

Article

# Prediction of Critical Flashover Voltage of High Voltage Insulators Leveraging Bootstrap Neural Network

M. Tahir Khan Niazi <sup>1</sup>, Arshad <sup>2,\*</sup>, Jawad Ahmad <sup>3</sup>, Fehaid Alqahtani <sup>4</sup>, Fatmah AB Baotham <sup>5</sup> and Fadi Abu-Amara <sup>6</sup>

<sup>1</sup> Department of Electrical Engineering, HITEC University Taxila, Punjab 47080, Pakistan; tahirniazai@yahoo.com

<sup>2</sup> Institute for Energy and Environment, University of Strathclyde, Glasgow G1 1XQ, UK

<sup>3</sup> School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, UK; Jawadkhattak@ieee.org

<sup>4</sup> Department of Computer Science, King Fahad Naval Academy, Al Jubail 35512, Saudi Arabia; fehaidb00282066@gmail.com

<sup>5</sup> Faculty of Computing and Information Technology, King Abdul Aziz University, Jeddah 21431, Saudi Arabia; fbaothman@kau.edu.sa

<sup>6</sup> Computer & Information Science Department, Higher Colleges of Technology, Abu Dhabi 25026, UAE; fabuamara@hct.ac.ae

\* Correspondence: arshadali.giki@gmail.com

Received: 19 August 2020; Accepted: 24 September 2020; Published: 2 October 2020



**Abstract:** Understanding the flashover performance of the outdoor high voltage insulator has been in the interest of many researchers recently. Various studies have been performed to investigate the critical flashover voltage of outdoor high voltage insulators analytically and in the laboratory. However, laboratory experiments are expensive and time-consuming. On the other hand, mathematical models are based on certain assumptions which compromise on the accuracy of results. This paper presents an intelligent system based on Artificial Neural Networks (ANN) to predict the critical flashover voltage of High-Temperature Vulcanized (HTV) silicone rubber in polluted and humid conditions. Various types of learning algorithms are used, such as Gradient Descent (GD), Levenberg-Marquardt (LM), Conjugate Gradient (CG), Quasi-Newton (QN), Resilient Backpropagation (RBP), and Bayesian Regularization Backpropagation (BRBP) to train the ANN. The number of neurons in the hidden layers along with the learning rate was varied to understand the effect of these parameters on the performance of ANN. The proposed ANN was trained using experimental data obtained from extensive experimentation in the laboratory under controlled environmental conditions. The proposed model demonstrates promising results and can be used to monitor outdoor high voltage insulators. It was observed from obtained results that changing of the number of neurons, learning rates, and learning algorithms of ANN significantly change the performance of the proposed algorithm.

**Keywords:** critical flashover voltage; Artificial Neural Networks (ANN); Gradient Descent (GD); Levenberg-Marquardt (LM); Conjugate Gradient (CG); Quasi-Newton (QN); Resilient Backpropagation (RBP); Bayesian Regularization Backpropagation (BRBP)

## 1. Introduction

Outdoor high voltage insulators are exposed to various types of stresses. Stresses include mechanical, electrical, thermal, and environmental stresses. To simulate the effect of these stresses in the laboratory, different types of techniques are used. High-voltage stresses and artificial rain and fog are a few examples. The critical flashover voltage of insulators depends on the insulator design,

surface roughness, orientation, rain, humidity, temperature, fogs, Ultraviolet (UV) radiations, wind speed, direction, and distance from the pollution source [1,2]. Although the performance of outdoor insulators is affected by many parameters, pollution deposition on the insulator surface is considered a major factor in the deteriorating performance of insulators. Pollution deposition on outdoor insulators surface may be due to industrial emissions, salt spray from the sea, and chemicals' emissions from vehicles and or agriculture. The change in the performance of outdoor insulators due to pollution deposition depends on the type of pollution constituents. Generally, pollution deposited on the insulator surface is classified into two major types: inert pollution and active pollution. The effect of active and inert pollution on the insulator performance is different resulting in errors of flashover voltage calculations [3].

Intelligent techniques such as fuzzy logic [4], Support Vector Machine (SVM) [5], Artificial Neural Networks (ANN) [6–8], Hidden Markov Model (HMM) [9], K-means clustering [10], Discrete Wavelet Transform (DWT) [11], S-Transform [12], have been extensively used in electrical power system and high voltage engineering problems. These intelligent systems can be successfully utilized for the condition monitoring of high-voltage outdoor insulators to increase the reliability of power system transmission and distribution as well as minimize human efforts and cost [13].

With the increase in transmission line voltages and increased distance of renewable power sources from the loads, the importance of research on the pollution performance of insulators has significantly increased. The mechanism of flashover in high voltage porcelain, glass, and ceramic insulators under contamination has been studied extensively in the past [14–16]. Many researchers have proposed mathematical models to predict the critical flashover voltage under uniform and non-uniform pollution [17,18]. An improved mathematical model has been proposed in Reference [19] to estimate pollution flashover voltage of ceramic insulators based on dimensional analysis of the flashover influencing parameters. Shahabi et al. [20] studied the flashover process of outdoor insulators by adding a random value to the discharge length to account for wind speed, direction, and thermal convection on the discharge. Palangar et al. [21] proposed an improved dynamic model for predicting the critical flashover parameters of ceramic insulators by incorporating capacitance in the equivalent circuit of the dry band.

