Differential Protection of Generator by Using Neural Network, Fuzzy Neural and Fuzzy Neural Petri Net

Prof. Dr. Abduladhem A. Ali *abduladem1@yahoo.com*
College of Eng. /Computer Dept.
University of Basrah
Basrah, Iraq

Asst. Prof. Dr. Abaas H. Abaas *abbashafeth@yahoo.com*
College of Eng. /Electrical Dept.
University of Basrah
Basrah, Iraq

Ahmed Thamer Radhi *dr.ahmed197750@yahoo.com*
Technical Institute/Al-Amarah/Elect. Dept.
Foundation of Technical Education
Missan, Iraq

Abstract

This paper deals with the applications of Artificial Intelligence techniques for detecting internal faults in Power generators. Three techniques are used which are Neural Net (NN), Fuzzy Neural Net (FNN) and Fuzzy Neural Petri Net (FNPN) to implement differential protection of generator. MATLAB toolbox has been used for simulations and generation of faults data for training the programs for different faults cases and to implement the relays. Results of different fault cases are presented and these results are compared among the three implemented techniques of relays and with the conventional differential relay of generator.

Keywords: Differential Protection, Generator Internal Faults, Neural Net, Fuzzy Neural and Fuzzy Neural Petri Net.

1. INTRODUCTION

Synchronous generator is the most important element of power system. Generators do experience short circuits and abnormal electrical conditions. In many cases, equipment damage due to these events can be reduced or prevented by proper generator protection. Generators need to protect from abnormal conditions, when subjected to these conditions, damage or complete failure can occur within seconds, thus requiring automatic detection and tripping. All faults associated with synchronous generators may be classified as either insulation failures or abnormal running conditions [1, 2]. An insulation failure in the stator winding will result in an inter-turn fault, a phase fault or a ground fault, etc. At present the generators are protected against almost all kind of faults using differential methods of protection. Differential relays, in particular the digital ones, are used to detect stator faults of generators. Electric power utilities and industrial plants use electromechanical and solid-state relays for protecting synchronous generators [3]. With the advent of digital technology have made significant progress in developing protection systems based on digital techniques [4,5]. Protection relaying is just as much a candidate for application of pattern recognition. The majority of power system protection techniques are involved in defining the system state through identifying the pattern of current waveforms measured at the relay location. This means that the development of adaptive protection can be essentially treated as a problem of pattern recognition. Artificial Intelligences (AIs) are powerful in pattern recognition and classification. They possess excellent features such as generalization capability, noise immunity, robustness and fault tolerance. AI-based techniques have been used in power system protection and encouraging results are obtained [6, 7]. Artificial neural network is a kind of network structure based on modern biology nervous system research, which shows great application potential on equipment diagnosis by its capabilities of parallel distributed processing, associative memory and self learning. Through learning on multiple types of fault samples, a single NN can memorize characteristics of such faults, thus a single NN can implement diagnosis of most fault types [8]. In fuzzy neural network (FNN), both fuzzy logic and neural network combinations have found extensive applications. This approach involves merging fuzzy systems and neural networks into an integrated system to reap the benefits of both. FNN is an efficient structure capable of learning from examples.

Petri Nets (PNs) [9] are based on the concept that the relationships between the components of a system, which exhibits asynchronous and concurrent activities, could by represented by a net. Petri nets are basically developed for describing and analyzing information flow, and they are excellent tools for modeling asynchronous concurrent systems such as computer systems and manufacturing system, as well as power system protection. The basic concept of PN incorporated into a traditional FNN is used to organize a FNPN system to be translated further into neural nets to adding the learning abilities of NN to the PN. The new structure of FNPN model is trained by the back-propagation way with multi-layered feed-forward nets of ANN which makes FNPN model give appropriate output when input sample is different. In this paper, a generator differential protection schemes by using NN, FNN and FNPN are introduced. The proposed schemes have the ability to detect the fault with higher sensitivity.

2. NEURAL NETWORK

The ANN theories have been applied to pattern recognition, pattern classification, learning, optimization, etc., Rumethart, et al. [10] had proposed a neural network technique called Back Propagation (BP) with multi-layered perceptrons. The technique has been successfully applied to adaptive pattern recognition problem.

The back-propagation approach can also be used in power systems. Some applications have been made in solving electrical problems such as transient stability [11], high impedance fault detection [12], fault location in EHV transmission line, fault location estimator for underground cable, and differential protection of power transformer [13]. The neural network with no feedback connections from one layer to another or to itself is called a "Feed forward Neural Network". A general node model is given in **Figure (1)** to illustrate the idealized model operation.

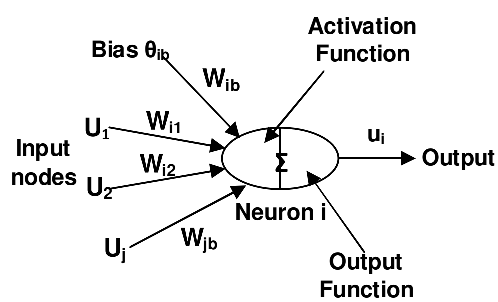

FIGURE 1: Idealized Neuron Operation

Defining output of unit j at the previous layer as u_j, the activation or total input of unit i at the present layer can be written as:

$$S_i=\sum_j Wiju j+\theta_{ib} \tag{1}$$

Where: W_{ij} is the weight of the connection from unit j to unit i.
θ_{ib} is the node threshold (or bias vector).

