

**MODELING OF BREAKDOWN VOLTAGE OF WHITE  
MINILEX PAPER IN THE PRESENCE OF VOIDS UNDER AC AND DC  
CONDITIONS  
USING NEURAL NETWORKS**

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF

**Bachelor of Technology**

**In**

**Electrical Engineering**

**By**

Abhishek Padhy (107EE040)

Anish (107EE062)

Asish Kumar Pati (107EE063)

Under the Guidance of

**Prof. Sanjeeb Mohanty**



**Department of Electrical Engineering**

**National Institute of Technology**

**Rourkela**

2011



**National Institute of Technology**

**Rourkela**

## **Certificate**

This is to certify that the thesis entitled “**Modeling of Break Down Voltage of White Minilex Paper in the presence of voids under AC and DC conditions using Neural Networks**” submitted by Shri Abhishek Padhy, Shri Anish and Shri Asish Kumar Pati in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Electrical Engineering at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Date :

Prof. Sanjeeb Mohanty

Place: Rourkela

Dept. of Electrical Engg.

National Institute of technology

Rourkela-769008

# ACKNOWLEDGEMENT

We would like to convey our deep gratitude to our project guide Prof. Sanjeeb Mohanty who has always been our mentor for carrying out the project. We also extend our sincere thanks to Prof. B.D. Subudhi, Head of our Department, for his invaluable guidance.

It is our pleasure to refer “Microsoft Word” exclusive of which the compilation of this report would have been impossible. Also it would not have been possible to complete the project without the simulation software” MATLAB (version 7.1)”.

A project of this kind could never have been carried out without reference to and inspiration from the works of others whose details are mentioned in references section. We acknowledge our indebtedness to all of them.

Last but not the least, our sincere thanks to all of our friends who have patiently extended all sorts of help for accomplishing this undertaking.

AbhishekPadhy

Anish

Asish Kumar Pati

# ABSTRACT

Insulating materials contain various voids due to which upon passing of an electrical signal above a threshold level they start deteriorating and thus breakdown occurs. Hence finding out the breakdown voltage of an insulator is of utmost importance. In this project the Neural Network method has been employed to model the desired breakdown voltages under both AC and DC conditions. With the help of neural networks a relationship between the input parameters and breakdown voltage has been established. The insulating material under consideration is White Minilex Paper. The voids are created artificially with varied dimensions. The values of the mean square error and mean absolute mean error are calculated and these results show the effectiveness of the model.

# CONTENTS

Certificate	2
Acknowledgement	3
Abstract	4
Contents	5
1. Introduction	6-8
1.1 Breakdown Voltage	
1.2 Objectives of Thesis	
1.3 Background	
2. Neural Network Theory	9-17
2.1 Neural Networks	
2.2 Artificial Neural Network	
2.3 Learning	
2.4 Types of Neural Networks	
3. Proposed theory for the modeling	18-26
3.1 Proposed MFNN Model	
3.2 Back Propagation Algorithm	
4. Modeling results	27-38
4.1 Tables 4.1 – 4.10	
4.2 Figure 4.1 and 4.2	
4.3 Conclusions	
Appendix	39-45
References	46

# **CHAPTER 1**

## **INTRODUCTION**

## 1.1 Breakdown Voltage

The breakdown voltage of any insulator is that amount of minimum voltage above which the insulator starts behaving like a conductor. Thus the purpose of an insulator is defeated and hence it is of great importance to calculate the breakdown voltage of any insulator. Breakdown voltage is a characteristic of an insulator that defines the maximum voltage difference that can be applied across the material before the insulator collapses and conducts. In solid insulating materials, this usually creates a weakened path within the material by creating permanent molecular or physical changes by the sudden current. Within rarefied gases found in certain types of lamps, **breakdown voltage** is also sometimes called the "striking voltage".<sup>[1]</sup>

The second meaning of the term more specifically refers to the breakdown of the insulation of an electrical wire or any other electrical component. Such breakdown usually results in short circuit or blown fuse. This occurs at the breakdown voltage. Actual insulation breakdown is more generally found in high-voltage applications, where sometimes it causes the opening of a protective circuit breaker. Electrical breakdown is also associated with the failure of solid or liquid insulating materials that are used inside transformers or capacitors in the electricity distribution grid. Electrical breakdown also occurs across the insulators that are suspended in overhead power lines, within underground power cables, or lines arcing to nearby trees. Under sufficient electrical stress, electrical breakdown can occur within solids, liquids, gases or vacuum. However, the specific breakdown mechanisms are quite significantly different

for each, particularly in different kinds of dielectric mediums. All this leads to catastrophic failure of the instruments causing immense losses.

## **1.2 Objectives of the Thesis**

This work is aimed at finding out the breakdown voltage of a typical insulator called White Minilex paper under various dimensions of the voids present. The modeling is a very effective tool to predict results and hence MATLAB has been used for modeling of the breakdown voltage. Artificial Neural networks method is employed for the MATLAB code (In Appendix 1). An elaborate theory on Neural Networks is proposed in the second chapter.

## **1.3 Background**

As the project involves the Neural Network theory, books on Neural Networks have been studied. Also a brief knowledge on insulator breakdown is acquired. Also a knowledge of the MATLAB programming is required for the modeling task. Also various related papers on modeling of breakdown voltages were studied.



## **CHAPTER 2**

# **NEURAL NETWORKS**

## 2.1 Neural Networks

The work on Neural Networks was inspired from the way the human brain works. Our brain is a highly complex, nonlinear and parallel computer-like device. It has the capability to organize its structural constituents known as neurons, so as to perform certain computations (e.g. pattern recognition, perception, and motor control) many times faster than the fastest digital computer in existence today.<sup>[2]</sup> This ability of our brain has been adapted into processing units for further excellence in the field of artificial intelligence. The theory of modern neural networks begun by the pioneering works done by McCulloch (a psychiatrist ) and Pitts ( a Mathematician) in 1943.

