Electric Power Systems Research 94 (2013) 113–121

Contents lists available at SciVerse ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Estimation of seasonal variation of ground resistance using Artificial Neural Networks

Fani E. Asimakopoulou[a], George J. Tsekouras[b], Ioannis F. Gonos[a,*], Ioannis A. Stathopulos[a]

[a] School of Electrical and Computer Engineering, National Technical University of Athens, 9 Iroon Polytechniou Street, GR-15780, Zografou Campus, Athens, Greece
[b] Department of Electrical Engineering and Computer Science, Hellenic Naval Academy, Terma Hatzikiriaku, GR-18539, Piraeus, Athens, Greece

ARTICLE INFO

Article history:
Received 23 February 2012
Received in revised form 13 July 2012
Accepted 19 July 2012
Available online 10 August 2012

Keywords:
Soil resistivity measurements
Ground resistance
Artificial Neural Networks
Back propagation algorithm

ABSTRACT

Objective of this paper is the development of a methodological approach for estimating the ground resistance by using Artificial Neural Network. The value of the ground resistance greatly depends on the grounding system and the properties of the soil, where the system is embedded. Given that the value of soil resistivity fluctuates during the year, the ground resistance does not have one single value. The approach proposed in this paper, takes advantage of the capability of Artificial Neural Networks (ANNs) to recognize linear and non-linear relationships between various parameters. By taking into account measurements of resistivity and rainfall data accrued for previous days, the ground resistance is estimated. On that purpose ANNs have been trained and validated by using experimental data in order to examine their ability to predict the ground resistance. The results prove the effectiveness of the proposed methodology.

1. Introduction

Purpose of any grounding system is to ensure safe and proper operation of the electric installation by allowing rapid identification and clearing of faults. Low ground resistance at individual structures improves safety and reduces back flashover stress from lightning surge currents.

The ground resistance (R_g) is defined by the size of the grounding system and soil resistivity (ρ), within which the grounding system is embedded. The value of soil resistivity varies significantly with location, depending on the nature of the soil, the amount of salts dissolved in it, the moisture content, the temperature and the compactness of the soil. Additionally, soil resistivity of the upper soil layer is subjected to seasonal variation due to ice or drought [1,2]. Since these parameters vary throughout the year, the grounding system cannot be characterized by a single value of ground resistance [3–6]. Therefore, these values should be monitored on a yearly basis, a time-consuming and cost-demanding task.

At this point an approach based on Artificial Neural Networks (ANNs) can be useful, since ANNs can model relationships between quantities without requirement of knowledge of the exact formula among them.

The aim of this paper is the study, analysis and modeling of changes in ground resistance of grounding systems over time, using Artificial Neural Network techniques. The results are based on extended experimental measurements of existing grounding system arrangements, throughout the year.

So far, ANNs have been successfully used by Salam et al. [7] for modeling and predicting the relationship between the length of the buried electrode and the grounding resistance. Amaral et al. [8] successfully attempted to map the relationship among the soil resistivity, grounding resistance, frequency, and current peak. This paper is complementary to the research presented in [9–12], in an attempt to delve into the problem and its parameters by using different training algorithms and by using an extended set of input data.

The rest of the paper is organized as follows: Section 2 refers to the experimental procedure of the measurements (ρ, R_g, rainfall). In Section 3 the ANN training algorithms are presented (in Appendix the respective mathematical base is shown), the training procedure is described and the results are presented. Moreover, a sensitivity analysis regarding the influence of the ANN's parameter selection on the performance of training algorithm is performed. Finally, Section 4 concludes the paper.

2. Soil resistivity and ground resistance measurements

The variation of depth method, the two-point method and the four-point method are the methods for measuring the soil resistivity. Among them the four-point method (Wenner) is the most

* Corresponding author. Tel.: +30 2107723539; fax: +30 2107723504.
E-mail addresses: faniassim@yahoo.com (F.E. Asimakopoulou), tsekouras_george_j@yahoo.gr, tsekouras@snd.edu.gr (G.J. Tsekouras), igonos@ieee.org (I.F. Gonos), stathop@power.ece.ntua.gr (I.A. Stathopulos).

0378-7796/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.epsr.2012.07.018

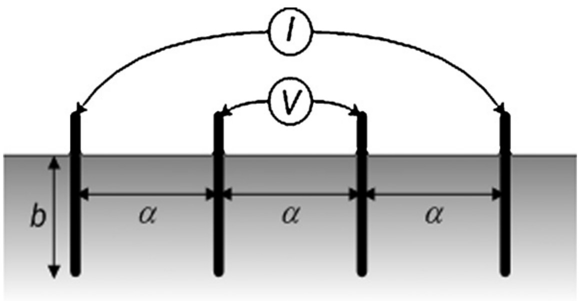

Fig. 1. Wenner method for measurement of apparent resistivity.

accurate in practice [2]. Within the scope of our experiment the Wenner method has been implemented.

The measurements of the soil resistivity were conducted in the area of Athens from October up to July [5,10], whereas the meteorological data were provided by the National Meteorological Authority of Hellas.

As shown in Fig. 1, four electrodes 45 cm in length are driven in line, in a depth b at equal distances α from each other. A test current (I) is injected at the two terminal electrodes and the potential (V) between the two middle electrodes is measured. The ratio V/I gives the apparent resistance R (in Ω). The apparent soil resistivity (ρ) is given by the following formula [2]:

$$\rho = \frac{4 \cdot \pi \cdot a \cdot R}{1 + 2 \cdot a/\sqrt{a^2 + 4 \cdot b^2} - a/\sqrt{a^2 + b^2}} \tag{1}$$

Measurements have been carried out on a 40 m line for two values of α (1 and 2 m) (Fig. 1) [5,10].

The ground resistance of a vertical rod, driven 1.5 m into the soil and having a diameter of 16 mm was measured according to the fall of potential method and the 62% rule [2]. The distance between the current electrode and the electrode being tested is 40 m, while the potential electrode is placed 24 m away from the electrode under test (Fig. 2).

The measurements were repeated at scheduled time intervals.

In Figs. 3 and 4 the seasonal variation of the resistivity for different distances between the four electrodes as well as the ground resistance are presented.

3. Artificial Neural Network methodology for the estimation of ground resistance

ANNs constitute a useful tool in the field of establishing relationships between quantities, that otherwise would have been difficult to model. A typical ANN is composed by three layers, the input, the

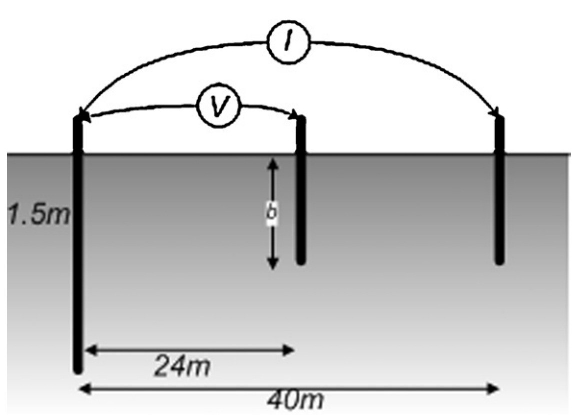

Fig. 2. The fall of potential method for measurement of ground resistance.