The output u_i of the unit i is expressed by unit i input S_i:

$$u_i=f(S_i) \tag{2}$$

Where, f(x) is usually, but not necessary the sigmoid function such as:

$$f(x)=1/(1+\exp(-x)) \quad (-\infty<x<+\infty) \tag{3}$$

The outputs of the hidden layer units i are then transmitted to the inputs of next layer units through another weighted connections. **Figure (1)** shows the clearly relationship given by Eqn. (1) and Eqn. (2). The error back-propagation algorithm is one of the most important and widely used learning techniques for neural networks. The learning rule is known as back-propagation, which is a kind of gradient descent technique with back error (gradient) propagation. The object here is to "train" the network to find a way of altering the weights and thresholds so that the error is to be reached to the minimum. Compare the final output signals with a target signals, total squared error, E_p is produced which is the sum of squared difference between the desired output t_p and actual output u_{ip},

$$E_p=1/2\sum_i (t_p - u_{ip})^2 \tag{4}$$

Where: t_p is a target signal of the unit u_i at the output layer, and
u_{ip} is an actual output signal of the unit u_i at the output layer.

The weights adjustment could be done by minimizing E_p in a gradient descent start at the output unit and the weight change (Δw_{ij}) work backward to the hidden layers recursively.

The weights are adjusted by:

$$w_{ij}(t+1)=w_{ij}(t)+\Delta w_{ij} \tag{5}$$

Where: $w_{ij}(t)$ is the weight from unit j to unit i at time t, and Δw_{ij} is the weight adjustment. The new weight $w_{ij}(t+1)$ is straightforward to the next layer repeatedly.

3. FUZZY NEURAL NETWORK

Fuzzy neural network (FNN) considered as a special type of neural network [14], every layer and every node have its practical meaning because the FNN has the structure which is based on both fuzzy rules and inference, **Figure (2)** show the structure of FNN. In the following items each layer will be described:

1. Input Layer: Transmits the input linguistic variables x_n to the output without changed.

2. Hidden Layer I: Membership layer represents the input values with the following Gaussian membership functions [14]:

$$\mu_j^i=\exp(-(x_j-c_{ij})^2/2s^2_{ij}) \tag{6}$$

Where c_{ij} and s_{ij} (i=1, 2, …., n; j=1, 2,…, m), respectively, are the mean and standard deviation of the Gaussian function in the j^{th} term of the i^{th} input linguistic variable x_n to the node of this layer.

3. Hidden Layer II: Rule layer implements the fuzzy inference mechanism, and each node in this layer multiplies the input signals and outputs the result of the product.

The output of this layer is given as [14]:

$$\varnothing_i=\Pi_j^n \mu_j^i \tag{7}$$

Where Ø_i represent the i^{th} output of rule layer.

4. Output Layer: The nodes in this layer represent output linguistic variables. Each node $Y_o(O=1, \ldots, N_o)$, which computes the output as [14]:

$$Y_o=\Sigma_i^m w_i^o \text{Ø}_i \tag{8}$$

The main goal of learning algorithm is to minimize the mean square error function [14]:

$$E=1/2(Y_o-Y_p)^2 \tag{9}$$

Where Y_o is the actual output and Y_p is the desired output.

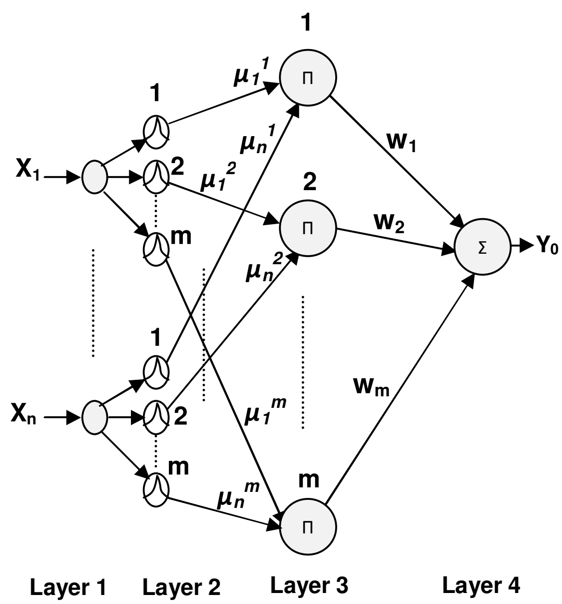

FIGURE 2: The Structure of FNN

The gradient descent algorithm gives the following iterative equations for the parameter values [14]:

$$w_i(k+1)=w_i(k)-\eta_w\partial E/\partial w_i \tag{10}$$

$$c_{ij}(k+1)= c_{ij}(k)-\eta_c\partial E/\partial c_{ij} \tag{11}$$

$$s_{ij}(k+1)= s_{ij}(k)-\eta_s\partial E/\partial s_{ij} \tag{12}$$

Where η is the learning rate for each parameter in the system, i=1, 2, …., n and j=1, 2,…, m.

