

# Power system security monitoring through classification approach using Multilayer Feed Forward Neural Network

<sup>1</sup>Ishita Bhatt, <sup>2</sup>Astik Dhandhia

<sup>1</sup>Asst. Professor, <sup>2</sup> Asst. Professor

<sup>1</sup>Department of Electrical Engineering,

<sup>1</sup>Aditya Silver Oak Institute of Technology, Ahmedabad, India

**Abstract:** The power system is a complex network with numerous equipment's interconnected which are forced to operate under highly stressed conditions closer to their limits. One of the major aspect for the secure operation of the system can be achieved through security assessment, context to which the power system static security assessment is necessary to evaluate the security status under contingency scenario. The conventional method of security assessment involves solving the set of nonlinear load flow equations. But the complexity and computation time makes them infeasible for real time security assessment of large power system networks. This necessitates the need for an efficient approach to assess the security status in short period of time. Thus, it is necessary to design an effective security assessment model. Multi-layer feed forward artificial neural network (MLFFN) is proposed to implement the classification approach for power system static security assessment. The contingency classification, is done based on the composite security index which is capable of accurately differentiating the secure and non-secure cases. For each contingency case as well as for base case condition, the composite security index is computed using the full Newton Raphson load flow analysis. The proposed artificial neural network (ANN) model take features selected using single ranking method and the probable contingencies as the input, assessing the system security by classifying the credible contingencies as secure and insecure.

**Index Terms – Feature selection, classification, power system security, Neural Network.**

## I. INTRODUCTION

In competitive electricity markets, customers expect a least-cost and high-quality supply of electric energy, which may require additional investments and more sophisticated operation techniques for enhancing power systems security. The sophistication could in part mitigate severe consequences in the event of cascaded power system contingencies, which might otherwise result in dramatic property and human losses, and severely impede the growth in the national economy making power system security a major issue of concern. Conventional method of security assessment involves solving the set of nonlinear load flow equations. But the complexity and computation time makes them infeasible for real time security assessment of large power system networks. However, the accuracy and the speed of the security evaluation depends on the type of approach used. This necessitates the need for an efficient approach to assess the security status in short period of time. Thus, it is necessary to design an effective security assessment model.

This factor motivated to design a quick and efficient model that predict the system severity for the power system security assessment by contingency classification approach. Further, an attempt was made to use pattern recognition technique for the security assessment problem by classification approach, but the design of a powerful pattern recognition system with good input features and classifier model is a drawback. The work presented is attempts to significantly reduce the computation time required for security assessment so that the analysis can be converted from offline to online use. From the above discussion, it is clear that for the security assessment and the enhancement, there is a need and scope to develop fast and efficient algorithmic techniques.

## II. CLASSIFICATION APPROACH

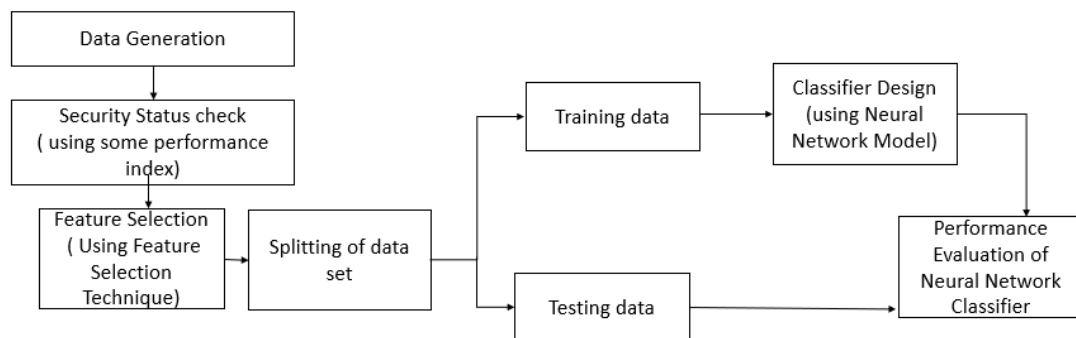


Figure 1 Classification Approach

## III. FEATURE SELECTION USING SINGLE RANKING METHOD

As a number of variable increase in the pattern, it is found that model becomes complicated and it takes more time for training, so feature selection technique is employed for selection of selection for the large power system. In pattern recognition, feature selection is a special form of dimensionality reduction. Feature selection involves simplifying the number of resources required to describe a large set of data accurately. Generally, the number of variables describing the power system state in a pattern vector is very large. Feature selection is the process of selecting a subset of original features, by removing redundant and irrelevant variables. Features may be selected by engineering judgment. But such selections will be subjective with the possibility of important variables getting rejected. In this paper, we have used Single Ranking and Correlation Coefficient approach. The general procedure for single ranking and correlation coefficient approach is as under, Calculate  $F_i$  for all  $i$  such that  $0 \leq i \leq n$ .

