12

Artificial Intelligence Techniques Applied to Electromagnetic Interference Problems Between Power Lines and Metal Pipelines

Dan D. Micu[1], Georgios C. Christoforidis[2] and Levente Czumbil[1]
[1]*Technical University of Cluj-Napoca,*
[2]*Technological Educational Institution of West Macedonia, Kozani,*
[1]*Romania*
[2]*Greece*

1. Introduction

European ecological regulations meant to protect nature and wild life along with construction cost reduction policies generated a set of government regulations that limit the access to new transmission and distribution corridors. As a result, gas, water or oil supply pipelines are forced to share the same distribution corridors with Electrical Power Lines (EPL), AC Railway Systems or Telecommunication Lines (figure 1).

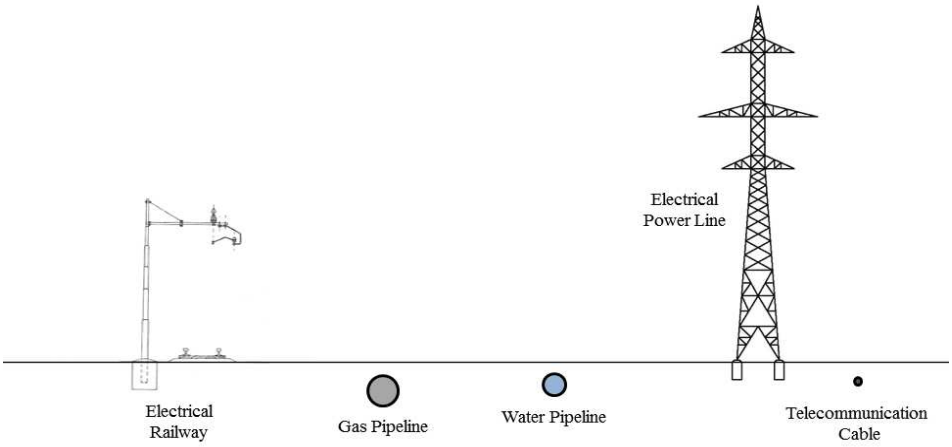


Fig. 1. Common distribution corridor. Right of way.

The electromagnetic fields generated by high voltage electrical power lines produce AC interference in the nearby metallic structures. Therefore, in many cases these underground or above ground utilities supply Metal Pipelines (MP) are exposed to effects of induced AC currents and voltages (CIGRÉ 1995, Dawalibi & Southey 1989). Especially in case of power line faults, the resulting AC voltage in unprotected pipelines may reach thousands of volts.

This could be dangerous on both the operating personnel (that may be exposed to electric shocks), and to the structural integrity of the pipeline, due to corrosion. Underground steel pipelines are in permanent contact with the electrolyte solution from the soil, so proper protection measures are necessary in order to limit the induced current densities, which are the cause of electrochemical corrosion. Uncontrolled corrosion of supply pipelines may cause gas or/and oil spills with very serious ecological and economic implications. (Baboian 2002, Collet et al. 2001)

In order to provide proper protection for pipelines the AC interference has to be well known. Normally, the electromagnetic interference between electrical power lines and nearby metallic pipelines could be generated by any of the following three types of couplings (CIGRÉ 1995):

- **Inductive Coupling**: Aerial and underground pipelines that run parallel to or in close proximity to transmission lines or cables are subjected to induced voltages by the time varying magnetic fields produced by the transmission line currents. The induced electromotive forces (EMF) causes current circulation in the pipeline and voltages between the pipeline and surrounding earth.
- **Conductive Coupling**: When a ground fault occurs in the electrical power system the current flowing through the grounding grid produce a potential rise on both the grounding grid and the neighbouring soil with regard to remote earth. If the pipeline goes through the "zone of influence" of this potential rise, then a high difference in the electrical potential can appear across the coating of the pipeline metal.
- **Capacitive Coupling**: Affects only above ground pipelines situated next to overhead power lines. It occurs due to the capacitance between the power line and the pipeline. For underground pipelines the effect of capacitive coupling may be neglected, because of the screening effect of the earth.

In case of interferences between EPLs under normal operating conditions and underground pipelines, only the inductive coupling described by the self and mutual impedance matrix has to be taken into consideration. Conductive and capacitive interference may be, also, neglected when a ground fault happens significantly away from the common corridor.

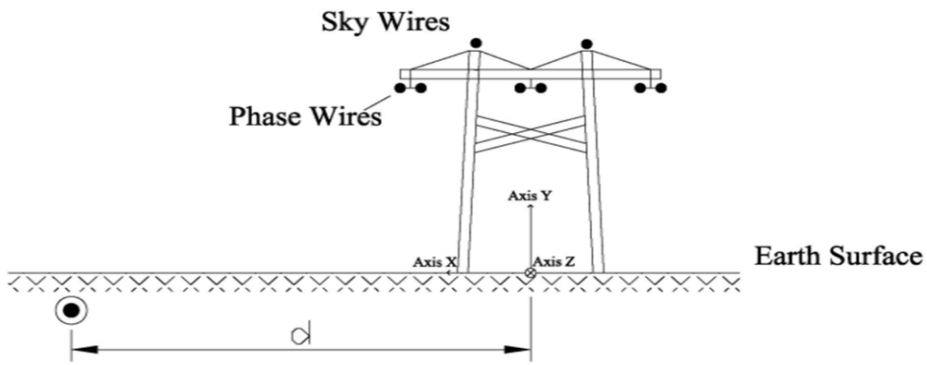


Fig. 2. Cross section of an interference problem between an EPL and an underground MP.

To evaluate the self and mutual impedance between the conductors (phase wires, sky wires and pipelines) the magnetic vector potential has to be evaluated on the cross section (figure 2) of these conductors as presented in (Christoforidis et al. 2003, 2005).

Thus, taking into account the cross section of the studied problem, the z-direction component of the magnetic vector potential A_z and of the total current density J_z are described by the following system of differential equations:

$$\begin{cases} \dfrac{1}{\mu_0\mu_r}\cdot\left[\dfrac{\partial^2 A_z}{\partial x^2}+\dfrac{\partial^2 A_z}{\partial y^2}\right]-j\omega\sigma A_z+J_{sz}=0 \\ -j\omega\sigma A_z+J_{sz}=J_z \\ \iint\limits_{S_i} J_z ds=I_i \end{cases} \tag{1}$$

where σ is the conductivity, ω is the angular frequency, μ_0 is the magnetic permeability of free space ($\mu_0 = 4\cdot\pi\cdot 10^{-7}$ H/m), μ_r is the relative permeability of the environment, J_{sz} is the source current density in the z-direction and I_i is the imposed current on conductor i of S_i cross section.

To solve this differential equation system, the finite element method (FEM) is recommended to be used. FEM effectively transforms the electromagnetic interference problem into a numerical one. Although FEM yielded solutions are very accurate, regarding to the problem complexity, the computing time of this method increases with the geometry, its mesh, material characteristics and requested evaluation parameters.

The FEM calculations are used in the described method, as a means of calculating the self and mutual impedances of the conductors present in the configuration. Generally, if there exist n conductors in the configuration and assuming that the per unit length voltage drop on every conductor is known for a specific current excitation, the mutual complex impedance between conductor i and another conductor j carrying a certain current, where all other conductors are imposed to carry zero currents, is given by:

$$Z_{ij}=\frac{V_i}{I_j}\ \ (i,j=1,2,\ldots,n) \tag{2}$$

Similarly, the self-impedance of conductor i may be calculated using (2), by setting i=j.

The procedure is summarized below (Papagiannis et. al. 2000):

- By applying a sinusoidal current excitation of arbitrary magnitude to each conductor, while applying zero current to the other conductors, the corresponding voltages are calculated.
- The self and mutual impedances of the j conductor may be calculated using (2).

The above procedure is repeated n times, so as to calculate the impedances for n conductors.

Applying FEM calculations for the solution of linear electromagnetic diffusion equation (first relation from system (1)), the values for source current density in the z-direction (J_{szi}) on each conductor i having a conductivity of σ_i are obtained. Therefore, equation (2) becomes:

$$Z_{ij} = \frac{V_i}{I_j} = \frac{J_{szi} / \sigma_i}{I_j} \quad (i, j = 1, 2, \ldots, n) \tag{3}$$

Following the above procedure, effectively linking electromagnetic field variables and equivalent circuit parameters, the self and mutual impedances per unit length of the problem are computed.