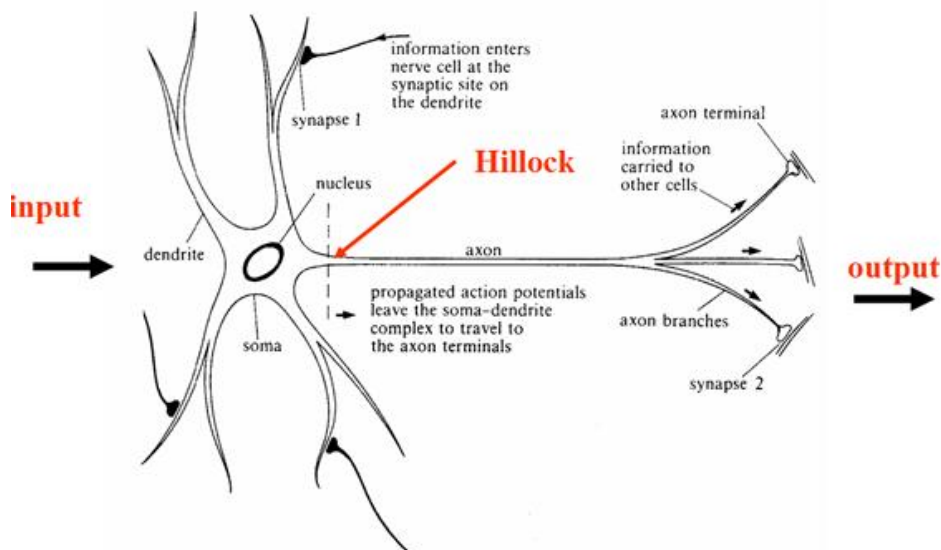


Fig.2.1 : Neural network of the human brain.<sup>[3]</sup>

## 2.2 Artificial Neural Networks

It is mathematical or a computational model derived from the aforementioned human brain neural network consisting of artificial neurons. These neurons are the functioning units of the network. Neural networks have gained excellent reputation in recent times for non-linear computation and to model complex relationships between inputs and outputs.

In artificial neural networks there is a function  $f(x)$  that is a composition of various other functions  $g_i(x)$ , which further are a mixture of other functions. This is usually pictorially represented by arrows depicting the dependency of variables on each other. A relation is as follows:

$$f(x) = K(\sum_i w_i g_i(x)) \quad (2.1)$$

Where  $w_i$  are the various weights provided to the functions,  $K$  is known as the activating function. The figure is the functional view: the input  $x$  is transformed into a 3-dimensional vector  $h$ , which is then transformed into a 2-dimensional vector  $g$ , which is finally transformed into  $f$ .

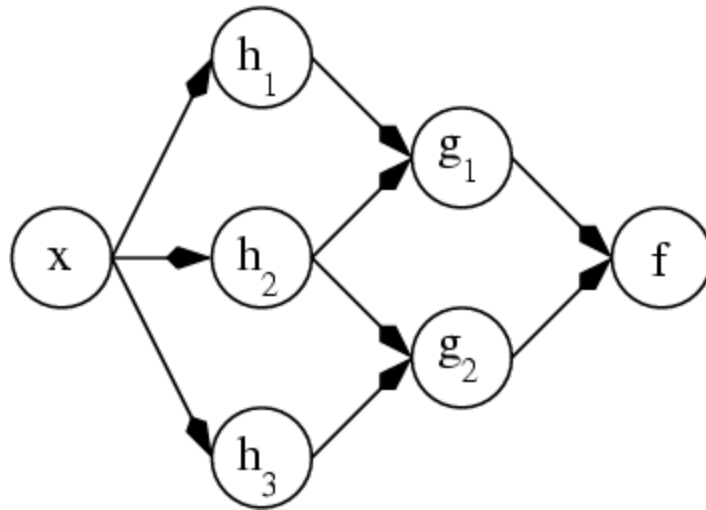


Fig. 2.2 The ANN dependency graph

## 2.3 Learning

One of the most attractive features in neural networks is the possibility of learning. Given a specific task to solve, and a class of functions,  $F$  learning means using a set of observations to find  $f^* \in F$  which solves the task at hand. The neural network not only learns from its environment but also improves its performance through learning. Learning paradigms are employed to carry out the process of learning in neural networks. The major paradigms are described below.

## 1. Supervised learning

Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consists of a set of training examples. In supervised learning, each example is a pair consisting of an input object (usually a vector) and a desired output value. A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier (for discrete output) or a regression function (for a continuous output). The inferred function should predict the correct output value for any valid input object. The parallel task in human and animal psychology is often referred to as concept learning.

## 2. Unsupervised learning

In machine learning, unsupervised learning is referred to the problem of trying to find hidden structure in unlabeled data. As the examples given to the learner are unlabeled, there is no error or reward signal to be evaluated as a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning. Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods used to preprocess data.

Approaches to unsupervised learning include clustering, blind signal separation using feature extraction techniques for dimensionality reduction. Among neural network models, the self-organizing map (SOM) and adaptive resonance theory (ART) are commonly used unsupervised learning algorithms.

### 3. Reinforced Learning

Inspired by behaviorist psychology, reinforcement learning is the area of machine learning in artificial neural networks, concerned with how an agent ought to take actions in an environment so as to maximize the notion of cumulative reward. Reinforcement learning is different from standard supervised learning in a manner that correct input/output pairs are never presented, nor are sub-optimal actions explicitly corrected. Further, there is a focus on on-line performance, which involves finding a balance between exploration of uncharted territory and exploitation of current knowledge.