$$F(x_j) = \frac{|\mu_j^s - \mu_j^{is}|}{\sigma_j^s + \sigma_j^{is}}$$

Where  $\mu_j^s$  is the mean of the  $j^{\text{th}}$  attribute insecure class (insecure class) pattern vector;  $\sigma_j^s$  is the standard deviation of the  $j^{\text{th}}$  attribute insecure class (insecure class) pattern vector. This method is chosen for the reason that it is very simple and relies only on the general characteristics of the data without involving any mining algorithm. An acceptable simple criterion for selecting a variable as a feature is that it should provide more information about the decision or classification than other variables.

## IV. MULTILAYER FEED FORWARD NEURAL NETWORK

Neural networks are commonly used for classification problems and regression problems. In classification problems, the objective is to determine which class (out of several possibilities) that an input belongs to. For example, say we want to have a network learn to distinguish patterns as secure and insecure class. So, we would provide the network with a number of patterns, and for each pattern, we would tell the network whether the pattern is secure (0) or insecure (1).

In regression problem, the objective is to learn a real-valued target function. An example would be to learn the exact composite security index value. The type of problem one is trying to solve (i.e., whether you're solving a regression problem or a classification problem) will determine exactly how to structure the network.

The application of the artificial neural networks has gained major importance in many engineering fields, because of the fact that these models can easily handle complex and non-linear problems. ANNs are massively parallel-interconnected networks of simple elements intended to interact with the real world in the same way as the biological nervous system. They offer an unusual scheme, based programming standpoint and exhibit higher computing speeds compared to other conventional methods. The ANNs are characterized based on their topology, such as, the number of interconnections, the node characteristics that are classified by the type of nonlinear elements used and the kind of learning rules employed. The ANN architecture is designed with well-organized topology of Processing Elements (PEs) called neurons. In Multilayer Feed Forward Neural network (MFNN) the neurons are arranged in layers and only neurons in adjacent layers are connected.

The MFNN architecture consists of three layers, which consists of input layer, hidden layer and an output layer as shown in figure 4. This structure is called multilayer because it has a layer of processing units (i.e., the hidden units) in addition to the output units. These networks are called feedforward because the output from one layer of neurons feeds forward into the next layer of neurons. There are never any backward connections, and connections never skip a layer. Typically, the layers are fully connected, meaning that all units at one layer are connected with all units at the next layer. So, this means that all input units are connected to all the units in the layer of hidden units, and all the units in the hidden layer are connected to all the output units.