By using FEM to calculate the impedances of the problem instead of classic formulae, (e.g. Carson's formulae), one can deal effectively with more complex situations, such as multilayer earth or terrain irregularities.

Nevertheless, the study of electromagnetic interference between power lines and underground pipelines using FEM for different system configurations requires expensive computing time. This is because each new problem geometry taken under consideration involves a new mesh development and new calculations. To solve system equation (1) for a given problem geometry (EPL-MP separation distance, soil resistivity) with an Intel Core2 Duo T6400 (2.0 GHz/2.0 GHz) processor PC it takes from 20 to 50 minutes depending on mesh discretization. Therefore, any scaling method of the results from one configuration case to another may be of interest if it provides less computing time.

A first attempt in applying artificial intelligence techniques to scale EPL-MP interference results was made in (Satsios et al. 1999a, 1999b). A Fuzzy Logic Block (FLB) was implemented to evaluate the Magnetic Vector Potential (MVP) for an EPL-MP interference problem where phases to earth fault occurred. The input values were the geometrical parameters of the studied problem configuration (separation distance, soil resistivity) and the coordinates of the point where the MVP should be calculated. However, the implemented FLB provide relatively good results for MVP, the main disadvantage of this method consists in determination of the optimal parameters, which describes the fuzzy logic rule base. An iterative technique based on conjugate gradient has been used to optimize the fuzzy rule base parameters. Later on a Genetic Algorithm technique had been proposed in (Damousis et al. 2002) to determine the optimal parameters and rule base configuration

Another approach in using artificial intelligence techniques in the study of electromagnetic interferences between power lines and underground metal pipelines was introduced by Al-Badi (Al-Badi et al. 2005, 2007). A feed-forward Neural Network with one output layer and one hidden layer was proposed to evaluate the induced AC interference in an underground pipeline exposed to electromagnetic fields generated by an electrical power line in case of a phase to earth fault. The input values of this Neural Network were the fault current, the soil resistivity, the separation distance and a fourth parameter which indicates the presence of mitigation wires. The main advantage of this NN solution was that it provided directly the value of the induced AC voltages. However, the limitation of this model consists in the fact that the results are obtained for a specific common distribution corridor length.

In this chapter two artificial intelligence techniques are presented. These were applied by the authors in some EPL-MP electromagnetic interferences studies (Micu et al. 2009, 2011). The first one is a neural network alternative to the EPL-MP interference problem presented in (Satsios et al. 1999a, 1999b). The advantage of the proposed alternative consists in the accuracy of the obtained results and in the shorter training time. The second is a neural

network solution used to evaluate the self and mutual impedance matrix, which describe the inductive coupling between an electrical power line and an underground pipeline in the presence of a three vertical layer earth. For such a case, no analytical formula exists so far, and this means that lengthy calculations using FEM must be made. The obtained impedance matrix can be used to build and solve the equivalent electrical circuit model and thus evaluate the induce AC voltage. This equivalent electrical circuit approach permits to solve more complex problem geometries where for example the separation distance between EPL and MP varies along the common distribution corridor.

2. Magnetic vector potential evaluation using neural networks

The first attempt of the authors to use neural networks based artificial intelligence techniques in the study of electromagnetic interference problems was focused on finding an easier method to identify the optimal solution to scale the results from a set of known problem geometries to any other new problem geometries in case of specific EPL-MP interference problems.

2.1 Studied electromagnetic interference problem

The studied electromagnetic interference problem, presented in figure 3, refers to an underground metallic gas pipeline which shares for 25 km the same distribution corridor with a 145 kV EPL at 50 Hz frequency. The power line consists of two steel reinforced aluminium conductors per phase. Sky wire conductors have a 4 mm radius and the gas pipeline has a 0.195 m inner radius, a 0.2 m outer radius and a 0.1 m coating radius. The characteristics of the materials in this configuration have the following properties: the soil is assumed to be homogeneous; MP and sky wires have an σ =7.0E+05 S/m conductivity and a μ_r=250 relative permeability.

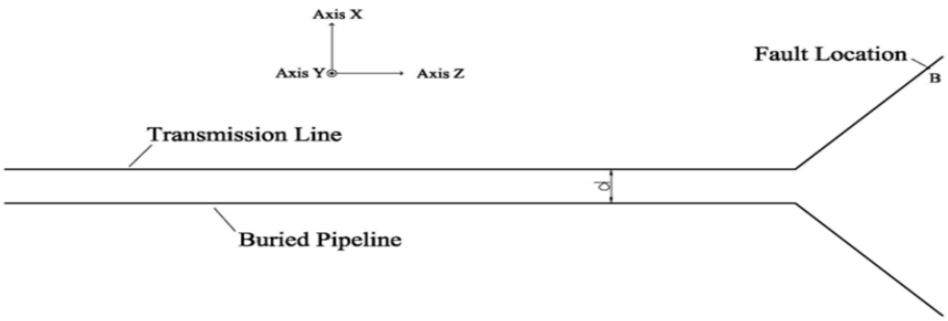


Fig. 3. Top view of the parallel exposure.

It is assumed that a phase to ground fault occurs at point B, far away outside the common EPL–MP distribution corridor. The earth current associated with this fault has a negligible action upon the buried pipeline. This fact allows us to assume only the inductive coupling caused by the flowing fault current in the section where the power lines runs parallel to the buried gas pipeline. End effects are neglected, leading to a two dimensional (2D) problem, presented in figure 4, were the magnetic vector potential has to be evaluated.

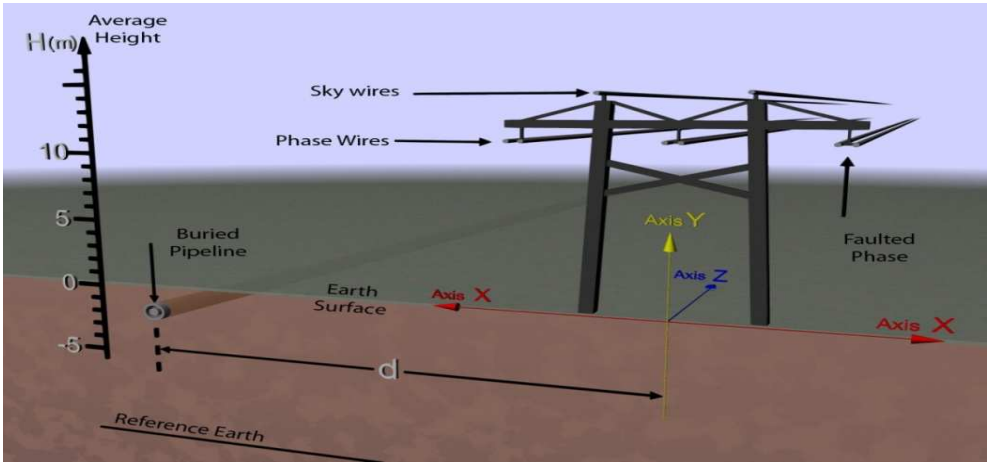


Fig. 4. Cross section of the studied EPL-MP interference problem.

