

# Finite Element Method Combined with Neural Networks for Power System Grounding Investigation

Liviu Neamt, Oliviu Matei, Olivian Chiver  
Electrical, Electronic and Computer Engineering Department  
Technical University of Cluj-Napoca  
Cluj-Napoca, Romania

**Abstract**—Even in homogenous soil and for simple geometrical structure the analytical design of a grounding system is a complex and not very accurate procedure. Using Finite Element Analysis (FEA) it can perform a precise design for complex grounding systems but with important hardware resources and time consumption. This paper proposes a methodology for power system grounding design, directed to ensure the advantages of the FEA but without its disadvantages. This is realized by adding the function emulation using neural networks. The vertical rod, buried in inhomogeneous soil is the subject of this presentation. Consequently, the first step was to perform FEA for a large number of configurations: different types of vertical rods connected to the surface, buried at various depths in different double-layer soil structures. Then, the results have been interpreted through a multi-layer perceptron (MLP) with one hidden layer. A compromise between the number of inputs and precision have been tested, in order to define a minimum number of FEA required to obtain an acceptable grounding system design, i.e. a desired grounding resistance, for any combinations of the geometrical and material parameters. The validation of the methodology was done based on data reported in various research works.

**Keywords**—neural network; finite element analysis; power systems; grounding

## I. INTRODUCTION

Numerical simulation of the electromagnetic field lays on the basis of modern CAD in electrical engineering. Finite Element Method (FEM) is the most used tool for this and permits, not just the design, but also the optimization and the validation of the equipment behavior in the field. The main disadvantage consists in hardware resources and time needed for simulations and the lack of generalization, i.e. for every configuration it must be performed another FEA

The targeted sustainable smart grid concept, which ensures the continuity and quality of the energy supplies, must be realized in order to guarantee the safety of the human being and the installations. One of the key for achieving this is the power system grounding.

The professional design of power system grounding is conducted analytical in homogenous soil, for simple structures configurations, according to the theoretical computation of the electromagnetic field [1], [2] and again for more complex devices, but using simplified relations imposed by standards and regulations, [3-5]. For inhomogeneous soils, analytical relations are very complicate, if exist, inaccurate and

determinable for simplified structures, material parameters and variable behavior [1-5]. All these difficulties can be easily hurdle using FEA [6-11]. The problem here is that the results cannot be generalized, so for every configuration it means another simulation.

This paper intend to structure a methodology, based on FEA and neural network to generalize the FEA result, meaning the grounding resistance value, for any variation of the geometrical and material parameter of the base configuration. For an easy and logic presentation, the methodology is depicted using a usable grounding structure, i.e. a single vertical rod, with variable length, buried in double layer, horizontally layered soil and connected to the surface.

Therefore, an initial configuration, changed successively regarding the imposed variation limits for parameters (geometrical and material) will constitute the FEA models. In this step, a large number of models will be analyzed, asking for great hardware and time resources.

All the results, i.e. the grounding resistances, in terms of: rod length, the thickness of the first soil layer and the ratio between upper and inferior layer resistivity, will enter in the next stage, meaning the neural network generalization. Creation, optimization of the neural network, reducing the number of inputs required to maintain a desired precision are the goals of this last step.

As final result, for the above grounding system, the methodology offers, virtually instantaneously, a value of the grounding resistance for any combination of the material and geometrical parameters, offering in this way a key to a close optimum configuration.

## II. VERTICAL ROD GROUNDING STRUCTURE

Standard regulations provide minimum allowed cross section for different materials for electrodes and also suggest some recommendations for different configurations. Based on these, the basic configuration is a Zn coated steel cylindrical rod, length  $L = 1 \div 3$  m, diameter  $d = 2$  inches, buried at a depth  $h = 1$  m, in a double layered soil with resistivity,  $\rho_{up}$  and  $\rho_{inf}$ , as depicted in Fig. 1.

