ISSN 0976 – 1411

Available online at www.internationaleJournals.com

International eJournals

International eJournal of Mathematics and Engineering 72 (2010) 715 - 727

INTERNATIONAL eJOURNAL OF MATHEMATICS AND ENGINEERING

www.InternationaleJournals.com

CONTINGENCY EVALUATION OF ELECTRICAL POWER SYSTEM USING ARTIFICIAL NEURAL NETWORK

Shekhappa G. Ankaliki[1], A. D. Kulkarni[2], T. Ananthapadmanabha[3]

1 Department of Electrical & Electronics Engineering, Hirasugar Institute of Technology Nidasoshi-591 236, Belgaum (Dist), Karnataka. (State) Email:sankaliki@rediffmail.com
2 Department of Electrical & Electronics Engineering, National Institute of Engineering, Mysore, Karnataka (State), India
3 Department of Electrical & Electronics Engineering, National Institute of Engineering, Mysore, Karnataka (State), India

---

Abstract

This paper presents application of Artificial Neural Network (ANN) based contingency analysis of power system. The ANN has been chosen because of its high adaptation parallel information processing capability. Another feature that makes the ANN more suitable for this type of problems is its ability to augment new training data without the need for retraining. In this Multilayer Feed Forward network is used for contingency analysis in planning studies where the goal is to evaluate the ability of a power system to support a projected range of peak demand under all foreseeable contingencies. This work involves selection of network design, preparation of input patterns, training & testing. In order to generate the training patterns three system topologies were considered. Training data are obtained by load flow studies (NR Method) for different system topologies over a range of load levels **using software simulation package (Mipower)** and the results are compiled to form the training set. For training the ANN back propagation algorithm is used. The proposed algorithm is applied to a sample six bus power system and the numerical results are presented to demonstrate the effectiveness of this proposed algorithm in terms of accuracy. It is concluded that the trained ANN can be utilized for both off-line simulation studies and on line estimation of line flows and voltages.

Key words: Contingency Evaluation, Load flow study, Artificial Neural Network.

1.0 Introduction

Contingency evaluation is one of the most important tasks encountered by planning and operation engineers of bulk power systems. In planning, contingency analysis is used to examine the performance of a power system and the need for new transmission expansion due to load growth or generation expansion. In operation, contingency analysis assists engineers to operate the power system at a secure operating point where equipment are loaded within their safe limits and power is delivered to customers with acceptable quality standards. In general, the state of a system is determined based on its ability to meet the expected demand under different contingency levels. In this type of analysis the objective is to find over loads or voltage violations under such contingency and the proper measures that are needed to alleviate these violations. Finding these contingencies and determining the corrective actions often involves exhaustive load flow calculations. The necessity for such a tool is increasingly critical due to the emerging complexity of power systems that results from network expansions and the fact that power systems are pushed to operate at their limits due to financial and environmental constraints.

2.0 Review of Basic Methods

There exist many methods for contingency evaluation of bulk power systems [1-3]. AC Load flow method, PSC (Power Supply Capacity) Calculations, DC Load flow method and Sensitivity Analysis & Distribution factor. AC Load flow and Power Supply Capacity (PSC) calculations [1] are most accurate methods. The basic concept of the PSC method is to determine the maximum amount of power each bus can deliver subject to generation capacity, power flow constraints and equipment rating. These two approaches involve a huge number of AC load flow calculations to determine line flow and bus voltage for each contingency. This computation poses a challenging task even for today's fast computers and efficient algorithms. The analysis and interpretation of these calculations present an even harder problem. Another deficiency is that contingency analysis uses fast converging load flow algorithms, such as Fast Decoupled Newton Raphson (FDNR) algorithm that has poor convergence characteristics when dealing with heavily loaded power system. There are many other techniques that simplify contingency analysis. DC load flow is one of the most popular methods that are used to reduce the computational efforts required by the AC power flow to an acceptable level [2]. However, it can only provide a good estimate of the MW flow under each contingency. Therefore voltage violation and line overload due to excessive VAR flow can't be detected using this method. Another technique uses sensitivity analysis and distribution factor [3] but it is not guaranteed to provide accurate line flow solution since it is based on a linear model to approximate the solution especially in highly loaded power system where the non-linearity is a significant factor.

