

# Power system steady state calculations using artificial neural networks

Muzaffar Khudayarov<sup>1</sup>, Nuriddin Normamatov<sup>2</sup>

<sup>1,2</sup>Tashkent State Technical University named after Islam Karimov, Department of Power Plants, Networks and Systems. 100095, 2 University street, Tashkent, Uzbekistan

**Abstract.** The power systems steady-state problem are described by a system of nonlinear equations, and for their solution are widely used iterative techniques such as the Newton-Raphson and others. Recently, techniques based on the use of genetic algorithms, the theory of fuzzy sets, artificial neural networks have been applied to solve this problem. In this article feedforward neural networks are used for calculating the steady-state regimes. The modeling results were obtained with the results of calculations using the Newton-Raphson method.

## 1 Introduction

The power systems steady-state calculations are a very urgent task of the electric power industry both at the design stage and during its operation. The accuracy and reliability of the results obtained ensure the correct functioning of the power systems.

As you know, the steady-state calculation is the determination of the parameters of the regimes with known circuit parameters and parameters of loads and generation [1]. In this case, the problem can be represented as a system of high-order nonlinear algebraic equations or in matrix form, which causes certain difficulties. Also, for its solution, various iterative techniques are traditionally used [2], which is associated with great mathematical and computational difficulties. In addition, the use of iterative techniques for operational calculations is also ineffective due to the large time costs associated with the computational cumbersomeness of these methods.

That is why the search for the most optimal methods for calculating steady-state regimes is an important task. From this point of view, a promising direction is associated with the use of modern intelligent data processing tools such as genetic algorithms, fuzzy sets, artificial neural networks and other.

Recently, techniques based on the use of such modern technologies as genetic algorithms [3, 4, 5], theory of fuzzy sets [6, 7], artificial neural networks [8, 9, 10, 11] are widely used for solving various energy problems.

Given the noted in this article, 2 types of feedforward neural networks (feedforward and cascadeforward) are used for calculating steady-state regimes.

### 1.1. Traditional approach to the problem of calculating the steady state

At present, iterative methods (or methods of successive approximations) have found wide practical application in the calculation of power systems steady-state calculations. Iterative methods include: simple iteration method, Gauss-Seidel method, Newton-Raphson method and its modifications, gradient method, etc [1, 2].

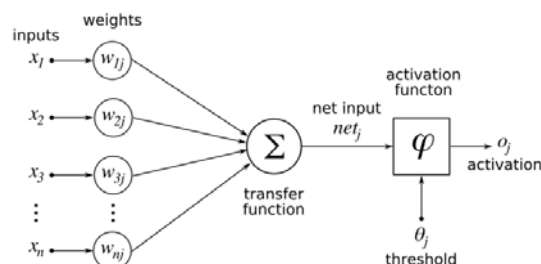
Simple iteration and Gauss-Seidel methods are not always sufficient to solve nonlinear systems of computational power equations, therefore, in practice, these methods are rarely used and are used to calculate the mode of simple electrical networks.

From a theoretical point of view, Newton-Raphson method is the most attractive, but its practical application runs into certain difficulties. In particular, the problem of finding a good initial approximation, the need to solve a linear system of equations at each iteration step, etc.

But despite these shortcomings, Newton-Raphson method is the main one for calculating steady-state regimes due to the high convergence rate (with good initial approximations, it is enough to perform 3-5 iterations) and are used in modern software tools like DIgSILENT PowerFactory, Etap, RASTR, Mustang, etc.

### 1.2. About artificial neural networks

Artificial neural networks (ANNs), usually simply called neural networks (NNs), are computing systems vaguely inspired by the biological neural networks. ANN's basic processing unit is a artificial neuron, the model of which is shown in Fig. 1. Inputs, a transfer function, an activation function, a threshold, and an output. In every neuron except for the input neurons, signals from the previous layer ( $x_1, x_2, \dots, x_n$ ) are multiplied by an associated adaptive weight ( $w_{1j}, w_{2j}, \dots, w_{nj}$ ), which indicates the connection strength of the neuron with a particular input, and the transfer (summation) function is then applied to the weighted signals  $net_j$ .