The FEA for these configurations are presented in details in [12]. The results kept in these step, are the grounding resistances for 300 configurations generated by variations of the next variables:  $L = 1 \div 3$  m, the thickness of the upper layer  $t_{up} = 0 \div \infty$ , the ratio between upper and inferior layer

resistivity  $\rho_{up}/\rho_{inf} = 0.2 \div 5$ .

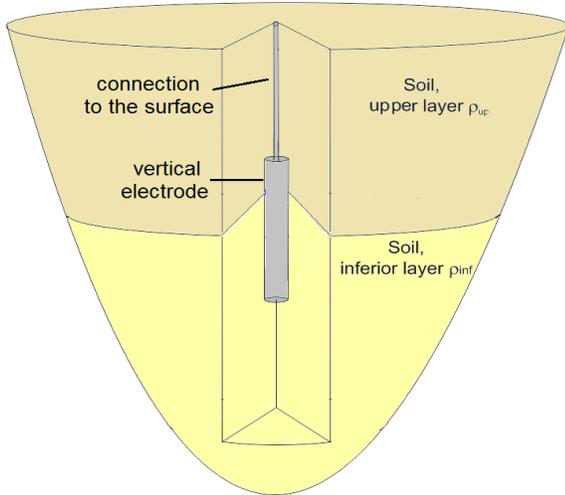


Fig. 1. Vertical grounding electrode with connection to the surface

### III. NEURAL NETWORK STRUCTURE

The grounding resistance has been emulated by building a neural network. It is well known the capability of neural networks to approximate functions, a concept called “regression” [13]. For such tasks, a simple multi-layer perceptron has been proved as a good choice, according to Schürmann [13]. However, the complexity of the network is crucial for its behavior. A trade-off regarding its size is always needed. A small architecture may prove inefficient to approximate the desired function, whereas a larger network may over-learn the training set, being unable to generalize on extra input data. Baum and Haussler give some principles in [14].

Therefore several experiments have been run for choosing:

- ✓ The right size of the neural network;
- ✓ The transfer function;
- ✓ The learning algorithm;
- ✓ The training epochs.

The data consisted of 300 sets, with: vertical electrode length, the thickness of the upper layer and the ratio between upper and inferior layer resistivity as inputs for the network and the resistance as the desired output. For avoiding the over-training, which is a common problem in neural design (see [15] and [16]), we have set aside 20% of the data for cross validation and other 20% for testing. This means that the training has been done using only 60% of the available data. For a better learning, the data has been shuffled as suggested by Ikegaya in [17].

A fully-connected multi-layer perceptron has been used, like in Fig. 2.

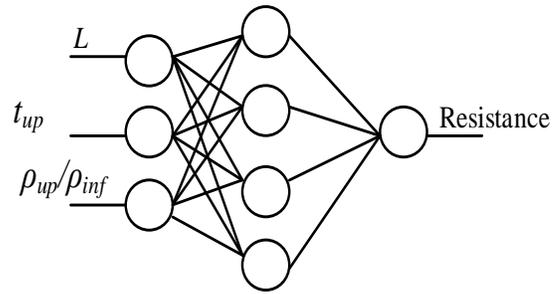


Fig. 2. The general architecture of the neural network tested within the experiments

The neural network has:

- ✓ three inputs, each corresponding to: vertical electrode length,  $L$ , the thickness of the upper layer,  $t_{up}$ , and the ratio between upper and inferior layer resistivity,  $\rho_{up}/\rho_{inf}$ ;
- ✓ a hidden layer with variable number of neurons;
- ✓ and one output – the desired grounding resistance.

As transfer function we have tested the sigmoid and the hyperbolic tangent. The sigmoid function is defined as:

$$S = \frac{1}{1 + e^{-t}} \quad (1)$$

and looks like in Fig. 3.

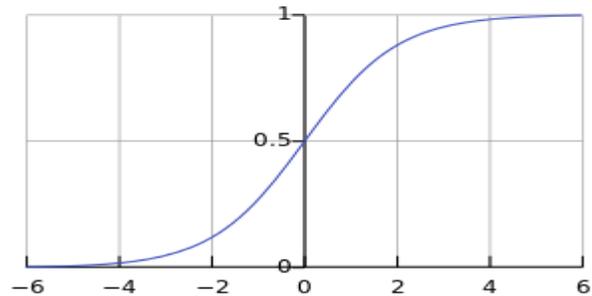


Fig. 3. The shape of the sigmoid function

The hyperbolic tangent ( $\tanh$ ) has a shape like in Fig. 4.