Apart from mathematical and numerical modeling, many researchers have proposed intelligent systems such as ANN for flashover voltage prediction [6,8,22]. Salem et al. [22] combined Adaptive Neuro Fuzzy Inference System (ANFIS) with ANN and used insulator height, diameter, form factor, creepage distance along with Equivalent Salt Deposit Density (ESDD) as input parameters to train the model. In Reference [23], the authors applied dimensional analysis to the proposed ANFIS-based ANN network by establishing a relationship between critical flashover voltage and leakage current. The arc constant of the mathematical model for obtaining the test data was optimized using a Genetic Algorithm (GA) for improved results.

Another important intelligent technique used for flashover prediction is SVM, which offers the advantage of global optimality. Least Square SVM (LS-SVM) was proposed in Reference [24] for prediction of pollution severity and critical flashover voltage based on insulator diameter, height, ESDD, and form factor. Ming-Yuan et al. [25] estimated insulator leakage current using SVM by finding correlation between weather conditions and leakage current. Different meteorological parameters were combined with leakage current parameters generated from different types of insulators. Gencoglu et al. [26] proposed LS-SVM for prediction of flashover voltage by generating the training data set from numerical models based on Finite Element Method (FEM). The LS-SVM parameters were tuned using a grid search algorithm for improved accuracy.

Saranya et al. [27,28] proposed a new method for condition monitoring of outdoor insulators by identifying insulator arc faults using phasor angle measurements. The insulator arcs have been classified using SVM to support the design of improved protection schemes for smart grids. A modified LS-SVM scheme has been proposed by applying a fixed set of support vectors to predict the critical

flashover voltage under polluted conditions [5]. The Quadratic Renyi Criterion (QRC) is used to select support vectors from the training data set.

The existing literature demonstrates considerable work on the application of intelligent systems in predicting the flashover voltage of outdoor high voltage insulators. However, there are specific gaps in the current knowledge which need to be further investigated. The existing ANN algorithms used the Gradient Descent (GD) algorithm due to its faster convergence and lower computation time by compromising the prediction accuracy. The current literature also considered insulator height, diameter, form factor, and ESDD as input parameters for flashover prediction, while the flashover voltage also depends on environmental conditions such as temperature, humidity, and non-soluble pollution. Apart from that, fixing the number of neurons, learning rates, and the number of hidden layers significantly changes the prediction accuracy of ANN, which needs to be investigated. One of the major limitations of existing ANN-based prediction models is that most of them rely on data from mathematical models which are based on a particular assumption. Additionally, current mathematical models are applicable to porcelain and glass insulators and cannot be applied to polymeric insulators without modification due to the different flashover mechanism of polymeric insulators as compared to porcelain and glass insulators.

## 2. Materials and Methods

ANN and other machine learning algorithms have been used to predict critical flashover voltage, leakage current, and ESDD. However, there are some limitations of the existing literature such as; (1) use of insulator dimensions and pollution severity as input parameters for learning and ignoring the environmental conditions (humidity and temperature); (2) using a single learning algorithm for training, for example, GD in most cases; (3) the training data set is either small or generated from mathematical models. This paper presents an intelligent system for flashover voltage prediction of polymeric insulators using experimental results as a training data set for training the ANN. The experimental results of critical flashover voltage are obtained under controlled environmental conditions. To increase the sample space and accuracy of the proposed model, bootstrapping is applied to the actual data set. The proposed NN model is tested for different learning algorithms such as GD, Levenberg-Marquardt (LM), Conjugate Gradient (CG), Quasi-Newton (QN), Resilient Backpropagation (RBP) and Bayesian Regularization Backpropagation (BRBP). The number of neurons in the hidden layer, the number of hidden layers, as well as learning rate, are varied to obtain the optimum parameters. The prediction accuracy of each model is tested using Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Regression Value (R) and Normalized Mean Square Error (NRMSE).

### 2.1. Experimental Setup and Test Methods

High voltage tests were performed on rectangular samples of HTV silicone rubber under controlled environmental conditions. The clean fog method (solid layer) based on modified IEC 60507 was used to apply soluble and non-soluble pollution on the insulator samples. The test setup and sample configuration are shown in Figures 1 and 2, respectively. The insulator samples were energised using a power frequency 0–100 kV test transformer. Before energising, samples were placed in the climate chamber for a considerable amount of time to make sure no dry bands were present, and the samples were properly wetted. Initial tests were performed on a uniformly polluted sample to determine the probable flashover voltage. Once the probable flashover voltage was determined, the remaining tests were performed by applying voltage in steps of 5% of the probable flashover voltage. Each step was maintained for 2 min, and if no partial arcs appeared, the voltage was increased further. In the case of appearance of a partial arc, the voltage was kept constant at that step until the partial arc vanished or lead to flashover. This process was repeated for each sample. As silicone rubber loses its hydrophobicity under energization, the sample was replaced after every two tests. This helped in maintaining the uniform pollution layer and the hydrophobic nature of silicone rubber.