**Fig.1.** The model of artificial neuron

This process can be mathematically expressed as:

$$net_j = \sum_{i=1}^n x_i \cdot w_{ij} \quad (1)$$

The output ( $o_j$ ) of the neuron is then obtained by applying an activation function ( $\varphi$ ) to the net input ( $net_j$ ), and added the threshold of the neuron ( $\theta_j$ ) to the aggregate signals:

$$o_j = \varphi(net_j) + \theta_j \quad (2)$$

An activation function ( $\varphi$ ) is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1 etc.

The most commonly used are Sigmoid or Logistic Activation Function, so named for their curve looks like a S-shape. The main reason why we use sigmoid function is because it exists between (0 to 1).

$$\varphi(x) = \frac{1}{1+e^{-x}} \quad (3)$$

Hyperbolic tangent Activation Function  $th(x)$  is also like logistic sigmoid but better. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (S-shaped)

$$\varphi(x) = th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

ANNs can be classified into different types depending on the architecture and information flow procedure. There are one layer and multilayer networks. Among them, the

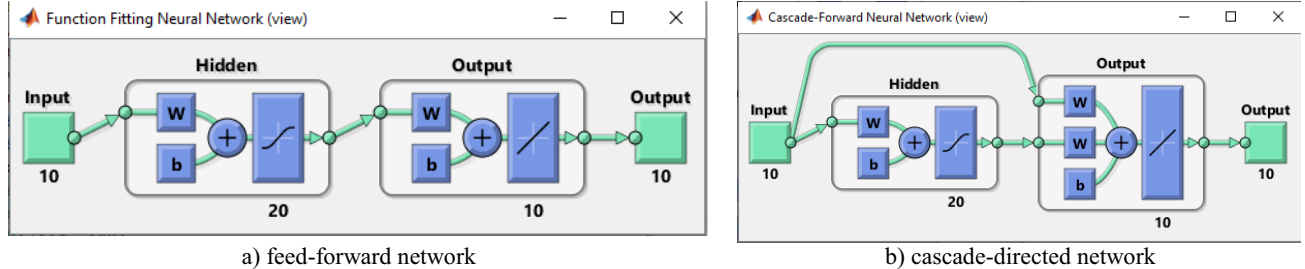
multilayer feedforward network consisting of an input layer, one or more hidden layer(s), and an output layer is the most commonly used network, where all of the neurons in each layer only have connections to the neurons of successive layers, not to neurons in the same layer.

To create neural networks, we use the Neural Network Toolbox library of the Matlab package [12], where various types of neural networks are implemented. In particular, feedforward neural network and cascadeforward neural network are considered (Fig. 2). The main difference between a cascade-directed network and a feed-forward network is that it has a connection from the input and each previous level to subsequent levels.