Taking the partial derivative of the error function given by Eqn. (9), gets the following equations:

$$\partial E/\partial w_i=(Y_o-Y_p)\ \text{Ø}_i \tag{13}$$

$$\partial E/\partial c_{ij}=(Y_o-Y_p)\ \text{Ø}_i\ w_i(x_j-c_{ij})/s^2_{ij} \tag{14}$$

$$\partial E/\partial s_{ij}=(Y_o-Y_p)\ \varnothing_i\ w_i(x_j-c_{ij})^2/s^3_{ij} \tag{15}$$

Hence, the new value of w_i, c_{ij} & s_{ij} after adaptation is equal to:

$$w_i(k+1)=w_i(k)-\eta_w(Y_o-Y_p)\varnothing_i \tag{16}$$

$$c_{ij}(k+1)= c_{ij}(k)-\eta_c(Y_o-Y_p)\ \varnothing_i\ w_i(x_j-c_{ij})/s^2_{ij} \tag{17}$$

$$s_{ij}(k+1)= s_{ij}(k)-\eta_s(Y_o-Y_p)\ \varnothing_i\ w_i(x_j-c_{ij})^2/s^3_{ij} \tag{18}$$

4. FUZZY NEURAL PETRI NET

The structure of the proposed Fuzzy Neural Petri Net is shown in **Figure (3)**. The network has the following three layers [15]:

1- An input layer composed of n input places.
2- A transition layer composed of hidden transitions.
3- An output layer consisting of m output places.

The input place is marked by the value of the feature. The transitions act as processing units. The firing depends on the parameters of transitions, which are the thresholds, and the parameters of the arcs (connections), which are the weights. The marking of the output place reflects a level of membership of the pattern in the corresponding class.

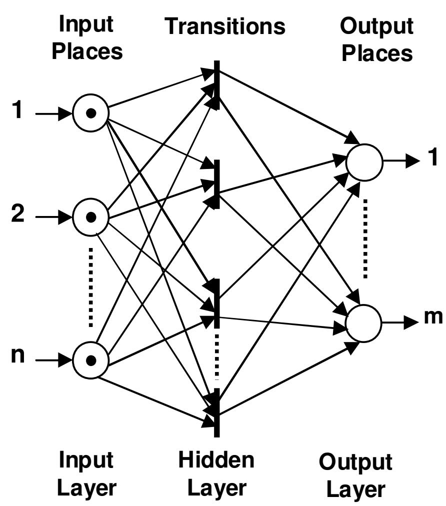

FIGURE 3: The Structure of FNPN

The specifications of the network for a section of the network is shown in **Figure (4)** are as follows [15]: P_j is the marking level of j-th input place produced by a triangular mapping function. The top of the triangular function is centered on the average point of the input values. The length of triangular base is calculated from the difference between the minimum and maximum values of the input.

The height of the triangle is unity. This process keep the input of the network within the period [0, 1]. This generalization of the Petri net will be in full agreement with the two-valued generic version of the Petri net.

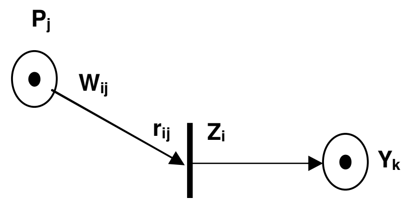

FIGURE 4: A section of the net outlines the notations

$$P_j=f(\text{Input}(j)) \tag{19}$$

Where f is a triangular mapping function shown in **Figure (5).**

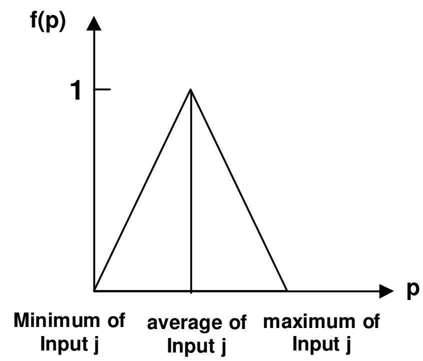

FIGURE 5: The triangular mapping function

W_{ij} is the weight between the i-th transition and the j-th input place;
r_{ij} is a threshold level associated with the level of marking of the j-th input place and the i-th transition;
Z_i is the activation level of i-th transition and defined as follows [15]:

$$Z_i=\mathop{\mathrm{T}}_{j=1}^{n}[W_{ij}S(r_{ij}\rightarrow P_j)],\ \text{j=1, 2,…., n; i=1, 2,…., hidden} \tag{20}$$

Where, "T" is a t-norm, "S" denotes an s-norm, while $\rightarrow$ stands for an implication operation expressed in the form:

$$a\rightarrow b=\sup\{c\in[0,1], aTc\leq b\} \tag{21}$$

Where a, b are the arguments of the implication operator confined to the unit interval.

In the case of two-valued logic, Eqn. (21) returns the same truth value as:

$$a\rightarrow b=\begin{cases} b, \ if \quad a\rangle b \\ 1, \ otherwise \end{cases}=\begin{cases} 0, \ if \quad a=1 \quad and \quad b=0 \\ 1, \ otherwise \end{cases} \qquad a, b\in[0, 1]$$

If T-norm is defined as a multiplication operator (Π) then

$$r_{ij} \rightarrow P_j = \begin{cases} \dfrac{p_j}{r_{ij}}, & if \quad r_{ij} \rangle p_j \\ 1, & otherwise \end{cases}$$

$$Z_i = \prod_{j=1}^{n} W_{ij} \vee \begin{cases} \dfrac{p_j}{r_{ij}}, & if \quad r_{ij} \rangle p_j \\ 1, & otherwise \end{cases}$$

Y_k is the marking level of the k-th output place produced by the transition layer and performs a nonlinear mapping of the weighted sum of the activation levels of these transitions (Z_i) and the associated connections V_{ki}, such as given:

$$Y_k = f\left(\sum_{i=1}^{No.of\ Transition} V_{ki} Z_i \right), j=1, 2, \ldots, n \tag{22}$$

Where "f" is a nonlinear monotonically increasing function from [0, 1].