2.2 Fuzzy logic implementation

A first attempt in applying artificial intelligence techniques to reduce the computational time needed by FEM to evaluate the MVP values for different problem geometries have been made in (Satsios et al. 1999a, 1999b). The presented Fuzzy Logic Block had as input values the separation distance, d, between EPL and MP, the soil resistivity, ρ, and the coordinates (x,y) of the point where MVP is wanted to be evaluated:

$$\begin{aligned} &\text{The } j^{th} \text{ rule from the presented Fuzzy Logic Block's rule base:} \\ &R^j: \textbf{IF } d, x, y \text{ and } \rho \text{ belong to the } j^{th} \text{ membership function, } \mu_d^j, \mu_x^j \mu_y^j \text{ and} \\ &\mu_\rho^j \text{ correspondingly} \\ &\textbf{THEN } A^j = \lambda_0^j + \lambda_d^j \cdot d + \lambda_x^j \cdot x + \lambda_y^j \cdot y + \lambda_\rho^j \cdot \rho \end{aligned} \tag{4}$$

The proposed Fuzzy Logic Block showed relatively good results for MVP's amplitude and phase evaluation according to the training database created by calculating MVP with FEM for a set of known problem geometries (figure 5):

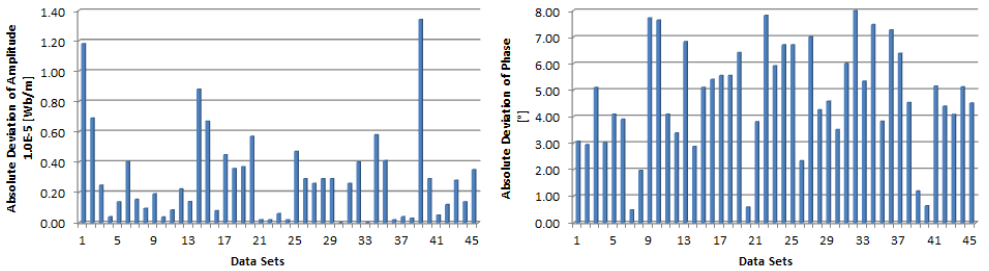


Fig. 5. Absolute evaluation error provided by the presented Fuzzy Logic Block.

To identify the proper rule base and the optimal parameters for each rule an iterative technique has been applied using gradient based relations like:

$$\nabla_{\overline{\alpha_\Delta^j}} J^P = \frac{\mu^j}{\sum_j \mu^j}\left[A^p(d,x,y,\rho) - A^p_{FEM}(d,x,y,\rho)\right]\cdot\left[A^j - A^p(d,x,y,\rho)\right]\cdot\frac{\Delta^p - \overline{\alpha_\Delta^j}}{\left(\sigma_\Delta^j\right)^2} \tag{5}$$

and

$$\nabla_{\overline{\sigma_\Delta^j}} J^P = \frac{\mu^j}{\sum_j \mu^j}\left[A^p(d,x,y,\rho) - A^p_{FEM}(d,x,y,\rho)\right]\cdot\left[A^j - A^p(d,x,y,\rho)\right]\cdot\frac{\Delta^p - \overline{\alpha_\Delta^j}}{\left(\sigma_\Delta^j\right)^3} \tag{6}$$

where $\Delta \in \{d,x,y,\rho\}$ and $A^p_{FEM}(d,x,y,\rho)$, $A^p(d,x,y,\rho)$ are the MVP values obtained with FEM and respectively.

In the following, in order to improve the accuracy of the obtained results and to simplify the implementation process of the applied artificial intelligence technique, the authors propose an alternative by using a Neural Network solution instead of the presented Fuzzy Logic Block presented in (Satsios et al. 1999a, 1999b).

2.3 MatLab implementation of prosed neural networks

To identify the optimal neural network solution different feed-forward and layer recurrent architectures were evaluated. To implement these neural network architectures the *Neural Network Toolbox* of the *MatLab* software application was used. This software was chosen because it enables the creation of almost all types of NN from perceptrons (single layer networks used for classification) to more complex architectures of feed-forward or recurrent networks. To create a feed-forward neural network in *MatLab* the following function can be called from command line:

$$\text{net = newff(P,T,S,TF,BTF,BLF,PF)} \tag{7}$$

where:

- P – is a RxQ1 matrix of Q1 representative R-element input vectors;
- T – is a SNxQ2 matrix of Q2 representative SN-element target vectors;
- S – is a vector representing the number of neurons in each hidden layer;
- TF – is a vector representing the transfer function used for each layer;
- BTF – is the back propagation function used to train the NN;
- BLF – is the weight/bias learning function;
- PF – is performance evaluation function.

A similar function can be called to create a layer recurrent neural network:

$$\text{net = newlrn(P,T,S,TF,BTF,BLF,PF)} \tag{8}$$

Once a neural network is created, to train it, the following *Matlab* function can be used:

$$\text{train(net,P,T)} \tag{9}$$

where:

- net – is the neural network that has to be trained;
- P – is a RxQ1 matrix of Q1 representative R-element input vectors;
- T – is a SNxQ2 matrix of Q2 representative SN-element target vectors;

These to functions also provide a pre-processing of the four input parameters: d the separation distance between EPL and MP (which varies between 70 m and 2000 m), ρ the soil resistivity (which varies between 30 Ωm and 1000 Ωm) and (x,y) the coordinates of the point where the MVP is wanted to be evaluated (which varies between 0 and 2100 m, respectively between 0 m and -30 m); by scaling them in the range of [-1,+1].

To train the different NN architectures the Levenberg-Marquardt training method and the descendent gradient with momentum weight learning rule has been implemented. As training data base a set of MVP values evaluated with FEM and presented in (Satsios et al. 1999a, 1999b) were used. These MVP values were calculated in different points up to 15 different problem geometries (soil resistivity, separation distance) obtaining a set of 37 input/output pairs used to train the proposed NN. Table 1 presents some of the training data sets.

No	d [m]	x [m]	y [m]	ρ [Ω*m]	MVP	
					Amp. 10^{-5} [Wb/m]	**Phase [°]**
1	70	70	-15	30	36.1	-22.8
5	800	818.25	-13.5	30	3.88	-82.61
9	400	384.81	-7.82	70	17.2	-44.46
14	70	40	0	100	55.9	-18.53
18	1000	1022.5	0	100	7.23	-67.27
23	300	290.26	-15.8	500	35.5	-26.74
28	700	670	-22.5	700	26	-33.74
30	150	150.55	-16.99	900	53	-19.7
33	1500	1499.1	-17.48	900	15.6	-46.35
37	2000	2030	-5	1000	12.2	-52.73

Table 1. Training data sets.

Once the NN are trained they can provide automatically the corresponding output values for any combination of input parameters by applying the following *MatLab* function:

$$\text{sim(net,X)} \tag{10}$$

where *net* is the implemented neural network and *X* is a set of input values.

2.4 Results obtained with feed-forward neural networks

To determine precisely the magnetic vector potential in each point of the studied domain, the amplitude and the phase of the MVP has to be evaluated. Considering the different variation range: $10^{-6}\div10^{-4}$ [Wb/m] for amplitude, and -180°÷180° for phase, the authors chose to implement two different neural networks - one for amplitude and one for phase -

instead of implementing a single NN that would provide both amplitude and phase. Also the output values for NN which returns the MVP amplitude were scaled from $10^{-6}\div10^{-4}$ to $0.1\div100$, so the final output values has to be multiplied by 10^{-5} to obtain the actual MVP amplitude value in [Wb/m].

Initially, feed-forward neural networks with one output layer and one hidden layer were tested (figure 6). Some of the obtained results were already presented in (Micu et al. 2009) and (Czumbil et al. 2009). In the following a more detailed study is presented.

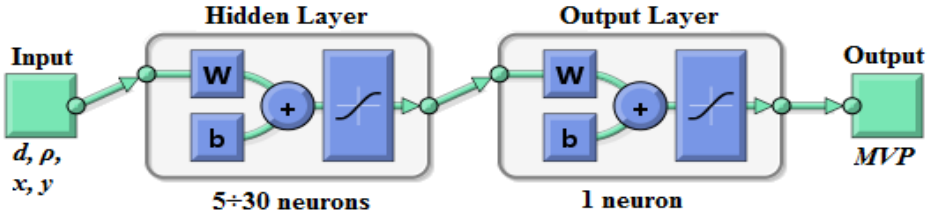


Fig. 6. Implemented feed-forward network architecture.

The number of neurons in the hidden layer was varied from 5 to 30 with a step of 5 neurons. The transfer function of the output layer was set to *purelin* (the linear transfer function) and the transfer function on the hidden layer was varied between *tansig* (the hyperbolic tangent sigmoid function), *logsig* (the logarithmic sigmoid function) and *purelin*. Also, the performance evaluation function was varied between *mse* (mean square error), *msereg* (mean square error with regularization performance) and *sse* (sum squared error).

After the implementation and training, the proposed feed-forward networks were submitted to a testing process. The error between the output values generated by NN and the magnetic vector potential evaluated with FEM for the training data sets was determined. Analysing the obtained errors, it was concluded that none of the tested architectures having the *purelin* transfer function on the hidden layer, had provided acceptable results. The average evaluation error was around 10% and the achieved maximum error was greater than 25%. For all the other NN architectures, the evaluation error for the training data sets was neglectable.

