African Journal of Engineering Research
Vol. 3(1), pp. 1-13, January 2015
ISSN: 2354-2144
Review

Online monitoring and accidents diagnosis system for research reactors

Magy Mohamed Kandil

Egyptian Nuclear and Radiological Regulatory Authority (ENRRA), Egypt.

Accepted 6 January, 2015

ABSTRACT

In a nuclear research reactors plant, a fault can occur in a few milliseconds, so locating the fault might be of utmost importance due to safety, and other important reasons. Accordingly, there is an increasing demand for automated systems to diagnose such failures. This paper proposes a new hybrid algorithm which considered as a novel method to detect faults in Egyptian research reactor in real-time. The new hybrid algorithm generates and processes many common faults by recognizing and modeling evolution in stochastic phenomenon. It focuses on detection of fault in data sequences by optimizing the back-propagation algorithm (BP), to determine the different common faults. Both differential evolution (DE) algorithms and neural networks, which are inspired by computation in biological systems, are emerged as established techniques for optimization and learning. So, this paper presents a software implementation of a neural network that had been optimized by Differential Evolution Algorithm to obtain the optimum construction of back-propagation algorithm (OBP). The new hybrid algorithm diagnoses the Multi-Purpose Research Reactor of Egypt accidents patterns, to avoid the risk of occurrence of a nuclear accident. Experimental results show that the proposed algorithm is very successful in detecting faults.

Keywords: Differential evolution algorithms, accidents, Egyptian research reactor, artificial neural networks, differential evolution algorithm, fault diagnosis.

E-mail: Magy_kandil@yahoo.com.

INTRODUCTION

This paper deals with the design of a Computerized Monitoring and Diagnosis System Egyptian Nuclear Research Reactor based on real time plant specific safety parameters. The system is made by proposing a new hybrid algorithm which carries out early detection and identification of accidents that might affect the reactor using optimized Back-Propagation algorithm. The graphical programming language matlab is used for creating human operator interface, networking, embedding the diagnosis procedure, handling and storage of data. The methodology presented in this paper can be adapted to any nuclear research reactor. In order to detect anomalies as soon as possible it is important to characterize various statistical features of signals acquired under normal operating conditions.

Early detection has two specific features. First, if operations staff can detect the anomalies at the earliest stage possible, negative effects on plant operation can be reduced. Second, a short detection time and high background noise level leads to large statistical errors during anomaly discrimination.

Neural networks (NNs) have more advantages than any other artificial learning methods, since they are able to deal with several data types. Back Propagation (BP) algorithm (Admitted and Pate, 2009) is one of the most common supervised training methods. The main attribute which distinguishes BP from traditional econometric methods is its ability to generate non-linear relationships between a vector of input variables and a dependent. Back-Propagation also has the ability to model any complex system. Although BP training has proven to be efficient in many applications, its convergence tends to be slow, and yields to suboptimal solutions since it convergences to local minima (Fiszelew et al., 2007).

The training algorithm in the BP can be viewed as the optimization of the error with respect to the weights. A

local optimization technique is almost always employed for training and as a consequence our training algorithm usually reaches a local minimum. Furthermore, the particular local minimum will determine the quality of the neural network solution. On one hand, if the minimum is close to the global one the performance will be acceptable and the training will be successful. On the other hand, there are minima that result in poorly trained networks and unsuccessful convergence.

Attempts to speed up training and reduce convergence to local minima have been made in many context of gradient descent such as Sadeq et al. (2000), Yao (1991) and Hamacher (2006). The major algorithms are based on adapting the weights, learning rate, step size and bias to dynamically adapt BP algorithm during its training cycle. There are a number of review papers in this area (for example, Yao (1993)), as well as methodology studies (Binachini et al., 2001; Sprave, 1994).

The factors that determine the final local minimum are mainly the particular weight initialization and the training algorithm. Furthermore, the weight initialization influences the speed of convergence, the probability of convergence and the generalization.

Under the point of speed of convergence, a particular initialization value can be closer or farther than another different value to the same final local minimum. So, the number of iterations of the training algorithm and the convergence time will vary depending on the weight initialization. Considering the probability of successful convergence, it is clear from the initial discussion that a particular weight initialization value can lead the training algorithm to an acceptable local minimum or to a false local minimum. In one case, we will consider that the neural network converged successfully, and in the other that the neural network did not converge.

The probability of successful convergence depends on the weight initialization scheme. Finally, the third effect is the generalization performance of the neural network. Considering the case of two successful convergences, two different local minima were reached. In this case, we have considered that the performance of both local minima is acceptable, but it can be different and therefore the generalization performance will also be different.

Alternatively, the detection of more than one minimizer can be achieved by modifying the objective function, so as to contain information concerning the position of the previously detected minimizers in its new form. In this context, Goldstein and Price proposed an efficient algorithm for the minimization of algebraic functions, which exploits higher order derivatives of the involved polynomials (Laurene, 1999). The technique was later generalized for non-polynomial problems using a transformation, which involves the Hessian of the objective function (Sindu et al., 2014).

However, the numerical, computation of the Hessian is not always feasible and in any case, it is computationally expensive. Thus, in its general form, this approach is not widely applicable. Similar algorithms were proposed by Laurene (1999), but, after the detection of a few minimizers, the objective function becomes very flat and, thus, minimization is becoming increasingly difficult.

An interesting approach is the *tunneling* technique, proposed by Hamacher and Wenzel (1999) for one-dimensional (1-D) functions, and generalized or the multidimensional case by Hamacher and Wenzel (1999) and Wu (1996). However, the hyper surface constructed by the tunneling algorithm becomes very flat as the number of the detected minimizers increases, hindering further exploration of the search space, after the detection of a few minimizers.

A different technique for avoiding local minimizers is *simulated annealing* that combines local search with Monte Carlo techniques and simulates the annealing processes which are used to reveal the low temperature state of materials (IAEA, 1994).

Differential Evolution and BP Algorithms are two techniques for optimization and learning, each with its own strengths and weaknesses, and have generally evolved along separate paths, so recently there have been attempts to combine them.

From the above discussion, we can conclude that the weight initialization is a very important issue. However, the usual way to initialize the weights is at random. This fact seems to be paradoxical because we leave an important topic at random. In the bibliography, we can find several papers on weight initialization for the Multilayer Feed forward. In some of them a new weight initialization scheme is proposed. Therefore, the new hybrid structural to form an optimizing Back-Propagation algorithm is initializing the weights using the Evolutionary Algorithm (Korbicz et al., 2004). DE enables the parallel evolution of a population of DE models which exhibit estimated optimality with respect to multiple functions of errors by using Evolutionary training function. This stage is very important to speed up the training, reduce convergence to local minima and improving the non-linear generalization approach of BP. The hybrid structural optimizes and adapts the weights, and biases to dynamically adapt BP algorithm during its training cycle. The a new hybrid algorithm adapts by implementing the DE domain for initializing the weights to recognize the different fault patterns, with little or no a priori knowledge of the form.

