Eleco 2003 3rd Int. Conf. on Electrical and Electronics Engineering, Bursa, Turkey, pp. 235-239, December 3-5, 2003

DETERMINATION OF CRITICAL IMPULSE BREAKDOWN VOLTAGE BY ARTIFICIAL NEURAL NETWORK

Özcan Kalenderli[1] Suna Bolat[2] Hasbi Ismailoglu[3]

[1, 2]*Electrical & Electronics Faculty, Istanbul Technical University, Istanbul, Türkiye*
[3]*Electrical Engineering Department, Kocaeli University, Kocaeli, Türkiye*

Keywords: *High voltage, Impulse voltage, Breakdown voltage, Artificial Neural Network*

ABSTRACT

The paper presents the determination of critical impulse breakdown voltage and standard deviation using an Artificial Neural Network (ANN) algorithm. The results, obtained using ANN method, are compared with those of conventional methods. The algorithm performs fast and gives acceptable results.

1. INTRODUCTION

There are many publications those describe the methods to determine the critical impulse breakdown voltage (50% breakdown voltage) and standard deviation from breakdown probability data for a given insulation [1-4]. Almost all of these papers assume that; when the voltage applied to the test object is changed, the probability of breakdown will also change. The relationship between the applied voltage and its breakdown probability is therefore approximated by the cumulative normal distribution function.

The methods about to determine critical breakdown voltages and standard deviations, described in the literature, are generally based on the up and down method [5-10]. And some of them utilize the concepts of the graphical techniques using normal probability paper or maximum likelihood [4] to estimate the parameters of breakdown probability distribution. Another method is based on curve fitting method of first order, by least squares method [11-15].

In this study, critical breakdown voltages and standard deviations are determined by training an Artificial Neural Network (ANN). Using ANN algorithm for the problem instead of conventional methods, one can reach to the results very fast, without computational difficulties. After the ANN trained for once, it can give the breakdown voltage and standard deviation of any breakdown probability.

For the last couple of years extensive research works has been carried out on the application of ANN in various fields. So, the literature on ANN is growing very rapidly. ANN have been adapted successfully on a very wide range of applications including machine vision, speech processing, sonar analysis, radar analysis, pattern recognition, robotic control etc. In electrical power systems, ANN have been used accurately for load forecasting, security evaluation, capacitor control, alarm processing etc. In high voltage techniques, applications of ANN have been reported for pattern recognition of partial discharges, optimization of electrode and insulator contours, and lightning prediction [16-22].

In this study a new approach to determine critical impulse breakdown voltage based on ANN is presented. The multilayer feedforward network is used for supervised learning with resilient back propagation.

2. ARTIFICIAL NEURAL NETWORKS

An artificial neural network consists of a set of processing elements called neurons that interact by sending signal to one another along weighted connections [22, 23]. The connection weights, which can be determined adaptively, specify the precise knowledge representation. It is not possible to specify the weights beforehand, because the knowledge is distributed over the network. Therefore, a learning procedure is necessary in which the strengths of the connections are modified to achieve the desired form of activation function.

The learning procedure is divided into three types: supervised, reinforced and unsupervised. The type of error signal used to train the weights in the network define these three types of learning. In supervised learning, an error scalar is provided for each output neuron by an external ‘teacher’, while in reinforced learning the network is given only a global punish/reward signal. In unsupervised learning, no external error signal is provided, but instead internal errors are generated between the neurons, which are then used to modify weights [24].

In supervised learning the weights, connecting neurons are set on the basis of detailed error information supplied to the network by an external teacher. In most cases the network is trained using a set of input-output pairs, which are examples of the mapping that the network is required to learn to compute. The learning process may therefore be viewed as fitting a function, and its performance can thus be judged on whether the network can learn the desired function over the interval represented by the training set, and to what extent the network can successfully generalize away from the points that it has been trained on.