The learning process depends on minimizing certain performance index in order to optimize the network parameters (weights and thresholds). The performance index used is the standard sum of squared errors. The errors are the difference between the marking levels of the output places and the target values. The training set (p, t), which is the marking levels of the input places (denoted by p) and the required marking of the output places (target "t"), are presented to the network in order to optimize the parameters. The performance index is as follows:

$$E = \frac{1}{2} \sum_{k=1}^{m} (t_k - Y_k)^2 \tag{23}$$

Where:

t_k is the k-th target;
Y_k is the k-th output.

The updates of the parameters are performed according to the gradient method:

$$param(iter+1) = param(iter) - \alpha \nabla_{param} E \tag{24}$$

Where $\nabla_{param} E$ is a gradient of the performance index E with respect to the network parameters, α is the learning rate coefficient, and iter is the iteration counter.

The nonlinear function associated with the output place is a standard sigmoid described as:

$$Y_k = \frac{1}{1 + \exp(-\sum Z_i V_{ki})} \tag{25}$$

The flow chart of algorithms learning is shown in **Figure (6)**.

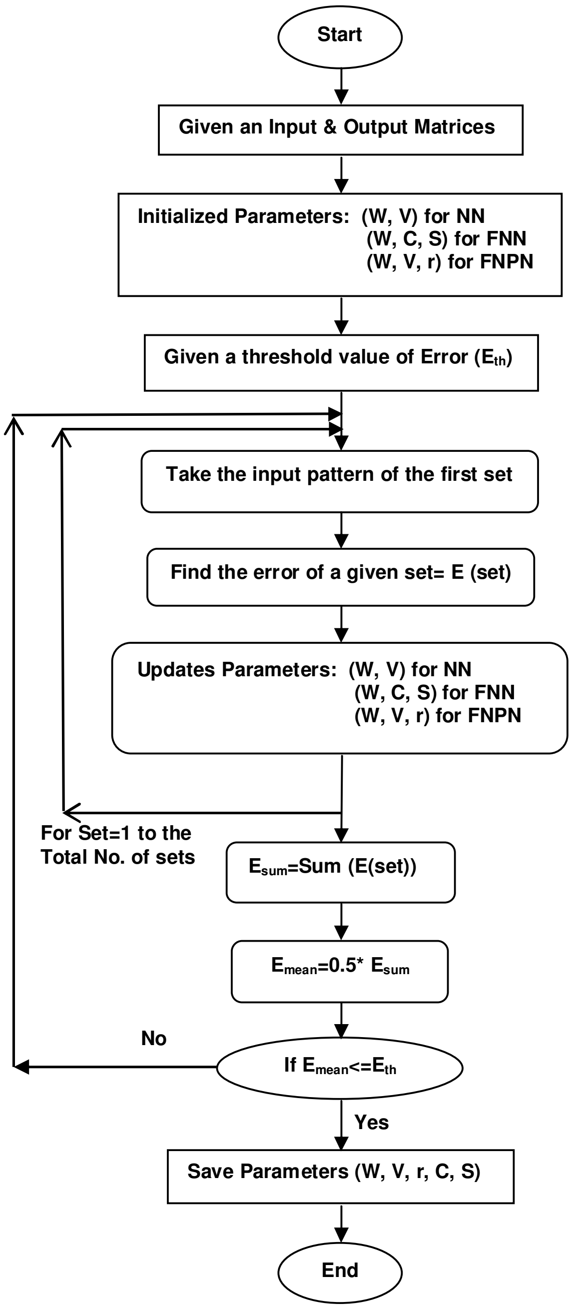

FIGURE 6: The Learning Mechanization of the proposed Algorithms

5. PROPOSED RELAYING APPROACH OF GENERATOR

Power generator is the major component in power system and any fault will effect on the availability of the power. Differential protection is the most common system employed for the protection of stator windings against earth faults and phase to phase faults makes use of circulating current principle. The sensitivity of such protection for earth fault depends upon the resistance in neutral to earth connection. The resistance of the neutral may not cause the relay to operate, the magnitude of the unprotected zone depend upon the value of resistance employ in neutral earth and the relay setting. The value of this protected zone about (80%-85%) of the winding, which mean (20%-15%) of the winding near neutral point can not cause tripping [16]. This approach produce a protective relay to solve this problem and to protect power generator, where the input variables of the proposed relay are differential current [16] and third harmonic method [17]. The differential current at two ends of protected generator are compared, under normal operation conditions these currents are equal, but may be differ when a fault occurs in the protected section.

The difference of the current under fault condition is made to flow through the relay operation coil. Differential protection protects the stator windings against earth faults and phase to phase faults [16, 18].Differential third harmonic current provide stator ground fault protection. The compared third harmonic component in the neutral of machine and in the terminal give a pilot for normal and earth fault operation condition. Power supervision is incorporated allowing sensitive settings on machine that have their third harmonic content varying significantly as exported power changes. The third harmonic component differs as the location of earth fault change.