## 2.4 Types of Neural Networks

### 1. Feedforward Neural networks

A feedforward neural network is an artificial neural network where connections between the units do *not* form a directed cycle. This is different from recurrent neural networks. The feedforward neural network was the first and arguably simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

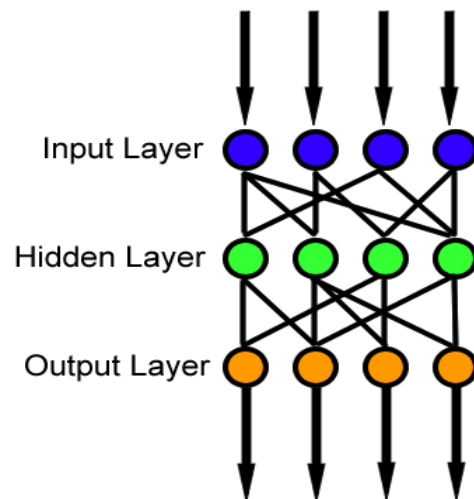


Fig. 2.3 A feedforward neural network

## 2. Radial Basis function Networks

A radial basis function network is an artificial neural network that uses radial basis functions as activation functions. It is a linear combination of radial basis functions. They are used in function approximation, time series prediction, and control. Radial basis functions are powerful techniques for interpolation in multidimensional space. An RBF is a function which is built into a distance criterion with respect to a center. Radial basis functions are applied in the area of neural networks where they can be used as a replacement for the sigmoidal hidden layer transfer characteristic in a multi-layer perceptrons. Also typically RBF networks have two layers of processing: In the first, input is mapped onto each RBF in the 'hidden' layer. The RBF chosen is usually a Gaussian function. In regression problems the output layer is then a linear combination of hidden layer values representing mean predicted output. The interpretation of this output layer value is the same as a regression model in statistics. In classification problems the output layer is typically a sigmoid function of a linear combination of hidden layer values, representing a posterior probability. Performance in both cases is often improved by shrinkage techniques, known as ridge regression in classical statistics and known to correspond to a prior belief in small parameter values (and therefore smooth output functions) in a Bayesian framework.



### 3. Recurrent Networks

A recurrent neural network (RNN) is a class of neural network where connections between the units form a directed cycle i.e. There are loops. This creates an internal state of the network due to which it exhibits dynamic temporal behavior. Unlike feedforward neural networks, RNNs can use their internal memory to process various arbitrary sequences of inputs. This makes them applicable to tasks such as unsegmented connected handwriting recognition, where they have achieved the best known results.

Various other popular kinds of neural networks are Alive networks, Spiking neural networks, Cascading neural networks, Neuro-fuzzy networks, Compositional pattern-producing networks, etc. These are beyond the scope of this work.

## **CHAPTER 3**

# **PROPOSED THEORY FOR THE**

# **MODELLING**

### 3.1 PROPOSED MULTILAYER FEEDFORWARD NEURAL NETWORK (MFNN) MODEL

The MFNN structure used here consists of three layers, namely the input layer, the hidden layer and the output layer as shown in Figure 4. The input layer consists of 4 neurons corresponding to the four inputs  $t$ ,  $t_1$ ,  $d$  and  $\epsilon_r$ . The number of output neurons is decided by the number of estimated parameters, therefore one in the present model, corresponding to the breakdown voltage,  $V_b$ . The Back Propagation Algorithm (BPA) is used to train the network. The sigmoidal function represented by equation (1) is used as the activation function for all the neurons except for those in the input layer.

$$S(x) = 1 / (1 + e^{-x}) \tag{3.1}$$

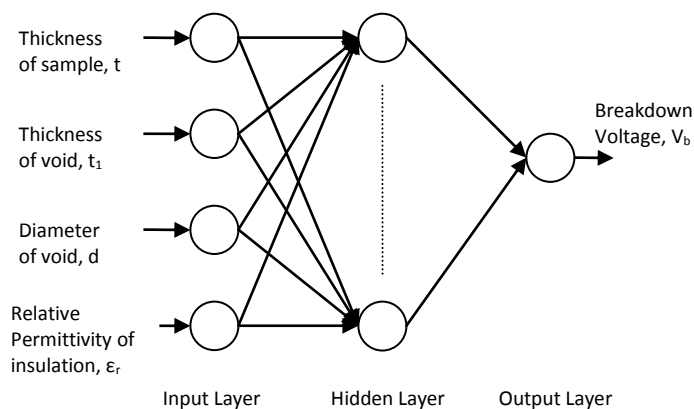


Fig. 3.1 : Multi-layer Feed-Forward Network (MFNN)

The choice of optimal number of hidden neurons,  $N_h$  is the most interesting and challenging aspect in designing the MFNN. There are various schools of thought in deciding the value of  $N_h$ . Simon Haykin has specified that  $N_h$  should lie between 2 and  $\infty$ . Hecht-Nielsen uses ANN interpretation of Kolmogorov's theorem to arrive at the upper bound on the  $N_h$  for a single hidden layer network as  $2(N_i+1)$ , where  $N_i$  is the number of input neurons. But generally this value should be decided very judiciously depending on the requirement of a problem. A large value of  $N_h$  may reduce the training error associated with the MFNN, but at the cost of increasing the computational complexity and time. For example, if one gets a tolerably low value of training error with certain value of  $N_h$ , there is no point in further increasing the value of  $N_h$  to enhance the performance of the MFNN. The input and the output data are normalized before being processed in the network as follows:

In this scheme of normalization, the maximum values of the input and output vector components are determined as follows:

$$n_{i,\max} = \max(n_i(p)) \quad p=1, \dots, N_p \quad i = 1, \dots, N_i \quad (3.2)$$

Where  $N_p$  is the number of patterns in the training set and  $N_i$  is the number of neurons in the input layer. Again,

$$o_{k,\max} = \max(o_k(p)) \quad p = 1, \dots, N_p, \quad k = 1, \dots, N_k \quad (3.3)$$

Where,  $N_k$  is the number of neurons in the output layer.