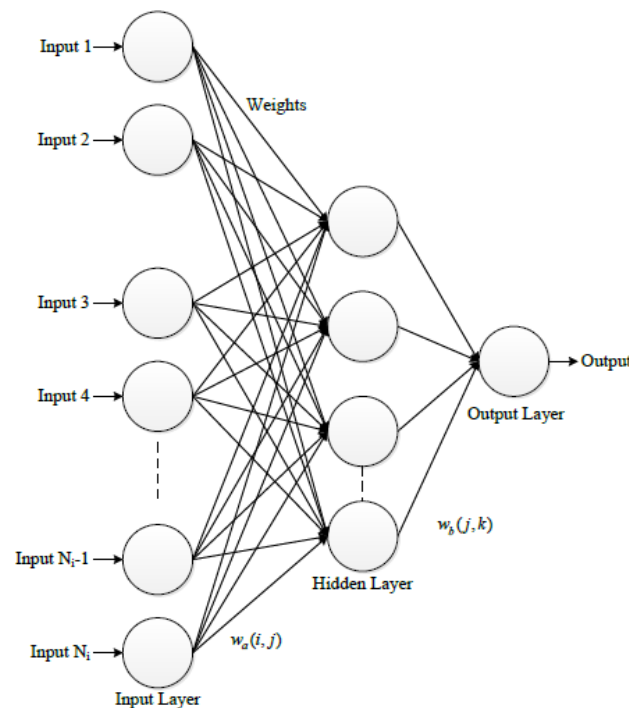


Figure 2 Multi-Layer Feedforward Neural Network<sup>[9]</sup>

Each connection between nodes has a weight associated with it. In addition, there is a special weight (called  $w_0$ ) that feeds into every node at the hidden layer and a special weight (called  $z_0$ ) that feeds into every node at the output layer. These weights are called the bias and set the thresholding values for the nodes. Initially, all of the weights are set to some small random values near zero. The training of our network will adjust these weights using the Backpropagation algorithm so that the output generated by the network matches the correct output.

### 1) 3.1.1 Processing at a neuron

Every node in the hidden layer and in the output layer processes its weighted input to produce an output. This can be done slightly differently at the hidden layer, compared to the output layer.

### A. Input units

The input data provided to network comes through the input units. No processing takes place in an input unit – it simply feeds data into the system. The value coming out of an input unit is labeled  $x_i$ , representing  $i$  input units. There is also a special input unit labeled  $x_0$ , which always has the value of 1. This is used to provide the bias to the hidden nodes.

### B. Hidden units

The connections coming out of an input unit have weights associated with them. A weight going to hidden unit  $z_j$  from input unit  $x_i$  would be labeled  $w_{ji}$ . The bias input node,  $x_0$ , is connected to all the hidden units, with weights  $w_{h0}$ . In the training, these bias weights,  $w_{h0}$ , are treated like all other weights, and are updated according to the backpropagation algorithm. The value coming out of  $x_0$  is always 1.

Each hidden node calculates the weighted sum of its inputs and applies a thresholding function to determine the output of the hidden node. The weighted sum of the inputs for hidden node  $z_j$  is calculated as:

$$\sum_{d=0}^i w_{dj} x_d \quad (1)$$

The thresholding function applied at the hidden node is typically either a step function or a sigmoid function. The general form of the sigmoid function is:

$$\text{Sigmoid}(a) = \frac{1}{1+e^{-a}} \quad (2)$$

The sigmoid function is sometimes called the “squashing” function, because it squashes its input (i.e.,  $a$ ) to a value between 0 and 1. At the hidden node, we apply the sigmoid function to the weighted sum of the inputs to the hidden node, so we get the output of hidden node  $z_j$  is:

$$z_j = \text{Sigmoid} \left( \sum_{d=0}^i w_{dj} x_d \right) = \frac{1}{1+e^{-\left(\sum_{d=0}^i w_{dj} x_d\right)}} \quad (3)$$

For  $h$  going from 1 to  $H$ , where  $H$  is the total number of hidden nodes.

### C. Output units

Now, following similar computation for the output nodes but difference being in the computation of output depends on the type of problem to be solved – either a regression problem or a classification problem. And, the calculation also depends on whether problem has 1 output unit or multiple output units. Starting out the same as with the hidden units, calculating the weighted sum. Label the weights going into output unit  $k$  from hidden unit  $j$  as  $w_{jk}$ . Just like the input layer, there is also a bias at the hidden layer. So, each output unit has a bias input from hidden unit  $z_0$ , where the input from  $z_0$  is always 1 and the weights associated with that input are trained just like all the other weights.

#### 2) 3.1.2 Selection of ANN Parameters

The learning rate of the Backpropagation Algorithm (BPA) is influenced by the momentum factor,  $\alpha_1$  and the learning rate parameter,  $\eta_1$ . The BPA contribute an approximation to the trajectory in the weight space calculated by the method of steepest descent [9]. If the considered value of  $\eta_1$  is very small, which results in slow rate of learning, while if the value of  $\eta_1$  is too large in order to speed up the rate of learning, the MFNN may become unstable. A simple method of increasing the rate of learning without making the MFNN unstable is by adding the momentum factor  $\alpha_1$ . Preferably, the values of  $\eta_1$  and  $\alpha_1$  should lie between 0 and 1.