5.1 Generator Simulation Model

Power generator with (22 kV, 23 MVA) is simulated by using MATLAB simulation software as shown in **Figure (7)**. The scenario of training and test the proposed approaches are generated during nominal power system operating conditions. Full load, 0.5 full load and 0.25 full load cases are taken to cover wide range of fault events. Fault type, fault location (i.e. internal and external protected zone), fault resistance (0, 5Ω) and fault inception time were changed to obtain training patterns covering a wide range of different power generator faults conditions.

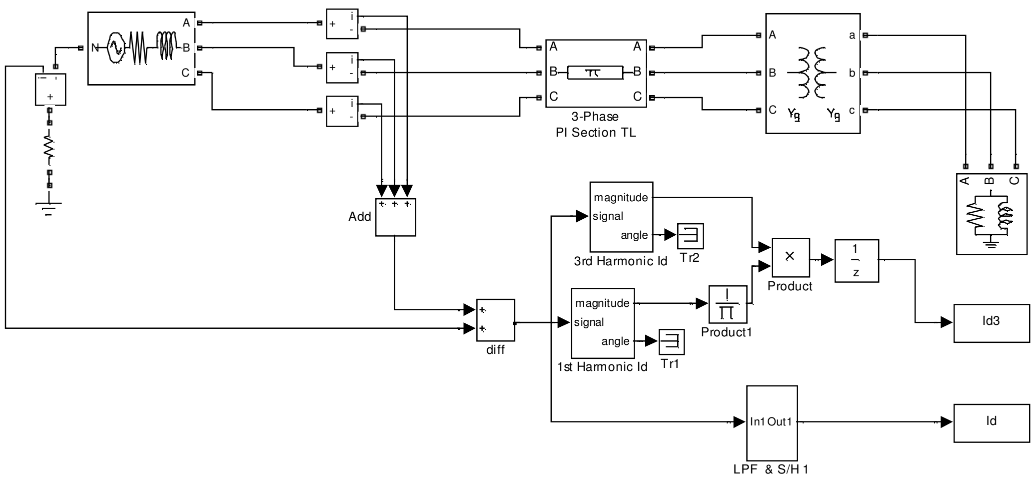

FIGURE 7: Simulated Model of Generator

5.2 Patterns Generation and Preprocessing

The differential current and differential third harmonic current were processed by simple 2nd-oredr low-pass filters. The filters had a cut-off frequency of 400 Hz which introduces just a small time delay. These two inputs are sampled at a rate of 20 samples per cycle and normalized to have a maximum value of +1 and a minimum value -1 as shown in **Figure (8)**. Multilayer feedforward network were chosen to process the prepared input data. The input layer contains 5 sample for each input, since there are two inputs in the model, therefore, the input layer required 10 input nodes. Various networks with different number of nodes in their hidden layer were studied. The output layer consists of only one node, which has value 1 if fault occurs to indicate tripping or 0 for no fault.

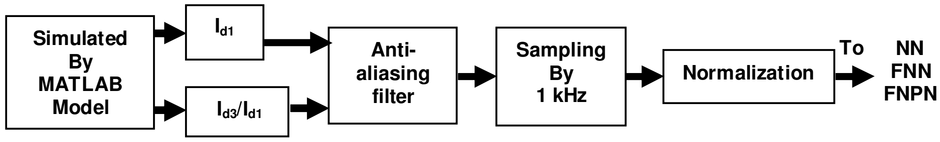

FIGURE 8: Training Pattern Generation Process

6. TEST RESULTS

The data of training and testing the proposed approaches are generated during nominal power system operating conditions by using MATLAB simulation software. Full load, 0.5 full load and 0.25 full load cases are taken to cover diversity of fault event. The types of faults that simulated are includes:

-Normal operation
-Normal with unbalance load
-Normal with non-linear load
-External L-G
-External L-L-G
-External L-L
-External symmetrical fault
-Internal L-G
-Internal L-L-G
-Internal L-L
-Internal symmetrical fault
-Inter-turn fault at 20% of winding

6.1 Full load Case

Table (1) explaining the magnitude and time delay of output for proposed approaches, NN relay, FNN relay, FNPN relay and Conventional differential relay for full load case. **Table (1)** shows that the proposed approaches have no output trip for (Normal operation and external fault cases). NN-relay has output trip for internal faults cases with time delay (2-4) ms, FNN-relay has output trip for internal faults cases with time delay (2-4) ms and FNPN-relay has output trip for internal faults cases with time delay (2-3) ms. Conventional differential relay has no output trip for normal operation and external fault cases, while for nternal fault cases has output trip with time delay (2-9) ms.
Example of this case is Internal L-G fault shown in **Figure (9)**, an Internal L-G fault (phase A-G) occurs at t=6ms, as seen from **Figure (9)** all relays detect the fault at t=6ms, NN-relay output trip after 2ms but FNN-relay and FNPN-relay are both output trip after 1ms of fault occur, hence FNN-relay and FNPN-relay are gives output trip faster than NN-relay.

TABLE 1: Results of Simulation for Full Load Case.