In this paper, for optimizing the Back-Propagation algorithm, the Evolutionary algorithm is adapted and applied for optimizing the initial weights of the back-propagation to skip from the local minimum and enforce them to global minimum. Inundation, the parameters of the BP algorithm are adapted to guarantee that the optimal solution is reached in a number of steps that is independent of the cost function by using Evolutionary algorithm.

Locating a fault in a nuclear research reactors plant, which can occur in a few milliseconds, might be the

utmost interest due to safety, and other reasons. The early detection of such plant's failure could prevent system malfunction or serious damage, which could also lead to disaster. Therefore, an intelligent fault detection (Hwang, 1993) and diagnosis system to deal with inaccurate information has also been greatly required.

One of the best structures that were obtained is two layers BP with correspondence values of weights and biases that are required to construct such network for layer 1and layer 2 respectively. The differential evolution algorithm improves the chances of finding a global solution, due to its random nature. This process involves a large number of complex arithmetical operations. However, the software implementations do not have the desired performance (Laurene, 1999).

One of the basic problems to implement neural network on reconfigurable software, is related to the neurons transfer functions. The problem of representing the arithmetic operations using software is related to using some transfer function like the sigmoid function (frequently used in the Multilayer Perceptron (MLP) model), is not easy to implement the design in the software environment by reaching the global minimum. Since the function reach to local minimum in most learning process, so, for our case, the sigmoid function has been substituted by the evolutionary function during the run of program of BP network that were used to obtain the optimum construction of such neural network.

The rest of the paper is organized as follows: presentation of the Egyptian Second Nuclear Research Reactor; explanation of the new hybrid algorithms which is our complete software optimized Neural Network architecture for fault diagnosis; results and discussions of the proposed system performance; conclusion; future work and finally the references are devoted.

EGYPTIAN SECOND NUCLEAR RESEARCH REACTOR

The Egyptian Second Nuclear Research Reactor, is a multipurpose (MPR), open pool type, 22 MW power, light water cooled and moderated and with beryllium reflectors, which had been mainly designed for radioisotopes production for medical and industrial purposes, semiconductors production, activation analysis, neutron radiography and beam tube experiments, basic and applied research in reactor physics and training. The operation of the reactor is controlled and monitored using: the suspension and control system (SCS) and the Reactor Protection System (RPS). The SCS provide process information to the operator in charge allowing him to control the process systems evolution and reactor power.

The RPS is basically a control system that generates the signals for the protective functions to be carried out by the safety systems. The RPS encloses all electrical and mechanical devices and Circuitry involved in generating those initiation signals associated with protective function carried out by the safety actuation systems. The reactor protection system is based on intelligent units combined with hardwired voting protective logic's are placed in the instrumentation room. The detector and sensors are placed as close as possible to the variables that they supervise. The following accidents diagnosed: 1- Loss of Flow Accident (LOFA), 2- Loss of Power Supply (LOPS), 3- Loss of Heat Sink (LOHS), 4- Small Loss of Coolant Accident (SLOCA), 5- Medium Loss of Coolant Accident (MLOCA), 6- Large Loss of Coolant Accident (LLOCA), 7- Uncontrolled Slow Reactivity Insertion (USRI), 8- Uncontrolled Fast Reactivity Insertion (UFRI) and 9- Normal case (Hwang, 1993).

THE NEW HYBRID ALGORITHM (OBP)

In the new purpose algorithm, a Differential Evolution algorithm was designed and employed to construct an optimum back propagation network (algorithm) (OBP) by optimizing the values of its initial weights that are required to construct such network. In the new purpose algorithm, a neural network is designed and trained to recognize the 9 accidents of the nuclear reactors. By the aid of reactor operation crew and Safety Analysis Report (SAR) of the reactor, also the Egyptian Nuclear and Radiological Regulatory Authority (ENRRA) experts, data sets was collected for the eight accidental cases listed below plus the normal operation case (Classes) as shown in Figure 1. So the total cases, which we have, are nine. The result is that each accident is represented as a 3-by-5 grid of Boolean values.

The nine 15-element input vectors are defined in the function *accmodels* as a matrix of input vectors called *accidents*. The target vectors are also defined in this file with a variable called, *targets*. Each input vector is a 15-element vector (3-by-5) (Figure 2), with a 1 in the position of the accident it represents, and 0's everywhere else. For example, the TR0 is to be represented by a 1 in the first element (as TR0 is the first accident of the accidents), and 0's in elements two through fifteen.

The network receives the 15 Boolean values as a 15-element input vector (3-by-5). It is then required to identify the accident by responding with a 9-element output vector. The 9 elements of the output vector each represent an accident. To operate correctly, the network should respond with a 1 in the position of the accident being presented to the network. All other values in the output vector should be 0. In addition, the network should be able to handle noise. In practice, the network does not receive a perfect Boolean vector as input. Specifically, the network should make as few mistakes as possible when recognizing vectors with noise of mean 0 and standard deviation of 0.2 or less.

TR0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
TR1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
TR2	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1
TR3	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1
TR4	1	1	1	1	1	1	1	0	0	1	0	0	1	1	1
TR5	0	0	0	0	1	1	1	0	0	0	1	0	1	1	1
TR6	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
TR7	1	1	0	0	1	1	1	1	1	1	1	0	0	0	0
TR8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 1. Sample of reactor accidents data patterns.

```
TR0 = [0 0 0 ...        .....        TR8 = [1 1 1 ...
       0 1 1 ...        .....               1 1 1 ...
       1 1 1 ...        .....               1 1 1 ...
       1 1 1 ...        .....               1 1 1 ...
       1 1 1]';         .....               1 1 1]';
```

Figure 2. Samples of (15 × 9) matrix of (3 × 5) bit maps for each accident.

The architecture of the new hybrid algorithm (OBP)

The program flowchart of the new hybrid algorithm is represented in Figure 3. The new hybrid algorithm needs 15 inputs and 9 neurons in its output layer to identify the accidents. The network is a two-layer log-sigmoid/log-sigmoid network. The log sigmoid transfer function was picked because its output range (0 to 1) is perfect for learning to output Boolean values

$$a_2 = f2\ (\ LW_{2,1}\ (\ f1\ (\ LW_{1,1p} + b^1\ ) + b^2) = y_j \qquad (1)$$

Where j is neurons in the output layer, the hidden (first) layer has 10 neurons.