### 3) 3.1.3 Weight Update Equations

The weights between the hidden layer and the output layer are updated based on the equation (4).

$$W_b(j, k, m+1) = W_b(j, k, m) + \eta_1 * \delta_k(m) * S_b(j) + \alpha_1(W_b(j, k, m) - W_b(j, k, m-1)) \quad (4)$$

Where,  $m$  is the total number of iterations, where  $j$  and  $k$  varies from 1 to  $N_h$  and 1 to  $N_k$  respectively.  $\delta_k(m)$  is the error for the  $k^{th}$  output at the  $m^{th}$  iteration,  $S_b(j)$  is the output from the hidden layer.

Similarly, the weights between the hidden layer and the input layer are updated as in equation

$$W_a(i, j, m+1) = W_a(i, j, m) + \eta_1 * \delta_j(m) * S_a(i) + \alpha_1(W_a(i, j, m) - W_a(i, j, m-1)) \quad (5)$$

Where  $i$  varies from 1 to  $N_i$ , as there are  $N_i$  inputs to the network,  $\delta_j(m)$  is the error for the  $j^{th}$  output after the  $m^{th}$  iteration and  $S_a(i)$  is the output from the first layer.

The  $\delta_k(m)$  in equation (4) and  $\delta_j(m)$  in equation (5) are related as,

$$\delta_j(m) = \sum_{k=1}^K \delta_k(m) * w_b(j, k, m) \quad (6)$$

### 4) 3.1.4 Evaluation Criteria

The Mean Square Error  $E_{tr}$  for the training patterns after the  $m^{th}$  iteration is defined as,

$$E_{tr} = \left( \frac{1}{N_P} \right) * \left[ \sum_{P=1}^{N_P} \{X_{1P} - X_{2P}(m)\}^2 \right] \quad (7)$$

Where  $X_{1p}$  is the actual value and  $X_{2p}$  is the estimated value of the CSI after the  $m^{th}$  iteration. The training is stopped when the least value of  $E_{tr}$  is obtained and this value does not change much with the number of iterations. The  $E_{tr}$  tells how well the network has adopted to fit the training data only, even if the data is contaminated.

## V. DATA GENERATION FOR TRAINING AND TESTING

In this work, to realize the effectiveness of the ANN models, the data is generated by performing large set of offline computations. This is achieved by considering the IEEE-30 bus and IEEE-57 bus systems, and varying the load condition from 80% to 120% of its base case. For each loading condition, under N-1 line outage contingency, the performances indices, the line flows and the voltages are computed using Newton-Raphson load flow analysis. The generated data is used to train the neural network models.

For IEEE-30 bus system, the total number of line outage cases considered are 37. The load is varied from 80% to 120% of its base case and the corresponding values of Composite Security Index (CSI) is computed for each loading condition. The total number of patterns generated are 820. Out of which, 762 sets of input-output patterns are used to train the network and the remaining 58 sets are used to test the network.

For IEEE-57 bus system, the total number of line outage cases considered are 73. The load is varied from 80% to 120% of its base case and the corresponding values of Composite Security Index (CSI) is computed for each loading condition. The total number of patterns generated are 1626. Out of which, 1470 sets of input-output patterns are used to train the network and the remaining 156 sets are used to test the network.

Once the neural network models are trained to obtain least mean square error, they are feasible to undergo testing to predict the performance indices. The models are used in the ranking module to predict security status in terms of

the performance indices for a particular operating condition for static security assessment by contingency ranking approach.

### B. 3.3 PREDICTION OF COMPOSITE SECURITY INDICES USING MFNN

The model predicts the Composite Security Index (CSI) as a function of different power system network operating conditions. The network is provided with both input data (training data) and desired output (target data), and it is trained in a supervised learning fashion using the back propagation algorithm. The back propagation algorithm performs the input to output mapping by making weight connection adjustment following the error between the computed output value and the desired output response. The training phase is completed after a series of iterations. In each iteration, output is compared with the desired response and a match is obtained.

The number of input parameters to the network is obtained using the feature selection technique single ranking method is used here, using which total 41 features are selected and the output parameter is composite security index. Figure 5 shows the flowchart for the prediction of performance indices using MFNN.