No	Case	NN_{Relay}		FNN_{Relay}		$FNPN_{Relay}$		Conventional Diff. Relay	
		R_{Out}	T (ms)	R_{Out}	T (ms)	R_{Out}	T (ms)	R_{Out}	T (ms)
1	Normal Operation	0.0003	-	0	-	0.0058	-	No Trip	-
2	Normal with unbalance load	0.0011	-	0	-	0.005	-	-	-
3	Normal with Non-linear load	0	-	0.0014	-	0.0001	-	-	-
4	External L-G	0.0017	-	0.0005	-	0	-	No Trip	-
5	External L-L-G	0	-	0.0067	-	0	-	No Trip	-
6	External L-L	0	-	0.0033	-	0.001	-	No Trip	-
7	External symmetrical fault	0.0087	-	0.01	-	0	-	No Trip	-
8	Internal L-G	1	3	0.9998	2	0.9997	2	Trip	5
9	Internal L-L-G	1	4	1	4	1	3	Trip	2
10	Internal L-L	1	3	1	4	1	3	Trip	3
11	Internal symmetrical fault	1	2	1	2	1	2	Trip	2
12	Inter turn at 20% of winding	1	4	0.9991	4	0.9994	3	Trip	9

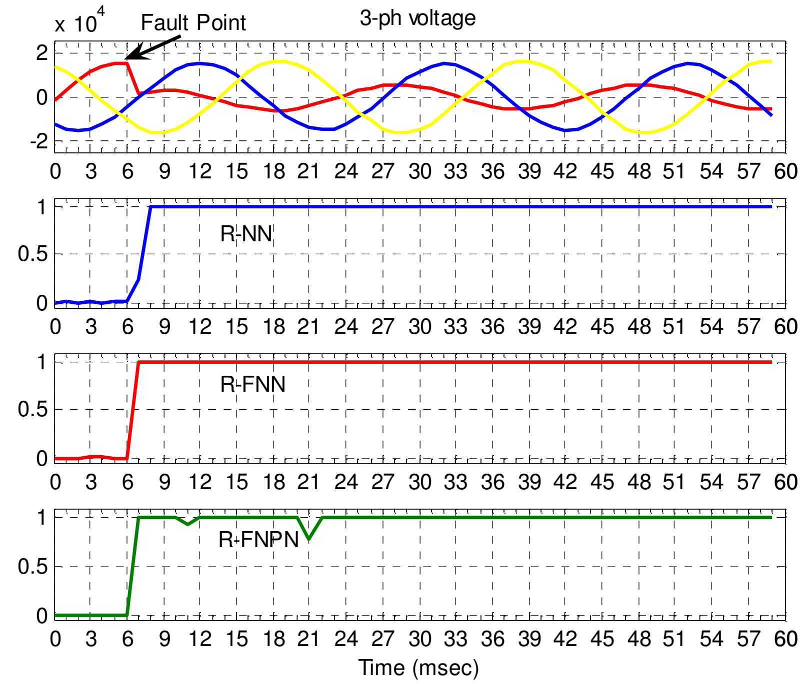

FIGURE 9: Relay Output for Internal L-G Fault for Full Load Case

6.2 Half Full load Case

Table (2) explaining the magnitude and time delay of output for proposed approaches, NN relay, FNN relay, FNPN relay and Conventional differential relay for half full load case. **Table (2)** shows that the proposed approaches have no output trip for (Normal operation, and external fault cases). NN-relay has output trip for internal faults cases with time delay (2-5) ms, FNN-relay has output trip for internal faults cases with time delay (2-4) ms and FNPN-relay has output trip for internal faults cases with time delay (2-3) ms. Conventional differential relay has no output trip for normal operation and external fault cases, while for internal fault cases has output trip with time delay (3-11) ms.

Example of this case is Internal L-L-G fault shown in **Figure (10)**, an Internal L-L-G fault (2-phase A-B-G) occurs at t=3ms, as seen from **Figure (10)**, NN-relay detect the fault after 2ms (i.e. at t=5ms) and output trip after 3ms (i.e. at=6ms), while FNN-relay and FNPN-relay are both detect the fault after 1ms (i.e. at t=4ms) and output trip after 2ms of fault occur (i.e. at t=5ms), hence FNN-relay and FNPN-relay are gives output trip faster than NN-relay.

No	Case	NN_{Relay}		FNN_{Relay}		$FNPN_{Relay}$		Conventional Diff. Relay	
		R_{Out}	T (ms)	R_{Out}	T (ms)	R_{Out}	T (ms)	R_{Out}	T (ms)
1	Normal Operation	0.0006	-	0.02	-	0.0036	-	No Trip	-
2	Normal with unbalance load	0.001	-	0	-	0.0038	-	-	-
3	Normal with Non-linear load	0.008	-	0	-	0.0016	-	-	-
4	External L-G	0.0036	-	0.0003	-	0	-	No Trip	-
5	External L-L-G	0	-	0.0001	-	0	-	No Trip	-
6	External L-L	0	-	0.0034	-	0	-	No Trip	-
7	External symmetrical fault	0	-	0.019	-	0.002	-	No Trip	-
8	Internal L-G	0.9994	2	0.9993	2	0.9981	2	Trip	5
9	Internal L-L-G	1	5	1	4	1	3	Trip	3
10	Internal L-L	1	3	0.9912	3	0.9995	3	Trip	3
11	Internal symmetrical fault	0.9999	3	0.9882	2	1	2	Trip	3
12	Inter turn at 20% of winding	1	4	0.9999	4	0.9991	3	Trip	11