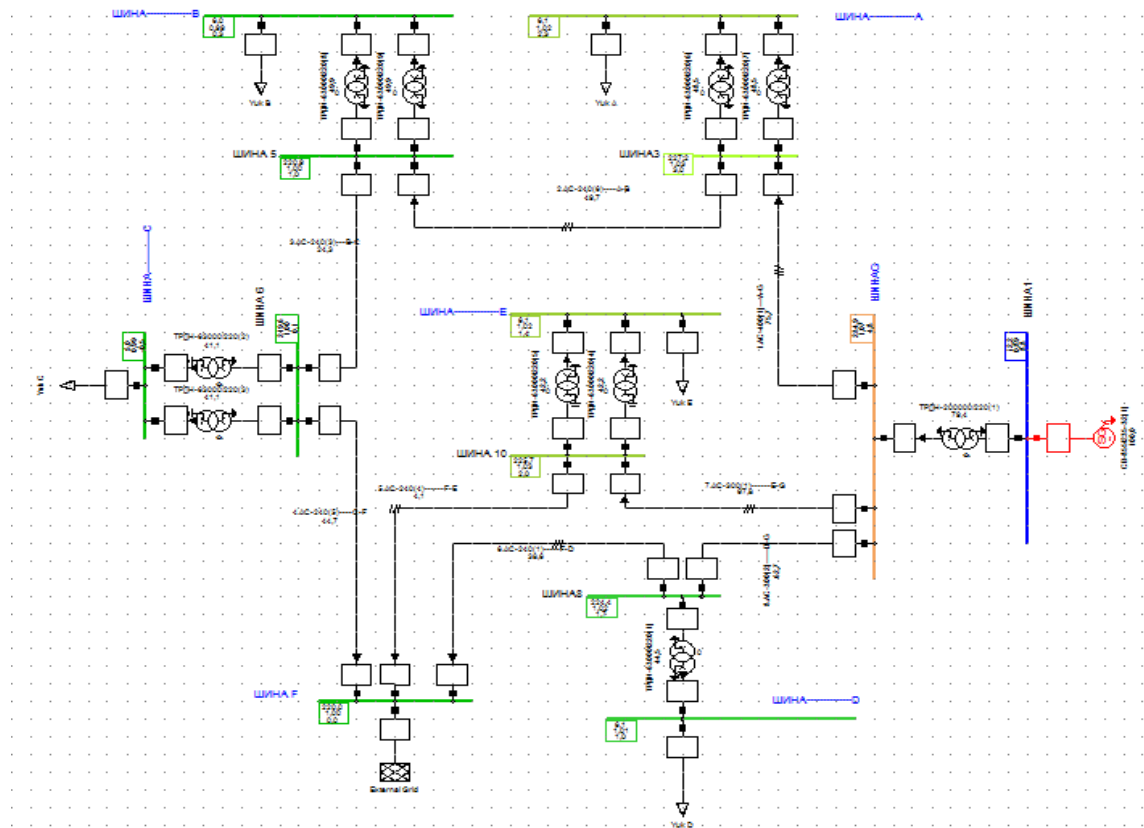
### 1.3. Generation of statistical data for ANN's modeling

To perform the necessary power systems steady-state calculations, a circuit was chosen that includes 5 load nodes, 1 generation unit and an infinite power bus (Fig. 3). The generating unit includes 3 generators of the CB-855 / 235-32 brand with a total power of  $S = 176.5$  MVA (active power  $P = 150$  MW;  $\cos(\phi) = 0.8$ ). The branch data is presented in Table 1.

In order to generate statistical data, the calculations of the steady state EES are performed using the DIGSILENT PowerFactory program, which is based on the Newton-Raphson method [13]. Using this program, a large number of calculations were performed, in a wide range of load changes (50–100% of the specified) in 5 load nodes. In each case, the module and the voltage phase of the load nodes are used as the result of the steady-state calculation.



**Fig 2.** Various types of neural networks



**Fig.3.** Calculation scheme of the electric power system

**Table 1.** Approved quarterly voltage schedule at control points of CA UPS

Branch	$U_n$ (kV)	Line type	Length (km)	$R_0$ , ohm	$X_0$ , ohm	$B_0 \cdot 10^{-4}$ , sim
G – B	220	AC-400	30	0.075	0.42	0.027
B – A	220	AC-240	42	0.121	0.435	0.026
A – C	220	AC-240	31	0.121	0.435	0.026
C – Sist	220	AC-240	35	0.121	0.435	0.026
G – E	220	AC-300	45	0.096	0.429	0.0264
E – Sist	220	AC-240	46	0.121	0.435	0.026
G – D	220	AC-300	55	0.096	0.429	0.0264
D – Sist	220	AC-240	43	0.121	0.435	0.026

On the basis of the entered data and the results of calculations, a statistical database of 150 input-output pairs was formed. In this case, the input variables are the active and reactive powers of the load bases ( $P_A, Q_A, P_B, Q_B, P_C, Q_C, P_D, Q_D, P_E, Q_E$ ), and the output variables are the module and voltage phase of the load bases ( $U_A, dU_A, U_B, dU_B, U_C, dU_C, U_D, dU_D, U_E, dU_E$ ).