If the network has trouble learning, then neurons can be added to this layer (Figure 4). The network is trained to output a 1 in the correct position of the output vector and to fill the rest of the result in the network's not creating perfect 1's and 0's. After the network is trained the output is passed through the competitive transfer function *compete*. This makes sure that the output corresponding to the accident most like the noisy input vector takes on a value of 1, and all others have a value of 0. The result of this post-processing is the output that is actually used.

The ANN training and testing for the new hybrid algorithm (OBP)

The first step, of the new hybrid algorithm is training the most important parameters of the BP algorithm such as initial weights and bias that are adapted to guarantee that the optimal solution is reached in a number of steps that is independent of the cost function by using evolutionary algorithm.

The second step of the new hybrid algorithm is creating a network that can handle noisy input vectors, it is best to train the network on both ideal and noisy vectors. To do this, the network is first trained on ideal vectors until it has a low sum squared error. Then the network is trained on 10 sets of ideal and noisy vectors. The network is trained on two copies of the noise-free accidents at the same time as it is trained on noisy vectors. The two copies of the noise-free accidents are used to maintain the network's ability to recognize ideal input vectors.

Unfortunately, after the training described above the network might have learned to recognize some difficult noisy vectors at the expense of properly recognizing a noise-free vector. Therefore, the network is again trained on just ideal vectors. This ensures that the network

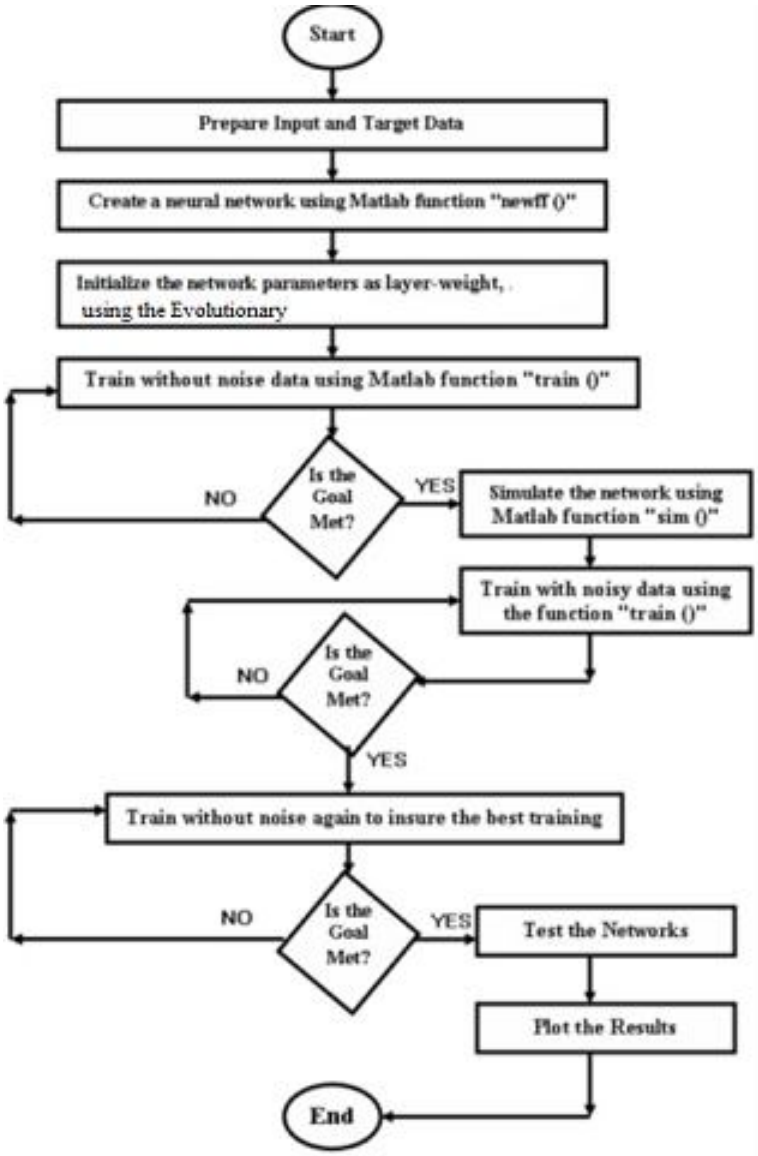

Figure 3. Program flowchart of the new hybrid algorithm.

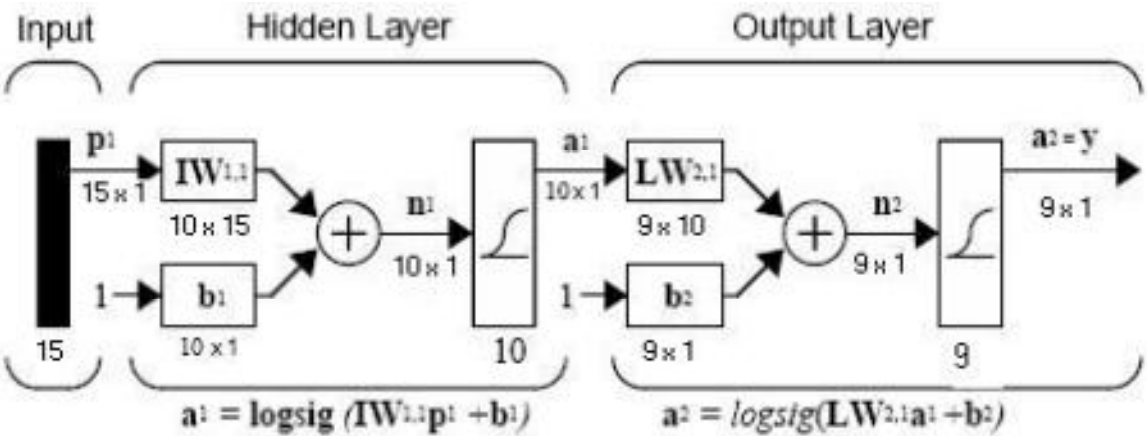

Figure 4. The neural network design for accidents patterns recognition.

responds perfectly when presented with an ideal accident.

All training is done using back propagation with both adaptive learning rate and momentum, with the function *'traingdx'.* The Error (Performance) is calculated in terms of Sum-Squared Error as in equation (2) which is used for training BP feed-forward neural networks is the mean sum of squares of the network errors.