No	d [m]	x [m]	y [m]	ρ [Ω*m]	MVP	
					Amp. 10^{-5} [Wb/m]	**Phase [°]**
1	70	40	-15	100	53.8	-19.34
2	70	81.66	-27.03	30	32.90	-25.57
3	400	392.25	-25.56	70	16.7	-46.05
4	300	281.66	-27.03	500	37.5	-25.93
5	700	690.36	-15.80	700	25.6	-34.07
6	1000	1007.50	0	70	5.68	-72.98
7	1000	1015	-30	100	7.16	-69.22
8	1500	1524.77	-6.93	900	15.40	-46.56

Table 2. Testing data sets.

Since the main goal was the implementations of a suitable NN, that would provide accurate solutions for any new problem geometry, a second testing database was used to select the optimal NN architecture. This second database is a totally new set of data, which was not applied to NN during the training process (table 2).

Analyzing the average and maximum evaluation errors obtained for the testing data sets, in case of the neural network which would evaluate the amplitude of the magnetic vector potential, two possible NN architectures were determined (AmpFfNN2 and AmpFfNN7). The first one (AmpFfNN2) has 10 neurons with *tansig* transfer function on the hidden layer and uses an *mse* performance evaluation function. In this case the obtained average evaluation error for the testing data set is 0.71% with a maximum 1.72% achieved evaluation error. Figure 7 presents the evaluation error obtained for both training and testing data sets.

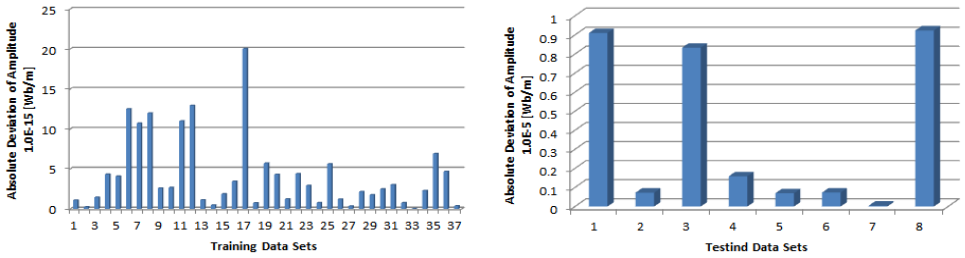


Fig. 7. Absolute evaluation error for AmpFfNN2 network.

The second possible solution (AmpFfNN7) for the amplitude network has 5 neurons with *logsig* transfer function on the hidden layer and uses an *mse* performance evaluation function. In this case the obtained average evaluation error for the testing data set is 0.77% with a maximum 2.50% achieved evaluation error. Figure 8 presents the evaluation error obtained for both training and testing data sets.

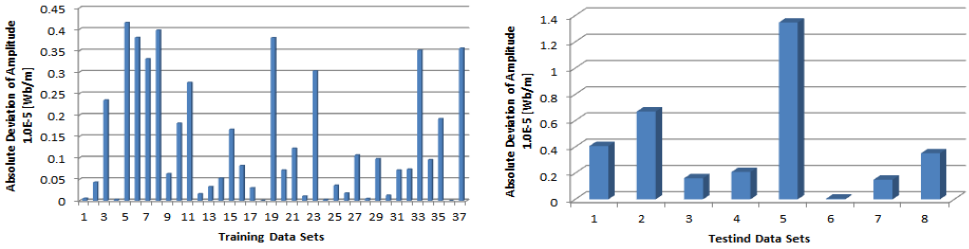


Fig. 8. Absolute evaluation error for AmpFfNN7 network.

Comparing the result for both possible amplitude NN architectures it can be observed that AmpFfNN2 provides better results for both training and testing data sets.

Based on the obtained maximum and average evaluation errors for neural networks implemented for MVP phase evaluation, two possible optimal NN architectures were determined (PhaseFfNN38 and PhaseFfNN43). The first one (PhaseFfNN38) has 10 neurons with *tansig* transfer function on the hidden layer and uses an *sse* performance evaluation

function. In this case the obtained average evaluation error for the testing data set is 1.55% with a maximum 4.02% achieved evaluation error. Figure 9 presents the evaluation error obtained for both training and testing data sets.

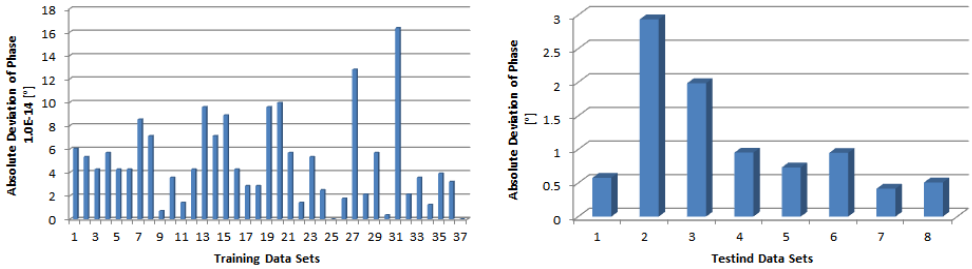


Fig. 9. Absolute evaluation error for PhaseFfNN38 network.

The second possible solution (PhaseFfNN43) for the amplitude network has 5 neurons with *logsig* transfer function on the hidden layer and uses an *sse* performance evaluation function. In this case the obtained average evaluation error for the testing data set is 1.19% with a maximum 5.47% achieved evaluation error. Figure 10 presents the evaluation error obtained for both training and testing data sets.

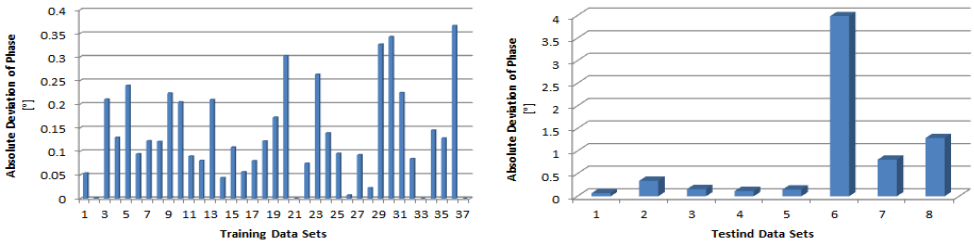


Fig. 10. Absolute evaluation error for PhaseFfNN43 network.

Comparing the result for both possible phase NN architectures it can be observed that generally PhaseFfNN43 provides better results for the testing data sets. However, considering the fact that for the training data sets PhaseFfNN43 provides evaluation errors in range of 0.25 degrees while PhaseFfNN38 provides almost none existing evaluation errors, the optimal NN solution could be considered PhaseFfNN38.

2.5 Results obtained with recurrent neural networks

To find the best NN solution which would provide the most accurate results different layer recurrent architecture were also tested. Some of the results were presented in (Micu et al. 2011) but a more detailed study is given in the following.

A layer recurrent feed-forward neural network with one output layer and one hidden layer is considered (figure 11). The number of neurons in the hidden layer was varied from 5 to 30 with a step of 5 neurons. The transfer function of the output layer was set to *purelin* (the linear transfer function) and the transfer function on the hidden layer was varied between *tansig* (the

hyperbolic tangent sigmoid function) and *logsig* (the logarithmic sigmoid function). Also performance evaluation function was varied between *mse* (mean square error), *msereg* (mean square error with regularization performance) and *sse* (sum squared error).

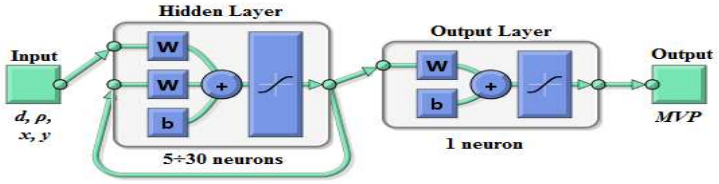


Fig. 11. Implemented layer recurrent network architecture.

Comparing the average and maximum evaluation errors obtained for the testing data sets, in case of the neural network which evaluates the amplitude of MVP, three different NN architectures (AmpLrnNN8, AmpLrnNN19 and AmpLrnNN43) were determined as possible solutions.