Further, the resulting database is divided into training and control samples, which are necessary for training of the neural network and control of the trained network. The volume of statistics for the training and control sample can be changed. The proportion of 70% to 30% is taken as a basis.

#### 1.4. Selection of ANN structures

The choice of a suitable ANN structure includes the choice of the number of layers, the choice of the activation

function, the number of inputs and outputs, and the number of neurons in each layer.

For our task, a three-layer feedforward network is used which works well enough for this task. At the same time, the following are considered: a feedforward neural network and a cascade forward neural network.

As an activation function, a hyperbolic tangent is used for the hidden layer, and a linear one for the output layer. The number of neurons in the input layer and the output layer is 10 based on the amount of input and output data. The number of neurons in the hidden layer is determined by the results of experiments. The best values are obtained when the number of neurons in the hidden layer is 20.

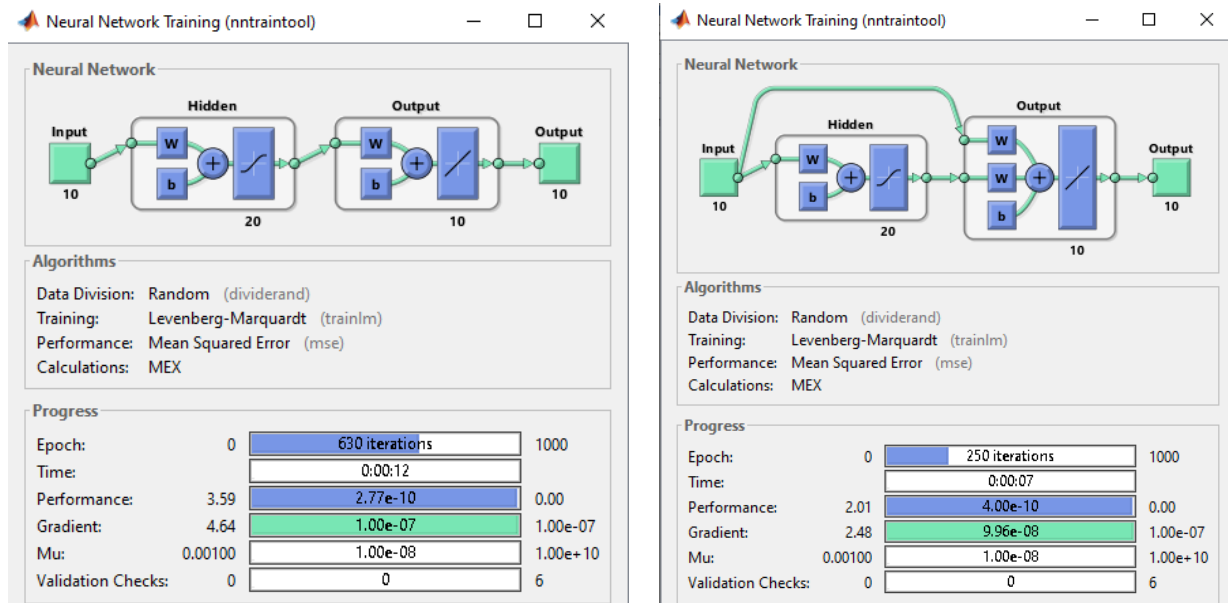
#### 1.5. Training of ANN

Feedforward neural network training is the process of determining the values of network weights based on

examples that form a training sample. In this case, the goal of training feedforward neural networks is to determine such a vector of weights  $w$  so that functional (2) takes a minimum value. There are many methods for solving this problem.