2.1. MULTILAYER FEEDFORWARD NETWORK

The simplest network capable of supervised learning is a two-layer feedforward network consisting of an input layer and an output layer. Each neuron of the input layer receives a signal from all input neurons along connections with modifiable weights. But such two-layer feedforward networks can compute only linearly separable functions. However, it has also been shown that a feedforward network with more than one layer of adaptive weights can compute very complex functions.

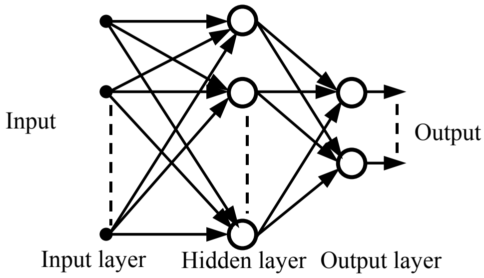

Figure 1. Schematic multilayer feed-forward ANN

The neurons in the network can be divided into three layers: input layer, output layer and hidden layers (Figure 1). It is important to note that in feedforward networks, signals can only propagate from the input layer to the output layer via one or more hidden layers. It should also be noted that only the nodes in the hidden layers and the output layer, which perform activation function, are called ordinary neurons. Since the nodes in the input layer simply pass on the signals from the external source to the hidden layer, they are often not regarded as ordinary neurons.

2.2. RESILIENT PROPAGATION ALGORTIHM

Resilient propagation (Rprop) algorithm is one of the faster back propagation learning algorithms. When the learning process starts, an input pattern is presented to the input neurons for each training set. This pattern is then propagated forward through the entire network, yielding an output pattern from the output neurons. This output pattern is compared with the corresponding target output. The connection weights are then modified according to the deviation of the real output from the target output. This process is repeated until the deviation does not exceed a certain threshold for each training set. The connection weights of the feedforward network are modified in the standard back propagation algorithm on the basis of the minimization of the error by steepest descent. In standard back propagation, the weights are updated proportional to the computed derivative,

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \tag{1}$$

where w_{ij} is the weight of the connection from the i-th unit in the predecessor layer to the j-th unit in the actual layer and η is called the learning rate, usually a real number in the interval (0-1). To avoid oscillations, a momentum term can be added to eq. (1) in the back propagation algorithm. Rprop algorithm considers the local topology of error function E. Although, standard back propagation uses two factors, such as learning rate η and partial derivative of the error function $\partial E/\partial w_{ij}$, Rprop introduces a so called ‘update value’ that determines only the step width of the weight change for each connection weight. Only the sign of the partial derivative $\partial E/\partial w_{ij}$ is taken to determine the direction of the weight change,

$$w_{ij}^{t+1} = w_{ij}^{t} \begin{cases} -\Delta_{ij} & \left[\dfrac{\partial E}{\partial w_{ij}}\right]^{(t)} > 0 \\ +\Delta_{ij} & \left[\dfrac{\partial E}{\partial w_{ij}}\right]^{(t)} < 0 \\ 0 & \left[\dfrac{\partial E}{\partial w_{ij}}\right]^{(t)} = 0 \end{cases} \tag{2}$$

where t is the iteration number and Δ_{ij} is a ‘personal’ update value for each connection weight, such as,

$$\Delta_{ij}^{t} = \begin{cases} -\Delta_{ij}^{t-1}\eta^{+} & \left[\dfrac{\partial E}{\partial w_{ij}}\right]^{(t-1)}\left[\dfrac{\partial E}{\partial w_{ij}}\right]^{(t)} > 0 \\ -\Delta_{ij}^{t-1}\eta^{-} & \left[\dfrac{\partial E}{\partial w_{ij}}\right]^{(t-1)}\left[\dfrac{\partial E}{\partial w_{ij}}\right]^{(t)} < 0 \\ -\Delta_{ij}^{t-1} & \left[\dfrac{\partial E}{\partial w_{ij}}\right]^{(t-1)}\left[\dfrac{\partial E}{\partial w_{ij}}\right]^{(t)} = 0 \end{cases} \tag{2}$$