TABLE 2: Results of Simulation for Half Full Load Case

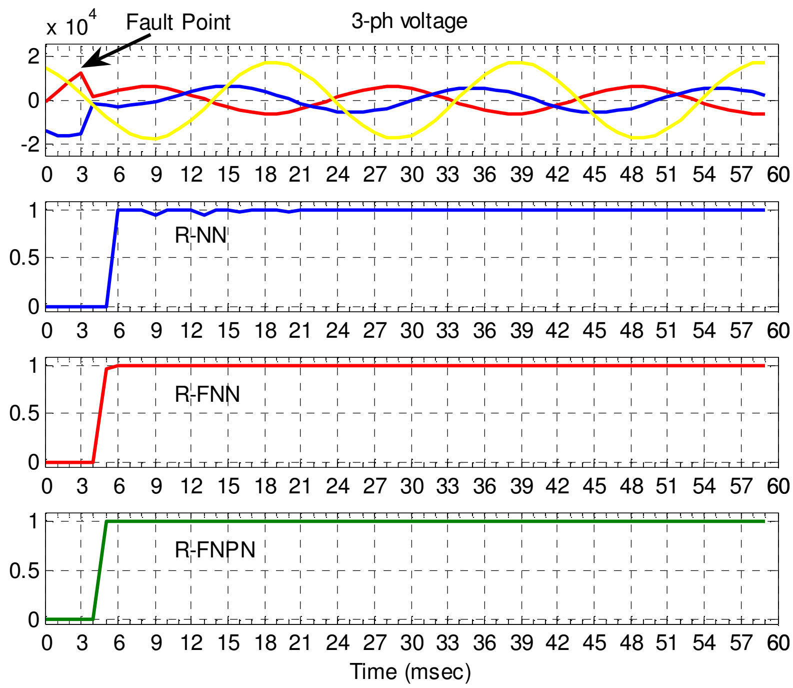

FIGURE 10: Relay Output for Internal L-L-G Fault for Half Full Load Case

6.3 Quarter Full load Case

Table (3) explaining the magnitude and time delay of output for proposed approaches, NN relay, FNN relay, FNPN relay and Conventional differential relay for quarter full load case. **Table (3)** shows that the proposed approaches have no output trip for (Normal operation, and external fault cases). NN-relay has output trip for internal faults cases with time delay (3-6) ms, FNN-relay has output trip for internal faults cases with time delay (3-5) ms and FNPN-relay has output trip for internal faults cases with time delay (2-5) ms. Conventional differential relay has no output trip for normal operation and external fault cases, while for internal fault cases has output trip with time delay (3-12) ms.

Example of this case is Internal L-L-L-G fault shown in **Figure (11)**, an Internal L-L-L-G fault (3-phases A-B-C-G) occurs at t=10ms, as seen from **Figure (11)**, NN-relay detect the fault after 1ms (i.e. at t=11ms) and output trip after 2ms (i.e. at t=12ms), while FNN-relay and FNPN-relay are both detect the fault at t=10ms) and output trip after 1ms of fault occur (i.e. at t=11ms), hence FNN-relay and FNPN-relay are gives output trip faster than NN-relay.

No	Case	NN_{Relay}		FNN_{Relay}		$FNPN_{Relay}$		Conventional Diff. Relay	
		R_{Out}	T (ms)	R_{Out}	T (ms)	R_{Out}	T (ms)	R_{Out}	T (ms)
1	Normal Operation	0.0011	-	0	-	0.0093	-	No Trip	-
2	Normal with unbalance load	0.0001	-	0	-	0.0068	-	-	-
3	Normal with Non-linear load	0	-	0	-	0	-	-	-
4	External L-G	0.086	-	0.0233	-	0.001	-	No Trip	-
5	External L-L-G	0	-	0.0021	-	0	-	No Trip	-
6	External L-L	0	-	0.0032	-	0	-	No Trip	-
7	External symmetrical fault	0	-	0.0196	-	0	-	No Trip	-
8	Internal L-G	1	6	0.9999	5	0.9376	5	Trip	7
9	Internal L-L-G	0.9908	6	1	5	1	4	Trip	3
10	Internal L-L	1	3	1	4	1	2	Trip	3
11	Internal symmetrical fault	1	3	1	3	1	4	Trip	3
12	Inter turn at 20% of winding	1	4	0.9978	5	1	4	Trip	12