One of the popular methods is the backpropagation method. Despite its widespread use, its main disadvantages

are slow convergence and the negative impact of local minima. There are methods that do not have these disadvantages, among which the Levenberg-Marquardt method [14] is known, which gave the best results when solving the problem of calculating the steady state (Fig.4).



**Fig.4.** Training process of ANN

### 1.6. Comparative evaluation of ANN modeling results

A control sample is used to assess the quality of modeling. The input data of the load nodes from the control sample are fed to the input of the trained feedforward neural networks and the parameters of the steady state are calculated. The results obtained are compared with the results of calculations obtained in the DIGSILENT

PowerFactory program using the Newton-Raphson method. On Tables 2-6 shows the comparison of the estimated and actual bus voltages obtained by ANN-based algorithm and conventional Newton-Raphson method for Bus 1 – Bus 5.

The graphical comparison between the Newton-Raphson method and the ANN-based algorithm results (statistic data № 4) for Bus Voltages module under different operating conditions are shown in Fig. 5 and for Bus Voltage phase in Fig. 6.

**Table 2.** Comparative evaluation for Bus 1

Bus 1	Nyuton – Rafson (N-R)		cascadeforward (CFNN)		feedforward (FFNN)		Error (%)			
	U (kV)	dU (grad)	U (kV)	dU (grad)	U (kV)	dU (grad)	N-R& CFNN		N-R& FFNN	
							U (kV)	dU (grad)	U (kV)	dU (grad)
P <sub>1</sub> ,Q <sub>1</sub>	6,0994	1,3168	6,0992	1,3158	6,0235	1,3186	0,00	0,07	1,24	0,21
	6,0484	0,3482	6,0488	0,3488	6,0164	0,3524	0,01	0,16	0,54	1,03
	6,0173	1,0240	6,0171	1,0236	6,0179	1,0231	0,00	0,04	0,01	0,05
	6,1035	2,0485	6,0153	2,0935	6,0162	2,3895	1,44	2,20	0,01	14,14
	6,0490	1,5948	6,0172	1,5987	6,0156	1,5276	0,53	0,24	0,03	4,45

**Table 3.** Comparative evaluation for Bus 2

Bus 2	Nyuton – Rafson (N-R)		cascadeforward (CFNN)		feedforward (FFNN)		Error (%)			
	U (kV)	dU (grad)	U (kV)	dU (grad)	U (kV)	dU (grad)	N-R& CFNN		N-R& FFNN	
							U (kV)	dU (grad)	U (kV)	dU (grad)
P <sub>2</sub> ,Q <sub>2</sub>	5,8299	-0,8551	5,8310	-0,8551	5,8388	-0,8516	0,02	-0,01	0,13	-0,40
	5,8031	-1,4863	5,8024	-1,4866	5,8315	-1,4764	0,01	-0,02	0,50	-0,69
	5,8322	-0,8432	5,8347	-0,8485	5,8327	-0,8436	0,04	-0,63	0,03	-0,58
	5,7594	-0,9535	5,8310	-0,9947	5,8312	-0,9982	1,24	-4,32	0,00	-0,35
	5,8364	-1,2123	5,8328	-1,2112	5,8312	-1,2167	0,06	-0,09	0,03	-0,45