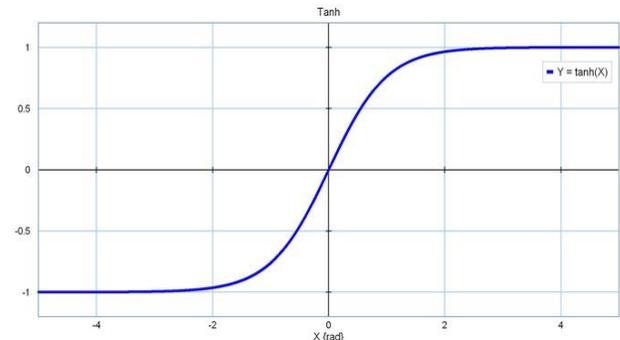


Fig. 4. The shape of  $\tanh$  function

Tanh is defined as:

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2)$$

The learning rules considered during our experiments were: step, momentum, quickprop, delta-bar-delta, conjugate-gradient, Levenberg-Marquardt and resilient backpropagation (rprop), as defined in [18].

Gradient descent learning rules (*Step*) estimate the way to the minimum error of the network. The algorithm searches for the descending slope of the function with various steps. Tweaking the steps is an important aspect of the approach, as smaller steps would result in longer times to reach the optimum, whereas larger steps could overshoot the bottom, causing it to rattle or even diverge.

The *Momentum* provides the gradient descent with some inertia, moving downwards based on some average estimates of that direction.

The *Quickprop* implements Fahlman's quickprop algorithm. It is a gradient search procedure, however very fast in various problems. It is also very accurate. It makes use of the second order derivative for accelerating the search, unlike the *step* or *gradient* methods.

*Delta-Bar-Delta* is a search method which makes use of the sign of the current update with respect to the previous one. If the two updates are both of the same sign, it increases the learning rate linearly. If the updates have different signs, this is an indication that the weight has been moved too far. When this happens, the learning rate decreases geometrically to avoid divergence.

*Conjugate gradient* is also a second order method (like *quickprop*), which means that it approximates the second derivatives of the performance surface to determine the weight update.

The *Levenberg-Marquardt* (LM) algorithm is one of the most appropriate higher-order adaptive algorithms known for minimizing the minimum square root (MSE). It is also a second order method. The LM makes use of the so called Gauss-Newton approximation that keeps the Jacobian matrix and discards second order derivatives of the error (see [19]).

*Resilient backpropagation* (*Rprop*) is able to outperform most other local (i.e., first-order) learning techniques because it is able to adapt the step sizes of each individual weight instead of using the same step size for all weights. A detailed description of the algorithm can be found at [21].

#### IV. RESEARCHING THE BEST NEURAL ARCHITECTURE

The first experiments were meant to determine the right learning algorithms. The number of hidden neurons was set to 10, and sigmoid as transfer function, as this seems to be a reasonable architecture, according to [12].

The results are concluded in table I. Training MSE are the mean square errors of the neural network in the training phase. The training epochs represents the number of epochs needed for training. This is determined when there are no

improvements in the training MSE and before the cross validation starts increasing, as this is the moment when the neural network starts overlearning the training set and behaving poorly on a test set.

“Step”, “Quickprop”, “Deltabar” and “Rprop” present the training MSE, number of training epochs and test MSE when the network uses learning rule: step, quickprop, deltabar, respectively rprop; columns “Mom.”, “Conj. grad.”, “L.M.” show the values when the network is trained with momentum; conjugate gradient, respectively Levenberg-Marquardt.