The first one (AmpLrnNN8) has 10 neurons with *logsig* transfer function on the hidden layer and uses an *mse* performance evaluation function. In this case the obtained average evaluation error for the testing data set is 0.35% with a maximum 1.08% achieved evaluation error. Figure12 presents the evaluation error obtained for both training and testing data sets.

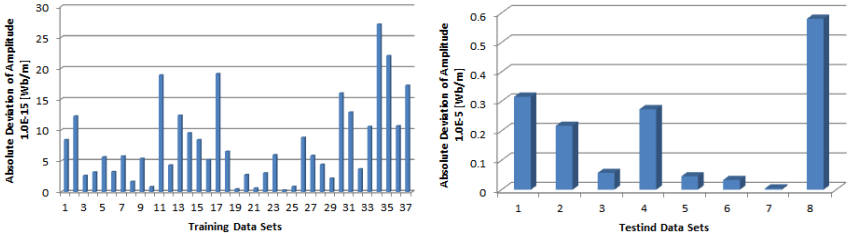


Fig. 12. Absolute evaluation error for AmpLrnNN8 network.

The second possible solution (AmpLrnNN19) for the amplitude network has 5 neurons with *logsig* transfer function on the hidden layer and uses an *msereg* performance evaluation function. In this case the obtained average evaluation error for the testing data set is 0.65% with a maximum 1.12% achieved evaluation error. Figure 13 presents the evaluation error obtained for both training and testing data sets.

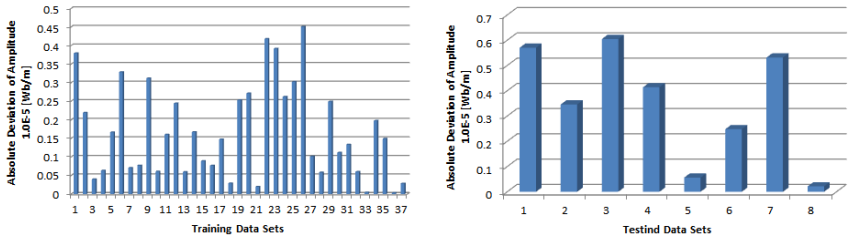


Fig. 13. Absolute evaluation error for AmpLrnNN19 network.

The third possible solution (AmpLrnNN43) for the amplitude network has 5 neurons with *tansig* transfer function on the hidden layer and uses an *sse* performance evaluation function. In this case the obtained average evaluation error for the testing data set is 0.47% with a maximum 1.26% achieved evaluation error. Figure 14 presents the evaluation error obtained for both training and testing data sets.

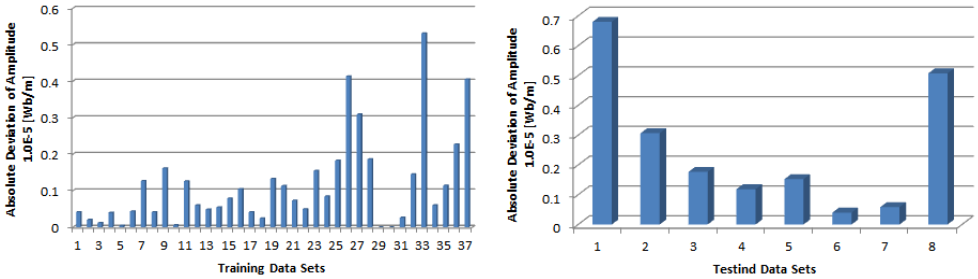


Fig. 14. Absolute evaluation error for AmpLrnNN43 network.

Comparing the results from figures 12, 13 and 14 it can be observed that the most accurate solutions will be obtained for AmpLrnNN8 neural network.

Studying the maximum and average evaluation errors obtained for neural networks implemented to evaluate the phase of MVP, three NN architectures (PhaseLrnNN8, PhaseLrnNN19 and PhaseffNN44) were determined as possible optimal solution.

The first one (PhaseLrnNN8) has 10 neurons with *logsig* transfer function on the hidden layer and uses a *mse* performance evaluation function. In this case the obtained average evaluation error for the testing data set is 1.16% with a maximum 4.21% achieved evaluation error. Figure 15 presents the evaluation error obtained for both training and testing data sets.

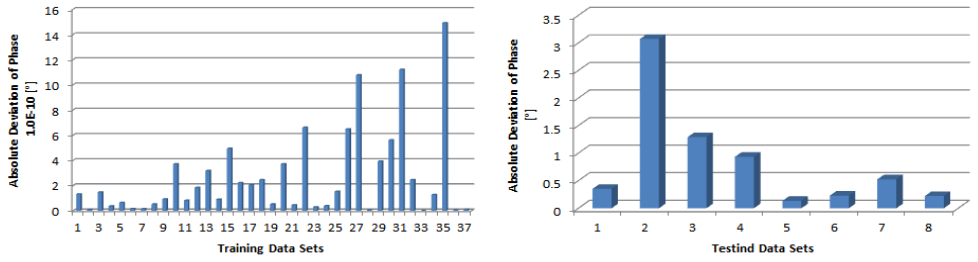


Fig. 15. Absolute evaluation error for PhaseLrnNN8 network.

The second possible solution (PhaseLrnNN19) for the amplitude network has 5 neurons with *tansig* transfer function on the hidden layer and uses a *msereg* performance evaluation function (ffNN2). In this case the obtained average evaluation error for the testing data set is 1.19% with a maximum 3.02% achieved evaluation error. Figure 16 presents the evaluation error obtained for both training and testing data sets.

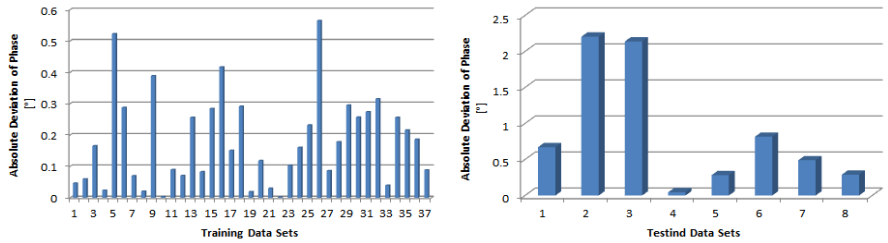


Fig. 16. Absolute evaluation error for PhaseLrnNN19 network.

The third possible solution (PhaseLrnNN44) for the amplitude network has 10 neurons with *logsig* transfer function on the hidden layer and uses a *sse* performance evaluation function. In this case the obtained average evaluation error for the testing data set is 1.18% with a maximum 3.58% achieved evaluation error. Figure 17 presents the evaluation error obtained for both training and testing data sets.

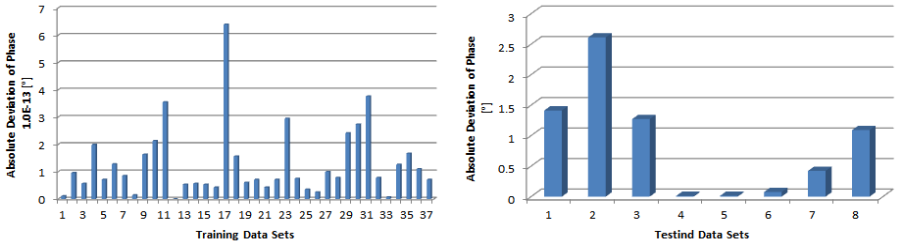


Fig. 17. Absolute evaluation error for PhaseLrnNN43 network.

Analysing the results shown in figures 15, 16 and 17 the authors concluded that the optimal layer recurrent NN architecture solution to evaluate the phase of the magnetic vector potential it is PhaseLrnNN19 network structure.

2.6 Discussions

Based on the maximum and average evaluation errors obtained for the implemented NN architectures the authors has concluded that it should be used a NN structures that have even *tansig* either *logsig* transfer function implemented on the hidden layer. Generally, a *tansig* transfer function will provide much better training results then a *logsing* function, but for totally new input values could provide less accurate results. Also it was observed that a higher number of neurons did not necessary provide more accurate results and instead of predicting the MVP values for new problem geometries it would identify the closest training input/output pair.