**Table 4.** Comparative evaluation for Bus 3

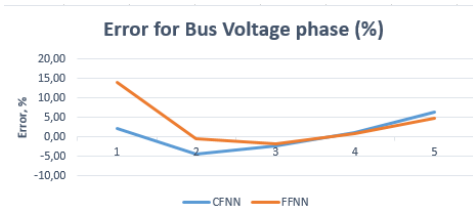
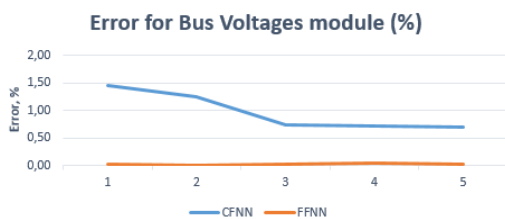
Bus 3	Nyuton – Rafson (N-R)		cascadeforward (CFNN)		feedforward (FFNN)		Error (%)			
	U (kV)	dU (grad)	U (kV)	dU (grad)	U (kV)	dU (grad)	N-R& CFNN		N-R& FFNN	
							U (kV)	dU (grad)	U (kV)	dU (grad)
P <sub>3</sub> ,Q <sub>3</sub>	5,9082	-1,0172	5,9086	-1,0177	5,9144	-1,0165	0,01	-0,04	0,10	-0,11
	5,8807	-1,5577	5,8808	-1,5577	5,9126	-1,5236	0,00	0,00	0,54	-2,19
	5,9118	-1,3089	5,9117	-1,3089	5,9122	-1,3084	0,00	0,00	0,01	-0,04
	5,8665	-1,2069	5,9103	-1,2356	5,9115	-1,2152	0,75	-2,38	0,02	-1,66
	5,9026	-1,6647	5,9116	-1,6626	5,9107	-1,6658	0,15	-0,13	0,02	-0,19

**Table 5.** Comparative evaluation for Bus 4

Bus 4	Nyuton – Rafson (N-R)		cascadeforward (CFNN)		feedforward (FFNN)		Error (%)			
	U (kV)	dU (grad)	U (kV)	dU (grad)	U (kV)	dU (grad)	N-R& CFNN		N-R& FFNN	
							U (kV)	dU (grad)	U (kV)	dU (grad)
P <sub>4</sub> ,Q <sub>4</sub>	6,1067	0,8933	6,1055	0,8949	6,0597	0,8550	0,02	0,18	0,75	4,46
	6,0299	0,4552	6,0920	0,4385	6,0585	0,4576	1,03	3,69	0,55	4,37
	6,0595	0,7764	6,0589	0,7765	6,0585	0,7739	0,01	0,01	0,01	0,32
	6,0158	1,4141	6,0584	1,4296	6,0607	1,4176	0,71	1,10	0,04	0,84
	6,0230	1,2547	6,0599	1,2894	6,0598	1,2937	0,61	2,76	0,00	0,33

**Table 6.** Comparative evaluation for Bus 5

Bus 5	Nyuton – Rafson (N-R)		cascadeforward (CFNN)		feedforward (FFNN)		Error (%)			
	U (kV)	dU (grad)	U (kV)	dU (grad)	U (kV)	dU (grad)	N-R& CFNN		N-R& FFNN	
							U (kV)	dU (grad)	U (kV)	dU (grad)
P <sub>5</sub> ,Q <sub>5</sub>	6,0756	0,5984	6,0748	0,5983	6,0589	0,5996	0,01	0,02	0,26	0,21
	6,1263	0,1526	6,0786	0,1565	6,0583	0,1987	0,78	2,53	0,33	27,01
	6,0591	0,8548	6,0598	0,8542	6,0591	0,8548	0,01	0,07	0,01	0,07
	6,0165	0,7169	6,0584	0,7619	6,0593	0,7985	0,70	6,27	0,01	4,80
	6,0884	0,8265	6,0592	0,8235	6,0578	0,8985	0,48	0,35	0,02	9,10



**Fig.5.** The graphical comparison between the Newton-Raphson method and the ANN-based algorithm results

## 2 Conclusion

The use of feedforward neural networks for calculating the power systems steady-state calculations is a worthy alternative to traditional calculation techniques.

Compared to classical methods, calculating the steady state of an already trained ANN requires insignificant computational and time resources, which is very important in operational control.

As the results of calculations have shown, in most cases, the calculation error is relatively small, which indicates a successfully trained and correctly formed ANN for a specific power systems schemes.