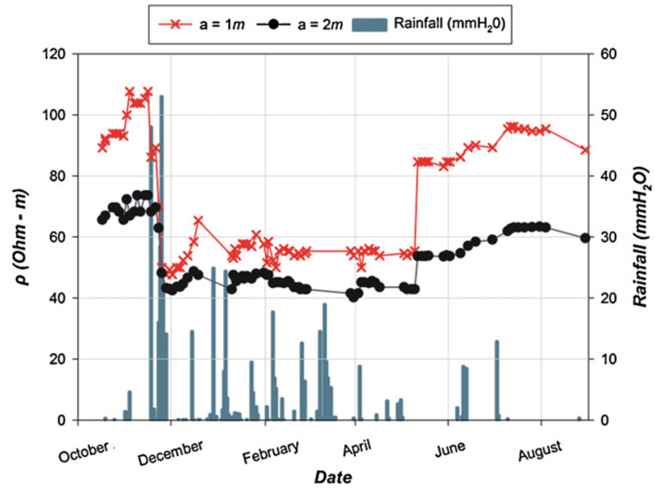

Fig. 3. Seasonal variation of the soil resistivity for distance between the electrodes 1 m and 2 m.

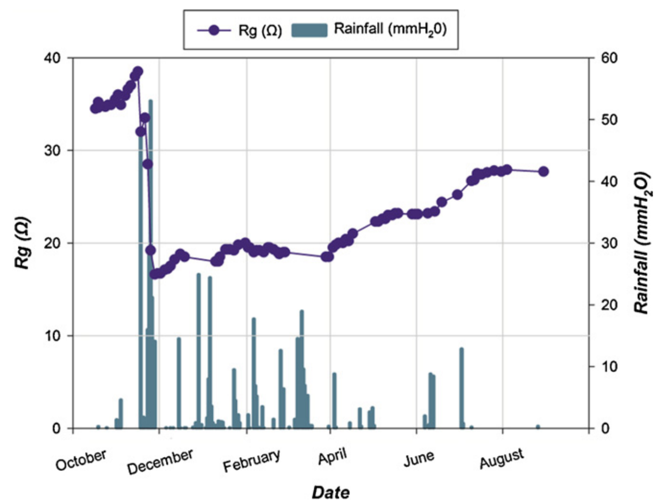

Fig. 4. Seasonal variation of the ground resistance.

hidden and the output layer. In Fig. 5 a schematic diagram of the ANN structure of our problem is presented. The input layer (input vector $(I_1 ... I_5)$) comprises the apparent soil resistivity measurements ρ_a (in Ωm) for electrode distances at a = 1 m and a = 2 m, the average rainfall during the preceding week, the rainfall during the day on which the ground resistance is estimated (in mm) and the average resistance during the preceding week (in Ω). The output layer (output variable) of the ANN is the ground resistance (in Ω).

The ground resistance of the rod is estimated by applying the methodology presented in Fig. 6.

Before proceeding to the training of the ANN, the input data are normalized in order to achieve convergence and avoid saturation problems of the algorithm according to the expression:

$$\hat{x} = a + \frac{b - a}{x_{\max} - x_{\min}}(x - x_{\min}) \tag{2}$$

where $\hat{x}$ is the normalized value for variable x, $x_{\min}$ and $x_{\max}$ are the lower and the upper values of variable x, a and b are the respective values of the normalized variable.

Following the experimental data are divided into three sets:

- The training set (53 cases), which is used for training until the network has learned the relationship between the inputs and the output.

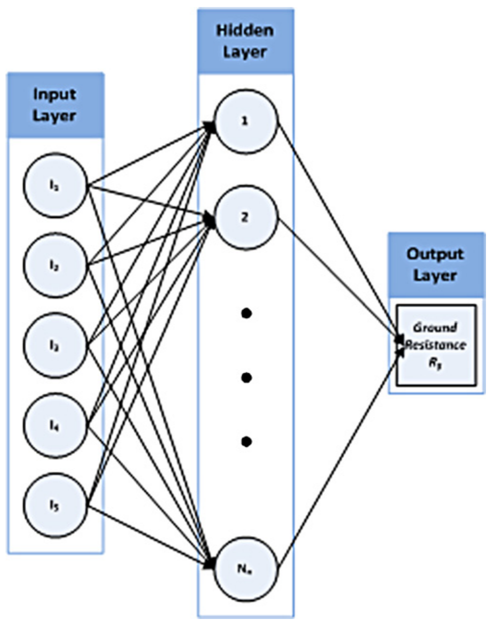

Fig. 5. ANN structure [11].

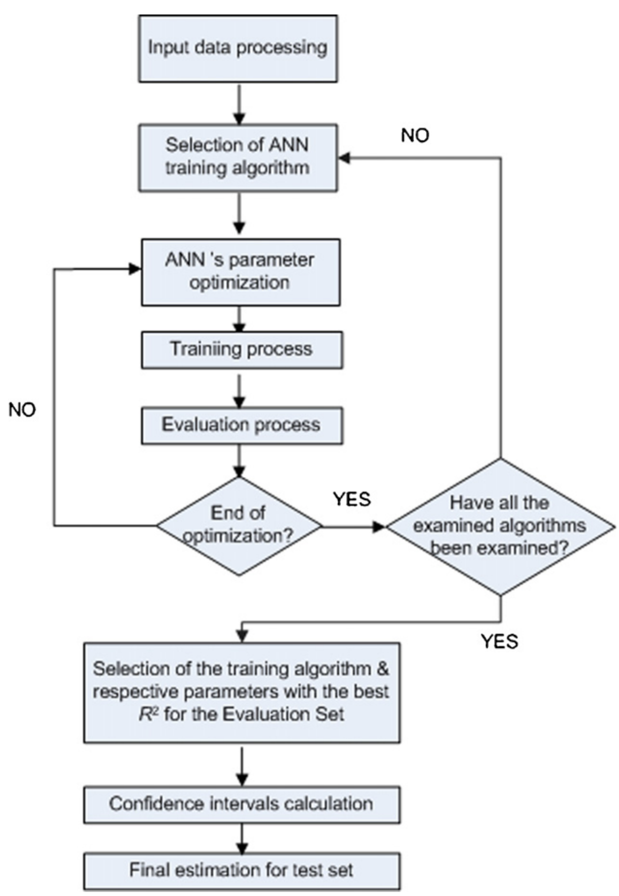

Fig. 6. Flowchart of the ANN methodology for the estimation of the ground resistance [11,12].

- The evaluation set (14 cases), which is used for the selection of the ANN parameters (number of the neurons in the hidden layer, type and parameters of the activation functions, learning rate, momentum term). The parameters are selected so that the maximum correlation index (R^2) between the actual and the estimated values for the evaluation set is achieved.
- The test set (10 cases) which verifies the generalization ability of the ANN by using an independent data set.