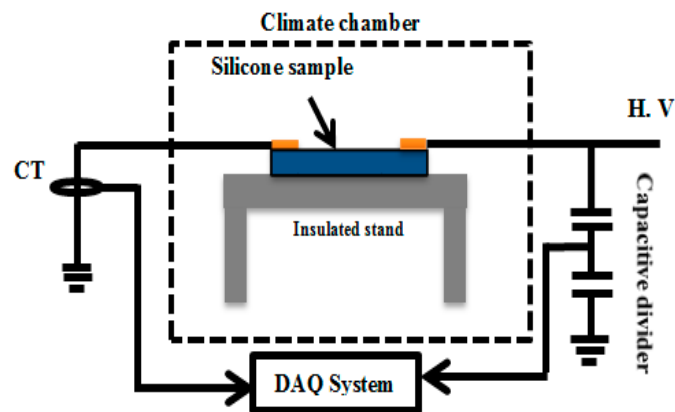


Figure 1. The high-voltage test setup.

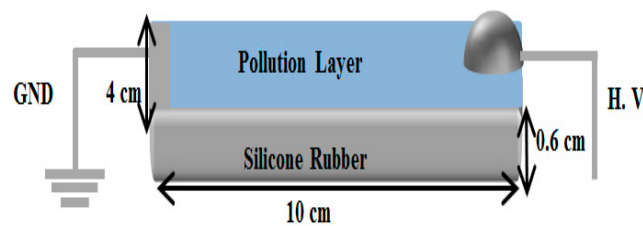
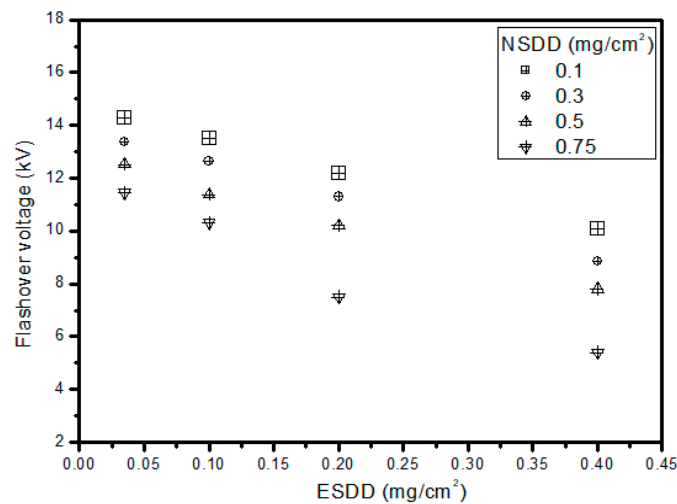


Figure 2. The electrode setup and sample configuration.

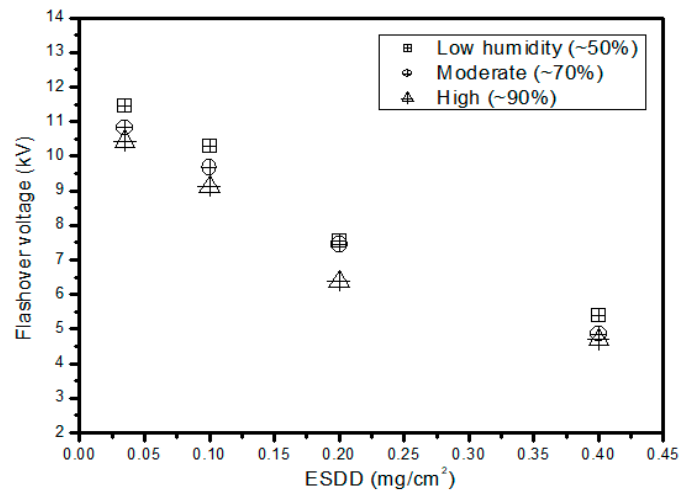
### Experimental Results

Air pollution deposited on the insulator surfaces can be broadly classified into two major types: active and inert. Active pollution is represented with ESDD, while inert pollution is represented with NSDD. NSDD is the non-soluble part of pollution such as dust, cement, or sand, which does not dissolve in water but forms a thick layer on the surface of the insulator, which may affect the flashover behavior. The effect of ESDD and NSDD is different on flashover voltage of polymeric insulators as presented in Reference [1]. Figure 3 shows the relationship between critical flashover voltage and ESDD at different values of NSDD. A total of 16 tests were performed at different combinations of ESDD and NSDD. The results show that as the value of ESDD and NSDD increases, the critical flashover voltage decreases. This is mainly due to the increase in leakage current due to the increased conductivity of the pollution layer, as well as the increased thickness of the pollution layer when NSDD is increased. The increase in the thickness of the pollution layer resists the recovery of hydrophobicity and facilitates uniform wetting of the pollution layer, resulting in increased leakage current. The temperature and humidity were kept constant during these tests to minimize the effect of environmental conditions.