Studying the results provided by the identified optimal NN architectures in case of layer recurrent networks (AmpLrnNN8 for MVP amplitude evaluation, respectively PhaseLrnNN19 for MVP phase evaluation) and comparing to the MVP solutions provided by the optimal feed-forward architectures (AmpFfNN2 and respectively PhaseFfNN38) it can be observed that even if the studied problem does not necessary require the implementation of recurrent neural networks, we can get more accurate solutions by using recurrent networks.

Comparing the results obtained with the implemented neural networks (figure 7÷10 and 12÷15) with those provided by the fuzzy logic block presented in (Satsios et al. 1999a, 1999b) (figure 5) we can observe a 50% or more accuracy increase in determining MVP amplitude and phase, depending on the implemented neural network architecture.

Once the magnetic vector potential is evaluated, the self and mutual impedance matrix, which describes the inductive coupling between the electrical power line and underground pipeline, can be evaluated using the relationships presented in (Christoforidis et al. 2003, 2005). After that the equivalent electrical circuit of the studied EPL-MP problem can be solved to obtain the induced AC voltage.

3. Self and mutual inductance matrix evaluation in case of a three layer earth

Considering the accuracy of the results for the MVP obtained from the implemented neural networks, the authors started to develop a neural network solution to evaluate directly the self and mutual impedance matrix, which describe the inductive coupling. In this case a more complex EPL-MP interference problem had been chosen for study.

3.1 Studied electromagnetic interference problem

An underground gas transportation pipeline runs in the same right of way with a 220kV/50Hz electrical power line (figure 18). In order to study more realistic problem geometries is considered a multilayer soil (three vertical layers) with different resistivities.

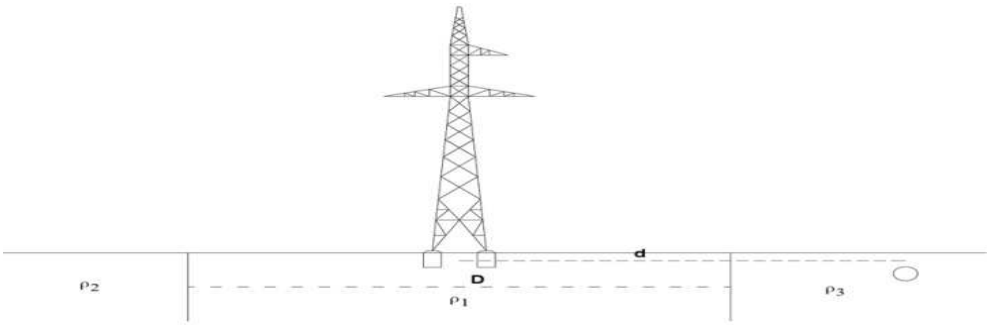


Fig. 18. Right of way configuration

The pipeline is considered to be buried at 2 m depth and having an 0.195 m inner radius, 5 mm thickness and 5 cm bitumen coating. The EPL phase wires are placed on triagonal single circuit IT.Sn102 type towers with one sky wire.

3.2 Neural network implementation

In order to do not redo each time the finite element calculation when different problem geometry has to be studied, the authors had decided to implement a neural network solution to evaluate the self and mutual inductance matrix for any possible problem geometries. As input values the following geometrical parameters has been selected:

- d - distance between EPL and MP (which varies between 0 m and 1000 m);
- ρ_1 - resistivity of middle layer earth (which varies between 30 Ωm and 1000 Ωm);
- ρ_2 , ρ_3 ($\rho_2 = \rho_3$) - resistivity of left and right side earth layer (which varies as ρ_1);
- D - earth middle layer width (which varies between 50 m and 1100 m).

During the pre-processing stage of the proposed neural network solutions implementation all the input parameters were automatically scaled by MatLab in the [-1,+1] range.

The outputs of the proposed neural network will be the impedance matrix elements. Considering the fact that the impedance matrix is a symmetrical matrix, the output values are the matrix elements above the main diagonal. For the proposed EPL-MP problem with one pipeline, three phase wires and one sky wire the matrix elements will be: Z_{11} , Z_{12} , Z_{13} , Z_{14} , Z_{15} , Z_{22} , Z_{23} , Z_{24} , Z_{25} , Z_{33} , Z_{34} , Z_{35} , Z_{44} , Z_{45} , Z_{55} , where Z_{ii} represents the self impedance of conductor i and Z_{ij} the mutual impedance between conductor i and j ($i = \overline{1,3}$ represents EPL phase wires, $i = 4$ represents EPL sky wire and $i = 5$ represents MP).

After analysing in detail the impedance matrices for different EPL-MP problem geometries the authors concluded that in order to increase NN results accuracy and to reduce training time it will be better to implement different NN for the real and imaginary part of each impedance. Also for accuracy improvement were implemented 3 different networks for the impedance matrix elements: NN1 for all conductors self impedances (Z_{11} , Z_{22} , Z_{33} , Z_{44} , Z_{55}), NN2 for the mutual impedances between the pipeline and the conductors(Z_{15} , Z_{25} , Z_{35} , Z_{45}), NN3 for the other mutual impedance elements.

Case No.	d [m]	D [m]	ρ_2 [Ω*m]	ρ_1 [Ω*m]	ρ_3 [Ω*m]	Case No.	d [m]	D [m]	ρ_2 [Ω*m]	ρ_1 [Ω*m]	ρ_3 [Ω*m]
8	5	60	500	50	500	2301	20	550	30	250	30
373	100	60	500	750	500	2532	100	550	100	500	100
875	20	120	100	750	100	2914	5	1050	10	250	10
1231	500	120	100	30	100	3274	100	1050	500	1000	500
1391	0	240	50	10	50	3545	750	1050	30	750	30
1891	250	240	500	30	500	4320	750	1500	50	10	50
2134	0	550	50	250	50	4442	1000	1500	250	1000	250

Table 3. Training EPL-MP problem geometries.

To train the proposed NN a training data base was created based on the impedance matrices obtained using FEM for different EPL-MP problem geometries. In order to create a useful training database approximately 5000 different EPL-MP problem geometries were simulated varying the EPL-MP separation distance from 0 m to 1000 m, the earth layers resistivity from 10 Ωm to 1000 Ωm and the middle layer width from 50 m to 1500 m. Table 3 presents some of the different EPL-MP problem geometries used to train the proposed NN.

Case No.	d [m]	D [m]	ρ_2 [Ω*m]	ρ_1 [Ω*m]	ρ_3 [Ω*m]	Case No.	d [m]	D [m]	ρ_2 [Ω*m]	ρ_1 [Ω*m]	ρ_3 [Ω*m]
1	310	800	900	850	900	16	490	1100	300	400	300
2	15	400	850	450	850	17	170	700	300	350	300
3	105	1100	550	550	550	18	150	700	500	500	500
4	350	900	500	800	500	19	240	500	80	750	80
5	250	800	150	150	150	20	125	800	300	600	300
6	60	800	500	900	500	21	420	100	550	20	550
7	340	400	600	150	600	22	75	400	350	700	350
8	65	400	650	350	650	23	105	1200	250	950	250
9	170	800	650	750	650	24	100	700	650	850	650
10	55	1000	900	400	900	25	85	400	140	160	140
11	40	200	600	800	600	26	300	400	900	100	900
12	115	800	800	800	800	27	145	300	350	900	350
13	120	900	750	350	750	28	15	500	140	700	140
14	135	400	180	500	180	29	100	1300	300	180	300
15	310	800	900	850	900	30	10	1000	200	750	200

Table 4. Training EPL-MP problem geometries.

In order to find the optimal neural network solution which will provide the most accurate results, the authors have implemented and tested different NN architectures. To test the implemented neural networks, the training database and a totally different data set that was not applied during the training process, were used. The error between the solutions provided by each implemented NN and the self and mutual impedance matrices are determined to identify the optimal architecture. Table 4 presents the randomly generated EPL-MP problem geometries used to test the implemented NN

3.3 Tested feed-forward architectures

To identify the optimal solution for each of the proposed neural networks, different feed-forward architectures with one output layer and two hidden layers were implemented (figure 19). Based on the experience gained after implementing the neural network for MVP calculation, the transfer function for the output layer has been chosen to be *purelin* (linear function) and *tansig* (hyperbolic tangent sigmoid function) for the hidden layers. The number of neurons was varied from 5 to 30 for the first hidden layer and from 5 to 20 for the second hidden layer. The performance evaluation function was set to *mse* (mean square error) and the descendent gradient with momentum weight learning rule was selected to train the neural networks using the Levenberg-Marquardt method.