The ANN is trained by applying the back propagation algorithm and its variations, which are presented in Fig. 7 and analyzed in Appendix. During the execution of the training algorithm the free parameters (weights) of the network are adjusted in order for the average error function between the estimated and the actual value to be minimized. The average error function for all N patterns is given by (3):

$$G_{av} = \frac{1}{2N}\sum_{n=1}^{N}\sum_{j\in C}(d_j(n) - y_j(n))^2 \tag{3}$$

where C is the set of neurons, $d_j(n)$ the desirable output and $y_j(n)$ the actual output of the j-neuron.

The weights are adjusted by random presentation of every input vector (stochastic mode) according to the following criteria:

(1) the stabilization of the weights (4)
(2) the minimization of the error function (5) and
(3) the maximum number of epochs criterion (6),

which are respectively described by the following expressions:

$$\left|w_{kv}^{(l)}(ep) - w_{kv}^{(l)}(ep-1)\right| < \text{limit}_1, \forall k, v, l \tag{4}$$

$$\left|\text{RMSE}\ \ (ep) - \text{RMSE}\ \ (ep-1)\right| < \text{limit}_2 \tag{5}$$

$$ep \geq \max\_\text{epochs} \tag{6}$$

where $w_{kv}^{(l)}$ is the weight between l-layer's k-neuron and $(l-1)$-layer's v-neuron, $\text{RMSE} = \sqrt{1/(m_2 \cdot q_{out})\sum_{m=1}^{m_2}\sum_{k=1}^{q_{out}} e_k^2(m)}$ is the root mean square error of the evaluation set with m_2 members and q_{out} neurons of the output layer (in this case $q_{out} = 1$), max_epochs is the maximum number of the epochs.

Two variations have been applied for each training algorithm. In the first one (a) all the above criteria are applied, whereas in the second variation (b) only the first and the third criterion are applied.

The parameters of each ANN algorithm are optimized according to the procedure described in [9–12]. These parameters are: the number of neurons of the hidden layer, the formula and parameters of the activation function of the hidden and output layer, and various other parameters depending on the training algorithm.

After optimizing the parameters of every training algorithm, the one, which presents the highest correlation index (R^2) between the experimental and the estimated values of ground resistance for the evaluation set, is selected. It is noted that:

$$R^2 = r_{y-\hat{y}}^2 = \frac{\left(\sum_{i=1}^{n}((y_i - \bar{y}_{real})\cdot(\hat{y}_i - \bar{y}_{est}))\right)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_{real})^2 \cdot \sum_{i=1}^{n}(\hat{y}_i - \bar{y}_{est})^2} \tag{7}$$

where y_i is the experimental value of the ground resistance, $\bar{y}_{real}$ the mean experimental value of the respective data set, $\hat{y}_i$ the estimated value, $\bar{y}_{est}$ the mean estimated value of the data set, n the population of the set.

In Table 1 the optimized parameters, activation functions for the hidden (f_1) and the output layer (f_2) of each variation are presented. For all the variations of the training algorithm, the maximum number of epochs is selected to be equal to 7000. In the same table, the average error (G_{av}) and the correlation (R^2) between the estimated and the measured values of ground resistance for the evaluation and the test set are tabulated.

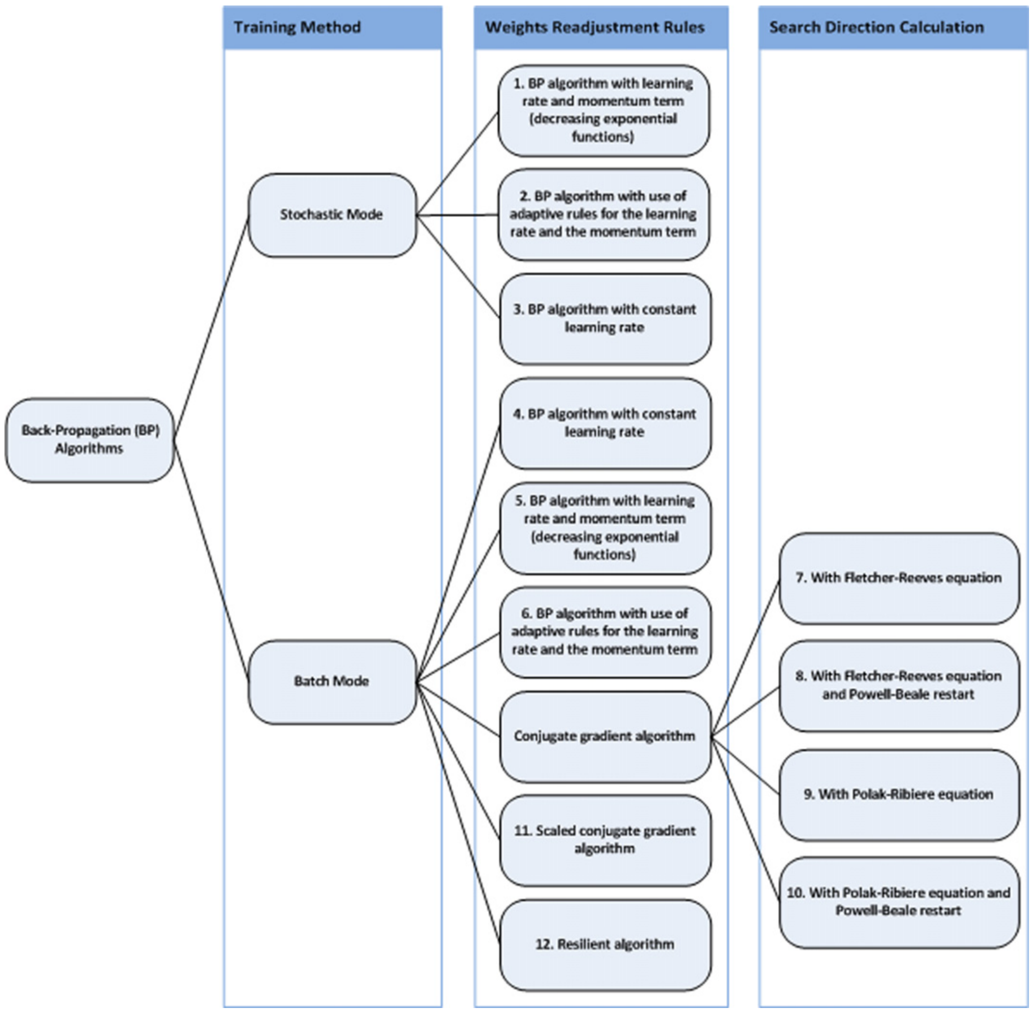

Fig. 7. Training algorithms.

From Table 1 it can be easily observed that the training algorithm which provides the highest correlation between the estimated and the measured data for the evaluation set is the batch mode conjugate gradient algorithm with Fletcher–Reeves equations with two termination criteria (7b).

The performance of the ANN for the optimum combination of parameters for training algorithm 7b is presented in Table 2 where the experimental values of the ground resistance and those estimated by the ANN for the test set are tabulated. In the same table the mean absolute percentage error (MAPE) given by (8) is recorded.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \qquad (8)$$

where y_i is the actual value and $\hat{y}_i$ is the estimated value of the ground resistance.

In Figs. 8 and 9 the measured and the estimated values of the ground resistance for the best training algorithm for the evaluation and the test set along with the confidence intervals with 5% probability are presented.