Normalized by these maximum values, the input and output variables are given as follows:

$$n_{i,nor}(p) = \frac{n_i(p)}{n_{i,max}} \quad p = 1, \dots, N_p, \quad i = 1, \dots, N_j \quad (3.4)$$

And

$$o_{k,nor}(p) = \frac{o_k(p)}{o_{k,max}} \quad p = 1, \dots, N_p, \quad i = 1, \dots, N_k \quad (3.5)$$

After normalization, the input and output variable lay in the range of 0 to 1.

### **i) Choice of ANN parameters**

The learning rate,  $\eta_1$  and the momentum factor,  $\alpha$  also have a very significant effect on the learning speed of the BPA. The BPA provides an approximation to the trajectory in the weight space computed by the method of steepest descent . If the value of  $\eta_1$  is made very small, this results in slow rate of learning, while if the value of  $\eta_1$  is too large in order to speed up the rate of learning, the MFNN may become unstable (oscillatory). A simple method of increasing the rate of learning without making the MFNN unstable is by adding the momentum factor  $\alpha$ . Preferably the values of  $\eta_1$  and  $\alpha$  should lie between 0 and 1.

## ii). Weight Update Equations

The weights between the hidden layer and the output layer are updated based on the equation (2) as follows

$$w_b(j,k,m+1) = w_b(j,k,m) + \eta_1 * \delta_k(m) * S_b(j) + \alpha * (w_b(j,k,m) - w_b(j,k,m-1)) \quad (3.6)$$

Where  $m$  is the number of iterations,  $j$  varies from 1 to  $N_h$  and  $k$  is the number of estimated output parameters. The value of  $k$  is 1 as only breakdown voltage  $V_b$  is to be estimated.  $\delta_k(m)$  is the error for the  $k^{\text{th}}$  output at the  $m^{\text{th}}$  iteration.  $S_b(j)$  is the output from the hidden layer.

Similarly, the weights between the hidden layer and the input layer are updated as follows :

$$w_a(i,j,m+1) = w_a(i,j,m) + \eta_1 * \delta_j(m) * S_a(i) + \alpha * (w_a(i,j,m) - w_a(i,j,m-1)) \quad (3.7)$$

Where  $i$  varies from 1 to 4 as there are 4 inputs to the network,  $\delta_j(m)$  is the error for the  $j^{\text{th}}$  output after the  $m^{\text{th}}$  iteration and  $S_a(i)$  is the output from the first layer. The  $\delta_k(m)$  in equation (6) and  $\delta_j(m)$  in equation (7) are related as :

$$\delta_j(m) = \sum_{k=1}^K \delta_k(m) * w_b(j,k,m) \quad (3.8)$$

### iii). Evaluation Criterion

The Mean Square Error  $E_{tr}$  for the training patterns after the  $m^{\text{th}}$  iteration is defined as

$$E_{tr}(m) = \left( \sum_{p=1}^P (V_{b1p} - V_{b2p}(m))^2 \right) * (1/P) \quad (3.9)$$

Where,  $V_{b1p}$  is the experimentally value of the breakdown voltage taken for training purpose,  $V_{b2p}(m)$  is the modeled value of the breakdown voltage after  $m^{\text{th}}$  iteration and  $P$  is the number of training patterns. The training is stopped when the least value of  $E_{tr}$  has been obtained and this value does not change with the number of iterations.

### iv). Mean Absolute Error

The Mean Absolute Error  $E_{ts}$  is a good performance measure for judging the accuracy of the ANN System. The  $E_{tr}$  tells how well the network has adopted to fit the training data only, even if the data is contaminated. On the other hand, the  $E_{ts}$  indicates how well a trained network behaves on a new data set not included in the training set.

The  $E_{ts}$  for the test data expressed in percentage is given by :

$$E_{ts} = (1/S) * \left( \sum_{s=1}^S |(V_{b4s} - V_{b3s})| / (V_{b3s}) * 100 \right) \quad (3.10)$$

Where,  $V_{b3s}$  is the experimental value of the breakdown voltage taken for testing purpose,  $V_{b4s}$  is the modeled value of the breakdown voltage after the testing input data is passed through the trained network and  $S$  is the number of testing patterns. <sup>[4]</sup>

## 4.2 Back Propagation Algorithm

Backpropagation is a common method of teaching artificial neural networks so as how to perform a given task. It was first described by Arthur E. Bryson and Yu-Chi Ho in 1969, but it wasn't until 1974 and later, through the work of Paul Werbos, David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams, that it gained popularity.

It is a supervised learning method which can be seen as a generalization of the delta rule. It requires a teacher that knows, or can find out, the desired output for any given input in the training set. It is most useful for feedforward networks. The term is actually an abbreviation for "backward propagation of errors". Backpropagation requires that the activation function used by the artificial neurons should be differentiable.



For better understanding, the backpropagation learning algorithm can be divided into two phases: 1.propagation and 2.weight update.

### **Phase 1: Propagation**

Each propagation module involves the following steps:

1. Forward propagation of a training data pattern's input through the neural network so as to generate the propagation's output activations.
2. Backward propagation of the propagation's output activations through the neural network using the training pattern's target in order to generate the deltas of all output and hidden neurons.

### **Phase 2: Weight update**

For each weight-synapse:

1. Multiply its output delta and input activation to get the gradient of the weight.
2. Bring the weight in the opposite direction of the gradient by subtracting a ratio of it from the weight.