Recently, artificial neural networks (ANNs) have been utilized for contingency screening [4 - 9]. However, most of these applications use ANNs as a tool to classify the system states under contingency to secure or insecure states. This approach is used mainly for real time operation at power control centers where the objective is to provide the operator with an indication about the state of the power system. Clearly, this formulation is not sufficient for planning purposes where there is a need for more elaborate studies to compare alternative expansion plans based on quantitative economic and reliability factors. In [4], a linear ANN structure was used with non-linear feedback loop as a tool to solve power flow problems. It estimates bus voltage magnitude and angle in a manner similar to standard power flow algorithms. The linear ANN structure is used to estimate required adjustments in bus voltage magnitude and angle based on power mismatch at each bus. This estimated voltage adjustment is then used to calculate the line flow using power flow equations. Calculated flows combined with net bus injections are fed back to the ANN and the process keeps repeating until reasonable error is reached. However, the use of linear model limits the mapping capability of the neural networks.

The Hopfield model was used to classify the contingency by learning to recognize the number and type of limit violations associated with each contingency [5]. It uses a linear programming technique to optimize the ANN classification accuracy. The violation pattern that results from each contingency is constructed using a binary matrix in which violations are assigned a binary code.

The application of Kohonen's Self-organizing feature map provides a fast contingency assessment tool for real time operations [6]. The operating point of the system is presented to the ANN as a vector of line active and reactive power flows obtained from running load flow under different conditions. The state of the system can be determined by estimating how far the operating point is away from the safe operating boundaries of the system, Kohonen's self organizing map was also used to identify similarities in system state variables (line flows and bus voltages) under different contingency [7]. The network is trained to produce a feature map that relates each contingency and pre-contingency state parameter to post contingency attributes. A modified version of this method is presented in [8], where a supervised ANN is used to provide rough estimates of post contingency line flows and bus voltages and an unsupervised ANN to that uses the outputs of the supervised ANN to classify contingency. Contingencies are classified in to different groups based on their impact on the system. A separate supervised ANN is used for each group of contingencies to provide a more accurate estimate of post contingency voltage and line flow patterns. One difficulty with this method is that it requires a large number of supervised ANNs. Neural Networks were also used for security assessment of large–scale power system [9]. The system is split up into small subsystems and each one is handled separately using different ANN. The basic principle of this approach is similar to those presented in [5-8], that is to apply the pattern recognition capability of ANNs to classify the system. The boundary buses are selected based on network sensitivity analysis. With the exception to [4], contingency evaluation is used to classify the system to either secure or insecure states, which is more useful in real time operation. Another important observation on these approaches is that ANNs are not trained on the relation between the system parameters that affect the power flow and bus voltage, such as bus load, generation distribution and system impedance. They employ ANNs for the mapping of a pre-contingency voltage and power flow patterns to a post contingency voltage and power flow patterns. This mapping is more suitable for on-line contingency analysis where the real objective is to provide the operator with a list of critical contingencies.

3.0 Proposed Approach

The proposed approach is the application of Artificial Neural Network for contingency Evaluation of Electrical Power System. ANN is more ideal for this type of problems is its ability to augment new training data without the need for retraining. Here feed forward network is used for contingency analysis in planning studies where the goal is to evaluate the ability of a power system to support a projected range of peak demand under all foreseeable contingencies. If a transmission expansion is necessary, then it must yield the maximum improvement to the system. For large-scale power system contingency evaluation, extensive studies need to be carried out considering the following factors;

- Number and type of possible contingencies and their combinations.
- Expected range of peak loads with a margin for forecasting error and
- Different generation scenarios based on efficiency and availability of generating units.

To optimize a transmission expansion plan, combinations of these factors need to be considered to cover all possible operating conditions. This process may produce a huge number of cases to be evaluated. A neural network needs to be trained on a limited set of cases that covers the operating boundary conditions for a given power system. To optimize the planned expansion, the trained (network) ANN is used for contingency evaluation under other operating conditions.

4.0 Artificial Neural Network

An Artificial Neural Network is a computing system made up of number of simple and highly interconnected processing elements which process information by its dynamic state response to external inputs. In recent times the study of ANN model has gain rapid and increasing importance because of the potential to offer solutions to some of the problems which have hitherto been intractable by standard serial computers in the areas of computer science and artificial intelligence. Instead of performing a program of instruction sequentially neural net models explore many computing hypothesis simultaneously using parallel net composed of many computational elements. No assumptions will be made because no relationships will be established. Computational elements in neural networks are non-linear models and are also faster. Hence the result comes through non-linearity due to which the result is more accurate than other methods. Because of these reasons neural networks find their applications in achieving human like performance in the fields such as speech processing, image reorganization, machine vision, robotic control etc.