Since algorithm7b has the best performance among the tested algorithms, whereas algorithm 2b has the best performance among the stochastic training algorithms, a sensitivity analysis is performed in order to study the influence of parameters such as the initialization mode of the synaptic weights and the influence of the number of epochs to the performance of each algorithm.

In Table 3 the variation of the correlation index between measured and estimated values of the ground resistance for consecutive applications of the training algorithm (resulting in difference initialization of the synaptic weights) is tabulated. From the results it can be stated that there is no evident influence on the performance of each training algorithm.

In Table 4 the influence of the number of epochs on the performance of the ANNs trained with algorithms 2b and 7b is examined. On that purpose the number of epochs varies from 50 to 10,000. As the number increases, the correlation between measured and

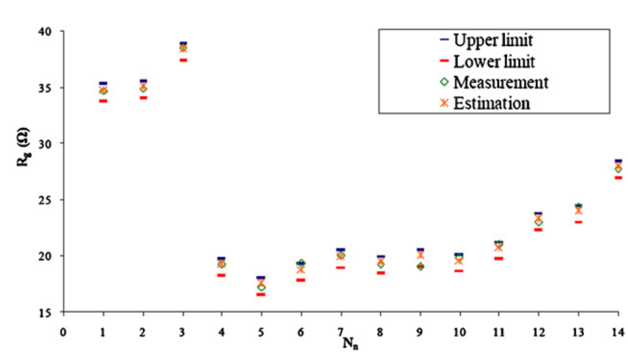

Fig. 8. Experimental and estimated values of the ground resistance for the evaluation set for batch mode conjugate gradient algorithm with Fletcher–Reeves equations (algorithm 7b).

Table 1
Parameters of the training algorithms, average error and correlation between the estimated and measured values of ground resistance. (Symbols of parameters are explained in Appendix.)

Algorithm	N_n	Algorithm parameters	Activation functions	$Gav \times 10^{-3}$ evaluation set	$Gav \times 10^{-3}$ test set	R^2 of the evaluation set	R^2 of the test set
1a	2	$\alpha_0 = 0.9$ $T_\alpha = 1400$ $\eta_0 = 0.7$ $T_\eta = 1000$	$f_1(x) = 1/(1+\exp(-1.9x))$ $f_2(x) = 0.3x$	0.52888	0.36193	0.99196	0.99533
1b	18	$\alpha_0 = 0.5$ $T_\alpha = 1500$ $\eta_0 = 0.9$ $T_\eta = 1600$	$f_1(x) = \tanh(1.9x)$ $f_2(x) = 1/(1+\exp(-0.4x))$	0.31276	0.35950	0.99533	0.99612
2a	23	$\alpha_0 = 0.4$ $T_\alpha = 1300$ $\eta_0 = 0.7$ $T_\eta = 1500$	$f_1(x) = \tanh(1.2x)$ $f_2(x) = \tanh(0.3x)$	0.19246	0.33767	0.99686	0.99531
2b	10	$\alpha_0 = 0.5$ $T_\alpha = 1500$ $\eta_0 = 0.5$ $T_\eta = 1400$	$f_1(x) = \tanh(1.4x)$ $f_2(x) = 0.2x$	0.23785	0.36424	0.99708	0.99545
3a	7	$\eta_0 = 1.3$	$f_1(x) = \tanh(1.3x)$ $f_2(x) = \tanh(0.2x)$	0.19324	0.35826	0.99686	0.99493
3b	9	$\eta_0 = 0.8$	$f_1(x) = \tanh(1.4x)$ $f_2(x) = 1/(1+\exp(-0.4x))$	0.20748	0.33615	0.99682	0.99574
4a	2	$\eta_0 = 3.2$	$f_1(x) = \tanh(1.8x)$ $f_2(x) = 1/(1+\exp(-0.4x))$	0.79253	0.55889	0.98843	0.99325
4b	9	$\eta_0 = 0.7$	$f_1(x) = \tanh(1.9x)$ $f_2(x) = 1/(1+\exp(-0.3x))$	0.19479	0.32615	0.99696	0.99570
5a	2	$\alpha_0 = 0.9$ $T_\alpha = 3000$ $\eta_0 = 0.9$ $T_\eta = 3000$	$f_1(x) = \tanh(1.6x)$ $f_2(x) = \tanh(0.5x)$	1.17586	0.65206	0.98216	0.99279
5b	2	$\alpha_0 = 0.9$ $T_\alpha = 3000$ $\eta_0 = 0.9$ $T_\eta = 3000$	$f_1(x) = \tanh(1.6x)$ $f_2(x) = \tanh(0.5x)$	1.17586	0.65206	0.98216	0.99279
6a	3	$\alpha_0 = 0.9$ $T_\alpha = 2800$ $\eta_0 = 0.9$ $T_\eta = 2800$	$f_1(x) = \tanh(1.7x)$ $f_2(x) = \tanh(0.5x)$	0.48363	0.46830	0.99253	0.99423
6b	3	$\alpha_0 = 0.9$ $T_\alpha = 2800$ $\eta_0 = 0.9$ $T_\eta = 2800$	$f_1(x) = \tanh(1.7x)$ $f_2(x) = \tanh(0.5x)$	0.48363	0.46830	0.99253	0.99423
7a	22	$s = 0.2$ $T_{bv} = 40$ $T_{\mathrm{trix}} = 50$ $e_{\mathrm{trix}} = 10^{-5}$	$f_1(x) = \tanh(1.4x)$ $f_2(x) = \tanh(0.3x)$	0.18559	0.37276	0.99704	0.99533
7b	22	$s = 0.2$ $T_{\mathrm{trix}} = 50$ $e_{\mathrm{trix}} = 10^{-5} T_{\mathrm{bv}} = 40$	$f_1(x) = \tanh(1.6x)$ $f_2(x) = \tanh(0.2x)$	0.17666	0.31379	0.99716	0.99562
8a	6	$s = 0.2$ $T_{bv} = 20$ $T_{\mathrm{trix}} = 50$ $e_{\mathrm{trix}} = 10^{-6}$ $\lim_{\mathrm{orthogonality}} = 0.9$	$f_1(x) = \tanh(1.8x)$ $f_2(x) = \tanh(0.4x)$	0.23956	0.36674	0.99616	0.99543
8b	6	$s = 0.2$ $T_{bv} = 20$ $T_{\mathrm{trix}} = 50$ $e_{\mathrm{trix}} = 10^{-6}$ $\lim_{\mathrm{orthogonality}} = 0.9$	$f_1(x) = \tanh(2.0x)$ $f_2(x) = 0.3x$	0.21616	0.34217	0.99668	0.99549
9a	5	$s = 0.2$ $T_{bv} = 20$ $T_{\mathrm{trix}} = 50$ $e_{\mathrm{trix}} = 10^{-5}$	$f_1(x) = \tanh(1.8x)$ $f_2(x) = \tanh(0.3x)$	0.23758	0.37167	0.99620	0.99531
9b	5	$s = 0.2$ $T_{bv} = 20$ $T_{\mathrm{trix}} = 50$ $e_{\mathrm{trix}} = 10^{-5}$	$f_1(x) = \tanh(1.8x)$ $f_2(x) = \tanh(0.3x)$	0.23777	0.37207	0.99620	0.99531
10a	5	$s = 0.2$ $T_{bv} = 20$ $T_{\mathrm{trix}} = 50$ $e_{\mathrm{trix}} = 10^{-5}$ $\lim_{\mathrm{orthogonality}} = 0.9$	$f_1(x) = \tanh(1.6x)$ $f_2(x) = \tanh(0.4x)$	0.26472	0.37439	0.99578	0.99531
10b	5	$s = 0.2$ $T_{bv} = 20$ $T_{\mathrm{trix}} = 50$ $e_{\mathrm{trix}} = 10^{-5}$ $\lim_{\mathrm{orthogonality}} = 0.9$	$f_1(x) = \tanh(1.8x)$ $f_2(x) = \tanh(0.3x)$	0.23834	0.37220	0.99620	0.99529
11a	9	$\sigma = 10^{-4}$ $\lambda_0 = 10^{-7}$	$f_1(x) = \tanh(1.9x)$ $f_2(x) = 0.3x$	0.20660	0.32483	0.99678	0.99560
11b	9	$\sigma = 10^{-4}$ $\lambda_0 = 10^{-7}$	$f_1(x) = \tanh(2.0x)$ $f_2(x) = \tanh(0.4x)$	0.32942	0.38109	0.99481	0.99564
12a	3	$\delta_1 = 0.2$ $\delta_2 = 5.0$	$f_1(x) = 1.9x$ $f_2(x) = (1+\mathrm{e}^{-0.3}x)^{-1}$	22.25339	20.43561	0.93842	0.82334
12b	3	$\delta_1 = 0.2$ $\delta_2 = 6.0$	$f_1(x) = 0.8x$ $f_2(x) = (1+\mathrm{e}^{-0.2}x)^{-1}$	18.55159	14.53134	0.92041	0.87589