This ratio influences the speed and quality of learning; it is called the learning rate. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction. <sup>[5]</sup>

Repeat the phase 1 and 2 until the performance of the network is good enough.

The MATLAB code used to train and test the data sets is given in Appendix I. The code provided is used for both AC and DC conditions by changing the ANN parameters accordingly and thus the breakdown voltage is found out.

## **CHAPTER 4**

# **MODELLING RESULTS**

Using the MATLAB code and the results for training and testing data sets under both AC and DC conditions to find out the breakdown voltage of white minilex paper are tabulated.

## 4.1 TABULATION

**Table 4.1** : Table to train the Neural Network program for AC breakdown voltage for white Minilex paper.

Sl. No.	Insulation thickness in mm	Void Depth in mm	Diameter of void in mm	Breakdown voltage in kV
1	0.125	0.125	3	2.3
2	0.125	0.125	4	2.2
3	0.125	0.125	5	2.3
4	0.18	0.125	1.5	2.2
5	0.18	0.125	2	2.2
6	0.18	0.125	4	2.2
7	0.26	0.125	1.5	2.3
8	0.26	0.125	3	2.33

**Table 4.2 :** Table for Mean square error by keeping the no. of hidden neurons and learning rate factor constant and varying the momentum factor (Under AC)

Sl.no.	Nh (No. of hidden neurons)	$\alpha$ (Momentum factor)	Eta (Learning rate factor)	MSE (Mean Square error)
1.	3	0.4	0.4	$1.3152 \times 10^{-4}$
2.	3	0.5	0.4	$9.371 \times 10^{-5}$
3.	3	0.6	0.4	$5.5887 \times 10^{-5}$
4.	3	0.65	0.4	$3.7929 \times 10^{-5}$
5.	3	0.7	0.4	$2.2281 \times 10^{-5}$

**Table 4.3:** Table for Mean square error (Under AC) by keeping the momentum factor and learning rate factor constant and varying the no. of hidden neurons

Sl.no.	Nh (No. of hidden neurons)	$\alpha$ (Momentum factor)	Eta (Learning rate factor)	MSE (Mean Square error)
1.	4	0.7	0.4	$1.2563 \times 10^{-5}$
2.	5	0.7	0.4	$9.2540 \times 10^{-6}$
3.	6	0.7	0.4	$3.6672 \times 10^{-6}$

**Table 4.4:** Table Mean Square error by keeping the no. of hidden neurons and momentum factor constant and varying the learning rate factor.

Sl.no.	Nh (No. of hidden neurons)	$\alpha$ (Momentum factor)	Eta (Learning rate factor)	MSE (Mean Square error)
1.	6	0.7	0.5	$2.1139 \times 10^{-6}$
2.	6	0.7	0.6	$1.3261 \times 10^{-6}$
3.	6	0.7	0.65	$1.0754 \times 10^{-6}$
4.	6	0.7	0.7	$8.356 \times 10^{-7}$

**Table 4.5:** Table to test the Neural Network program for AC breakdown voltage for white Minilex paper based on the data from tables 2, 3 and 4

Sl. No.	Insulation thickness in mm	Void Thickness in mm	Diameter of void in mm	Experimental Value of Breakdown voltage in kV	Modeled Value of Breakdown voltage in kV
1	0.125	0.125	1.5	2.2	2.1673
2	0.125	0.125	2	2.3	2.2132
3	0.18	0.125	3	2.3	2.2211
4	0.18	0.125	5	2.2	2.1738
5	0.26	0.125	2	2.2	2.1806
6	0.26	0.125	4	2.2	2.1806
7	0.26	0.125	5	2.2	2.1806

**RESULT :**

Mean Absolute Error (MAE) under AC conditions = 1.7871%

**Table 4.6:** Table to train the Neural Network program for DC breakdown voltage for white Minilex paper.

Sl.no.	Thickness of the material (mm)	Void Depth (mm)	Void Diameter (mm)	Mean value of Breakdown Voltage (Experimental) (kV)
1.	0.125	0.025	1.5	23.44
2.	0.125	0.025	2.0	22.88
3.	0.125	0.025	3.0	23.22
4.	0.125	0.025	4.0	24.44
5.	0.125	0.025	5.0	22.55
6.	0.18	0.025	1.5	23.55
7.	0.18	0.025	2.0	23.22
8.	0.18	0.025	3.0	24.44
9.	0.18	0.025	4.0	23.77
10.	0.18	0.025	5.0	22.88
11.	0.26	0.025	1.5	23.33
12.	0.26	0.025	2.0	23.00
13.	0.26	0.025	3.0	24.44
14.	0.26	0.025	4.0	23.77
15.	0.26	0.025	5.0	23.22
16.	0.125	0.125	1.5	24.44
17.	0.125	0.125	2.0	23.55
18.	0.125	0.125	3.0	22.55
19.	0.125	0.125	4.0	23.22
20.	0.125	0.125	5.0	23.77
21.	0.18	0.125	1.5	23.00
22.	0.18	0.125	2.0	24.33
23.	0.18	0.125	3.0	23.77
24.	0.18	0.125	4.0	22.88
25.	0.18	0.125	5.0	24.33
26.	0.26	0.125	1.5	23.22
27.	0.26	0.125	2.0	23.55
28.	0.26	0.125	3.0	23.44
29.	0.26	0.125	4.0	23.77
30.	0.26	0.125	5.0	22.88



**Table 4.7:** Table for Mean square error keeping the No. of hidden neurons and learning rate factor constant and varying the momentum factor (Under DC).