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \quad (2)$$

With these settings, the input vectors and target vectors will be randomly divided into three sets as follows:

i) 70% will be used for training.
ii) 15% will be used to validate that the network is generalizing and to stop training before overfitting.
iii) The last 15% will be used as a completely independent test of network generalization.

For this Fault Diagnosis problem, we train and test the data using BP only DE only and the new hybrid algorithm.

RESULTS AND DISCUSSION

Results for BP algorithm

Training results for BP network

In the back propagating algorithm, the training stopped when the validation error increased for six iterations, which occurred at iteration of 40. The training performance for gradient-descent algorithm reached to the goal with Mean Square Error (MSE) for the training dataset were 0.00671 after 35 epochs as shown in Figure 5.

This training stopped when the validation error increased for six iterations, which occurred at iteration 40. Performance in the training window produces; a plot of the training errors, validation errors, and test errors appears, as shown in Figures 6 and 7. In this Fault Diagnosis example, the results are not reasonable because although the final mean-square error in training is small, the test set error and the validation set error does not have similar characteristics.

Test results for BP network

After the network has been trained, the test computes the network outputs. The regression in the training window represents the performance of a linear regression between the network outputs and the corresponding targets. Figure 8 shows the linear regression results.

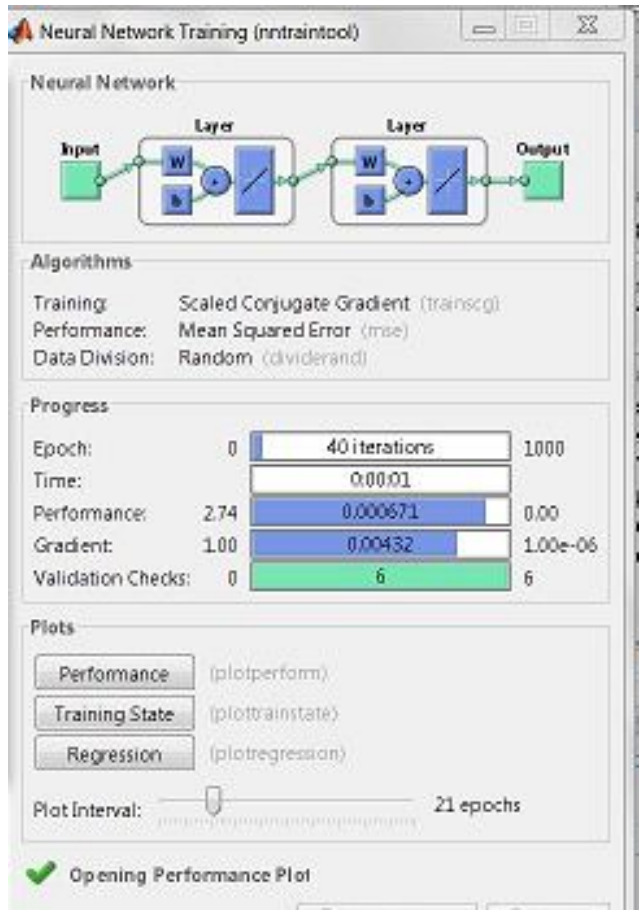

Figure 5. The results for training BP fault features.

As shown in Figure 8, the following regression plots display the network outputs with respect to targets for training, validation, and test sets. For a perfect fit, the data should fall along a 45 degree line, where the network outputs are equal to the targets. Perform some analysis of the network response. The fit is reasonably good for all data sets, with R values in each case of 0.93 or above. The blue bars represent training data, the green bars represent validation data, and the red bars represent testing data. The histogram can give an indication of outliers, which are data points where the fit is significantly worse than the majority of data.

In this case, it is seen that while most errors fall in small range, the errors in validation and test high that are visible on the testing regression plots is considered. The first corresponds to the point represented by the relation between the target and the output. It is a good idea to check the outliers to determine if the data is bad, or if those data points are different than the rest of the data set. If the outliers are valid data points, but are unlike the rest of the data, then the network is extrapolating for these points. From the results, we show that it should collect more data that looks like the outlier points, and retrain the network in this case.

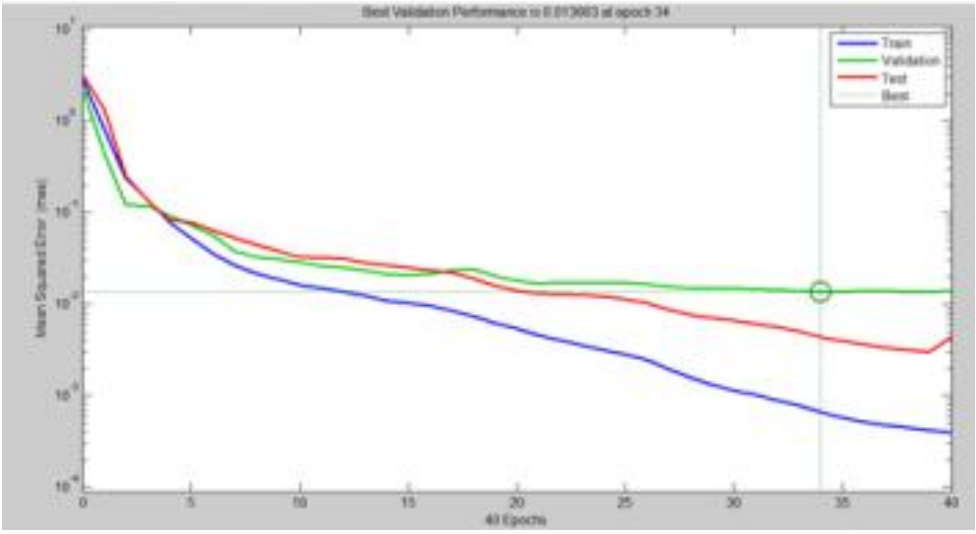

Figure 6. The training performance for BP algorithm only.