Table 2
Measured and estimated values of ground resistance for training algorithm 7b.

Ground resistance	Estimated by 7b	Measured
1	34.8	34.9
2	38.4	38.0
3	18.8	19.3
4	19.2	19.5
5	19.6	20.0
6	27.8	27.8
7	16.8	16.7
8	18.1	18.2
9	19.9	19.5
10	20.0	18.8
MAPE%	1.776	

estimated values for all sets increases. This behavior is more obvious for training algorithm 7b, whose correlation values remain constant above 5000 epochs. On the other hand, algorithm 2b provides higher correlation values than algorithm 7b for lower number of epochs. Above 500 epochs 7b algorithm performs better than algorithm 2b. Therefore, the value of 7000, which was chosen for the simulations, improves the performance of each algorithm, without increasing the computation time. It is also worth mentioning that the selection of values lower than 1000 epochs gives misleading results regarding the performance of the training algorithms.

4. Comparison to other ANN

One should always keep in mind that ANNs have the ability to learn the relationship between inputs and output according to the

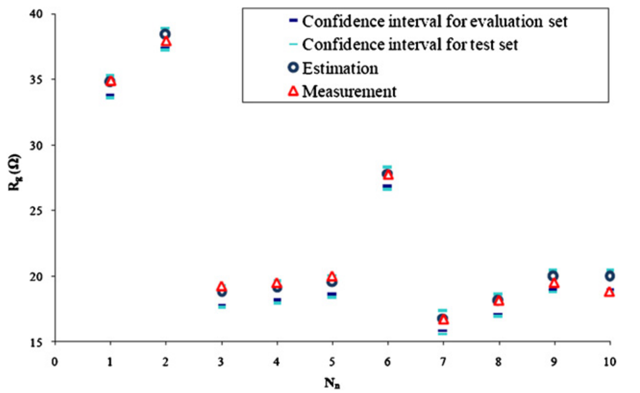

Fig. 9. Experimental and estimated values of the ground resistance for the test set for stochastic training algorithm with batch mode conjugate gradient algorithm with Fletcher–Reeves equations (algorithm 7b).

Table 3
Influence of random initialization of synaptic weights to the performance of training algorithms 2b and 7b.

Epochs	R^2					
	Training set		Evaluation set		Test set	
	2b	7b	2b	7b	2b	7b
1	0.99774	0.99780	0.99714	0.99716	0.99547	0.99562
2	0.99770	0.99772	0.99710	0.99722	0.99543	0.99547
3	0.99770	0.99802	0.99708	0.99772	0.99547	0.99608
4	0.99778	0.99778	0.99712	0.99700	0.99545	0.99537
5	0.99772	0.99768	0.99710	0.99698	0.99543	0.99586
6	0.99774	0.99784	0.99712	0.99728	0.99545	0.99554
7	0.99772	0.99774	0.99712	0.99714	0.99543	0.99572
8	0.99772	0.99754	0.99704	0.99690	0.99537	0.99582
9	0.99766	0.99782	0.99698	0.99710	0.99566	0.99556
10	0.99778	0.99750	0.99716	0.99666	0.99541	0.99529

patterns that have been presented and have been used for the training. Therefore, in case data for a different type of grounding system is used as test set it is expected that ANN will not be effective and retraining of the ANN is required. Besides, the sensitivity of the ANN on different training scenarios can be verified by taking into consideration that in [10] the same experimental data have been used however different training and test sets have been formed. In [10] the best training algorithm was stochastic training with learning rate and momentum term with two stopping criteria. The correlation being achieved was 99.78% for the evaluation set and 97.46% for the test set. Moreover, parameters such as the selection of the number of epochs and the initialization of the synaptic weights may influence in a smaller or larger scale the selection of the performance of each training algorithm.

Table 4
Influence of batch (epochs) number to training algorithms 2b and 7b.

Epochs	R^2					
	Training set		Evaluation set		Test set	
	2b	7b	2b	7b	2b	7b
50	0.97494	0.68847	0.96478	0.60115	0.98923	0.65542
100	0.97976	0.76718	0.97239	0.63652	0.99222	0.79872
500	0.98708	0.99592	0.98139	0.99489	0.99533	0.99519
1000	0.99395	0.99592	0.99291	0.99489	0.99556	0.99519
5000	0.99754	0.99780	0.99702	0.99716	0.99551	0.99562
7000	0.99766	0.99780	0.99708	0.99716	0.99545	0.99562
10,000	0.99774	0.99780	0.99714	0.99716	0.99547	0.99562

5. Conclusions

ANNs have been trained and validated for estimating the ground resistance of a grounding rod given the soil resistivity and rainfall data. The back propagation algorithm and several variations of this algorithm have been used and the estimated values of the ground resistance are found to be in good agreement with the experimental data. Among the training algorithms, the one with use of batch mode scale conjugate algorithm with Fletcher–Reeves equations and use of two stopping criteria provides the highest correlation index of the evaluation set. The effectiveness of the ANN in predicting the ground resistance is verified by the fact that the correlation index of the test set is respectively high.

Furthermore, the absence of limitations regarding number of the input and the output variables of the ANN makes possible the incorporation of experimental data for longer time periods, new parameters such as soil temperature, water content, type and size of the grounding system.