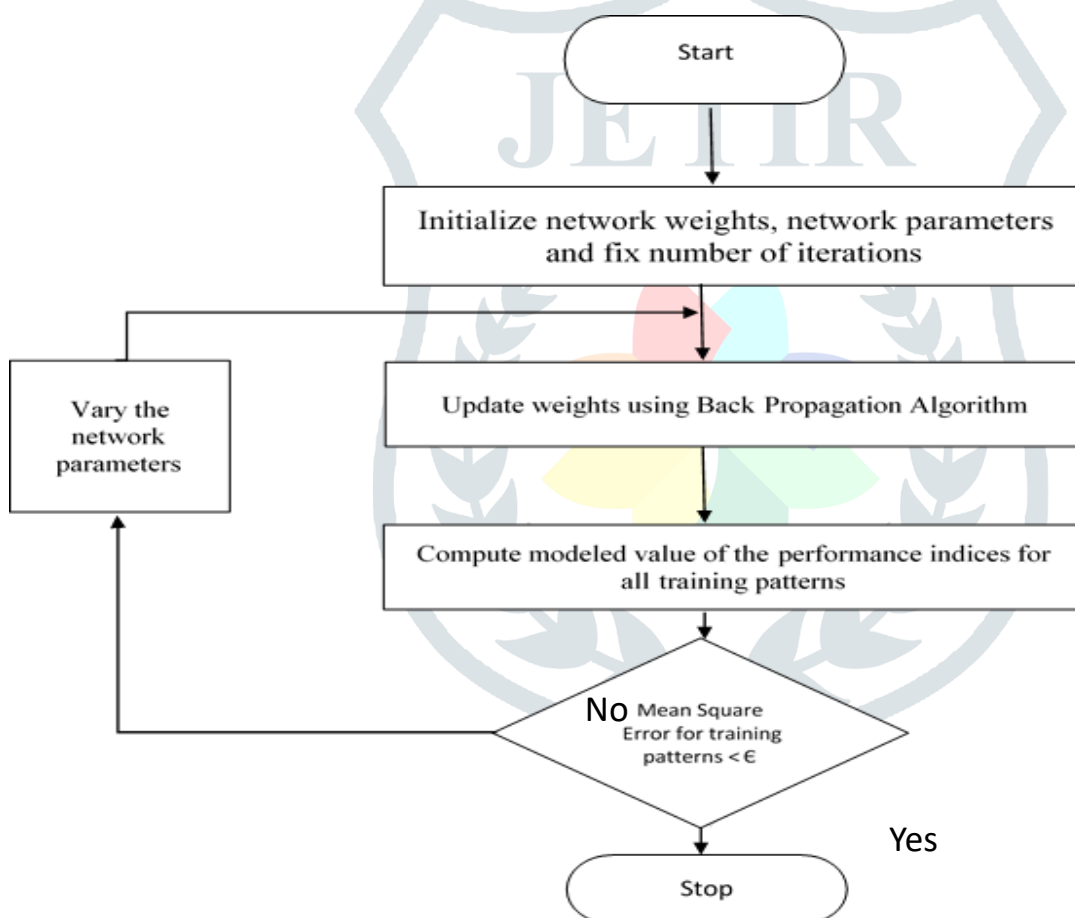


Figure 3 Flow chart for MFNN<sup>[7]</sup>

### C. 3.4 SIMULATION RESULTS

The performance of the neural network models depends on the optimal selection of network parameters. In this approach, the optimum network parameters are obtained based on mean square error  $E_{tr}$  for the training patterns.



The network is trained in a sequential order. In applying the BPA for the prediction of performance indices, the following important parameters are discussed.

1. Network parameters
2. Number of hidden neurons
3. Number of iterations

For BPA, the optimal values of the learning rate  $\eta_1$ , and momentum factor  $\alpha_1$  are obtained by performing simulation with different values of  $\eta_1$  and  $\alpha_1$ . Initially, the values of  $\eta_1=0.01$  is considered and varied to obtain the optimum value. The range of values of  $\eta_1$  and  $\alpha_1$  should lie between 0 and 1. The accuracy of the model depends on the optimal selection of the network parameters. To obtain optimal parameters, a series of simulations were carried out to obtain least mean square error (MSE). The simulations are carried to obtain least mean square error, by varying one parameter, with fixed values of other two parameters. The combination with least mean square error  $E_{tr}$  are selected as the optimal parameters.