4.1 Multilayer Feed Forward Neural Network

The Fig.1 shows the schematic representation of a Feed Forward Network, which is commonly used in ANN model. Processing elements in the ANN are called neurons. These neurons are interconnected by Information channels. Each neuron can have multiple inputs but only one output as shown in Fig. 2. Inputs to the neuron can be from external stimuli or from the output of other neurons. There is an interconnection strength called weight associated with each connection. When the weighted sum of the inputs to the neuron exceeds a certain threshold, the neuron is fired and output signal is produced. The neurons are divided into several layers; one input layer, one output layer and some hidden layers.

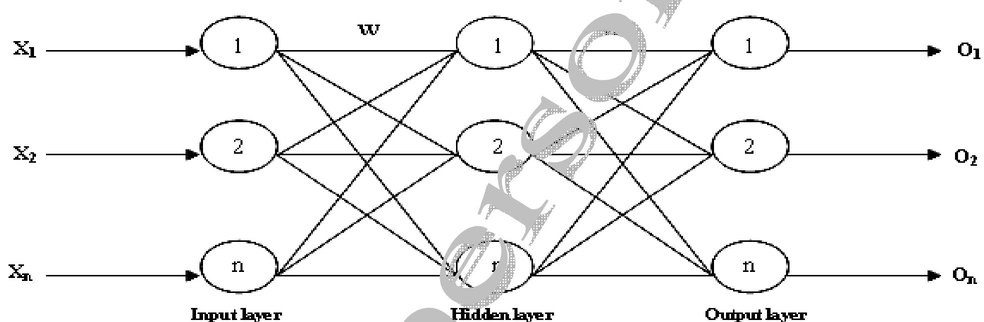


Fig.1. Multilayer Feed Forward Network

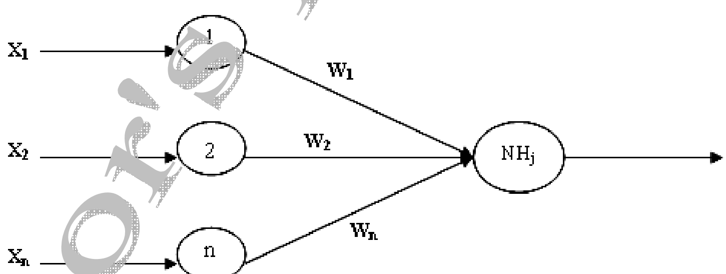


Fig. 2. Multiple Inputs and Single Output

These hidden layers are in between input and output layers. The neurons in the input layers takes the input signal and pass it to the hidden layers after giving some weightage to the signal. Only the neurons in the output and hidden layer perform activation function. The number of neurons in the input layer will be equal to the number of input signals. There is no hard and fast rule for selecting the number of neurons in the hidden layers. One assumption says that the number of neurons in the hidden layer should be equal to the square root of the product of input and output layer neurons. But the actual number of neurons in it depends upon the accuracy and fastness required. In this type of network the signals can only be propagated from input layer to hidden layer and from hidden layer to output layer, i.e. only in forward direction, hence the name Feed Forward Network.

4.2 Training the Network

The network can recognize input patterns only when the weights are adjusted or tuned via some kind of learning process called training. Collection of samples is divided into subsets. These subsets are presented to the network one at a time. If the outputs of these samples are known, then process is called supervised training. If the outputs are not known the process is called unsupervised training. One pass through this cycle is called epoch. The number of training samples in a subset of total samples is called epoch size. There are two methods of training the network:

i) Back propagation algorithm.

ii) Conjugate gradient algorithm.

The back propagation algorithm is the most frequently used method in training the network. This is also called generalized delta rule.

4.3 Generalized Delta Rule

An error signal proportional to the difference between what the output is (reference) and what is supposed to be (target) produced. Then the weights of the network are changed in proportion to the error times the input signal, which diminishes the error in the direction of gradient.