The effect of relative humidity on critical flashover voltage is shown in Figure 4. The relative humidity was varied within the climate chamber, while temperature and NSDD were kept constant. Samples with different ESDD values were tested. The critical flashover voltage decreased as humidity and ESDD increased. This may be due to the increase in pollution constituent dissolving in the humid air surrounding the insulator.

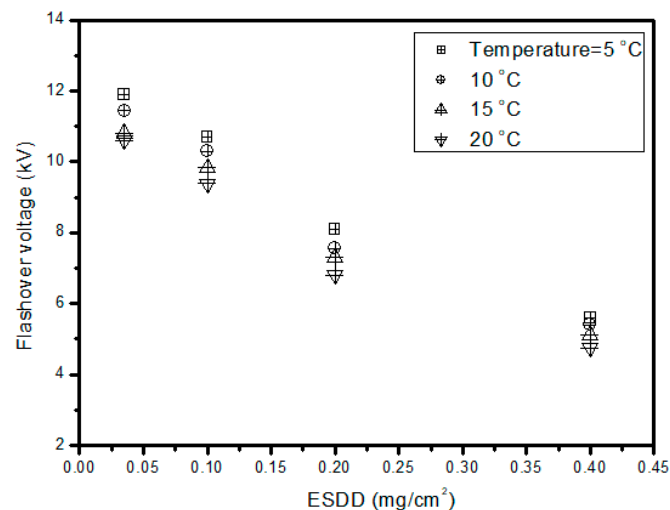


**Figure 3.** The relationship between NSDD and critical flashover voltage at moderate humidity and 10 °C temperature.



**Figure 4.** The relationship between relative humidity and critical flashover voltage at 10 °C temperature and NSDD of 0.75 mg/cm<sup>2</sup>.

Apart from humidity, inert, and active pollution, ambient temperature also affects the flashover process. The influence of high temperature on insulator performance in desert conditions has been investigated in the literature. However, here, the focus is on the effect of temperature under polluted and humid conditions, which influence the hydrophobicity loss and recovery process of polymeric insulators. The results of the critical flashover voltage at four different temperature values are shown in Figure 5. It can be observed that critical flashover voltage decreases with an increase in temperature and ESDD. There can be multiple explanations, such as a change in the hydrophobicity recovery process and conductivity of the pollution layer. However, the obtained results show that as the temperature increases, the conductivity of the pollution layer increases, which leads to an increase in leakage current, decrease in surface resistance, and critical flashover voltage.



**Figure 5.** The relationship between ambient temperature and critical flashover voltage at moderate humidity and NSDD of  $0.75 \text{ mg/cm}^2$ .

## 2.2. Proposed Artificial Neural Network Algorithm

Machine learning algorithms such as ANN can be effectively used in high voltage engineering to minimize cost and time of experimentation. In this work, we proposed a machine learning algorithm based on NN to predict the critical flashover voltage of outdoor high voltage insulators. Details about the proposed machine learning algorithm are given in the following section.

### 2.2.1. Bootstrapping Method

Bootstrapping, or sometimes called bagging, is a statistical technique to increase the sample space when a limited number of data samples are available for training machine learning algorithms. Apart from increasing the number of observations, bootstrapping also offers the advantage of improved accuracy as well as increased effectiveness of percentage estimation. A bootstrap sample is a random sample conducted with replacement; it means the number of times a random observation is selected from the real data. Rather than relying on the theory, which gives the sets of all possible estimates, the bootstrap generates estimates through re-sampling distribution named bootstrap distribution, and the standard deviation of all estimates is called the bootstrap standard error. There are two main reasons to use the bootstrap approach instead of large sample theory approach: one is the lack of large sample data, and the other is to workout with the standard error of the estimates.

In this technique, sampling is performed by extracting only one sample at a time from a given data, and the selected sample is returned to the data set. In this way, the sample appears more than once in the given test data in the next iteration. This method of sampling is known as sampling with replacement. The bootstrap method can be summarized as [29]:

- Select the number of samples which need to be extracted from given data
- Select the appropriate size of selected samples
- For each selected sample, perform sampling with replacement
- Compute the various statistical parameters of the given data
- Lastly, compute the mean of all statistical parameters.