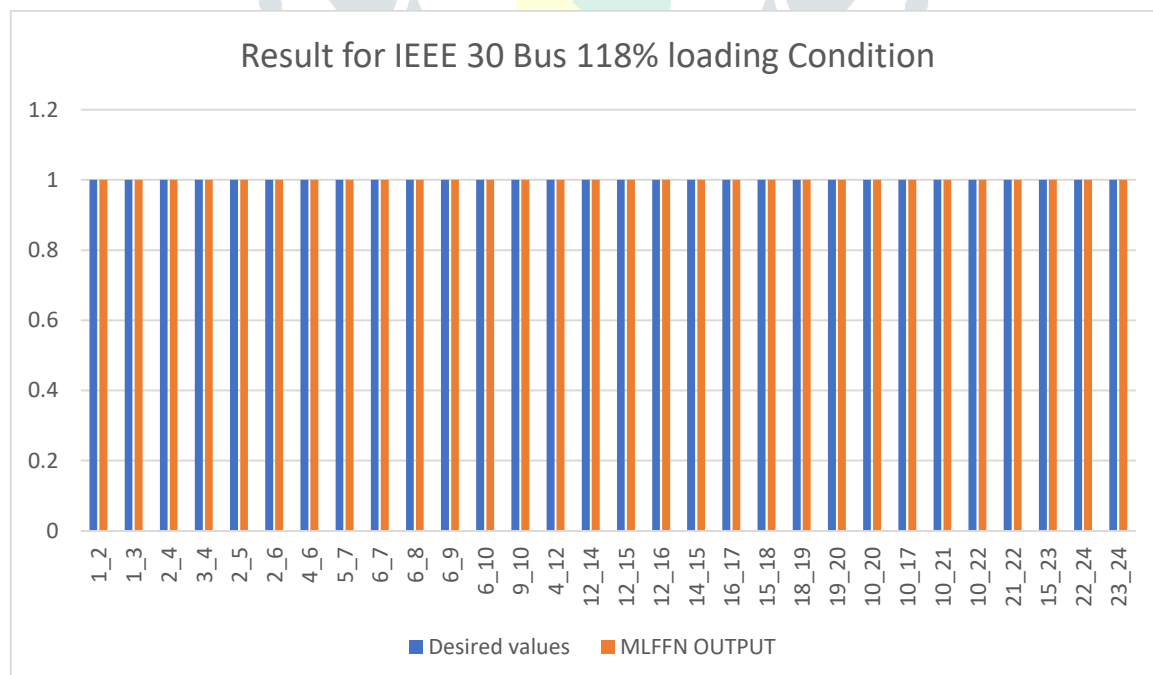
Here two neural networks are used, to have a cascade neural network approach in which first network is used to classify the data in two classes that is secure and insecure. While the second network is used to do regression on insecure patterns only and network is trained to obtain the accurate values of composite security index.