Of the two types of feedforward neural networks considered, when determining the voltage modulus, the feedforward neural network (maximum error - 0.75%) gives the best results, and when determining the angle of the cascadeforward neural network (maximum error - 6.3%).

At the same time, it should be noted that there is no possibility of universalizing the obtained feedforward neural networks for other EPS schemes.

## References

1. Idelchik V.I. *Elektricheskie sistemy i seti* [Electrical systems and networks]. Moscow: Energoatomizdat Publ., 1989, 592 p. (In Russian).
2. Ortega D., Reynboldt V. *Iteracionnye metody resheniya nelinejnyh sistem uravnenij so mnogimi neizvestnymi* [Iterative methods for solving nonlinear equation systems with many unknowns]. Moscow: World Publ., 1975, 560 p. (In Russian).
3. Gajendra Sahu and Kuldeep Swarnkar. Review of Reactive Power Optimization Using Evolutionary Computation Techniques. *Advance in Electronic and Electric Engineering*, 2014, no. 4, pp. 73–82.
4. Solopov R.V., Samulchenkov A.S. The use of a genetic algorithm for calculating the steady state of an electrical network. *Bulletin of Irkutsk State Technical University*. 2018. T. 22. № 2. C. 131–141. [DOI: 10.21285/1814-3520-2018-2-131-141](https://doi.org/10.21285/1814-3520-2018-2-131-141) (In Russian).
5. S. Pourya Hoseini Alinodehi, Sushil J. Louis, Sajjad Moshfe, Mircea Nicolescu. A Modified Steady State Genetic Algorithm Suitable for Fast Pipelined Hardware. 2017 IEEE Congress on Evolutionary Computation (CEC) 5-8 June 2017 [DOI:10.1109/CEC.2017.7969612](https://doi.org/10.1109/CEC.2017.7969612)
6. Draidi Abdellah; Labeled Djamel. Power flow analysis using adaptive neuro-fuzzy inference systems 2015 3rd International Renewable and Sustainable Energy Conference (IRSEC) 10-13 Dec. 2015 [DOI:10.1109/IRSEC.2015.7455102](https://doi.org/10.1109/IRSEC.2015.7455102)
7. Draidi Abdellah; Labeled Djamel Power flow analysis using adaptive neuro-fuzzy inference systems. 2015 3rd International Renewable and Sustainable Energy Conference (IRSEC) 10-13 Dec. 2015 [DOI:10.1109/IRSEC.2015.7455102](https://doi.org/10.1109/IRSEC.2015.7455102)
8. S. G. Ankaliki, S. G. Gollagi Power System Steady State Monitoring Using Artificial Neural Network. *Journal of Engineering and Technology*, Jan-Jun 2011, Vol 1, Issue 1, Page: 4-9
9. N Aishwarya, Manjula S Sureban. Analysis of Steady State Stability of Power System using Artificial Neural Network *International Journal of Scientific & Engineering Research* Volume 10, Issue 5, May-2019 ISSN 2229-5518
10. R. E. Bourguet, P. J. Antsaklis, "Artificial Neural Networks in Electric Power Industry," Technical Report of the ISIS (Interdisciplinary Studies of Intelligent Systems) Group, No. ISIS-94-007, Univ of Notre Dame, April 1994.
11. Krishna J, Srivastava L, Counterpropagation Neural Network for Solving Power Flow Problem, *International Journal of Intelligent Systems and Technologies*, Winter 2006, Page 57-62
12. Beale, M.H., Hagan, M.T. and Demuth, H.B. (2014) *Neural Network Toolbox™, User's Guide*. MATLAB®, MathWorks, Natick.
13. T.Kh. Nasyrov "Foundations of the general theory of normal and emergency modes of power systems." *Fan va Technology Publishing House* - 2015, 224 p. (In Russian).
14. Yu H. *Advanced Learning Algorithms of Neural Networks: PhD. dissertation.* / H. Yu. - USA: Graduate Faculty of Auburn University, 2011- 130 p.