Let the sum of the squared errors to be minimized be

$$E_p = \frac{(t_{pm} - O_{pm})^2}{2} \quad (1)$$

Where p = presentation number.

t_{pm} = target output for y^{th} component of p^{th} pattern.

O_{pm} = actual output for y^{th} component of p^{th} pattern.

To obtain a rule for adjusting the weight the gradient of E_p with respect to the weight W_{ym} is used. Where W_{ym} is the weight between Y^{th} & m^{th} neuron. From the descent gradient algorithm, the change in weight is proportional to the gradient of error and it should be in such a direction that the error is decreasing.

Hence, $\Delta W_{ym} \; \alpha \; \frac{-\partial E_p}{\partial W_{ym}}$

$$\Delta W_{ym} \; \alpha \; - \frac{\partial E_p}{\partial W_{ym}} * \frac{\partial O_{pm}}{\partial O_{pm}}$$

Then, error signal is defined as

$$\delta_{pm} = \frac{-\partial E_{pm}}{\partial O_{pm}} \quad (2)$$

Hence equation (2) becomes

$$\Delta W_{ym} \; \alpha \; \delta_{pm} * \frac{\partial O_{pm}}{\partial W_{ym}} \quad (3)$$

This can be manipulated

$$\Delta W_{ym} = \eta * \delta_{pm} * O_{py} \quad (4)$$

Where, η = adaptation gain = learning rate parameter

The error signal is defined in two ways:

(i) If neuron 'm' is one of the output layer

$$\delta_{pm} = (t_{pm} - O_{pm}) * O_{pm} * (1 - O_{pm}). \quad (5)$$

(ii) If neuron Y is not from the output layer

$\delta_{py} = O_{py} * (1-O_{pm}) * \Sigma\delta_{pm} * W_{ym}$ (6)

4.4 Back Propagation Algorithm

Step-1: A subset of training samples is presented to the network .The output of the neurons is computed using following equations. For each neuron in the input layer, the neuron output is the same as the neuron input for any neuron `m' in the hidden or output layer, the neuron input is $Net_{pm} = W_{my} * O_{py}$ (7)

Where y=1, 2…n. the neuron in the preceding layer

O_{py} =output of yth neuron in the preceding layer.

The output of neuron `m' is,

$$O_{pm} = \frac{1}{1+ \text{Exp}\{-(net_{pm} - \theta_{pm})/ \theta_{om}\}} \quad (8)$$

Where θ_{pm} =threshold; θ_{om} = abruptness of the transition

Step-2: The sum of the squared errors is generated using equation (1).

Step-3: If the error is greater than the tolerance limit, the error signals are generated using equations (5) and (6) otherwise go to step 6.

Step-4: The change in weight is calculated using equation (4)

To improve the convergence characteristic, a momentum term `α' is introduced as follows:

$$\Delta W_{ym\,(n+1)} = \eta * \delta_{\mathbf{pm}} * O_{py} + \alpha[W_{ym\,(n)} - W_{ym\,(n-1)}] \quad (9)$$

Where n =iteration count.

α = momentum gain.

η = adaptation gain

Then the new value of weight is

$$W_{ym\,(n+1)} = W_{ym\,(n)} + \Delta W_{ym\,(n+1)} \quad (10)$$

Step-5: The iteration count is incremented and step1 to 4 are repeated

Step-6: Presentation number is incremented and other subsets of training samples are presented to the network. If all the subsets are over, the program is terminated.

4.5 Activation Function

The activation function is a non linear function that, when applied to the net input of the neuron, determines the output of that neuron. A majority of ANN models used a sigmoid function as activation function. It may be defined as a continuous, real valued function whose domain is real and its derivative is always positive. The most commonly used sigmoid function is the logistic function given in the Fig.3.

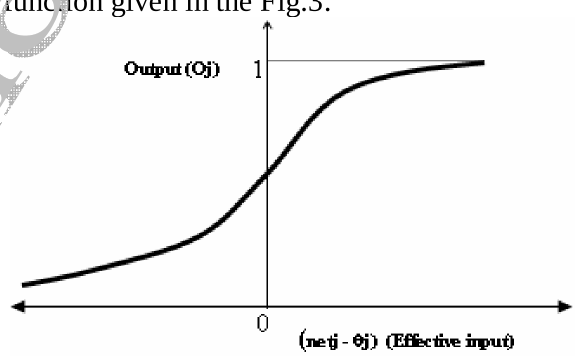


Fig. 3 Sigmoid Activation Function

It is defined by the equation

$$F(x) = \frac{1}{1 + \exp\{-(netj - \theta_j) / \theta_0\}} \quad (11)$$

This function yields an output that varies continuously from 0 to 1. The quantity θ_j serves as a "Threshold" and positions the transition region of the function. The quantity θ_0 determines the abruptness of the transition. The advantages of using this function as activation function are

i) Its derivative can be easily found.
$f^1(x) = f(x) * (1 - f(x))$

ii) Computer takes less time to evaluate this function.
Hence, the training speed will be higher.