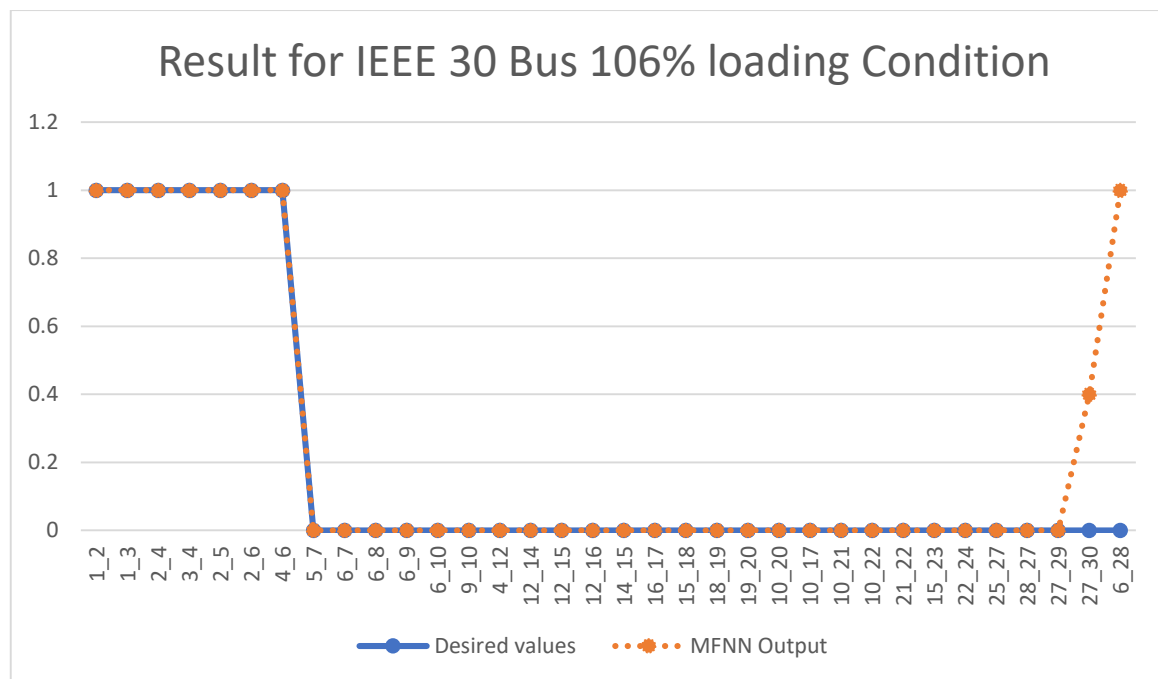
## VI. RESULTS AND DISCUSSION

### 1) 3.4.1 Results for IEEE-30 bus System

A 2 layered feed forward neural network is made for classification process, while the ranking module is 3 layered. For classification purpose all the CSI values are normalized between 0 and 1, where 1 indicates insecure class and 0 indicates secure class.

From Table 1, it can be observed that the parameters chosen are optimal network parameters using which when the classification is done it gives highly accurate results. There is no misclassification at 118% loading of base case and only 2 misclassification at 106% loading condition.





2)

3) 2 Results for IEEE-57 Bus System

A 2 layered feed forward neural network is made for classification process, while the ranking module is 3 layered. For classification purpose all the CSI values are normalized between 0 and 1, where 1 indicates insecure class and 0 indicates secure class.

From Table 4, it can be observed that the parameters chosen are optimal network parameters using which when the classification is done it gives highly accurate results. There is 4 misclassification cases out of 78 patterns at 98% loading of base case.

## II. REFERENCES

- [1] A. J. Wood and B. F. Wollenberg. *Power generation, operation and control*, John Wiley & Sons, 2009.
- [2] Pang, C.K., Prabhakara, F.S., El-Abiad, A.H. and Koivo, A.J., 1974. Security evaluation in power systems using pattern recognition. *IEEE Transactions on Power Apparatus and Systems*, (3), pp.969-976.
- [3] Da Costa, J.S. and Munro, N., 1984. Pattern recognition in power-system security. *International Journal of Electrical Power & Energy Systems*, 6(1), pp.31-36.
- [4] Sergios Theodoridis, Konstantinos Koutroumbas, *Pattern Recognition*, John Wiley & Sons, Prentice Hall, 3rd Edition, 2003.
- [5] Misra, R.K. and Singh, S.P., 2004. Steady-state security analysis using artificial neural network. *Electric Power Components and Systems*, 32(11), pp.1063-1081.
- [6] Sekhar, P. and Mohanty, S., 2016. An online power system static security assessment module using multi-layer perceptron and radial basis function network. *International Journal of Electrical Power & Energy Systems*, 76, pp.165-173.
- [7] Pudi Sekhar (2016). Assessment and Enhancement of Power System Security using soft computing & Data Mining approaches. (Doctoral Dissertation). National Institute of Technology, Rourkela.
- [8] Singh, S.N., Srivastava, L. and Sharma, J., 2000. Fast voltage contingency screening and ranking using cascade neural network. *Electric Power Systems Research*, 53(3), pp.197-205.



- [9] Misra, R.K. and Singh, S.P., 2010. Efficient ANN method for post-contingency status evaluation. *International journal of electrical power & energy systems*, 32(1), pp.54-62.
- [10] Sunitha, R., Kumar, S.K. and Mathew, A.T., 2013. Online static security assessment module using artificial neural networks. *IEEE Transactions on Power Systems*, 28(4), pp.4328-4335.
- [11] Kalyani, S. and Swarup, K.S., 2009. Study of neural network models for security assessment in power systems. *International Journal of Research and Reviews in Applied Sciences*, 1(2), pp.104-117.
- [12] Test CaseArchive.[Online].Available:<http://www.ee.washington.edu/research/pstca/>.
- [13] The Math Works Inc, MATLAB R2010a.