In case the behavior of a grounding system is similar to an already examined grounding system, a previously trained ANN could be used and is expected to produce satisfactory results. However, the performance of the ANN depends on the patterns which have been used for training. Therefore, a trained ANN might not produce accurate results when the input data differ greatly from those which has been used for train the ANN.

As a conclusion it can be stated that the work presented in this paper could be used as a guideline for further research on the applicability and the development of artificial intelligence techniques for grounding systems. In future work new scenarios for different grounding systems can be examined, while the sensitivity of the ANN to variations of the training and test sets should be investigated.

Acknowledgments

The authors would like to express their gratitude to Dr. V.T. Kontargyri and Mr. C.A. Alexandrou for their kind support concerning the application of the ANN methodology.

Appendix A. Mathematical modeling of ANN training algorithms

Steepest descent Back-Propagation algorithm: the weights' correction is calculated after the end of the respective epoch *ep*:

$$\Delta\vec{\mathbf{w}}(ep) = -\eta \cdot \nabla G\left(\vec{\mathbf{w}}(ep)\right) \tag{A.1}$$

where η is the learning rate and $G(n) = (1/2)\sum_{j \in C} e_j^2(n)$ is the sum of the square errors for all output neurons after the nth iteration. In case a momentum term (α) is added, then the equation for the weights' correction is:

$$\Delta\vec{\mathbf{w}}(ep) = -\eta \cdot \nabla G(\vec{\mathbf{w}}(ep)) + \alpha \cdot \Delta\vec{\mathbf{w}}(ep-1) \tag{A.2}$$

Adaptive Back-Propagation algorithm [14]: the learning rate and the momentum term are adaptively changed according to Eqs. (A.3) and (A.4):

$$\eta(ep) = \begin{cases} \eta(ep-1), & G_{av}(ep) > G_{av}(ep-1) \\ \eta(ep-1) \cdot \exp(-1/T_\eta), & G_{av}(ep) \le G_{av}(ep-1) \end{cases} \tag{A.3}$$

$$a(ep) = \begin{cases} a(ep-1), & G_{av}(ep) > G_{av}(ep-1) \\ a(ep-1) \cdot \exp(-1/T_a), & G_{av}(ep) \le G_{av}(ep-1) \end{cases} \tag{A.4}$$

where T_η, $\eta_0 = \eta(0)$, T_a and $a_0 = a(0)$ are the time parameter and the initial value of the learning rate and the momentum term respectively.

Resilient algorithm [14,13]: The weights' correction is given by the formula:

$$\Delta w_{ij}(ep) = \begin{cases} \delta_1 \cdot \Delta w_{ij}(ep-1), & \dfrac{\partial G_{av}}{\partial w_{ij}}(ep) \cdot \dfrac{\partial G_{av}}{\partial w_{ij}}(ep-1) > 0 \\ \Delta w_{ij}(ep-1), & \dfrac{\partial G_{av}}{\partial w_{ij}}(ep) \cdot \dfrac{\partial G_{av}}{\partial w_{ij}}(ep-1) = 0 \\ \dfrac{1}{\delta_2} \cdot \Delta w_{ij}(ep-1), & \dfrac{\partial G_{av}}{\partial w_{ij}}(ep) \cdot \dfrac{\partial G_{av}}{\partial w_{ij}}(ep-1) < 0 \end{cases} \tag{A.5}$$

where δ_1, δ_2 are the increasing and the decreasing factor of change in the value of the weights between two successive epochs.

Conjugate gradient algorithm [15]: the basic steps of this method are the following:

a. The first search direction $\vec{\mathbf{p}}_0$ is selected to be the negative of the gradient:

$$\vec{\mathbf{p}}_0 = -\nabla G(\vec{\mathbf{w}})\Big|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_0} \tag{A.6}$$

b. The error function is minimized along the search direction:

$$\Delta\vec{\mathbf{w}}_k = a_k \cdot \vec{\mathbf{p}}_k \tag{A.7}$$

where the parameter a_k is computed by arithmetic methods, such as the golden section and bisection.

c. The next search direction is selected according to:

$$\vec{\mathbf{p}}_{k+1} = -\nabla G(\vec{\mathbf{w}})\Big|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_{k+1}} + \beta_{k+1} \cdot \vec{\mathbf{p}}_k \tag{A.8}$$

where the parameter β_{k+1} is determined either by the Fletcher–Reeves equation (A.9) [16] or by the Polak–Ribière equation (A.10) [18].

$$\beta_{k+1} = \frac{\nabla G(\vec{\mathbf{w}})|^T_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_{k+1}} \cdot \nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_{k+1}}}{\nabla G(\vec{\mathbf{w}})|^T_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k} \cdot \nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k}} \tag{A.9}$$

$$\beta_{k+1} = \frac{\Delta(\nabla G(\vec{\mathbf{w}})|^T_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k}) \cdot \nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_{k+1}}}{\nabla G(\vec{\mathbf{w}})|^T_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k} \cdot \nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k}} \tag{A.10}$$

The second step and third step are repeated unless the algorithm has converged. In order to achieve faster convergence the algorithm should be restarted when the following criterion is fulfilled, as proposed by Powell and Beale [17]:

$$|\nabla G(\vec{\mathbf{w}})|^T_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k} \cdot \nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k+1}| \geq \lim_{\text{orthogonality}} \cdot ||\nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k+1}||^2 \tag{A.11}$$

where the orthogonality limit $\lim_{\text{orthogonality}}$ can range from 0.1 to 0.9 – preferably 0.2.

Scaled conjugate gradient algorithm [19]: Uses the Levenberg–Marquardt approach. The steps of the algorithm are the following:

The first direction search is initialized as in (A.6) as well as the vector of the weights and biases $\vec{\mathbf{w}}_0$ and the rest of the parameters (σ, λ_0, $\bar{\lambda}_0$ and *flag*) as:

$$0 < \sigma \leq 10^{-4} \quad 0 < \lambda_0 \leq 10^{-6} \quad \bar{\lambda}_0 = 0 \quad flag = 1 \tag{A.12}$$

If *flag* = 1, then:

$$\sigma_k = \sigma/||\vec{\mathbf{p}}_k|| \quad \vec{\mathbf{s}}_k = \frac{\nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k+\sigma_k\cdot\vec{\mathbf{p}}_k} - \nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k}}{\sigma_k}$$

$$\delta_k = \vec{\mathbf{p}}_k^T \cdot \vec{\mathbf{s}} \tag{A.13}$$

Parameter δ_k:

$$\delta_k = \delta_k + (\lambda_k - \bar{\lambda}_k) \cdot ||\vec{\mathbf{p}}_k||^2 \tag{A.14}$$

If $\delta_k \leq 0$, then the Hessian matrix is made positive:

$$\bar{\lambda}_k = 2\left(\frac{\lambda_k - \delta_k}{||\vec{\mathbf{p}}_k||^2}\right) \quad \delta_k = -\delta_k + \lambda_k \cdot ||\vec{\mathbf{p}}_k||^2 \quad \lambda_k = \bar{\lambda}_k \tag{A.15}$$

