ and $[W_{jk}]$ obtained after DE-based training are as follows:

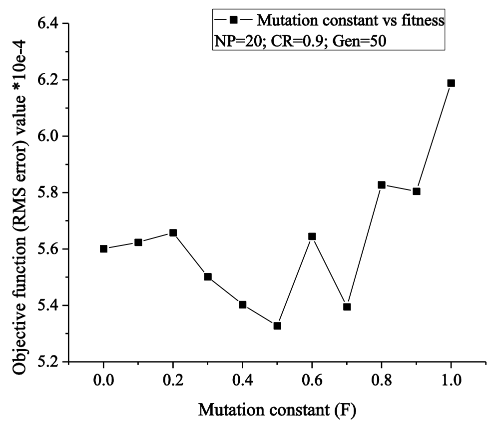
$$V_{ij} = \begin{bmatrix} 0.704 & 0.837 & 0.901 & 0.599 & 0.669 & 0.917 & 0.586 \\ 0.101 & 0.274 & 0.298 & 0.506 & 0.257 & 0.223 & 0.186 \\ 0.232 & 0.786 & 0.873 & 0.924 & 0.115 & 0.215 & 0.325 \\ 0.781 & 0.882 & 0.185 & 0.652 & 0.274 & 0.289 & 0.334 \\ 0.621 & 0.934 & 0.062 & 0.321 & 0.459 & 0.962 & 0.193 \\ 0.676 & 0.312 & 0.819 & 0.246 & 0.866 & 0.117 & 0.977 \\ 0.961 & 0.342 & 0.938 & 0.328 & 0.329 & 0.134 & 0.883 \end{bmatrix}$$

$$W_{jk} = \begin{bmatrix} 0.878 \\ 0.027 \\ 0.158 \\ 0.576 \\ 0.909 \\ 0.343 \\ 0.228 \end{bmatrix}$$

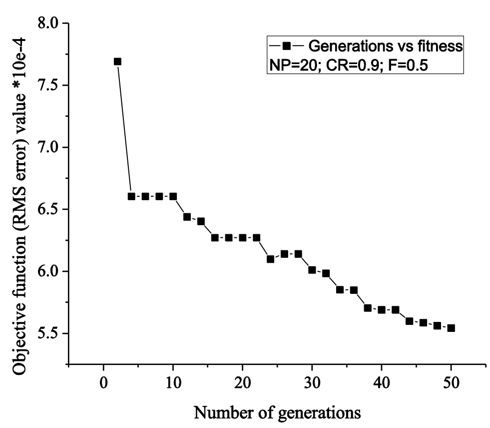
Moreover, the coefficients of transfer functions ct_1, ct_2, ct_3 and bias value b_1 are found to be equal to 0.611, 0.632, 0.562 and 0.00001, respectively. Once the offline training is over, the performance of the optimized network is tested on 10 test cases given in Appendix-A (which are different from the training cases and obtained through real experiments conducted by S.S. Panda et al. [1]).



(a) Crossover Rate vs Fitness



(b) Mutation Constant vs Fitness



(c) Maximum Number of Generations vs Fitness

Fig. 5 (a-c) Results of Parametric Study to Determine the DE Parameters

Fig. 6 shows the comparison of the predicted outputs by the DE-trained NN with their respective experimental values. It is seen from the scatter plot most of the model predicted values are close to the experimental values.

The percentage deviation in prediction of the response (that is, flank wear) for 10 test cases are shown in Fig. 7. It has been observed from the graph that the values of percentage deviation is found to lie in the range of (-10.54, 11.98) for the output – flank wear. Moreover, the average absolute percentage deviation in prediction of flank wear is found to be equal to 9.96. Thus the DE-trained NN is found to successfully modeled and predicted the flank wear in drilling with a reasonably good accuracy for the drilling process.