TABLE I. THE RESULTS OF THE EXPERIMENTS WITH 10 NEURONS IN THE HIDDEN LAYER, FOR DETERMINING THE PROPER LEARNING RULE

	Step	Mom.	Quick-prop	Delta-bar	Conj. grad.	L.M.	Rprop
Training MSE	0.47	0.475	0.485	0.275	0.47	<b>0.266</b>	<b>0.266</b>
Training epochs	142	3000	3000	1720	243	133	<b>123</b>
Test MSE	0.50	0.482	0.484	0.363	0.481	<b>0.345</b>	0.360

As observed in the table I, the best results, meaning lowest training and test MSE, are obtained using deltabar, Levenberg-Marquardt and rprop as learning rules. The number of training epochs (and thus the learning time) is the best for Levenberg-Marquardt and rprop and significantly higher for deltabar. However, as at this stage of experiments the quality is more important than the time, further experiments have been made taking into account these three algorithms.

We have run (meaning training and testing) the neural network having 5, 10 and 15 neurons in the hidden layer. The number of inputs and outputs remained the same throughout all our experiments.

The results are synthesized in table II. The second column shows the number of neurons in the hidden layer, namely 5, 10 and 15; the next columns presents the training MSE, the number of training epochs and the test MSE when the network uses delta-bar-delta, Levenberg-Marquardt, respectively rprop for learning.

TABLE II. THE RESULTS OF THE EXPERIMENTS FOR DETERMINING THE PROPER NUMBER OF NEURONS IN THE HIDDEN LAYER

	No. of hidden neurons	Deltabar	Levenberg-Marquardt	RProp
Training MSE	5	0.2966	<b>0.273</b>	0.2787
Training epochs	5	1836	<b>34</b>	35
Test MSE	5	0.3543	<b>0.3457</b>	0.35191
Training MSE	10	0.2757	<b>0.2661</b>	<b>0.2661</b>
Training epochs	10	1720	133	<b>123</b>
Test MSE	10	0.3631	<b>0.3455</b>	0.3607
Training MSE	15	0.2932	0.2674	0.2656
Training epochs	15	1730	52	<b>48</b>
Test MSE	15	0.3628	<b>0.3407</b>	0.3478

Based on the results shown in table II, a neural network with 5 hidden neurons and L.M as learning rule would be very suited. However, the experiments have shown that it is very unstable, even in the training phase, e.g. the learning curve and the cross validation curve cross each other many times, as shown in Fig. 5.

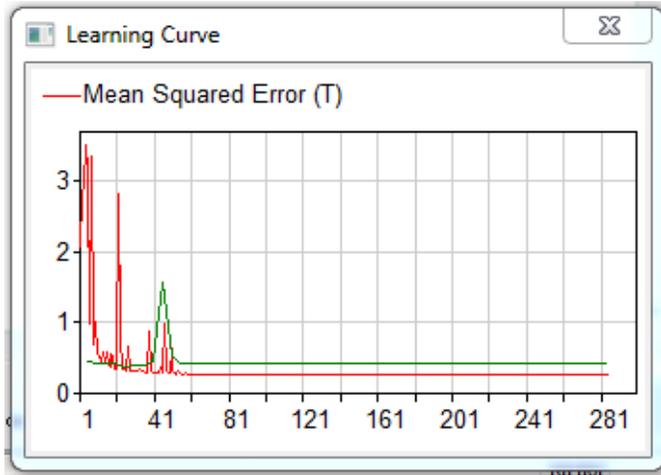


Fig. 5. The MSE for training (red) and cross validation (green) for a neural network with 5 hidden neurons

Therefore so far, the network with 10 hidden neurons and L.M. as the learning rule has the best behavior.

The next experiments were meant to determine the proper transfer function. The same set of tests have been run, but using *tanh* as transfer function, rather than sigmoid. The results are concluded in table III.

The lines represent the training mean square error, respectively training error, and cross validation MSE, respectively cross validation error, the number of training epochs and the testing MSE, respectively testing error. The columns represent the number of neurons in the hidden layer, the results for deltabar, Levenberg-Marquardt and rprop learning rules.