Whenever the partial derivative of the corresponding weight w_{ij} changes its sign, indicating that the last update value was too big, and the algorithm has jumped over a minimum, the update value Δ_{ij} is decreased by the factor η^{-}. If the derivative retains its sign, the update value is slightly increased in order to accelerate convergence in shallow regions. If there has been a change in the sign of the derivative, the adaptation process is skipped in the following step, in order to prevent too rapid decreased in step width. It has been seen that $\eta^{-} = 1.2$ and $\eta^{+} = 0.5$ give acceptable results. So, these values have been taken in the computation. Initially all the update values Δ_{ij} are set to initial value of $\Delta_0 = 0.07$. The upper value is taken as $\Delta_{max} = 50$.

2.3. NORMALIZATION OF INPUT-OUTPUT DATA

Since the input and output variables of the ANN have different ranges, the feeding of the original data to the network, leads to a convergence problem. It is obvious that the output of the ANN must fall within the interval of (0-1). In addition, input signals should be kept small in order to avoid a saturation effect of the sigmoidal function. So, the input-output patterns are normalized before training the network. Normalization by maximum value is done by dividing input-output variables to the maximum value of the input and output vector components. After the normalization, the input and output variables will be in the range of (0 to 1).

3. APPLICATION OF ANN AND COMPARISON OF THE RESULTS

An example will be given to illustrate the application of ANN and its results will be compared with those of the other methods.

The data, used in our study, received from [2], is shown in Table 1. Here, V_i is the peak value of the applied voltage and p_i is the breakdown probability at V_i. Applying the data to ANN, acceptable results are obtained very fast, because the Rprop algorithm converges much faster than the other learning algorithms.

Table 1. The data used in the study.

V_i (kV)	p_i
13.88	0.10
14.31	0.35
14.67	0.42
14.99	0.72
15.33	0.75

Data, applied to the network as input variables are voltages, those applied in the test and the output variables are the probabilities, obtained at applied voltages. Therefore, ANN includes only one input neuron and one output neuron.

The performance of Rprop algorithm is not very sensitive to the settings of the training parameters. Therefore, detailed studies have not been done to determine the parameters, but the effect of the number of neurons in the hidden layer has been taken into account. Number of the neurons in the hidden layer has changed from 2 to 10.

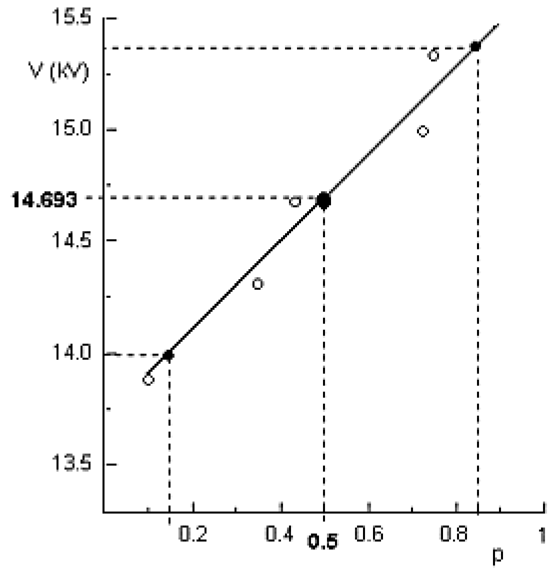

Figure 2. Determination of V_{50} and σ using ANN.

The results show that one hidden layer of 4 neurons gives the best training and test results accuracy. All the above-mentioned studies have been done with one hidden layer, and the results were evaluated after 100 iterations.