In this paper, a two-dimensional chaos map known as Tangent Delay Ellipse Reflecting Cavity Map System (TD-ERCS) was applied for the random selection of samples. This technique is used widely for the generation of random numbers and permutations. This type of chaos system is preferred

for the bootstrapping method because of its equiprobability and nonlinear nature [30]. TD-ERCS can be generalized as:

$$x_n = -\frac{2k_{n-1}y_{n-1} + x_{n-1}(\mu^2 - k_{n-1}^2)}{\mu^2 + k_{n-1}^2} \tag{1}$$

$$y_n = k_{n-1}(x_n - x_{n-1}) + y_{n-1} \tag{2}$$

where  $n = 1, 2, 3, \dots$

$$k_n = \frac{2k'_{n-m} - k_{n-1} + k_{n-1}(k'_{n-m})^2}{1 + 2k_{n-1}k'_{n-m} - k(k'_{n-m})^2} \tag{3}$$

$$k'_{n-m} = \begin{cases} -\frac{x_{n-1}}{y_{n-1}}\mu^2 & n < m \\ -\frac{x_{n-m}}{y_{n-m}}\mu^2 & n \geq m \end{cases} \tag{4}$$

$$y_0 = \mu \sqrt{1 - x_0^2} \tag{5}$$

$$k'_0 = -\frac{x_0}{y_0}\mu^2 \tag{6}$$

$$k_0 = -\frac{\tan\alpha + k'_0}{1 - k'_0\tan\alpha} \tag{7}$$

$$\begin{cases} \mu \in (0, 1) \\ x_0 \in [-1, 1] \\ \alpha \in (0, \pi) \\ m = 2, 3, 4, 5, \dots \end{cases} \tag{8}$$

Here  $\mu$ ,  $x_0$ ,  $\alpha$  and  $m$  are the seed parameters. These seed parameters are used as the key in random number generation from the TD-ERCS map. Random sequences are denoted by  $x_n$  and  $y_n$  in Equations (1) and (2). Machine learning algorithms were trained by taking 100 bootstrap samples, and 44 observations were made for each bootstrap sample. Given data was tested by using unselected observations. For each chosen sample, performance matrices as well as average value ( $\bar{y}$ ) were computed. Moreover, the deviation of each value from the average value was described in terms of standard deviation (STD). A schematic diagram of the bootstrapping method is shown in Figure 6.

$$\bar{y} = \frac{1}{B} \sum_{b=1}^B y \tag{9}$$

$$STD = \sqrt{\frac{\sum_{b=1}^B (y - \bar{y})^2}{B - 1}} \tag{10}$$

### 2.2.2. Artificial Neural Network

ANN is a specialized computer program that is trained through various learning algorithms for the identification of any linear or non-linear relationship between variables of interest in any raw data set. ANN is gaining importance in almost every field of life, ranging from business, social sciences, to engineering and sciences, mainly because of its exceptional large data handling and analyzing capability. A significant amount of research work has already been conducted, both for offline and online state monitoring, in power engineering through ANN [31,32]. In the implementation of ANN analysis, it is very crucial to devise a suitable ANN model with valid input and output variables. Proper scrutiny of data is very important as it ensures the preciseness of acquired results. Once the ANN model is developed, it can then be utilized for accurate estimation of an output variable by using a given set of input values.

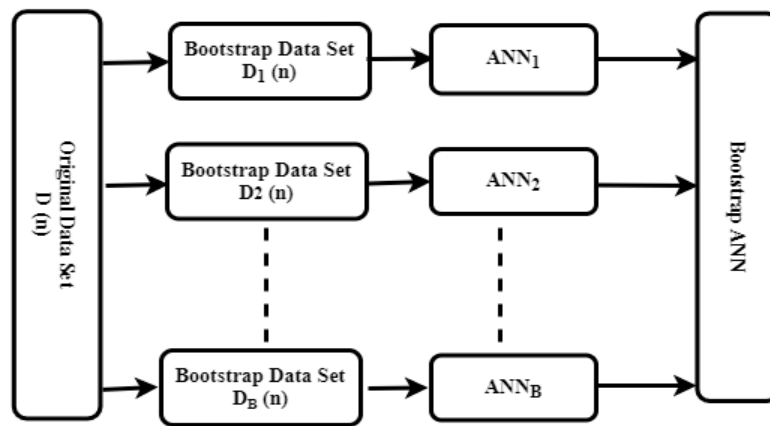


Figure 6. Bootstrap ANN architecture.

The main processing entity in the ANN model is the neuron. ANN contains many neurons which are linked to each other through specialized information-carrying pathways known as interconnections. There can be multiple inputs to a single neuron, and it can have one or more outputs. Generally, external stimuli or outputs of any other neuron act as the input to the given neuron. One possibility is that output of a neuron is fed back as the input to the same neuron. Each interconnection of neurons is associated with a weight. The output is produced only if the weighted sum of all neurons acting as input to a certain neuron crosses a predefined weighted sum limit. The ANN model contains three basic layers: the input layer, output layer, and one or more hidden layers. The number of neurons in each layer should be decided while implementing the ANN [33]. A schematic diagram of a typical ANN network is shown in Figure 7.

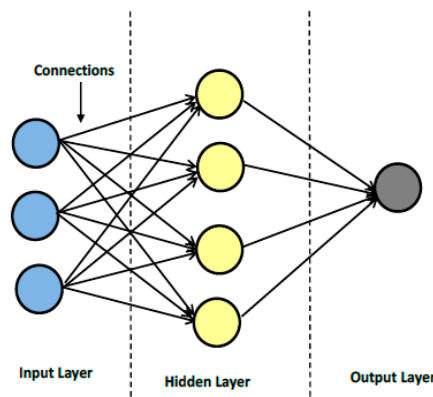


Figure 7. Generic diagram of an ANN network.