TABLE III. THE RESULTS OF THE EXPERIMENTS USING *TANH* AS TRANSFER FUNCTION

	No. of hidden neurons	Deltabar	Levenberg-Marquardt	RProp
Training MSE	5	0.0119	0.01	0.012
Training error	5	13.012%	12.50%	14.25%
CV MSE	5	0.0101	0.005	0.01
CV error	5	14.04%	13.14%	13.48%
Training epochs	5	1622	56	1604
Test MSE	5	0.14	0.002	0.005
Test error	5	79.25%	14.90%	19.23%
Training MSE	10	0.04	0.002	0.002
Training error	10	35%	<b>7.31%</b>	8.28%
CV MSE	10	0.02	0.003	0.003
CV error	10	25%	<b>7.77%</b>	7.42%
Training epochs	10	414	78	1245
Test MSE	10	0.04	0.0009	0.001
Test error	10	61.30%	<b>7.88%</b>	10.85%
Training MSE	15	0.03	0.01	0.0055
Training error	15	36.61%	20.06%	10.96%
CV MSE	15	0.0328	0.01	0.003
CV error	15	34.91%	13.00%	8.56%
Training epochs	15	324	268	1711
Test MSE	15	0.0099	0.01	0.004
Test error	15	29.76%	31.19%	16.88%

The chart in Fig. 6 concludes the number of training epochs for neural networks with 5, 10 and 15 hidden neurons, and sigmoid, respectively *tanh* as transfer functions, for L.M. and Rprop learning rules. The Delta-bar-delta has been skipped as the training times are much higher.

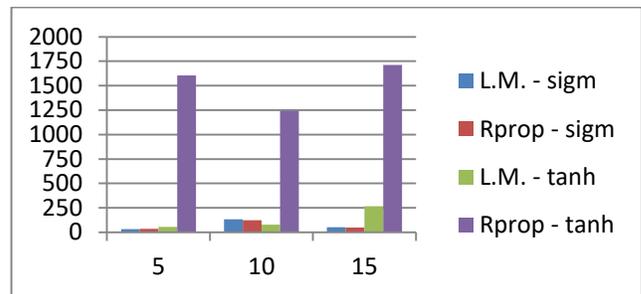


Fig. 6. The number of training epochs for various configurations of the neural network

The training MSE are depicted in Fig. 7.

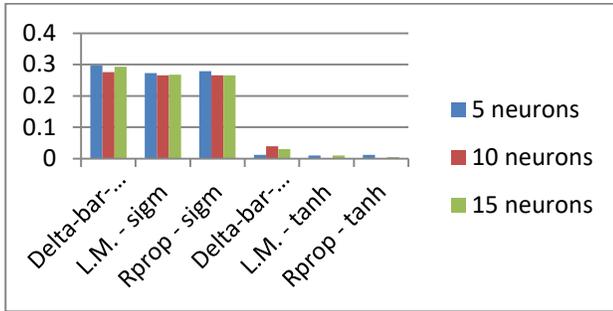


Fig. 7. The training MSE for the researched neural configurations

The MSE of the tests are depicted in Fig. 8

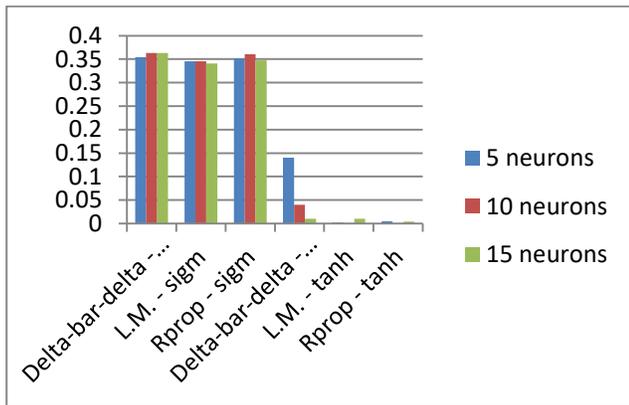


Fig. 8. The testing MSE for the researched neural configurations

From both Fig. 7 and Fig. 8, it is obvious that *tanh* is a better option for the transfer function.