4.6 Threshold

The threshold θ_j positions the transition region f the activation function. The effective input to the neuron will be ($net_j - \theta_j$). The values of these should also be learnt by the network. These are learnt by taking θ_j to be equivalent to another weight connecting the neuron 'j' to lower layer neuron, the output of which is always unity. The thresholds are also called bias for the neurons.

4.7 Inputs to the ANN

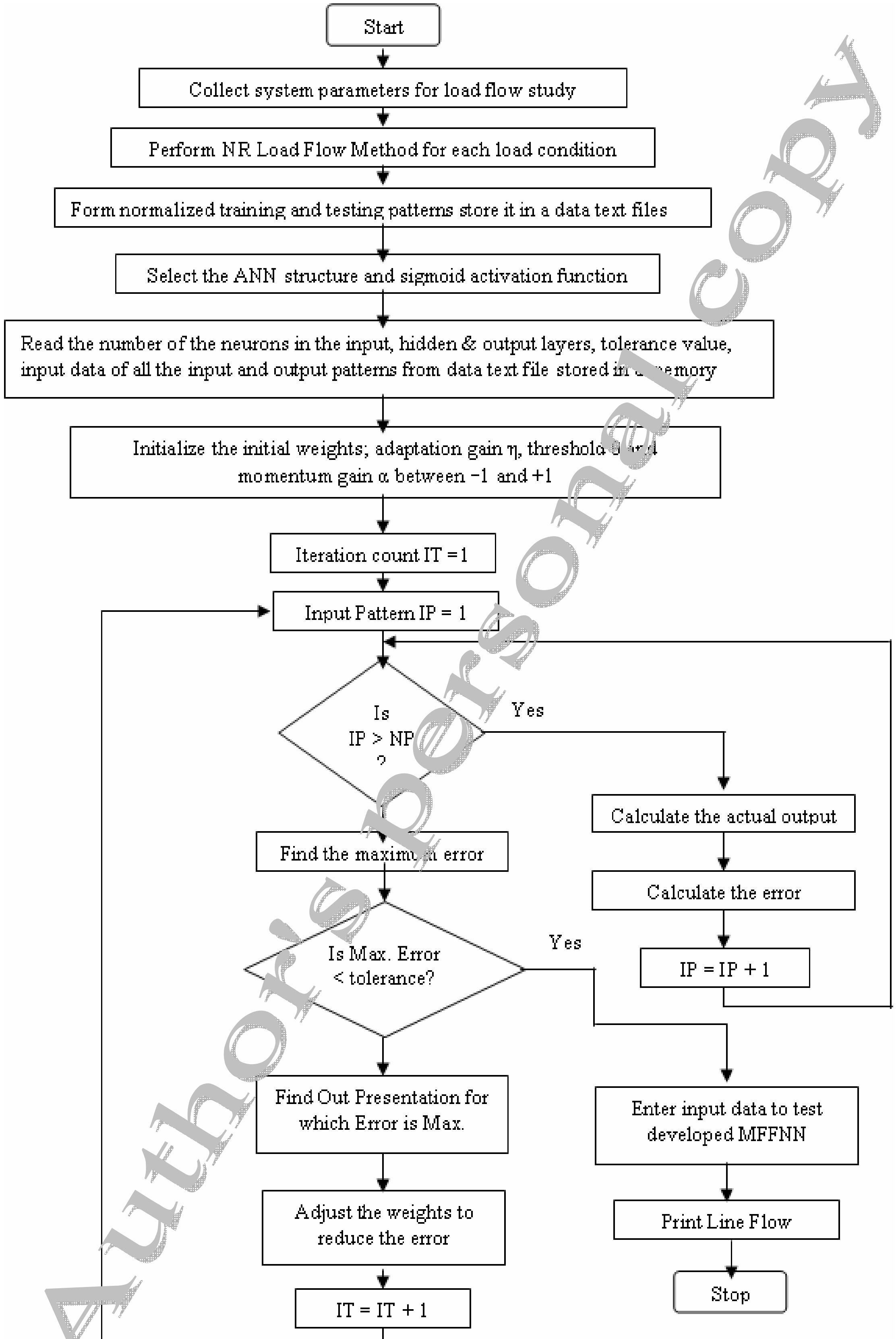
A large number offload patterns have been generated in a wide range of system operating conditions (60-110%) and AC load flow has been performed to obtain the line-flows for each case. The results are compiled to form the input patterns (i.e. the real and reactive power injections affecting a line-flow most) are selected.