TABLE 3: Results of Simulation for Quarter Full Load Case

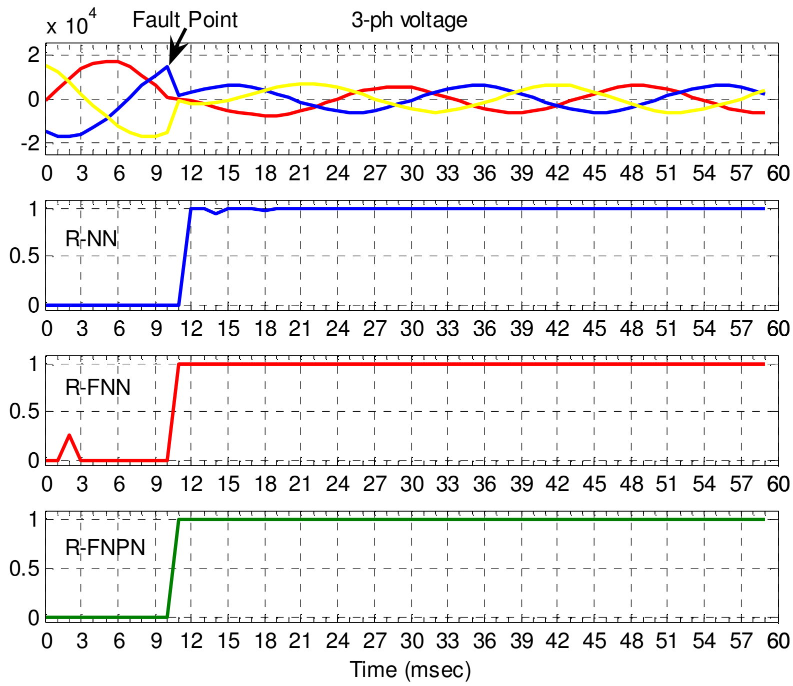

FIGURE 11: Relay Output for Internal L-L-L-G Fault for Quarter Full Load Case

7. CONCLUSION

The relaying approaches are proposed to protect power generator, by using basic principle as differential current and third harmonic differential current. These methods are used as an input to the NN-relay, FNN-relay and FNPN-relay, which have good solution for uncertainty cases. Comparison of these proposed approaches with conventional differential protection, which have the properties of good protection relays from speed of operation, sensitivity and reliability, where which have less than half cycle as an average to operate, these approaches have good sensitive to generator inter-turn fault at each point in the stator winding, with high reliability to distinguish between fault cases and non fault cases. The obtained results show that the proposed approaches represent a proper action and good performance. The test results also explained that FNN-relay and FNPN-relay are faster in operation than NN-relay in some fault cases.

8. REFERENCES

[1] Protective Relays Applications Guide. The English Electric Company Limited, Relay Division, Stafford, 1975.

[2] C. J. Mozina. "IEEE Tutorial on the Protection of Synchronous Generators." IEEE Tutorial Course, IEEE Power Engineering Society Special Publ., No. 95 TP102, 1995.

[3] A. I. Megahed, and o. p. Malik. "An Artificial neural network based digital differential scheme for synchronous generator stator winding protection." IEEE Transactions on Power Delivery, vol. 14, Jan. 1999.

[4] M. S. Sachdev (Coordinator)." Microprocessor Relays and Protections Systems." IEEE Totorial Course Text, Power Engineering Society Special Publ. No. 88 EH0269-1-PWR, IEEE, Piscatway, NJ, USA, 1988.

[5] G. Benmouyal, S. Barceloux, and R. Pelletier. "Field experience with a digital relay for synchronous generators." IEEE Transactions on Power Delivery, vol. 7, No. 4, October 1992.

[6] T. Dalstein and B. Kulicke. "Neural Network Approach to Fault Classification for High Speed Protective Relaying." IEEE Trans. On Power Delivery, vol. 10, No. 2, pp. 1002-1011, Apr. 1995.

[7] M. Kezonuic. "A Survey of Neural Net Application to Protective Relaying and Fault Analysis." Eng. Int. Sys. Vol. 5, No. 4, pp. 185-192, Dec. 1997.

[8] Z. Zhi-hua, and Ch. Shi-fu. "Neural Network Ensemble [J]." Chinese J. Computers, 25 (1), pp.1-8, 2002.

[9] T. Murata. "Petri Net: Properties, analysis, applications." in Proc. IEEE, 77, (4), pp. 541-580, 1989.

[10] D. E. Rumelhart, G. E. Hinton and R. J. Willians. "Learning Internal Representations by Error Propagation." in parallel Distributed Processing: Explorations in the Microstructure of Cognition (eds. Rumelhart and Mcclelland), vol. I, MIT Press, pp. 318-362, 1986.

[11] D. J. Sobajic and Y. H. Pao. "Artificial Neural Net Based Dynamic Security Assessment for Electric Power Systems."IEEE Trans. On Power Systems, Vol. 4, No. 1, pp. 220-224, Feb. 1989.

[12] A. F. Sultan, G. W. Swif and D. J. Fedirchuk. "Detection of High Impedance Arcing Faults Using A Multi-Layer."IEEE Trans., Power Delivery, vol. 7, No. 4, October 1992.

[13] Z. Moravej and D. N. Vishwakarma."ANN-based Harmonic Restraint Differential Protection of Power Transformer." vol. 84, June 2003.

[14] R. J. Wai, Ch. Ch. Chu."Robust Petri Fuzzy Neural Network Control for Linear Induction Motor Drive."IEEE trans., vol. 54, pp. 177-189, 2007.

[15] W. Pedrycz and F. Gomide."A Generalized Fuzzy Petri Net Model." IEEE Trans. On Fuzzy Systems, vol. 2, No. 4, November 1994.

[16] J. B. Gupta. A Course in Power System. 2003.

[17] D. Finney. "Dynamic Testing of Generator Protection Using a Model Generator Platform."GE Power Management 215 Anderson Avenue, Markham, Ontario, Canada L6E 1B3, Atlanta, GA, May 1-3, 2002.

[18] C. R. Mason. The Art & Science of Protective Relaying. 2002.