Training was continued, with the optimal number of neurons in the hidden layer, until the mean squared error met a fixed value of training accuracy, which is $\%1.10^{-3}$. The training was stopped after 43 iterations where the mean absolute error defined as the difference between expected breakdown voltage and the actual voltage, that is determined by the ANN, is less than 2.7% (Table 2).

Table 2. Comparison of the expected values and ANN outputs

Breakdown probabilities (pi)	Peak value of the applied voltages (V_i) [kV]	
	Expected values	ANN outputs
0.10	13.8800	13.8802
0.35	14.3100	14.3097
0.42	14.6700	14.6718
0.72	14.9900	15.0195
0.75	15.3300	15.2262

Training time for 43 iterations and the test is less than a second. Training process is shown in Figure 3. The results of the critical breakdown voltage (V_{50}) and the standard deviation (σ) are given in Table 3. The values of the first three methods were received from [2], and those of the last one were obtained by ANN, using the data given in Table 1.

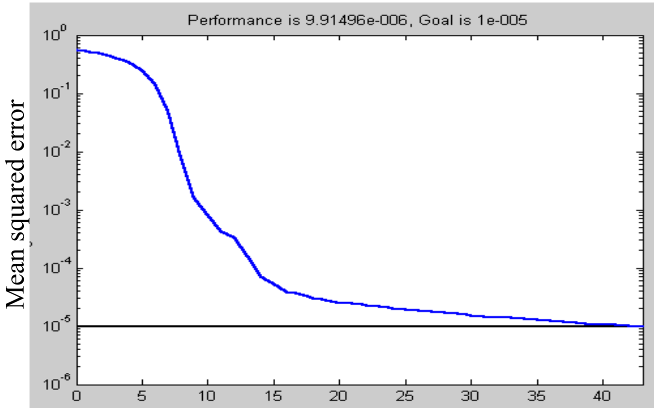

Figure 3. Change of mean squared error with respect to iteration number

The critical breakdown voltage V_{50} obtained by ANN is very close to the mean value of the other three methods. The standard deviation σ is a little smaller than those of others.

Table 3. Comparison of the results.

Method	V50 (kV)	σ (kV)
Maximum Likelihood	14.606	0.732
Graphical	14.730	0.712
Recursive Algorithm	14.713	0.727
Least Squares	14.693	0.694
Neural Network	14.712	0.708

4. CONCLUSIONS

This study presents a new technique based on artificial neural network to determine critical impulse breakdown voltage. A multi-layer feedforward network with resilient propagation learning algorithm is designed for the purpose.

Computational difficulties that arise in the other methods are avoided, and the calculated values are very close to those given in the literature. Also the determination time is very small with the learning process. The 50% breakdown voltage of an electrode configuration and its standard deviation can be easily obtained using ANN with acceptable results.

The method uses a very simple algorithm, and gives accurate values for the critical breakdown voltage and standard deviation, if the breakdown probability data lies within the range of p(V) = 10% and p(V) = 90%.