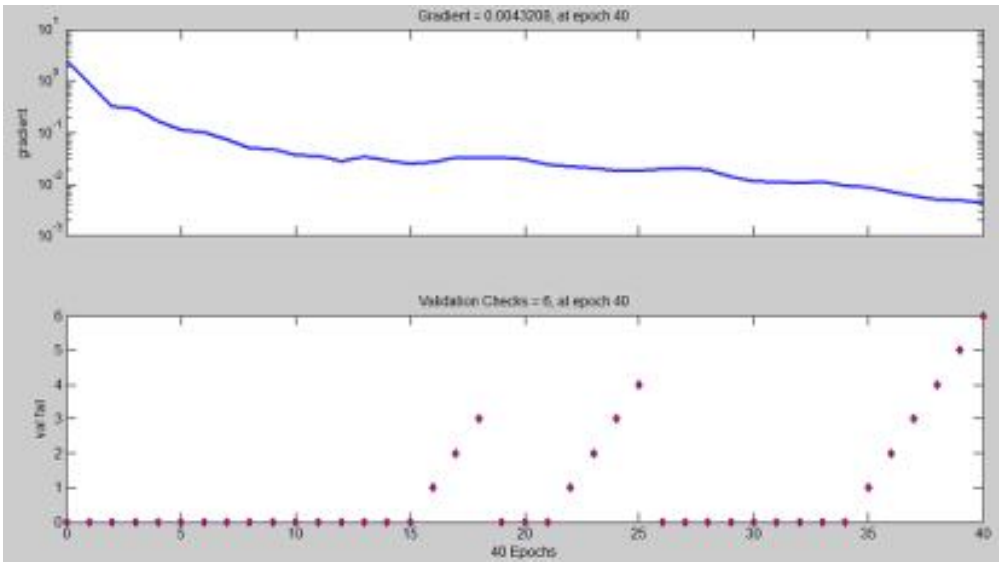

Figure 7. The training state for BP algorithm only.

Results for training differential evolutionary algorithm (DE)

Training DE network

In the DE algorithm, the training stopped when the validation error increased for six iterations, which occurred at iteration of 21. The training performance for gradient-descent algorithm reached to the goal with Mean Square Error (MSE) for the training dataset were 0.070497 after 15 epochs as shown in Figure 9.

This training stopped when the validation error increased for six iterations, which occurred at iteration 21. Performance in the training window, yields a plot of the training errors, validation errors, and test errors appears, as shown in Figures 10 and 11. In this FD example, the result is not reasonable because although the final mean-square error in training is small, the test set error and the validation set error does not have not similar characteristics.

Test results for DE network

After the network has been trained, the test computes the network outputs. The regression in the training window, represents the performance of a linear regression between the network outputs and the corresponding

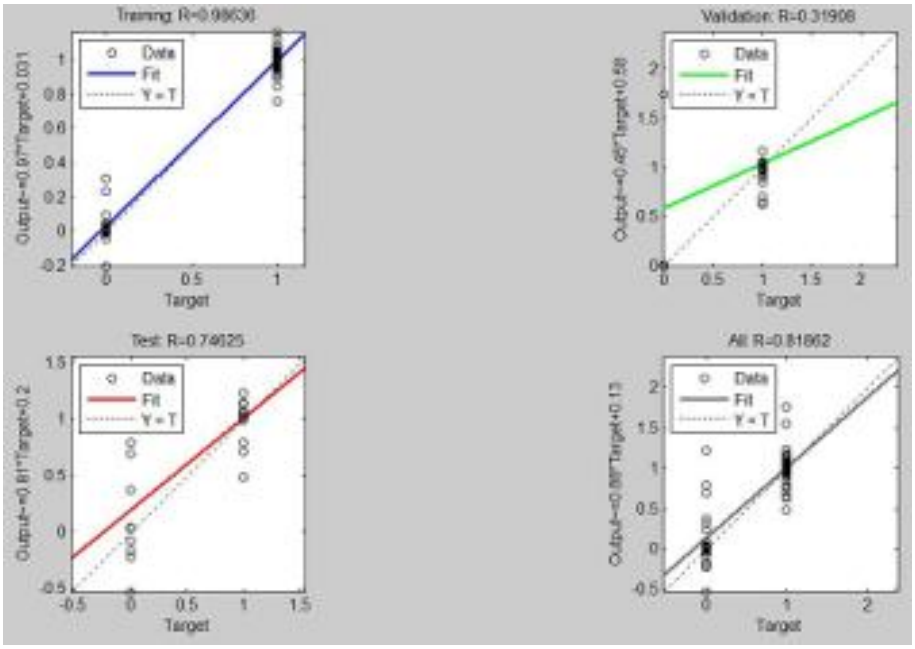

Figure 8. The regression window for BP algorithm.

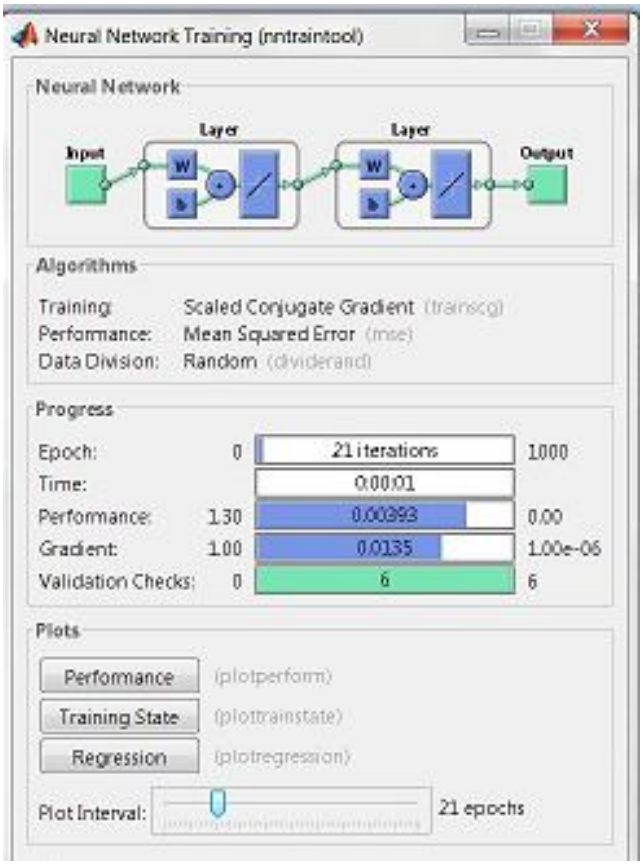

Figure 9. Results for training DE fault features.

targets. Figure 12 shows the linear regression results.

In this case, it is seen that while most errors fall in small range, the errors in validation and test that are visible on the testing regression plots are considered high. The first corresponds to the point represented by the relation between the target and the output. It is a good idea to check the outliers to determine if the data is bad, or if those data points are different than the rest of the data set. If the outliers are valid data points, but are unlike the rest of the data, then the network is extrapolating for these points. From the results we show that it should collect more data that looks like the outlier points, and retrain the network in this case.

Results for OBP algorithm

Training OBP *network*

In the DE algorithm, the training stopped when the validation error increased for six iterations, which occurred at iteration of 118. The training performance for gradient-descent algorithm reached to the goal with Mean Square Error (MSE) for the training dataset were 5.27e-12 after 118 epochs as shown in Figure 13.