Sl.no	Nh (No. of hidden neurons)	$\alpha$ (Momentum factor)	Eta (Learning rate factor)	MSE (Mean square error)
1.	3	0.4	0.4	$2.3948 \times 10^{-4}$
2.	3	0.5	0.4	$1.6828 \times 10^{-4}$
3.	3	0.6	0.4	$9.8450 \times 10^{-5}$
4.	3	0.65	0.4	$6.6363 \times 10^{-5}$
5	3	0.7	0.4	$3.8802 \times 10^{-5}$

**Table 4.8:** Table for Mean square error keeping the momentum factor and learning rate factor constant and varying the no. of hidden neurons (Under DC).

Sl.no.	Nh (No. of hidden neurons)	$\alpha$ (Momentum factor)	Eta (Learning rate factor)	MSE (Mean square error)
1.	4	0.7	0.4	$2.0299 \times 10^{-5}$
2.	5	0.7	0.4	$1.4483 \times 10^{-5}$
3.	6	0.7	0.4	$4.9604 \times 10^{-6}$

**Table 4.9:** Keeping the no. of hidden neurons and momentum factor constant and varying the learning rate factor, the values of Mean square error is as follows: (Under DC)

Sl.no	Nh (No. of hidden neurons)	$\alpha$ (Momentum factor)	Eta (Learning rate factor)	MSE (Mean square error)
1.	6	0.7	0.5	$2.686 \times 10^{-6}$
2.	6	0.7	0.6	$1.586 \times 10^{-6}$
3.	6	0.7	0.65	$1.2498 \times 10^{-6}$
4.	6	0.7	0.7	$9.9824 \times 10^{-7}$

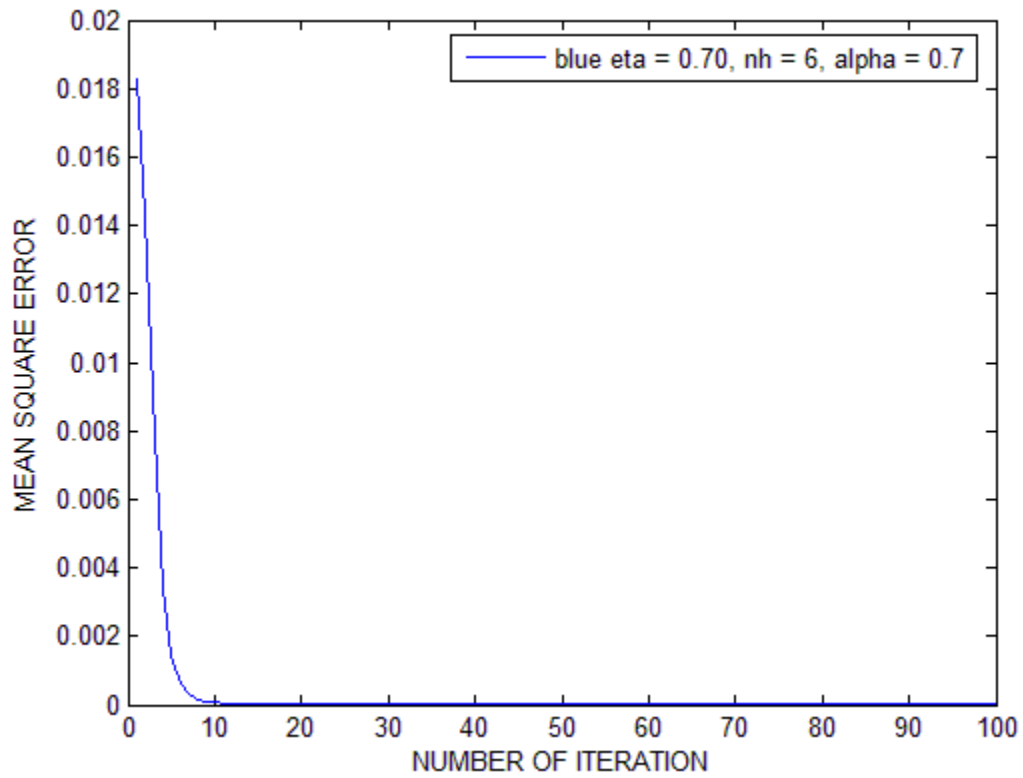
**Table 4.10:** Table to test the Neural Network program for DC breakdown voltage for white Minilex paper.

Sl.No.	Thickness of the material (mm)	Void Depth (mm)	Void Diameter (mm)	Mean value of Breakdown Voltage (Experimental) (kV)	Mean value of Breakdown Voltage (Modeled) (kV)
1.	0.125	0.125	1.5	24.44	23.5178
2.	0.125	0.125	2.0	23.55	23.1272
3.	0.125	0.125	3.0	22.55	22.4530
4.	0.125	0.125	4.0	23.22	22.9343
5.	0.125	0.125	5.0	23.77	23.2404
6.	0.18	0.125	1.5	23.00	22.8459
7.	0.18	0.125	2.0	24.33	23.5629
8.	0.18	0.125	3.0	23.77	23.3188
9.	0.18	0.125	4.0	22.88	22.7567
10.	0.18	0.125	5.0	24.33	23.5629
11.	0.26	0.125	1.5	23.22	23.0662
12.	0.26	0.125	2.0	23.55	23.2788
13.	0.26	0.125	3.0	23.44	23.1196
14.	0.26	0.125	4.0	23.77	23.4011
15.	0.26	0.125	5.0	22.88	22.8098