4.8 Training

For each line, supervised learning has been applied for accurate estimation of line-flows using Artificial Neural Network.

4.9 Solution Algorithm

The solution algorithm is given in the form of flow chart


Start
Collect system parameters for load flow study
Perform NR Load Flow Method for each load condition
Form normalized training and testing patterns store it in a data text files
Select the ANN structure and sigmoid activation function
Read the number of the neurons in the input, hidden & output layers, tolerance value, input data of all the input and output patterns from data text file stored in memory
Initialize the initial weights; adaptation gain η, threshold θ and momentum gain α between −1 and +1
Iteration count IT = 1
Input Pattern IP = 1
Is IP > NP ?
Yes
Calculate the actual output
Calculate the error
IP = IP + 1
Find the maximum error
Is Max. Error < tolerance?
Yes
Find Out Presentation for which Error is Max.
Adjust the weights to reduce the error
IT = IT + 1
Enter input data to test developed MFFNN
Print Line Flow
Stop

5.0 Case Study

To illustrate the proposed approach, IEEE-6 bus system is considered as shown in Fig. 4. In this work our goal is to examine the generalization capability of the ANN in the scope of being able to deal a large range of operating conditions and changes in network topology.

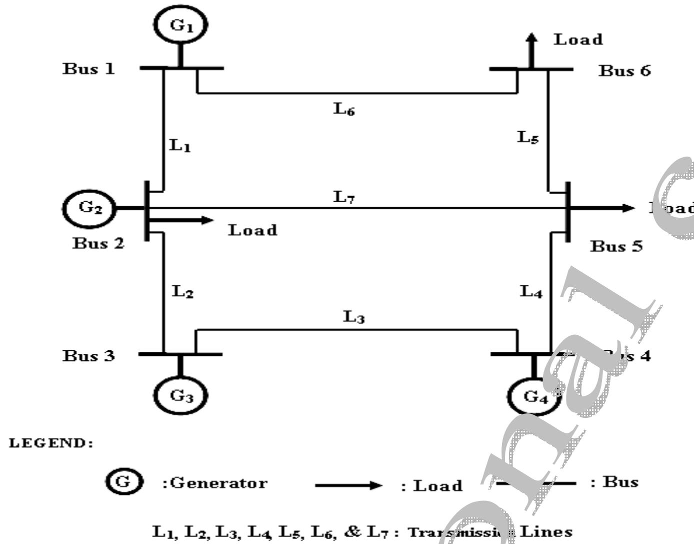


Fig. 4. Single Line Diagram of IEEE-6 Bus System

The **IEEE 6-Bus System** has four generators at buses 1, 2, 3, & 4 and loads at bus 2, 5, and 6. Bus 2 has both generator and load. In order to generate the training data patterns, three system topologies and load variations were used; Topology 1 with all lines in service; Topology 2 with line 2-5 Outage; Topology 3 with line 3-4 Outage. For each topology **16** different loading conditions are selected with loading level of the system in the range from **0.6 to 1.2** relative to the nominal operating point. Power factor of the loads are maintained at their nominal values. The training data set consists of **48×29** dimensional patterns labeled with their corresponding line flow values or bus voltages. The training and testing data are obtained by conventional **NR load flow** method using commercial **Mipower Software Package,** for different system topologies over a range of load levels. Although one MFFNN could be used to solve this problem, in this work three MFFNNs are used; two for line flows and other for bus voltage magnitude this produces better generalization result to reduce the over all number of weights needed to represent the over all relation. The line flow neural network is trained to map the Y- bus matrix, busload and generation injection to the line flow, while the bus voltage maps the same inputs to the bus voltage. The Y-bus represents the network topology under all possible transmission contingencies, while the busload and generation injection vectors represent possible variation in load and generation distribution patterns based on forecasted and generation scenarios.