The ANN model used in this work has four inputs (ESDD, NSDD, humidity, temperature) and one output (Flashover voltage), as shown in Figure 8. The number of neurons in the hidden layer and the number of hidden layers were varied to study the effect of varying the number of neurons and hidden layers on the performance of each algorithm. Apart from that, six different types of training algorithms were used.



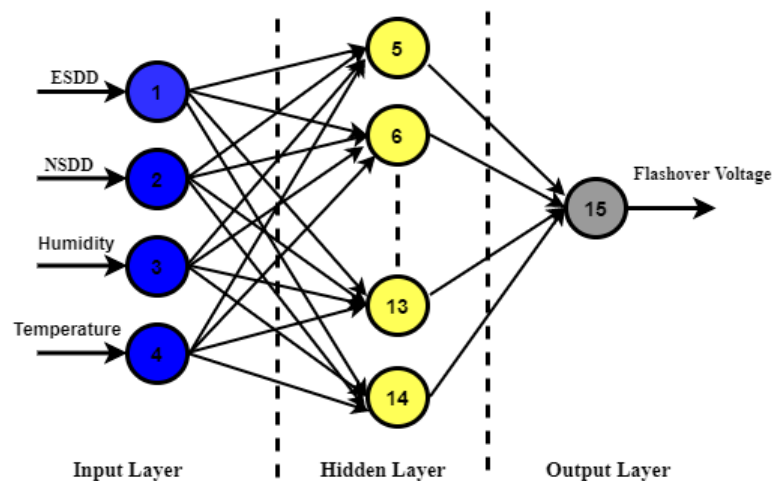


Figure 8. Schematic diagram of the proposed ANN network.

To avoid saturation while training the ANN model, it is important to perform normalization of the given data set. There are two different ways in which normalization can be performed. In the first method, normalization is achieved by considering only maximum values of input and output variables, while in the second method, both maximum and minimum values are considered. In this case, we used the first method of normalization as described below. If there are  $p = 1, 2, 3, \dots, n_p$  number of patterns,  $i = 1, 2, 3, \dots, n_i$  number of input values, and  $k = 1, 2, 3, \dots, n_k$  the number of output values. Then,

$$n_{i,max} = \max(n_i(p)) \quad (11)$$

$$O_{k,max} = \max(O_k(p)) \quad (12)$$

Therefore, normalized values are

$$n_{i,nor(p)} = \frac{n_i(p)}{n_{i,max}} \quad (13)$$

$$O_{k,nor(p)} = \frac{O_k(p)}{O_{k,max}} \quad (14)$$

After normalization, the input and output values will be between 0 and 1. The different types of learning algorithms used in this study such as GD, LM, CG, QN, RBP and BRBP are given in Appendix A.

### 3. Results

In this paper, various machine learning tools were applied to predict the critical flashover voltage of HTV silicone rubber outdoor insulators. A comparison between the predicted and actual value of flashover voltage obtained through the LM algorithm is shown in Figure 9. It can be observed from Figure 9 that forecasted values for flashover voltage are closer to the actual values. A similar comparison for the prediction of critical flashover voltages using machine learning techniques was done in Reference [34], which validates the results presented in Figure 9. For better visualization and comparison of these machine learning algorithms, it would be more appropriate to use some matrices for describing the accuracy and validity. In this paper, the accuracy and preciseness of the implemented algorithms were described in four matrices. These are Root Mean Square Error (RMSE), Normalized RMSE (N-RMSE), Mean Absolute Percentage Error (MAPE) and R value. RMSE is the square root of the average of squared errors, while NRMSE is the normalized value of RMSE. In MAPE,

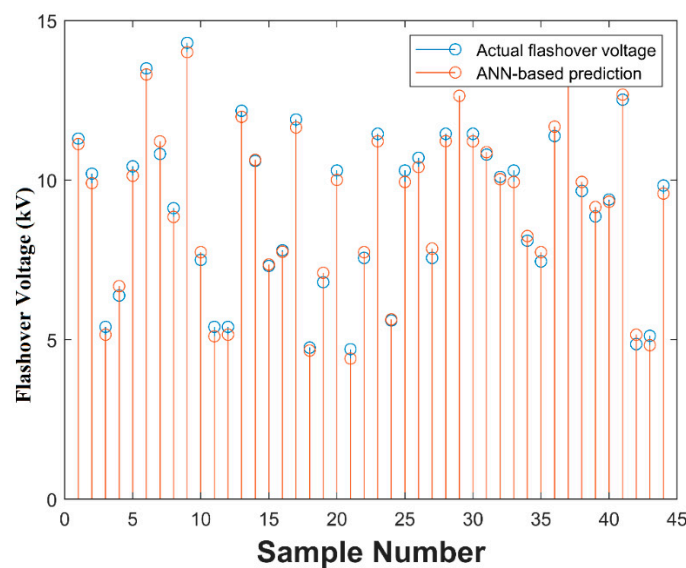
the percentage of the average of the error value is calculated. Mathematically, these matrices can be described as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (FV_{Ai} - FV_{Pi})^2} \tag{15}$$

$$NRMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{FV_{Ai} - FV_{Pi}}{FV_{Ai}}\right)^2} \tag{16}$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{FV_{Ai} - FV_{Pi}}{FV_{Ai}} \right| \times 100\% \tag{17}$$