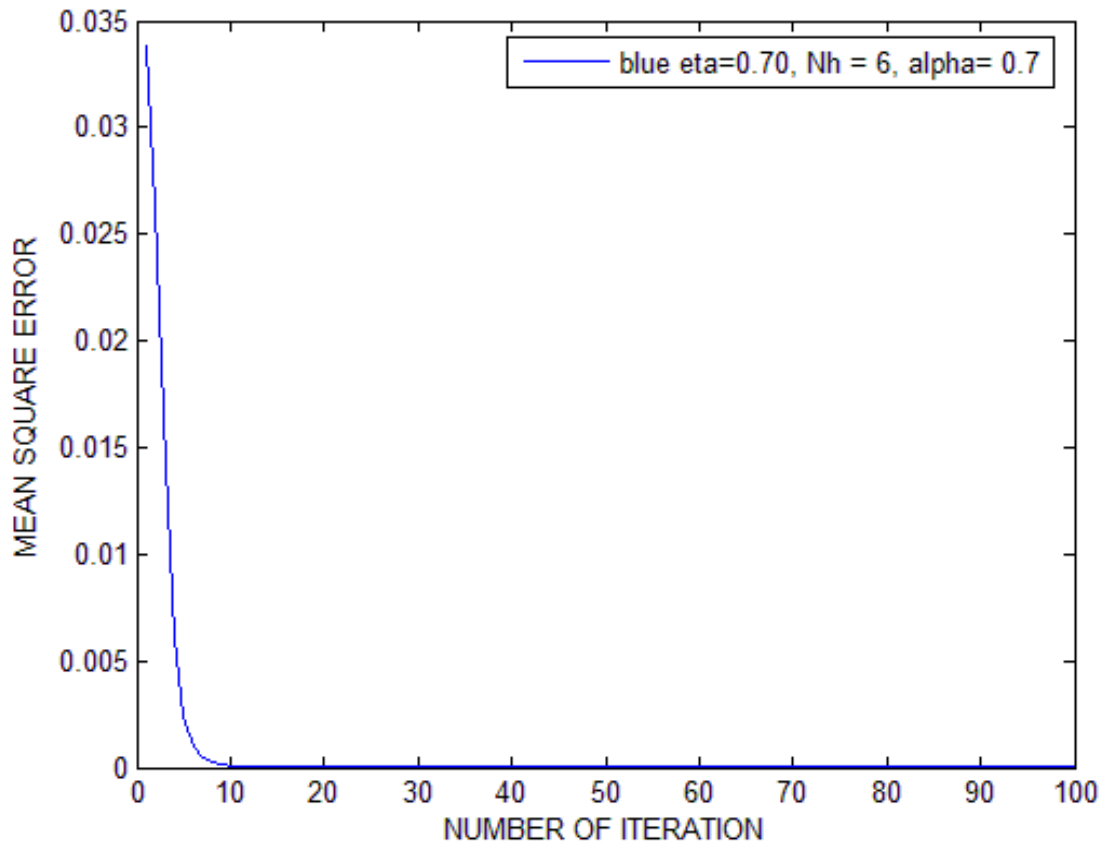
**RESULT:**

The Mean Absolute error under DC conditions = 3.589%

## 4.2 PLOTTING



**Fig 4.1:** Variation between the no. of iterations and the mean square error under AC condition.



**Fig 4.2:** Variation between the no. of iterations and the mean absolute error under DC condition. The ANN parameters are fixed for this plot.

## 4.3 CONCLUSION

In this work we have used one of the most sophisticated soft-computing techniques that is Artificial Neural Networks method. Under both AC and DC conditions the breakdown voltage of a typical White Minilex paper is predicted using MATLAB 7.1. The values of mean square error and the mean absolute error show the effectiveness of Multilayer Feedforward Neural Network in predicting the breakdown voltage.

- The mean square error under both AC and DC conditions are in the range of  $10^{-7}$ .
- The Mean absolute error under AC conditions was found out to be 1.7871%
- The Mean absolute error under DC conditions was found out to be 3.589%

# **APPENDIX**

## MATLAB code for modeling of breakdown voltage of White Minilex Paper under

### AC conditions.

```
clear all;
% ERROR FROM OUTPUT LAYER FEEDBACK TO THE HIDDEN LAYER
% To model the breakdown voltage of White Minilex under AC conditions
as a function of void
% diameter d(mm) & insulation thickness t(mm)
% Backpropagation Algorithm used for modeling
% t1 is void thickness = 0.125mm
% Input Training Patterns 8(8 by 2)(insulation thickness & void
% diameter)
t=[0.125;0.125;0.125;0.18;0.18;0.18;0.26;0.26];
t1=t/max(t);
d=[3;4;5;1.5;2;4;1.5;3];
d1=d/max(d);
x=[t1 d1];
% Output Training Patterns 8(8 by 1)(Breakdown Voltage Experimental)
bv=[2.3;2.2;2.3;2.2;2.2;2.2;2.3;2.3];
bv1=bv/max(bv);
% Define Learning Rate Parameter
Eta=0.99;
% Define Momentum Factor
alpha=0.9;
% Define number of hidden neurons
nh=4;
iter=500;
% Program for training the neural network & designing it by minimizing
the
% ms error of the training data &also testing the network
% Initialize the weights between input layer & hidden layer,hidden
layer &
% output layer
a1=-1;
b1=1;
rand('state',0);
for k=1:1
for j= 1:nh
for i=1:2
wa(i,j,1)=a1 + (b1-a1).*rand(1,1);
wb(j,k,1)=a1 + (b1-a1).*rand(1,1);
end;
end;
end;
for z=1:size(bv)
for m=1:iter
```



```

        % Output from first layer
sa(z)=x(z);
% Initialize xa (Activation to hidden layer)
for j=1:nh
xa(j)=0;
end;
% Activation to hidden layer
for j=1:nh
for i=1:2
xa(j)=xa(j)+wa(i,j,m).*sa(z);
end;
% Output from hidden layer
sb(j)=(1)/(1+exp(-xa(j)));
end;
% Initialize Activation to output layer
for k=1:1
xb(k)=0;
end;
% Activation to output Layer & output from output layer
for k=1:1
for j=1:nh
xb(k)=xb(k)+wb(j,k,m).*sb(j);
% Output from output Layer
bv2(z,m)=(1)/(1+exp(-xb(k)));
% Error term for mth iteration
delta2(z,k,m)=(bv1(z,:)-bv2(z,m))*bv2(z,m).*(1-bv2(z,m));
end;
end;
% Updating the weights between hidden layer & output layer
for k=1:1
for j=1:nh
if m-1<=0
wb(j,k,m+1)=0;
else
wb(j,k,m+1)=wb(j,k,m)+(Eta.*delta2(z,k,m).*sb(j))+(alpha.*(wb(j,k,m)-
wb(j,k,m-1)));
end;
end;
end;
% Error term for jth hidden unit(Initialize)
for j=1:nh
delta(z,j,m)=0;
end;
% Error term for jth hidden unit
for j=1:nh
for k=1:1
delta(z,j,m)=delta(z,j,m)+(delta2(z,k,m).*wb(j,k,m));
end;
delta1(z,j,m)=(delta(z,j,m)).*(sb(j).*(1-sb(j)));
end;
% Updating the weight between hidden layer & input layer