Fig. 9 displays the errors for training, cross-validation and tests. Please notice that there are two training errors for Delta-bar-delta, respectively for 5 and 10 hidden neurons very high (79.25%, respectively 61.30%). But for the sake of clearance, we preferred to scale down to 42% the Y axis of the chart, so that the important errors are visible too.

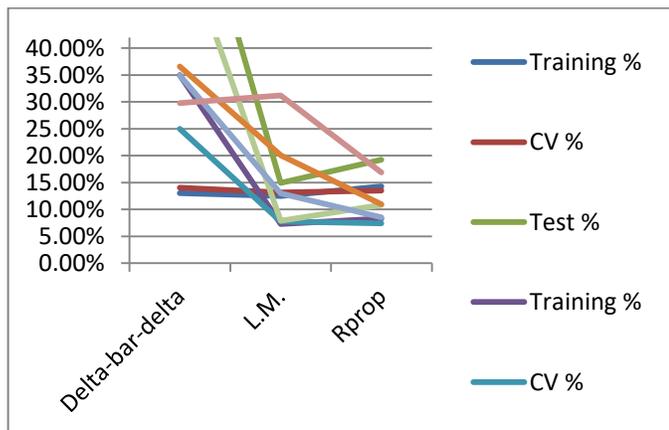


Fig. 9. The training, CV and test errors for the neural configuration with *tanh* as transfer function

## V. EXPERIMENTAL RESULTS

At this point we have decided that a neural network with 3 inputs, 10 hidden neurons and one output, using *tanh* as transfer function, respectively L.M. as learning rule is the best option in terms of quality (lowest error), learning time (shortest training time) and computation resources. We have done all these tests using a set of 300 data, out of which 60% (180 records) have been used for training, 20% (60 records) for cross validation and 20% (60 records) for testing. The records belonging to one set or the other have been chosen at random.

The question that rises at this stage is: what is the minimum size of the training set so that the quality of the output is still good (e.g. the error is less than 10%). For that, we have varied the size of the training set from 10% (meaning 30) to 60% (180 records). The training error, cross validation error and test error are depicted in Fig. 10.

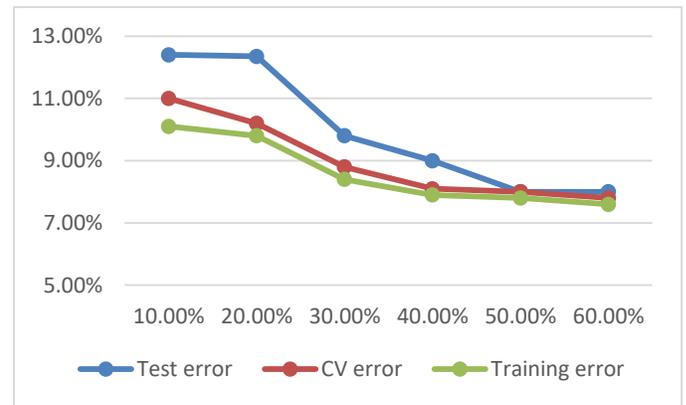


Fig. 10. The errors of the neural network varying the size of the training set

It is obvious that the errors are below 15% in all cases, however, the errors are less than 10% when the size of the training set is above 100 records (30% of the current available data).

This means that 100 FEA must be realized in order to have enough data to be able to generalize the results for all input data in designing process, with a guaranteed error below 10%.

For this grounding system configuration, a value of the grounding resistance for any combination of the material and geometrical parameters is obtained very fast, so the optimization process could be started up. Usually, the soil structure is known and the length of the electrode has to be chosen.

## VI. CONCLUSIONS

This article presented the experiments and the results obtained for configuring a neural network capable of emulating the grounding resistance of a vertical electrode buried in a two-layered soil.

The presentation was structured on two horizontally layered soil, which for zero or infinite thickness of the upper layer simulate also the homogenous soil. The number or/and

the arrangements of the layers could be modified, or/and the configuration of the grounding system could be altered (e.g. horizontally electrode, more simple electrodes, or a complex structure as it is for power substations) so the generality could be pushed further.