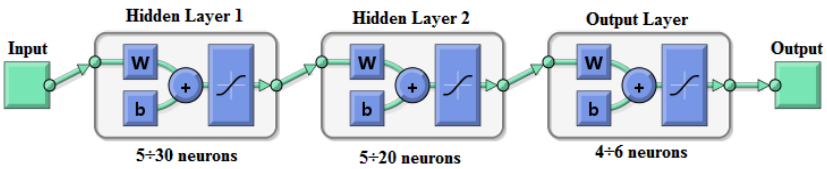


Fig. 19. Implemented feed-forward network architecture.

The training process took around 1 to 5 minutes for each implemented neural network. Once the proposed neural network architectures were trained, to identify the optimal solution, the error between the output values provided by NN and the finite element solutions was calculated and compared for each NN.

In case of NN used to evaluate the real part of the self-impedances, the optimal NN1 architecture is a feed forward NN with two hidden layers: 30 neurons on the first hidden layer and 10 neurons in the second hidden layer. The average error is 0.048% for the training data set and respectively 0.064% for the testing data set, with a maximum achieved error of 0.8% and 0.3% for the training and respectively the testing data sets. Figure 20 presents the error distribution for training and testing problem geometry data sets.

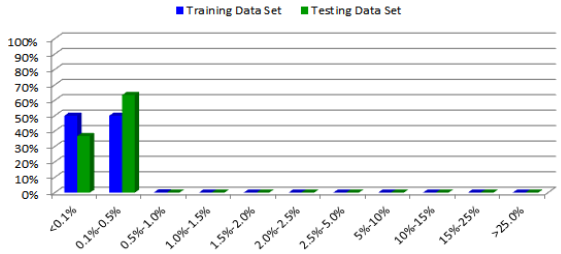


Fig. 20. Percentege error distribution for Real Part NN1.

For the NN used to evaluate the imaginary part of the self-impedances, the optimal NN1 architecture is a feed forward NN with two hidden layers: 20 neurons on the first hidden layer and 20 neurons in the second hidden layer. The average percentage error is 0.14% for the training data set and respectively 0.129% for the testing data set, with a maximum achieved percentage error of 5.24% and 0.85% for the training and respectively the testing data sets. Figure 21 presents the percentage error distribution for training and testing problem geometry data sets.

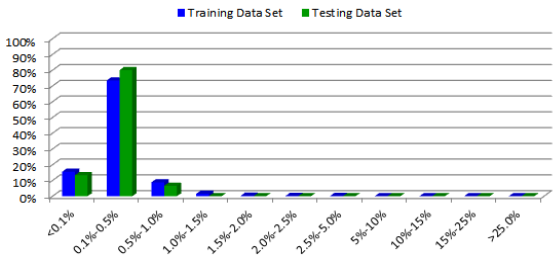


Fig. 21. Error distribution for Imaginary Part NN1.

In case of NN used to evaluate the real part of the mutual impedances between EPL conductors (phase wires and sky wires), optimal NN3 architecture is a feed forward NN with two hidden layers: 20 neurons on the first hidden layer and 20 neurons on the second hidden layer. The average error is 0.014% for the training data set and respectively 0.034% for the testing data set, with a maximum achieved error of 3.12% and 0.14% for the training

and respectively the testing data sets. Figure 22 presents the error distribution for training and testing problem geometry data sets.

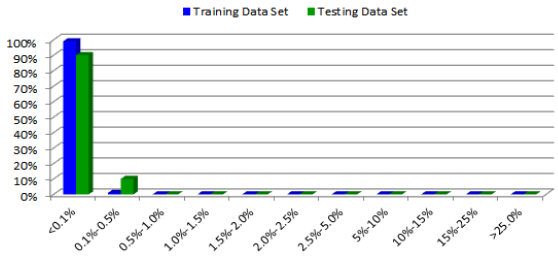


Fig. 22. Error distribution for Real Part NN3

For the NN used to evaluate the imaginary part of the mutual impedances between EPL conductors (phase wires and sky wires) optimal NN3 architecture is a feed forward NN with two hidden layers: 30 neurons on the first hidden layer and 20 neurons in the second hidden layer. The average error is 0.073% for the training data set and respectively 0.097% for the testing data set, with a maximum achieved error of 2.98% and 0.46% for the training and respectively the testing data sets. Figure 23 presents the error distribution for training and testing problem geometry data sets.

Unfortunately in case on the NN used to evaluate the real part of the mutual impedances between MP and other conductors, none of the tested NN architectures provided acceptable results. This was caused by the fact that the real part of mutual impedances between MP and other conductors varies in a very large range from 1E-11 to 1E-6.

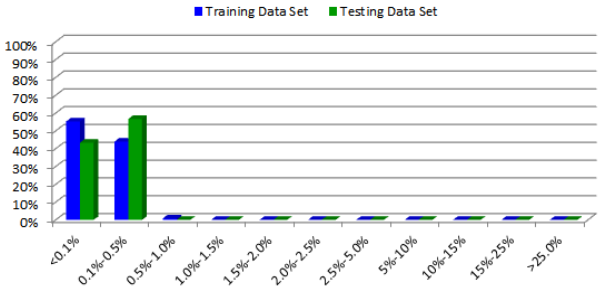


Fig. 23. Error distribution for Imaginary Part NN3.

After a complete analysis of the real and imaginary part of the mutual impedance between MP and other conductors another approach was used; two NN were implemented to evaluate the amplitude and argument of the mutual impedances, instead of the impedance real and imaginary part.

In case of NN used to evaluate the amplitude of the mutual impedances between MP and other conductors the optimal NN2 architecture it was identified as a feed forward NN with two hidden layers: 30 neurons on the first hidden layer and 25 neurons in the second hidden layer. The average error is 0.066% for the training data set and respectively 0.087% for the

testing data set, with a maximum achieved error of 17.95% and 0.43% for the training and respectively the testing data sets. Figure 24 presents the error distribution for training and testing problem geometry data sets.

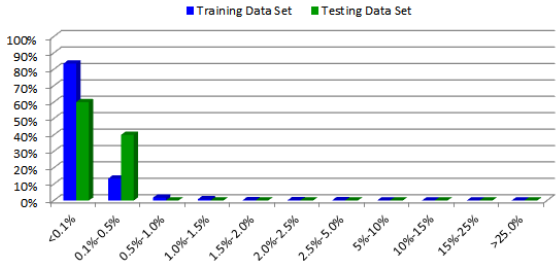


Fig. 24. Error distribution for Amplitude NN2.

For the NN used to evaluate the argument of the mutual impedances between MP and the other conductors the optimal NN2 architecture is a feed forward NN with two hidden layers: 20 neurons on the first hidden layer and 20 neurons in the second hidden layer. The average error is 0.249% for the training data set and respectively 0.313% for the testing data set, with a maximum achieved error of 6.71% and 1.43% for the training and respectively the testing data sets. Figure 25 presents the error distribution for training and testing problem geometry data sets.

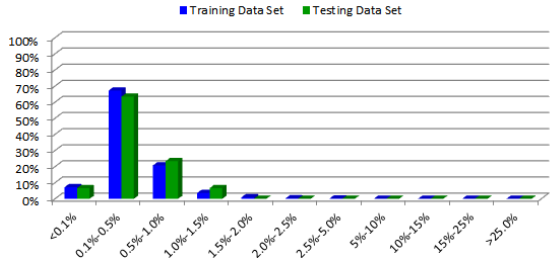


Fig. 25. Error distribution for Argument NN2.

After identifying the optimal architecture of the neural networks these two NN were unified in a virtual black box in order to evaluate mutual impedances real and imaginary part. This unification procedure has as secondary unwanted result a change in the final complex mutual impedance evaluation error. Thus the average error become 0.665% for the training data set and respectively 0.407% for the testing data set, with a maximum achieved error of 18.728% and 1.107% for the training and respectively the testing data sets. Figure 26 presents the global evaluation error distribution of complex mutual impedance for both training and testing problem geometry data sets.

Also, the authors implemented and tested some layer recurrent neural network architectures. But unfortunately because of the large training database with very different output values the training process proved to be very time consuming (more than one hour) and the obtained results had the same accuracy as the previous feed-forward networks.