The step size is calculated:

$$\mu_k = -\vec{\mathbf{p}}_k^T \cdot \nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k} \quad a_k = \frac{\mu_k}{\delta_k} \tag{A.16}$$

The comparison parameter is calculated:

$$\Delta_k = \frac{2 \cdot \delta_k \cdot G(\vec{\mathbf{w}})\Big|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k} - G(\vec{\mathbf{w}})\Big|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k+a_k\cdot\vec{\mathbf{p}}_k}}{\mu_k^2} \tag{A.17}$$

If $\Delta_k \geq 0$, then a successful reduction in error can be made:

$$\Delta\vec{\mathbf{w}}_k = a_k \cdot \vec{\mathbf{p}}_k \quad \vec{\mathbf{r}}_{k+1} = -\nabla G(\vec{\mathbf{w}})\Big|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_{k+1}} \quad \bar{\lambda}_k = 0 \quad flag = 1 \tag{A.18}$$

If $k \bmod N_w = 0$ (where N_w is the number of weights and biases), then the algorithm will be restarted:

$$\vec{\mathbf{p}}_{k+1} = -\nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_{k+1}} \tag{A.19}$$

else:

$$\beta_{k+1} = \frac{||\nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_{k+1}}||^2 - \nabla G(\vec{\mathbf{w}})|^T_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k} \cdot \nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k}}{\mu_k} \tag{A.20}$$

$$\vec{\mathbf{p}}_{k+1} = \nabla G(\vec{\mathbf{w}})|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_k} + \beta_{k+1} \cdot \vec{\mathbf{p}}_k \tag{A.21}$$

If $\Delta_k \geq 0.75$, then $\lambda_k = 0.25 \cdot \lambda_k$, else $\bar{\lambda}_k = \lambda_k$, *flag* = 0.
If $\Delta_k < 0.25$, then

$$\lambda_k = \frac{\lambda_k + \delta_k(1 - \Delta_k)}{||\vec{p}_k||^2} \tag{A.22}$$

If $\nabla G(\vec{\mathbf{w}})\Big|_{\vec{\mathbf{w}}=\vec{\mathbf{w}}_{k+1}} \neq \vec{\mathbf{0}}$, then $k = k + 1$ and step (2) is repeated, else the training process has been completed.

In case $\lambda_k = 0$, then the scaled conjugate gradient algorithm is identical to the conjugate gradient algorithm.

Newton algorithm [20]: Inversion of the Hessian matrix $\nabla^2 G(\vec{w})$ in order to estimate the weights and the biases comprises the basis of the method.

$$\Delta\vec{w}_k = -\nabla^2 G(\vec{w})|^{-1}_{\vec{w}=\vec{w}_k} \cdot \nabla G(\vec{w})|_{\vec{w}=\vec{w}_k} \tag{A.23}$$

Although this method is, usually, the fastest one, the inversion of the Hessian matrix according to the following formulas is complicated.

$$\text{Hessian matrix}: \quad \nabla^2 G(\vec{w}) = J(\vec{w})^T \cdot J(\vec{w}) + \sum_{j \in C} e_j(\vec{w}) \cdot \nabla^2 e_j(\vec{w}) \tag{A.24}$$

$$\text{Jacobian matrix}: \quad J(\vec{\mathbf{w}}) = \begin{bmatrix} \dfrac{\partial e_1}{\partial w_1} & \dfrac{\partial e_1}{\partial w_2} & \cdots & \dfrac{\partial e_1}{\partial w_{N_w}} \\ \dfrac{\partial e_2}{\partial w_1} & \dfrac{\partial e_2}{\partial w_2} & \cdots & \dfrac{\partial e_2}{\partial w_{N_w}} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial e_{p_C}}{\partial w_1} & \dfrac{\partial e_{p_C}}{\partial w_2} & \cdots & \dfrac{\partial e_{p_C}}{\partial w_{N_w}} \end{bmatrix}_{p_C \times N_w} \tag{A.25}$$

One of the basic variations for Newton method is quasi-Newton method, where the second term of (A.23) is omitted. Alternatively, in the one step secant algorithm only the diagonal elements of the matrix are stored, thus making the inversion of the matrix an unnecessary task. The algorithm requires greater number of iterations in order to converge. The computational complexity per iteration, however, is significantly compressed.

According to the Levenberg–Marquardt [21,22] method the weights are estimated by the following expression:

$$\Delta\vec{w}_k = -(J^T \cdot J + \lambda \cdot diag[J^T \cdot J])^{-1} \cdot \nabla G(\vec{w})|_{\vec{w}=\vec{w}_k}$$

$$\Rightarrow \Delta\vec{w}_k = -(J^T \cdot J + \lambda \cdot diag[J^T \cdot J])^{-1} \cdot J^T \cdot \vec{e}(\vec{w}_k) \tag{A.26}$$

Factor λ is given by the formula:

$$\lambda(k+1) = \begin{cases} \lambda(k) \cdot \beta, & G_{av}(k) > G_{av}(k-1) \\ \lambda(k), & G_{av}(k) = G_{av}(k-1) \\ \lambda(k)/\beta, & G_{av}(k) < G_{av}(k-1) \end{cases} \tag{A.27}$$

where parameter β takes significant values, such as 10.

Appendix B. Input data for ANN training

See Table B1.

Table B1
Experimental values.