5.1 Inputs to the Network

Input patterns to the neural network are obtained from NR Load Flow solution. Following are the variables;

i) Self and Mutual Admittances
ii) Complex Bus Loads
iii) Complex Bus Powers

An Artificial Neural Network having three layers with number of neurons in input layer **N_i=29**, chosen to be the same as that of the input variables, number of neurons in hidden layer **N_h=6** and output layer **N_o=1** neuron was selected. For training **MFFNN,** back propagation algorithm is used. After training, the least squared error (E) is reduced to **0.0001058 in 50000** presentations of training data set. The parameters of the learning process are **momentum gain α =0.25; threshold θ=0.87768089; adaptation gain η=0.8960400**. The performances of the MFFNNs for new cases not presented during the training session are shown in Tables-1a, 1b,1c, Table-2a, 2b, 2c, Table-3a, 3b. & 3c corresponding to different network topologies and particular operating condition.

6.0 Estimation of Active, Reactive Power Flows and Bus Voltages

The Tables-1a, 1b, 1c shows the comparison between actual active power flow by NR load flow and estimated active power flow by ANN in different lines and operating conditions.

Table-1a
Line Flow (P) Results from NR Contingency Evaluation Method and MFFNN Based Algorithm for Load Level of 0.6pu
Base Case: All Lines in service

Line No.	Line Flow by NR Method P_{pq} pu	Line Flow by MFFNN Based Algorithm, P_{pq} pu	Error in %
L_1	0.0316	0.0298	0.18
L_2	0.0337	0.0340	-0.03
L_3	0.0273	0.0321	-0.48
L_4	0.0374	0.0359	0.15
L_5	0.0337	0.0341	-0.04
L_6	0.0352	0.0355	-0.03
L_7	0.0327	0.0331	-0.04

Table-1b
Contingency Case I: Line 2-5 (L_7) Outage

Line No.	Line Flow by NR Method P_{pq} pu	Line Flow by MFFNN Based Algorithm, P_{pq} pu	Error in %
L_1	0.0370	0.0377	-0.07
L_2	0.0359	0.0364	-0.05
L_3	0.0327	0.0317	0.10
L_4	0.0359	0.0364	-0.05
L_5	0.0320	0.0376	-0.56
L_6	0.0370	0.0377	-0.07

Table-1c
Contingency Case II: Line 3-4 (L_3) Outage

Line No.	Line Flow by NR Method P_{pq} pu	Line Flow by MFFNN Based Algorithm, P_{pq} pu	Error in %
L_1	0.0396	0.0376	0.20
L_2	0.0324	0.0330	-0.06
L_4	0.0368	0.0374	-0.06
L_5	0.0396	0.0376	0.20
L_6	0.0347	0.0345	0.02
L_7	0.0320	0.0300	0.20

Table-2a
Line Flow (Q) Results from NR Contingency Evaluation Method and MFFNN Based Algorithm for Load Level of 0.6pu
Base Case: All Lines in service

Line No.	Reactive Line Flow by NR Method, Q_{pq} pu	Reactive Line Flow by MFFNN Based Algorithm, Q_{pq} pu	Error in %
L_1	0.0287	0.0286	0.01
L_2	0.0271	0.0315	-0.44
L_3	0.0313	0.0309	0.04
L_4	0.0286	0.0284	0.02
L_5	0.0339	0.0328	0.11
L_6	0.0271	0.0315	-0.44
L_7	0.0312	0.0297	0.15

Table-2b
Contingency Case I: Line 2-[illegible] (L_7) Outage

Line No.	Reactive Line Flow by NR Method, Q_{pq} pu	Reactive Line Flow by MFFNN Based Algorithm, Q_{pq} pu	Error in %
L_1	0.0351	0.0318	0.33
L_2	0.0342	0.0301	0.41
L_3	0.0331	0.0328	0.03
L_4	0.0320	0.0376	-0.56
L_5	0.0351	0.0318	0.33
L_6	0.0291	0.0301	-0.10