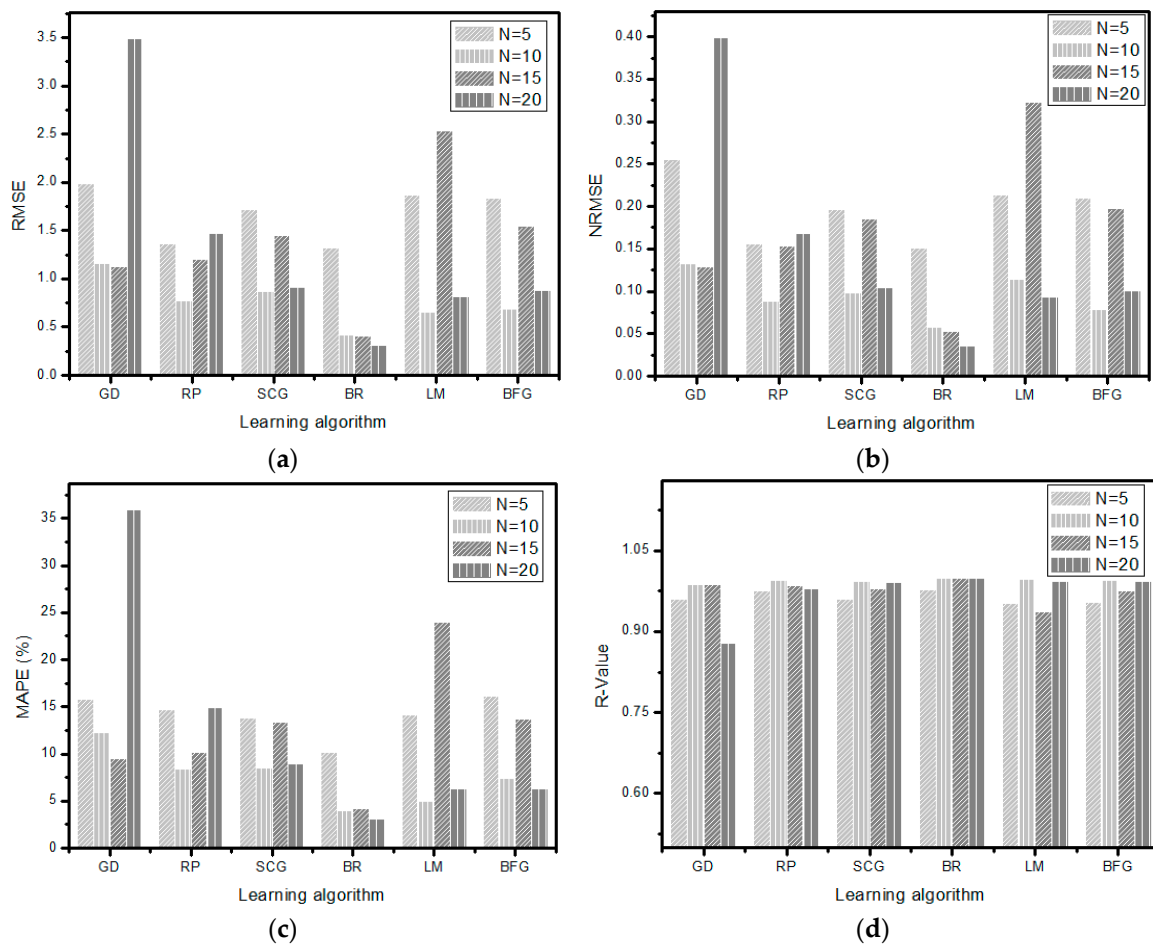
$$R = 1 - \left( \frac{\sum_{i=1}^n (FV_{Ai} - FV_{Pi})}{\sum_{i=1}^n FV_{Ai}} \right)^2 \tag{18}$$



**Figure 9.** Comparison of the predicted and actual flashover voltage using the LM learning algorithm and 10 neurons in the hidden layer.

Here, ‘*n*’ is the number of samples, ‘*FV<sub>Ai</sub>*’ and ‘*FV<sub>Pi</sub>*’ are actual and forecasted critical flashover voltage values. The values of these performance metrics must be close to some definite value. Usually, values of RMSE, NRMSE, and MAPE, which are approaching zero imply the efficient operation of a machine learning algorithm. In other words, the machine learning algorithm will be considered reliable only if its error values obtained through RMSE, NRMSE, and MAPE are approaching zero, while in terms of the R parameter, the machine learning would be rated as good enough if its error value in terms of R is closer to 1.