Previous day rainfall	Previous week rainfall	ρ (a = 1 m) (Ωm)	ρ (a = 2 m) (Ωm)	Previous week resistance (Ω)	R_g (Ω)
Training set					
0.00	0.04	91.00	66.49	34.77	34.7
0.00	0.04	91.00	66.49	34.77	34.9
0.00	0.01	93.82	69.17	34.83	35.5
0.60	0.01	93.82	69.17	34.83	36.0
4.60	0.01	93.82	69.17	34.83	34.9
0.00	1.13	100.23	68.28	35.47	35.9
0.00	1.13	100.23	68.28	35.47	36.6
0.00	1.13	100.23	68.28	35.47	37.0
0.00	0.00	103.82	70.06	36.50	38.5
48.00	0.00	103.82	70.06	36.50	32.0
0.10	7.20	99.71	71.85	36.17	33.5
16.00	7.20	99.71	71.85	36.17	28.5
53.00	7.20	99.71	71.85	36.17	19.2
14.10	19.23	69.73	60.25	27.07	16.6
0.00	19.23	69.73	60.25	27.07	16.7
0.00	0.00	48.19	42.88	16.67	17.1
0.00	0.00	48.19	42.88	16.67	17.2
0.10	0.00	48.19	42.88	16.67	17.5
0.10	0.04	50.76	43.95	17.27	18.8
0.10	2.10	56.14	47.66	18.50	18.5
0.00	0.76	54.22	45.69	18.13	19.3
0.10	0.76	54.22	45.69	18.13	19.3
9.50	1.37	57.68	46.85	19.30	19.2
2.20	1.37	57.68	46.85	19.30	19.8
0.00	1.11	58.83	47.13	19.50	20.0
2.20	1.11	58.83	47.13	19.50	19.5
17.70	2.86	55.88	47.75	19.67	19.0
5.20	2.86	55.88	47.75	19.67	19.2
0.00	2.86	55.88	47.75	19.67	19.2
0.10	2.29	52.55	45.03	19.13	19.0
0.00	2.29	52.55	45.03	19.13	19.5
1.47	0.21	55.63	45.12	19.33	19.3
0.00	0.21	55.63	45.12	19.33	19.0
6.40	2.75	54.09	43.29	19.03	19.0
0.00	2.75	54.09	43.29	19.03	19.0
0.00	0.05	54.63	40.85	18.50	19.5
0.20	0.05	54.63	40.85	18.50	19.8
0.00	1.29	53.60	43.93	19.77	20.0
0.00	1.29	53.60	43.93	19.77	20.4
0.00	1.29	53.60	43.93	19.77	20.2
0.00	0.12	55.66	45.14	20.20	21.0
0.00	0.07	54.25	43.20	22.30	22.6
0.00	0.07	54.25	43.20	22.30	22.6
0.00	0.07	54.25	43.20	22.30	23.0
0.00	0.00	64.89	46.48	22.73	23.0
0.00	0.00	64.89	46.48	22.73	23.2
0.00	0.00	64.89	46.48	22.73	23.2
0.00	1.63	86.17	54.65	23.20	23.4
0.00	1.22	89.25	57.06	23.40	24.4
0.00	0.00	90.02	58.40	24.40	25.2
0.00	0.00	95.80	62.69	27.10	27.4
0.00	0.00	95.80	62.69	27.10	27.6
0.00	0.00	94.64	63.22	27.80	27.7
Evaluation set					
0.00	0.04	91.00	66.49	34.77	34.70
4.60	0.01	93.82	69.17	34.83	34.90
0.00	0.00	103.82	70.06	36.50	38.50
53.00	7.20	99.71	71.85	36.17	19.20
0.00	0.00	48.19	42.88	16.67	17.20
0.00	0.76	54.21	45.69	18.13	19.30
0.00	1.11	58.83	47.13	19.50	20.00
0.00	2.86	55.88	47.75	19.67	19.20
0.00	0.21	55.63	45.12	19.33	19.00
0.20	0.05	54.63	40.85	18.50	19.80
0.00	0.12	55.66	45.14	20.20	21.00
0.00	0.00	64.89	46.48	22.73	23.00
0.00	1.22	89.25	57.06	23.40	24.40
0.00	0.00	94.64	63.22	27.80	27.70
Test set					
0.00	0.04	91.00	66.49	34.77	34.90
0.00	0.00	103.82	70.06	36.50	38.00
0.00	0.76	54.22	45.69	18.13	19.30
0.00	2.29	52.55	45.03	19.13	19.50
0.00	0.05	54.63	40.85	18.50	20.00
0.00	0.00	95.41	63.09	27.60	27.80
0.00	19.23	69.73	60.25	27.07	16.70
0.00	0.04	50.76	43.95	17.27	18.20
0.10	1.11	58.83	47.13	19.50	19.50
0.07	0.21	55.63	45.12	19.33	18.80

References

[1] ANSI/IEEE Std 80, Guide for safety in AC substation grounding, 2000.
[2] IEEE Std81, Guide for measuring earth resistivity, ground impedance and earth surface potentials of a ground system, 1983.
[3] J. Jasni, L.K. Siow, M.Z.A. Ab Kadir, W.F. Wan Ahmad, Natural materials as grounding filler for lightning protection system, in: Proc. ICLP, 2010, pp. 1101-1–1101-6.
[4] A.D. Galván, G.G. Pretelin, E.E. Gaona, Practical evaluation of ground enhancing compounds for high soil resistivities, in: Proc. ICLP, 2010, pp. 1233-1–1233-4.
[5] I.F. Gonos, A.X. Moronis, I.A. Stathopulos, Variation of soil resistivity and ground resistance during the year, in: Proc. ICLP, 2006, pp. 740–744.
[6] W.F. Wan Ahmad, M.S. Abdul Rahman, J. Jasni, M.Z.A. Ab Kadir, H. Hizam, Chemical enhancement materials for grounding purposes, in: Proc. ICLP, 2010, pp. 1233-1–1233-4.
[7] M.A. Salam, S.M. Al-Alawi, A.A. Maquashi, An artificial neural networks approach to model and predict the relationship between the grounding resistance and the length of the buried electrode in soil, Journal of Electrostatics 64 (2006) 338–342.
[8] F.C.L. Amaral, A.N. de Souza, M.G. Zago, A novel approach to model grounding systems considering the influence of high frequencies, CLAGTEE, Sao Pedro, 2003.
[9] F.E. Asimakopoulou, I.F. Gonos, G.J. Tsekouras, I.A. Stathopulos, An artificial neural network for estimating ground resistance, in: Proc. GROUND'2010 & 4th LPE, 2010 (paper 34).
[10] F.E. Asimakopoulou, E.A. Kourni, V.T. Kontargyri, G.J. Tsekouras, I.A. Stathopulos, Artificial neural network methodology for the estimation of ground resistance, in: Proc. 15th WSEAS Int. Conf. Syst., 2011, pp. 453–458.
[11] F.E. Asimakopoulou, G.J. Tsekouras, I.F. Gonos, I.A. Stathopulos, Artificial neural network approach on the seasonal variation of soil resistance, in: Proc. 7th Asia-Pacific Conf. Lightning Protection APL 2011, Chengdu, China, 2011, pp. 794–799.
[12] G.E. Asimakopoulou, V.T. Kontargyri, G.J. Tsekouras, F.E. Asimakopoulou, I.F. Gonos, I.A. Stathopulos, Artificial neural network optimization methodology for the estimation of the critical flashover voltage on insulators, IET Science, Measurement and Technology 3 (1) (2009) 90–104.
[13] P.S. Ghosh, S. Chakravorti, N. Chatterjee, Estimation of time-to-flashover characteristics of contaminated electrolytic surfaces using a neural network, IEEE Transactions on Dielectrics and Electrical Insulation 2 (6) (1995) 1064–1074.

[14] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the RPROP algorithm, in: Proc. IEEE Int. Conf. Neural Netw., vol. 1, San Francisco, 1993, pp. 586–591.
[15] L.M. Saini, M.K. Soni, Artificial neural network-based peak load forecasting using conjugate gradient methods, IEEE Transactions on Power Systems 17 (3) (2002) 907–912.
[16] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients, Computer Journal 7 (1964) 149–154.
[17] M.J. Powell, Restart procedures for the conjugate gradient method, Mathematical Programming 12 (1977) 241–254.
[18] E. Polak, Computational Methods in Optimization: A Unified Approach, Academic Press, New York, 1971.
[19] M.F. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993) 525–533.
[20] R. Battiti, First and second order methods for learning: between steepest descent and Newton's method, Neural Computation 4 (2) (1992) 141–166.
[21] K. Levenberg, A method for the solution of certain problems in least squares, The Quarterly of Applied Mathematics 2 (1944) 164–168.
[22] D. Marquardt, An algorithm for least squares estimation of nonlinear parameters, SIAM Journal on Application Mathematics 11 (1963) 431–441.