This training stopped when the validation error increased for six iterations, which occurred at iteration 118. Performance in the training window, yields a plot of the training errors, validation errors, and test errors appears, as shown in Figures 14 and 15. In this FD example, the result is reasonable because of the final mean-square error in training is very small and the test set error and the validation set error have similar characteristics.

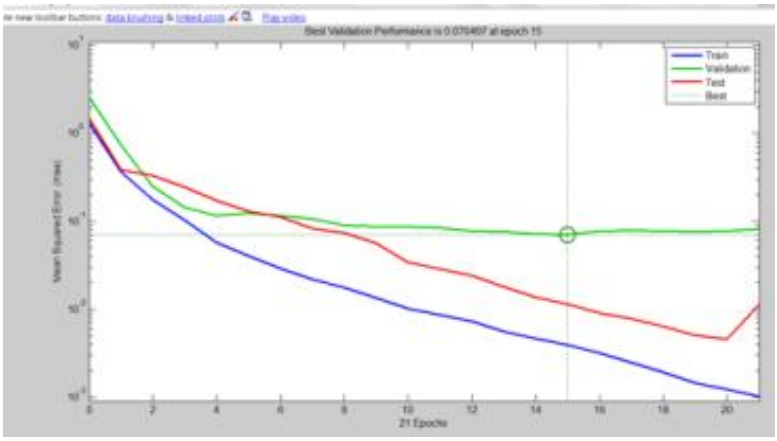

Figure 10. The training performance of the DE algorithm.

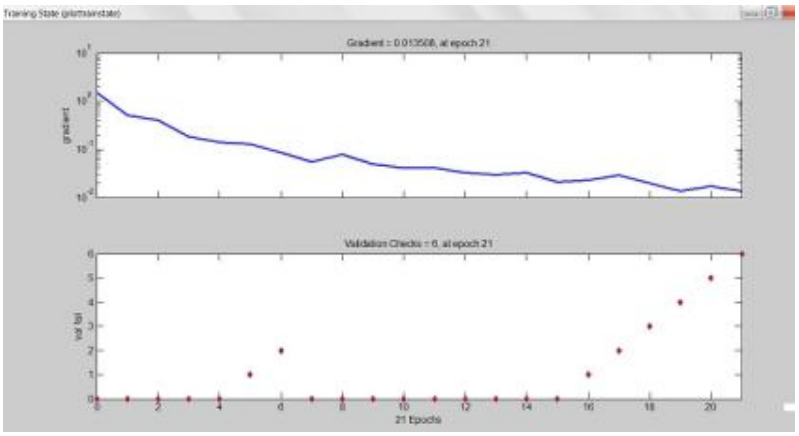

Figure 11. The training state for DE algorithm.

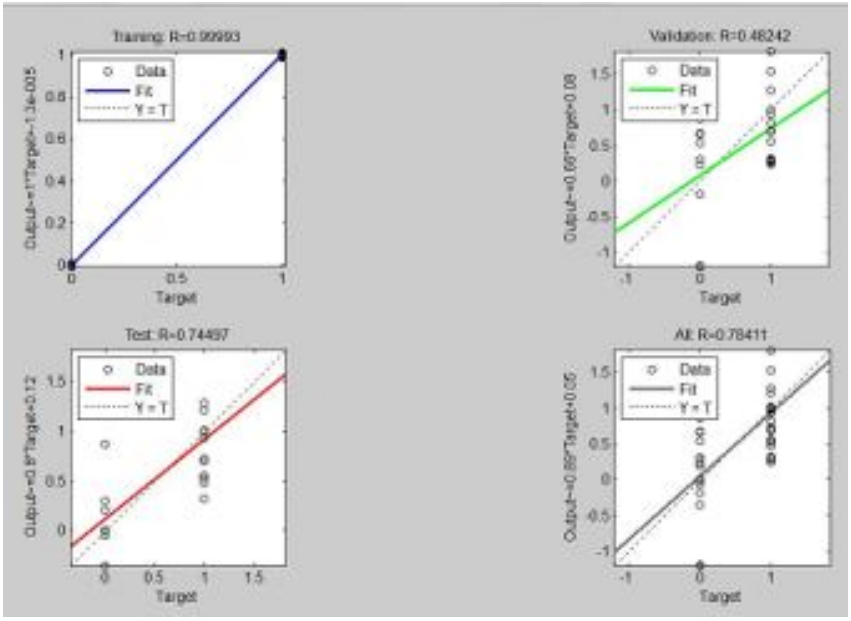

Figure 12. Regression window for DE algorithm.

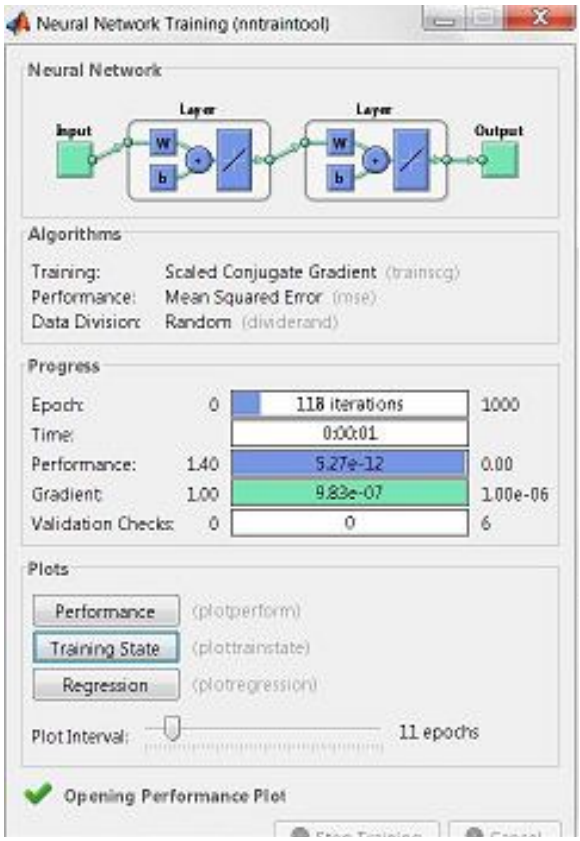

Figure 13. Results for training OBP fault features.