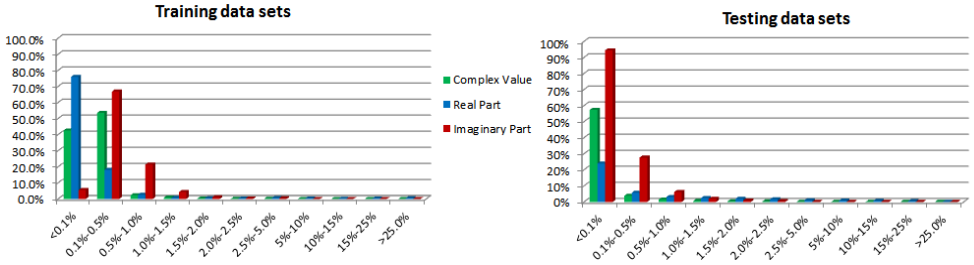


Fig. 26. Global error distribution for NN2 optimal networks.

4. Conclusions

In this chapter the authors implement neural networks based artificial intelligence techniques in the study of electromagnetic interference problems (right of way EPL-MP) focusing on finding an easier method to identify the optimal solution.

To solve the differential equation which describes the couplings between EPL and nearby MP usually a finite element method is used. This FEM calculation needs excessive computational time especially if different problem geometries of the same interference problem have to be studied.

In order to reduce computation time the authors proposed two neural networks based artificial intelligence techniques to scale the results from a set of known problem geometries to any new problem geometry. The proposed neural networks were implemented for specific EPL-MP interference problems. The first one evaluates the magnetic vector potential in case of a phase to earth EPL fault and second one determines the self and mutual impedance matrix in case of a three vertical layer earth.

The obtained results with the proposed neural network solutions proved themselves to be very accurate. Thus ad shown neural network based solution to study EPL-MP interference problems could be a very effective one, especially if we take into account the fact that the solutions provided are obtained instantaneously ones they are properly trained.

Also it has been shown that even there is a special requirement to use recurrent neural networks, these NN architectures could provide more accurate solutions than the basic feed-forward structures.

5. Acknowledgment

The work was supported by TE_253/2010_CNCSIS project – "Modelling, Prediction and Design Solutions, with Maximum Effectiveness, for Reducing the Impact of Stray Currents on Underground Metallic Gas Pipelines", No. 34/2010.

6. References

Al-Badi, A., Ellithy, K. & Al-Alawi, S., (2005), "An Artificial Neural Network model for Predicting gas pipeline Induced Voltage caused by Power Lines under Fault

Conditions", *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol. 24, No. 1, (2005), pp. 69-80,ISSN: 0332-1649.

Al-Badi, A., Ellithy, K. & Al-Alawi, S., (2007), "Prediction of Voltage on Mitigated Pipelines Paralleling Electric Transmission Lines using an Artificial Neural Network", *The Journal of Corrosion Science and Engineering*, Vol. 10, Preprint 28, (March 2007), ISSN: 1466-8858.

Baboian, R. (Ed.), (2002), *NACE Corrosion Engineer's Reference Book,* NACE International, ISBN: 1-57590-139-0, Huston, USA.

Christoforidis, G. C., Labridis, D. P. & Dokopoulos, S., (2003), "Inductive Interference Calculation on Imperfect Coated Pipelines due to Nearby Faulted Parallel Transmission Lines", *Electric Power Systems Research,* Vol. 6, No. 2., (August 2003), pp. 139-148, ISSN: 0378-7796.

Christoforidis, G. C., Labridis, D. P. & Dokopoulos, S., (2005), "A Hybrid method for calculating the Inductive Interference caused by Faulted Power Lines to nearby Buried Pipelines", *IEEE Trans. on Power Delivery,* Vol. 20, No. 2, (April 2005), pp. 1465-1473, ISSN: 0272-1724.

Czumbil, L. , Micu, D. D. & Ceclan, A., (2009), "Artificial Intelligence Techniques Applied to Electromagnetic Interference Problems", *IFMBE Proceedings*,ISBN:978-3-642-22585-7, Vol. 26, Cluj-Napoca, Romania, September 23-26, 2009

CIGRÉ Working Group 36.02, (1995), *Guide on the Influence of High Voltage AC Power Systems on Metallic Pipelines*.

Collet, E., Delores, B., Gabillard, M. & Ragault, I., (2001), "Corrosion due to AC Influence of Very High Voltage Power Lines on Polyethylene-Coated Steel Pipelines Evaluation of Riks-Preventive Measures", *Anti-Corrosion Methods and Materials*, Vol. 48, No. 4, (August 2001), pp. 221-226, ISSN: 0003-5599.

Damousis, I. G., Satsios, K. J., Labridis, D. P. & Dokopoulos, P. S., (2002), "Combiend Fuzzy Logic and Genetic Algorithm Techniques – Application to an Electromagnetic Field Problem", *Fuzzy Sets and Systems*, Vol. 129, No. 3, (August 2002), pp. 371-386, ISSN: 0165-0114.

Dawalibi, F. P. & Southey, R. D., (1989), "Analysis of electrical interference from power lines to gas pipelines. Part I – Computation methods", *IEEE Trans. on Power Delivery,* Vol. 4, No. 3, (July 1989), pp. 1840-1849, ISSN: 0272-1724.

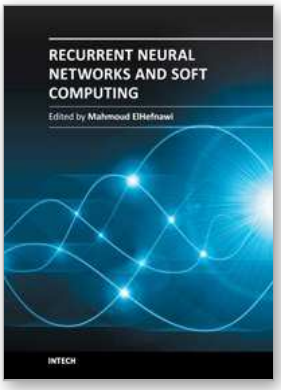
Micu, D. D., Czumbil, L., Christoforidis, G., Ceclan, A., Darabant, L. & Stet, D. (2009), "Electromagnetic interferences between HV power lines and metallic pipelines evaluated with neural network technique", *Proceedings of the 10th International Conference on Electrical Power Quality and Utilization (EPQU)*,ISBN: 978-1-4244-5171-5 , Lodz, Poland, September 15-17, 2009.

Micu, D. D., Czumbil, L., Christoforidis, G., & Ceclan, A. (2011), "Layer Recurrent Neural Network Solution for an Electromagnetic Interference Problem", *IEEE Trans. on Magnetics*, Vol. 47, No. 5, (May 2011), pp. 1410-1413, ISSN: 0018-9464.

Papagiannis, G. K., Triantafyllidis, D. G., Labridis, D. P., (2000) "A One-Step Finite Element Formulation for the Modelling of Single and Double Circuit Transmission Lines", *IEEE Trans. On Power Systems*, vol. 15, Nr. 1, 2000, pp.33-38.

Satsios, K. J., Labridis, D. P. & Dokopoulos, P. S., (1999), "An Artificial Intelligence System for a Complex Electromagnetic Field Problem: Part I – Finite Element Calculations and Fuzzy Logic Development", *IEEE Trans. on Magnetics*, Vol. 35, No. 1, (January 1999), pp. 516-522, ISSN: 0018-9464.

Satsios, K. J., Labridis, D. P. & Dokopoulos, P. S., (1999), "An Artificial Intelligence System for a Complex Electromagnetic Field Problem: Part II – Method Implementation and Performance Analysis", *IEEE Trans. on Magnetics*, Vol. 35, No. 1, (January 1999), pp. 523-527, ISSN: 0018-9464.



Recurrent Neural Networks and Soft Computing
Edited by Dr. Mahmoud ElHefnawi

ISBN 978-953-51-0409-4
Hard cover, 290 pages
Publisher InTech
Published online 30, March, 2012
Published in print edition March, 2012

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Dan D. Micu, Georgios C. Christoforidis and Levente Czumbil (2012). Artificial Intelligence Techniques Applied to Electromagnetic Interference Problems Between Power Lines and Metal Pipelines, Recurrent Neural Networks and Soft Computing, Dr. Mahmoud ElHefnawi (Ed.), ISBN: 978-953-51-0409-4, InTech, Available from: http://www.intechopen.com/books/recurrent-neural-networks-and-soft-computing/artificial-intelligence-techniques-applied-in-electromagnetic-interference-problems-between-hv-power

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821