Based on methodology depicted above we suggest that FEA combined with neural network analysis may be considered as the best computer aided investigation, not only for power system grounding systems, but also for many other systems used in engineering.

#### REFERENCES

- [1] O. Centea, Grounding devices from electrical installations, Prizele de pamant din instalatiile electrice, Ed. Academiei, Bucharest, 2006, pp.89-381.
- [2] J. He, R Zeng and B Zhang, Methodology and technology for power system grounding, Singapore: J. Wiley & Sons Oxford, 2013, pp.11-19.
- [3] IEEE 80-2000 Standard, Guide for safety in ac substation grounding, 2000.
- [4] IEEE 142-2007 Standard, Grounding of industrial and commercial power systems, 2007.
- [5] 1 RE-1p 30/2004 Guide book for design and execution of power system grounding, Indreptar de proiectare si executie a instalatiilor de legare la pamant, 2004.
- [6] F. P. Dawalibi, J. Ma and R. D. Southey, "Behaviour of Grounding Systems in Multilayer Soils", IEEE Trans. Power Delivery, Vol. 9, No. 1, pp. 334-342, January 1994.
- [7] J. A. Guemes and F. E. Hernando, "Method for Calculating the Ground Resistance of Grounding Grids FEM", IEEE Trans. on Power Delivery, vol. 19, no. 2, pp. 595-600, April 2004.
- [8] P. Hajebi, A. A. Heidari and A. Mirzaei, "Resistance to Earth of Grounding Grids in Two-layer soil structure using FEM and GA", PIERS Proceedings, Xi'an, China, March 22-26, 2010.
- [9] F. Rodriguez, J. A. Guemes, J. M. Ruiz and F. E. Hernando, "Determination of the ground Resistance and Distribution of Potentials in Grounding Grids using FEM", IEEE Transactions On Power Delivery, vol. 21, no 3, pp. 1261-1266, July 2006.
- [10] M. A. Salam, K. M. Jen and M.A. Khan, "Measurement and simulation of grounding resistance with two and four mesh grids," Power Electronics and Drive Systems (PEDS), 2011 IEEE Ninth International Conference on , pp.208-213, Dec. 2011.
- [11] I. A. Letia, O. Matei, "Hybrid Neural Approach in Manura Robot Localization", Advances in Electrical and Computer Engineering, vol. 5, no. 12, pp. 5-9, 2005.
- [12] L. Neamt, O. Chiver, C. Barz, C. Cristinel, Z. Erdei, "Considerations about power system grounding for different soil structure", Proceedings of the International Conference and Exposition on Electrical and Power Engineering, Iasi, Romania, 16-18 October 2014,.
- [13] J. Schürmann, Pattern classification: a unified view of statistical and neural approaches. Wiley, New York, 1996.
- [14] E. B. Baum, and D. Haussler. "What size net gives valid generalization?." Neural computation vol. 1, no.1, pp. 151-160, 1989.
- [15] A. Krogh, J. Vedelsby. "Neural network ensembles, cross validation, and active learning." Advances in neural information processing systems, pp. 231-238, 1995.
- [16] R. Setiono, "Feedforward neural network construction using cross validation." Neural Computation vol. 13, no. 12, pp. 2865-2877, 2001.
- [17] Y. Ikegaya, A. Gloster, et al. "Synfire chains and cortical songs: temporal modules of cortical activity." Science vol. 304 no. 5670 pp. 559-564, 2004.
- [18] L. C. Principe, N. R. Euliano, and W. C. Lefebvre. Neural and adaptive systems: fundamentals through simulations, Wiley, 2000.
- [19] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian learning." Proceedings of the 1997 international joint conference on neural networks. vol. 3. Piscataway: IEEE, 1997.
- [20] C. M. Bishop, Neural networks for pattern recognition, Oxford University Press, 1995.
- [21] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," Proceedings of the IEEE International Conference on Neural Networks (ICNN), pp. 586-591, San Francisco, 1993.