REFERENCES

1. S. Vibholm, P. Thyregod, The analysis of insulation probabilities by the up-and-down method, IEEE Trans. on Electrical Insulation, Vol. 21, No. 2, pp. 133-136, 1986.
2. O. E. Gouda, A. G. Zeitoun, E. A. Eisawy, New algorithm for determining critical flashover voltage and standard deviation from flashover probability data, ETEP Vol. 1, No. 5, pp. 233-241, 1991.
3. W. Hauschild, W. Mosch, *Statistical Techniques for High Voltage Engineering*, Peter Peregrinus Ltd., London, 1992.
4. F. Yildirim, C. Korasli, Statistical approach for determining breakdown voltage of gas-insulated cables, IEEE Trans. on Electrical Insulation, Vol. 27, No. 6, pp. 1187-1192, 1992.
5. E. Kuffel, A. Yializis, Impulse breakdown of positive and negative rod-plane gaps in SF_6-N_2 mixtures, IEEE Trans. on Power App. and Sys., Vol.PAS-97, No. 6, pp. 2359-2366, 1978.
6. I. Al-Bawy, O., Farish, Insulator flashover in SF_6 under impulse voltage conditions, IEE Proc.-A, Vol. 138, No. 1, pp. 89-97, 1991.
7. Z. Li, R. Kuffel, E. Kuffel, Volt-time characteristics in air, SF_6/air mixture and N_2 for coaxial cylinder and rod-sphere gaps, IEEE Trans. on Elec. Insul. Vol. EI-21, No. 2, pp. 151-155, 1986.
8. H. M. Ryan, Further breakdown characteristics in SF_6 and air, 3rd Int. Symp. on HV Eng., 32.14, 1979.
9. T. Shimazaki, Flashover characteristics and surface processes under negative impulse voltage in atmospheric air, IEEE Trans. on Elec. Insul., Vol. 27, No. 3, pp. 488-495, 1992.
10. IEC Publ. 60-1: High-voltage test techniques, Part 1: General definitions and test requirements, Appendix A: Statistical treatment of test results, 1991.
11. M. H. Shwehdi, F. Shahzad, Estimation of lightning impulse insulation strength of two series dielectrics using the least square technique, IEEE Elec. Insul. Magazine, Vol. 13, No. 2, pp. 22-27, 1997.
12. H. Ismailoglu, Breakdown in Compressed Gases Under Impulse Voltages, Ph.D. Thesis, Institute of Science and Technology, Istanbul Tech. Univ., 1996.
13. I. Gutmann, S. S. Wilks *Introductory Engineering Statistics*, Wiley, New York, 1965.
14. D. V. Griffiths and I. M. Smith, *Numerical Methods for Engineers*, Blackwell Sci. Publ., Oxford, 1991.
15. H. Ismailoglu, Ö. Kalenderli, M. Özkaya, Determination of impulse breakdown voltage using least squares method, 1997 Conf. on Elec. Insul. and Dielectric Phenomena (The 1997 CEIDP), Minnea-polis, USA, Vol. 1, pp. 246-249, Oct. 19-22, 1997.
16. P. K. Mukherjee, Optimization of HV electrode systems by neural network using a new learning method, IEEE Trans. on Dielectric and Electrical Insulation, Vol. 3, No. 6, pp. 737-742, 1996.
17. K. Bhattacharya, S. Chakravorti, P. K. Mukherjee, Insulator contour optimization by a neural network, IEEE Trans. on Dielectrics and Electrical Insulation, Vol. 8, No. 2, pp. 157-161, 1994.

18. S. Chakravorti, P. K. Mukherjee, Application of artificial neural networks for optimization of electrode contour", IEEE Trans. on Dielectric and Elec. Insul., Vol. 1, No. 2, pp. 254- 263, 1994.
19. M. M. A. Salama, R. Bartnikas, Determination of neural network topology for partial discharge pulse pattern recognition", IEEE Trans. on Neural Networks, Vol. 13, No. 2, pp. 446- 456, 2002.
20. S. Bolat, Ö. Kalenderli, E. Önal, A. Merev, Determination of impulse flashover voltage of HV insulators by artificial neural networks, ELECO'2002, Turkish Symposium on Electrical and Electronics Engineering, Bursa, pp. 154-158, 2002.
21. M. Oguz, Determination of Electrical Strength of Insulating Materials by Artificial Neural Networks, MSc. Thesis, Institute of Science and Technology, Istanbul Technical University, 2001.
22. S. Bolat, Electrode and Insulator Contour Optimization by Artificial Neural Networks, MSc. Thesis, Institute of Science and Technology, Istanbul Technical University, 2003.
23. S. Haykin, *Neural Networks: a Comprehensive Foundation*, Upper Saddle River, N.J., Prentice Hall, 1999.
24. S. H. Ngia, Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm, IEEE Trans. on Signal Processing, Vol. 48, pp. 1915-1927, 2000.