```

```

for j=1:nh
for i=1:2
if m-1<=0
wa(i,j,m+1)=0;
else
wa(i,j,m+1)=wa(i,j,m)+(Eta.*delta1(z,j,m).*sa(z))+(alpha.*(wa(i,j,m)-
wa(i,j,m-1)));
end;
end;
end;
end;
end;
% Calculation of Mean square error of the training data Etmse and root
mean
% square error of training data Etermse
Etmse(1:iter)=0;
k=1;
for m=1:iter
for z=1:size(bv)
Etmse(m)=Etmse(m)+((delta2(z,k,m)).^2)/8;
Etermse(m)=sqrt(Etmse(m));
bv3(z,m)=max(bv).*bv2(z,m);
end;
end;

% Testing processing
% Thickness(7 by 1)
t3=[0.125;0.125;0.18;0.18;0.26;0.26;0.26];
t4=t3/max(t3);
% Diameter of void(7 by 1)
d3=[1.5;2;3;5;2;4;5];
d4=d3/max(d3);
x3=[t4 d4];
% Experimental breakdown voltage(7 by 1)
bv7=[2.2;2.3;2.3;2.2;2.2;2.2;2.2];
bv8=bv7/max(bv7);
for z=1:size(bv7)
for m=1:iter
% Output from first layer
sal(z)=x3(z);
% Initialize xa (Activation to hidden layer)
for j=1:nh
xa1(j)=0;
end;
% Activation to hidden layer
for j=1:nh
for i=1:2
xa1(j)=xa1(j)+wa(i,j,m).*sal(z);
end;
% Output from hidden layer
sb1(j)=(1)/(1+exp(-xa1(j)));

```

```

end;
% Initialize Activation to output layer
for k=1:1
xb1(k)=0;
end;
% Activation to output Layer & output from output layer
for k=1:1
for j=1:nh
xb1(k)=xb1(k)+wb(j,k,m).*sb1(j);
% Output from output Layer
bv9(z,m)=(1)/(1+exp(-xb1(k)));
bv10(z,m)=max(bv7).*bv9(z,m);
% Error term for mth iteration
delta3(z,k,m)=(bv8(z,:)-bv9(z,m)).*bv9(z,m).*(1-bv9(z,m));
end;
end;
% Updating the weights between hidden layer & output layer
for k=1:1
for j=1:nh
if m-1<=0
wb(j,k,m+1)=0;
else
wb(j,k,m+1)=wb(j,k,m)+(Eta.*delta3(z,k,m).*sb1(j))+(alpha.*(wb(j,k,m)-
wb(j,k,m-1)));
end;
end;
end;
% Error term for jth hidden unit(Initialize)
for j=1:nh
delta4(z,j,m)=0;
end;
% Error term for jth hidden unit
for j=1:nh
for k=1:1
delta4(z,j,m)=delta4(z,j,m)+(delta3(z,k,m).*wb(j,k,m));
end;
delta5(z,j,m)=(delta4(z,j,m)).*(sb1(j).*(1-sb1(j)));
end;
% Updating the weight between hidden layer & input layer
for j=1:nh
for i=1:2
if m-1<=0
wa(i,j,m+1)=0;
else
wa(i,j,m+1)=wa(i,j,m)+(Eta.*delta5(z,j,m).*sal(z))+(alpha.*(wa(i,j,m)-
wa(i,j,m-1)));
end;
end;
end;
end;
end;
end;

```

```
% Calculation of MAE of test data

MAE=0;
for z=1:size(bv7)
e2(z)=bv7(z)-bv10(z,1);
    MAE=MAE +((100/7)*abs(e2(z)/bv7(z)));
end;
plot(Etmse,'b'),legend('Eta=0.8'),hold on

legend('blue eta=0.99')
Etmse
```

## **REFERENCES**

1. J. M. Meek and J. D. Craggs, Electrical Breakdown of Gases, John Wiley & Sons, Chichester, 1978.
2. Neural Networks by Simon Haykin (Second Edition, Chap. 1)
3. [http://www.codeproject.com/KB/recipes/NeuralNetwork\\_1.aspx](http://www.codeproject.com/KB/recipes/NeuralNetwork_1.aspx) (fig 1)
4. Neural Networks by Simon Haykin (Second Edition, Chap. 4 pg. 161-175)
5. "Artificial neural networks modeling of breakdown voltage of solid insulating materials in the presence of void" by S.Mohanty and S.Ghosh published in 2010.