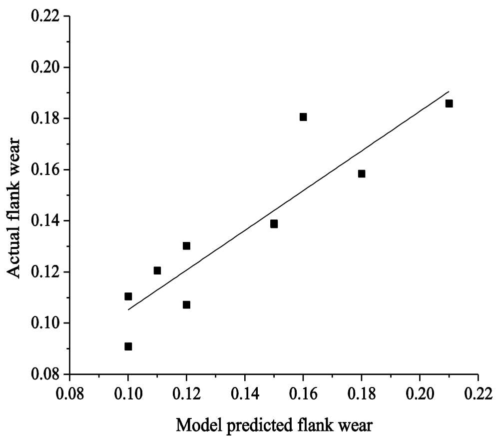


Fig. 6 Results of Parametric Study to Determine the DE Parameters

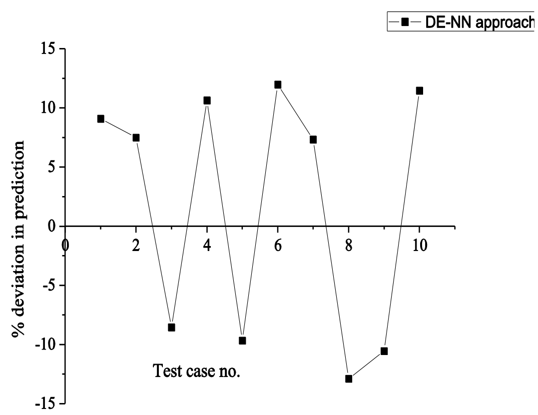


Fig. 7 Plot Showing Percentage Deviation in Prediction of Flank Wear

5. Conclusions

In the present paper an attempt is made to establish the input-output relationship of drilling process using feed forward neural network. Moreover, the optimal structure of neural network has been developed with the help of a popular global search and optimization algorithm, Differential Evolution. It is interesting to note that the accuracy in prediction of the response is tested for different test cases and found reasonably good prediction accuracy for the output. It could be due to the combined effect of steady learning capability of NN and global optimization feature of DE.

References

1. *Brinksmeier E (1990), "Prediction of Tool Fracture in Drilling, ANN", CIRP, Vol. 39(1), 97-100.*

2. *Oh Y T, Kwon W T and Chu C N (2004), "Drilling Torque Control using Spindle Motor Current and its Effect on Tool Wear", International Journal of Advanced Manufacturing Technology, Vol. 24, 327-334.*

3. *Zhang M Z, Liu Y B and Zhou H (2001), "Wear Mechanism Maps of Uncoated HSS Tools Drilling Die-Cast Aluminium Alloy", Tribology International, Vol. 34, 727-731.*

4. *Nede V and Himburg T (1986), "Prediction of tool Fracture in Drilling", Automatisiteerte Uberwachung Des Bohrens, VDI-Z, Vol. 128 (17), . 651-657.*

5. *Wardeny T I E I, Goa D and Elbestawi M A (1996), "Tool Condition Monitoring in Drilling using Vibration Signature Analysis", International Journal of Machine Tools and Manufacturing, Vol. 36 (6), 687-711.*

6. *Xiaoli L and Tso S K (1999), "Drill Wear Monitoring based on Current Signals", Wear, Vol. 231, 172-178.*

7. *Chowdary S K, Raju G (2000), "Investigation into Crater Wear in Drilling", International Journal of Machine Tolls and Manufacturing, Vol. 40, 887-889.*

8. *Vishy K, Tossapol K (2003), "Tool Condition Monitoring in Drilling using Artificial Neural Networks", Proc. of the Conference Artificial Intelligence, Perth 3-5, 293-301.*

9. *Tsao C C (2002), "Prediction of Flank Wear of Different Coated Drills for JISSUS Stainless Steel using Neural Networks", Journal of Material Processing Technology, Vol. 123, 354-360.*

10. *Yukum, Ertunc H M (2006), "Tool Wear Condition Monitoring in Drilling Process using Fuzzy Logic", ICONIP, Part III, LNCS 4234, 508-517.*

11. *Panda S S, Chakraborthy D and Pal P K (2008), "Flank Wear Prediction in Drilling using Back Propagation Neural Network and Radial Basis Function Network", Alied Soft Computing, Vol. 8, 858-871.*

12. *Alagumuthii N (2009), "Work Roll Grinding - Modeling and Optimization using Artificial Neural Networks and Genetic Algorithm", Journal of Manufacturing Engineering, Vol. 4(1), 45-51.*

13. *Desai C K and Shaikh A A (2006), "Drill Wear Monitoring using Artificial Neural Network with Differential Evolution Learning", Proc. Of IEEE International Conference on Industrial Technology, 2019-2022.*

14. *Storn R and Prince K (1997), "Differential Evolution – A Sample ans Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, Vol. 11(4), 1997, 341-359.*