Table-2c
Contingency Case II: Line 3-4 (L_3) Outage

Line No.	Reactive Line Flow by NR Method, Q_{pq} pu	Reactive Line Flow by MFFNN Based Algorithm, Q_{pq} pu	Error in %
L_1	0.0293	0.0314	-0.21
L_2	0.0306	0.0342	-0.36
L_4	0.0302	0.0297	0.05
L_5	0.0317	0.0311	0.06
L_6	0.0287	0.0286	0.01
L_7	0.0349	0.0306	0.43

Table-3a
Bus Voltage Results from NR Contingency Evaluation Method and MFFNN Based Algorithm for Load Level of 0.6pu
Base Case: All Lines in service

Bus No. P	Bus Voltage by NR Method, V_p pu	Bus Voltage by MFFNN Based Algorithm, V_p pu	Error in %
2	1.1039	1.1238	-1.99
3	1.0829	1.0701	1.28
4	1.0860	1.0407	4.53
5	1.0421	1.0822	4.01
6	1.0220	1.0000	2.20

Table-3b
Contingency Case I: Line 2-5 (L_7) Outage

Bus No. P	Bus Voltage by NR Method, V_p pu	Bus Voltage by MFFNN Based Algorithm, V_p pu	Error in %
2	1.0420	1.0400	0.20
3	1.0780	1.1120	-3.40
4	1.0418	1.0858	-4.20
5	1.0284	1.0708	-4.24
6	1.1063	1.1088	-0.25

Table-3c
Contingency Case II: Line 3-4 (L_3) Outage

Bus No. P	Bus Voltage by NR Method, V_p pu	Bus Voltage by MFFNN Based Algorithm, V_p pu	Error in %
2	1.0537	1.0091	4.46
3	1.0418	1.0838	-4.20
4	1.0114	1.0235	-1.21
5	1.0506	1.0972	-4.66
6	1.0406	1.0800	-3.94

7.0 Conclusion

The designed ANN model has been applied to predict the line-flows and bus voltages under changing operating condition of the power system. Once the ANN is trained, it predicts quick results for unknown load patterns. The computation of line-flows by load flow analysis takes long time, as it should be run for any change in load/generation. On the other hand, by the proposed method, once the training of the ANN is successfully completed, the prediction of the line-flows and bus voltages is almost instantaneous. This can be used for real time application. The outcome of this work can be used to examine the performance of a power system following a contingency and the need for new transmission expansion due to load growth or generation expansion. In operation, contingency analysis assist engineers to operate the power system at a secure operating point where the equipment are loaded within their safe limits and power is delivered to customers with acceptable quality standards.

8.0 References

[1] A. P. S. Meliooulose, C. S. Cheng, and Feng Xia, **"Performance Evaluation of Static Security Analysis Methods"** IEEE Transactions on Power Systems August 1994, Vol.9, PP 1441- 1449.

[2] J. L. Carpentier, P. J. Di Bono and P. J. Tournebise, **"Improved Efficient Method for DC Contingency Analysis using Reciprocity Properties"** IEEE Transactions on Power Systems, Vol.9, PP 76-84, February 1994.

[3] A. J. Wood and B. F. Wollenbeg, **"Power System Operation & Control"** John-Wiley & Sons, 1984.

[4] B. H. Chodhury and B.M. Ilamowski, **"Fast Power Flow with Capacity of Corrective Control Using Neural Network"** IEEE Transaction on Power System, August1992.

[5] J. C. Chow, R. Fichl, M. Kam, H. H. Yan and S. Ricciardi, **"An improved Hopfield Model for Power System Contingency Classification"** IEEE Transaction on Power System, August 1990.

[6] D. Niebur and A. J. Germond, **"Power Flow Classification for Static Security Assessment"** IEEE Transaction on Power System, November 1991.

[7] K. L. Lo, A. O. Ekwue, and D. T. Cheng, **"Application of self-organizing Neural Network to Static Security Assessment"** Artificial Neural Network, 26-28 June, 1995, IEE Conference Publication.

[8] G. Girrinione, M. Girrinione, and F. Piglione, **"A Neural Network Architecture for Static Security Mapping in Power System"** IEEE Transaction on Power System, May 1996.

[9] S. Weerasooriya, M. A. El-sharkawi, M. D. Damborg and R. J. Marks II, "**Towards Static Security Assessment of Large-scale Power System using Neural Networks"** IEE Proceedings-C, Vol.1, January 1992.