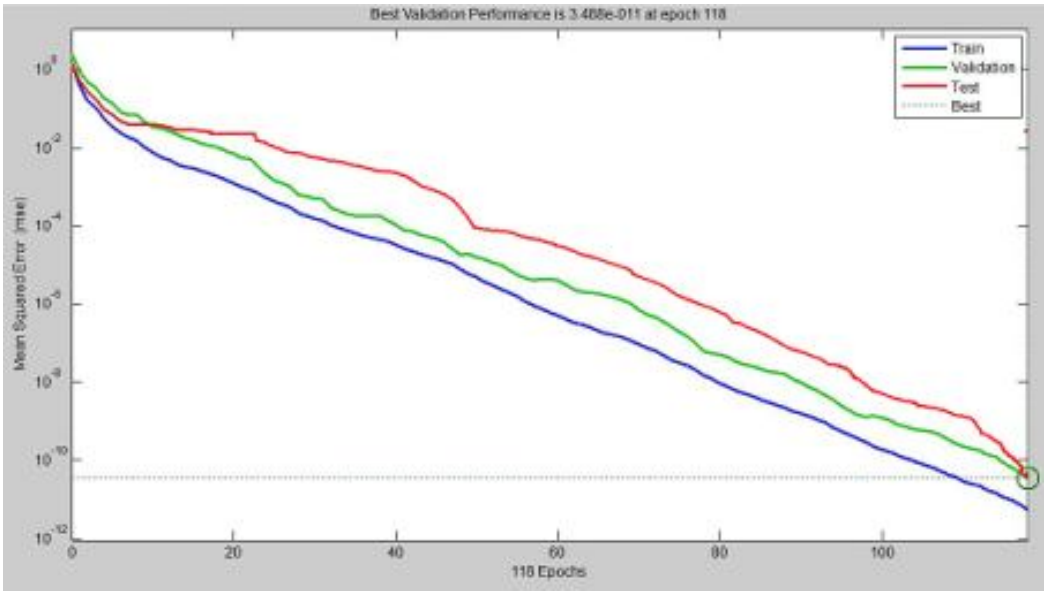

Figure 14. The training performance of OBP algorithm.

Test results for the OBP network

After the network has been trained, the test computes the network outputs. The regression in the training window represents the performance of a linear regression between the network outputs and the corresponding

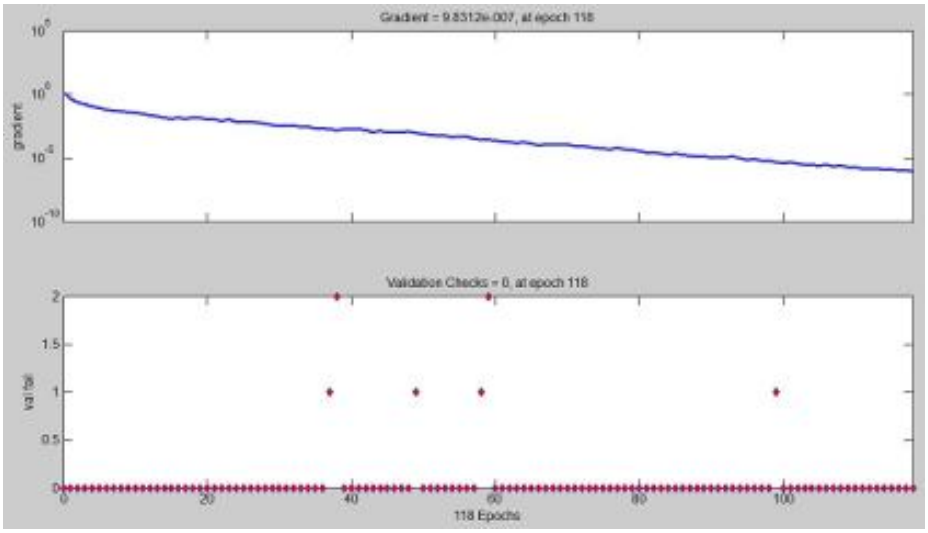

Figure 15. The training state for OBP algorithm.

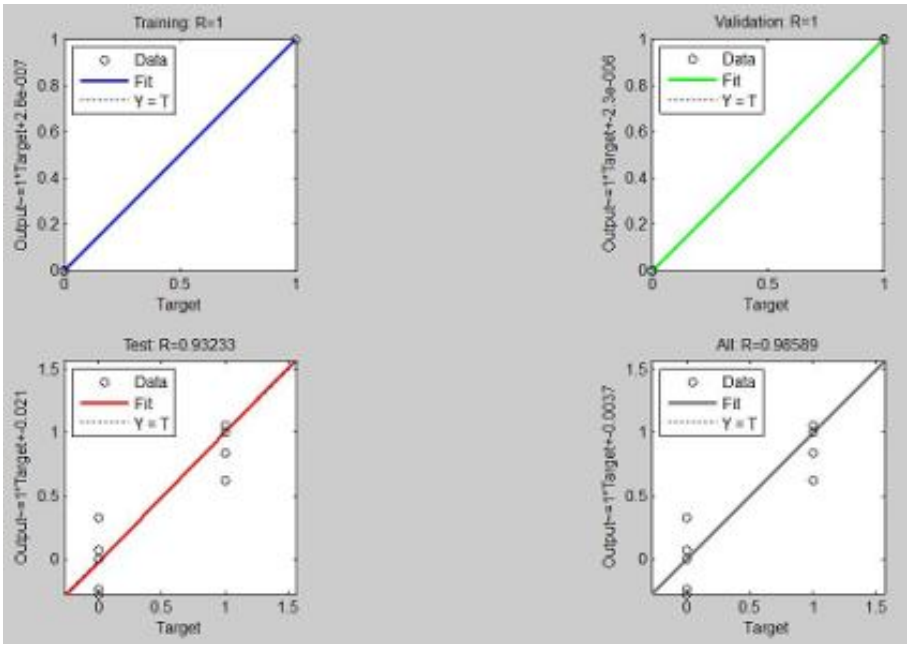

Figure 16. The regression window for OBP algorithm.

targets. Figure 16 shows the linear regression results.

In this case, it is seen that however most errors fall in small range, the errors in validation and test is considered small that are visible on the testing regression plots. The first corresponds to the point represented the relation between the target and the output. It is a good idea to check the outliers to determine if the data is bad, or if those data points are different than the rest of the data set. If the outliers are valid data points, but are unlike the rest of the data, then the network is extrapolating for these points.

Training results of all algorithm

From the previous training results we show that it should not collect more data since the new hybrid algorithm OBP give more accurate data as shown in Table 1.

Testing results of all algorithm

From the previous testing results we show that the new hybrid algorithm OBP give more accurate data as shown

Table 1. Training comparison between the new hybrid algorithm (OBP), the gradient-descent and Evolutionary algorithms.

Algorithm name	Best validation performance	Mean Error	Standard deviation	Success (%)
BP	0.013663	0.000671	0.0043208	74
DE	0.070497	0.00393	0.013508	81.8
OBP	3.488e-011	5.27e-12	9.8312e-07	99

Table 2. The performance CPU execution time and hit rate parameters.

Algorithm name	Average CPU time	Hit rate (%)
BP	0.160	74
DE	0.430	81
OBP	0.002	98.6

in Table 2, which represents comparison of the testing performance between the new hybrid algorithm and the gradient-descent and evolutionary algorithms.

Table 2 represents the validation performance for the new hybrid algorithm, the gradient-descent and evolutionary algorithms after testing them with 100 sampling of fault patterns.

From Table 2, the new optimized back-propagation has the best fault diagnosis accuracy (hit rate) 98.6% so that, the new hybrid algorithm increases the generalization of the NNs and reduces the diagnosis time.

The new hybrid structural optimized Back-Propagation algorithm forms a new hybrid fault diagnosis algorithm for speeding up the training, reducing convergence to local minima and improving the non-linear generalization approach of BP. The hybrid structural optimizes and adapts the *objective error function* to dynamically adapt BP algorithm during its training cycle. The new hybrid algorithm adapts the back-propagation by combing global and local searches algorithms to determine the local minimum of the gradient–descent within the Differential Evolutionary Neural Networks domain for recognizing the different fault patterns, with little or no *a priori* knowledge of the form. The results are reasonable because of the following considerations:

i) The final mean-square error is small.
ii) The test set error and validation set error, have similar characteristics.
iii) No significant over fitting has occurred by iteration 118 (where the best validation performance occurs).

CONCLUSION

Locating a fault in a nuclear research reactors plant, which can occur in a few milliseconds, might be the utmost interest due to safety, and other reasons. The early detection of such plant's failure could prevent system malfunction or serious damage, which could also lead to disaster. Therefore, an intelligent fault detection and diagnosis system to deal with inaccurate information has also been greatly required

The new structural forms a new hybrid real time faults diagnosis algorithm by optimizing BP algorithm for speeding up the training, reducing convergence to local minima and improving the non-linear generalization approach. The hybrid structural optimizes and adapts the parameters of the BP algorithm for optimizing the objective function of the BP algorithm during its training cycle. The new hybrid algorithm optimizes the BP by combining global and local searches algorithms to determine the local minimum of the gradient–descent within the Evolutionary Neural Networks domain for recognizing the fault patterns.

This new hybrid algorithm was designed to diagnose and predict the Multi-Purpose Research Reactor of Egypt accidents, to avoid the risk of occurrence of a nuclear accident.

The model needs less than 2 μs for processing the input values and presenting the results. This is a very fast implementation required for a critical field like the nuclear research reactors, compared to the protection BP only system that needs from 18 to 24 ms as a response time. So, more precise classification accidents can be processed in a shorter period of time.

RECOMMENDATION

In the future, a sequential Hardware Neural Network can be implemented using Xilinx FPGA family (Haggag, 2008). The implementation on FPGA provides the higher benefits of lower costs and higher results (Omondi and Rajapakse, 2006), where FPGA can be reprogrammed for an unlimited number of times; they can be used in innovative designs where hardware is always in dynamic change, or where hardware must be adapted to different

user applications requirements. In the future work we can implement the new hybrid algorithm as neural network's hardware model that can be integrated within the Reactor Protection System (RPS) as a future work, where valuable interesting results will obtain.

REFERENCES

Admitted LS, **Pate** SD, **2009**. Neuro–genetic cost optimization model: Application of textile spinning process. Int J Comput Theory Eng, 1(4): 1793-8201.

Binachini M, Fanelli S, Gori M, **2001**. Optimal algorithm for well-conditioned nonlinear systems of equation. Comput IEEE Trans on, 50(7):689-698.

Fiszelew A, Britos P, Ochoa A, Merlino H, Fernández E, García-Martínez R, **2007**. Finding optimal neural network architecture using genetic algorithms. Adv Comput Sci Eng Res Comput Sci, 27:15-24.

Haggag SS, **2008**. Design and FPGA Implementation of Multilayer Neural Networks with On-chip Learning. Atomic Energy Authority, Egypt 2nd Research Reactor. PhD, Menufia University.

Hamacher K, **2006**. Adaptation in stochastic tunneling global optimization of complex potential energy landscapes. Europhys Lett. 74(6):944.

Hamacher K, **Wenzel** W, **1999**. The scaling behaviour of stochastic minimization algorithms in a perfect funnel landscape. Phys Rev E, 59(1):938-941.

Hwang BC, **1993**. Fault Detection and Diagnosis of a Nuclear Power Plant Using Artificial Neural Networks. Simon Fraser University, March.

Korbicz J, Kowalczuk Z, Koscielny JM, Cholewa W, **2004**. Fault diagnosis: models, artificial intelligence, applications. Springer-Verlag Berlin Heidelberg.

Laurene F, **1999**. Fundamentals of Neural Network. New York: Thomas Publishing Co.

Omondi AR, **Rajapakse** JC, **2006**. FPGA implementations of neural networks. Springer, Dordrecht, The Netherlands.

Sadeq AM, Wahdan AA, Mahdi HMK, **2000**. Differential Evolution Algorithms and its Use with Back propagation Network. Faculty of Engineering, Ain Shams University, 35, 3.

Sindu P, Aravinth E, Raj Thilak K, **2014**. Inter-turn short circuit fault diagnosis using adaptive neuro-fuzzy interface system. Int J Multidisciplinary Curr Res, 2:915-918.

Sprave J, **1994**. Linear neighborhood evolution strategies. In Proc. of the 3-rd Annual Conf. on Evolutionary Programming, San Diego, CA, World Scientific.

The International Atomic Energy Agency (**IAEA**) Safety Series, **1994**. Safety in the Utilization and Modification of Research Reactors. IAEA publications, STI/PUB/961, ViDEa, Austria, December 1994, STI/PUB/961.

Wu Z, **1996**. The effective energy transformation scheme as a special continuation approach to global optimization with application to molecular conformation. Technical Report, Argonne National Lab., IL (United States), November.

Yao X, **1991**. Evolution of connectionist networks. In Preprints Int. Symp. AI, Reasoning and Creativity, Queensland, Australia, Griffith Univ., pp. 49–52.

Yao X, **1993**. Evolutionary artificial neural networks. Int J Intel Syst, 4:539-567.

Citation: Kandil MM, 2015. Online monitoring and accidents diagnosis system for research reactors. Afr J Eng Res, 3(1): 1-13.