A performance comparison, based on variation in the number of neurons in the hidden layers of different machine learning algorithms, is depicted in Figure 10. For the GD algorithm, the error value for RMS, NRMSE, and MAPE decreases with the increasing number of neurons from 5 to 15. However, a further increase in the number of neurons to 20 results in an increase of error values. The R-value for GD first increases from 5 to 15 neurons, and further increment to 20 neurons results in a decrement of the R value. Thus, increasing the number of neurons from 15 to 20 adversely affects the performance of GD. In the case of the RP algorithm, error values for RMS, NRMSE, and MAPE first decrease on increasing neurons from 5 to 10. Further increase in the number of neurons leads to an increase of error values (RMS, NRMSE and MAPE). A similar trend is followed by the R-value where the increase of the number of neurons beyond 10 decreases the R value.

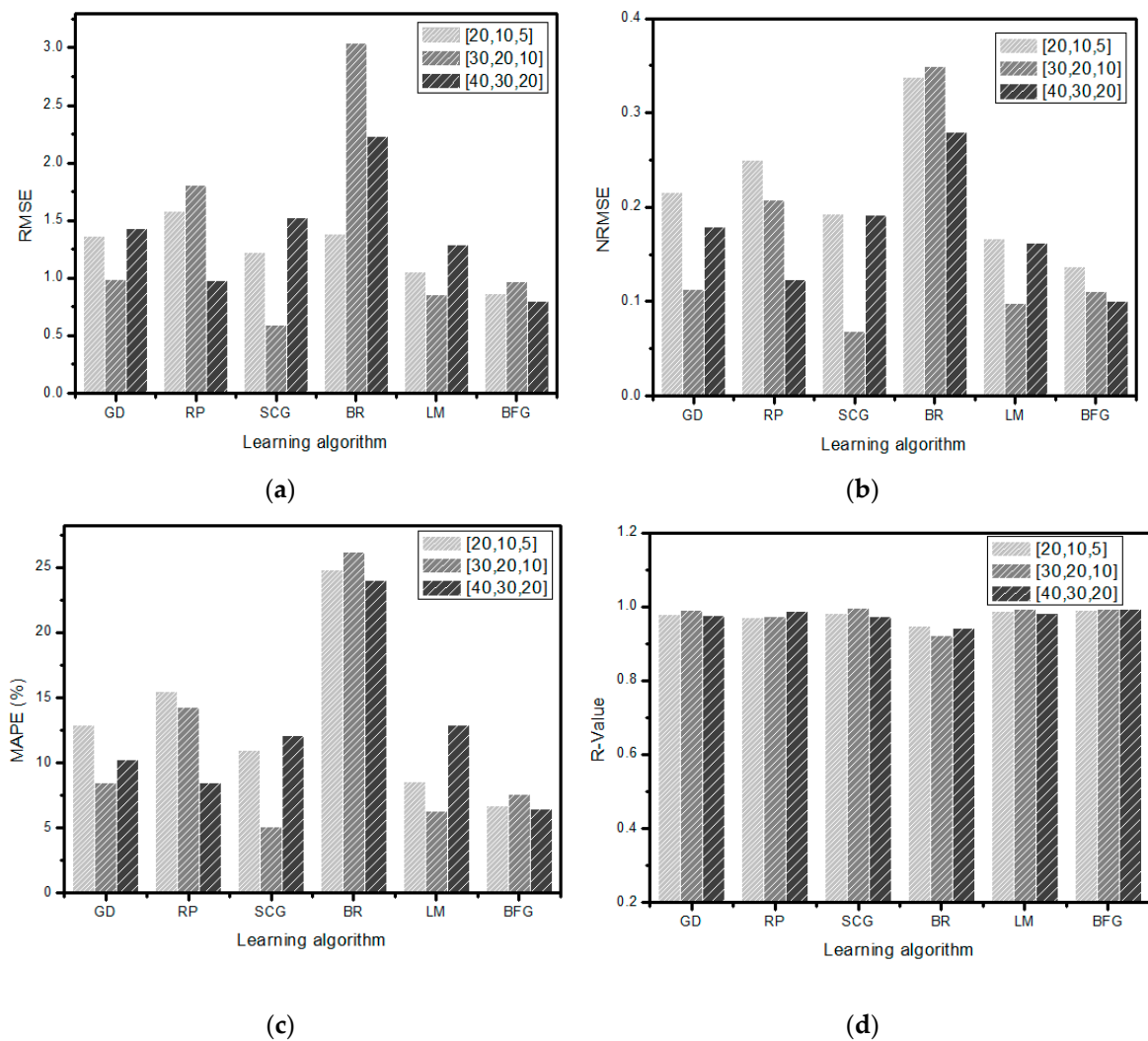


**Figure 10.** Performance parameters comparison of the different learning algorithms on the basis of changes in the number of neurons in the hidden layer. (a) RMSE; (b) NRMSE; (c) MAPE (%); (d) R value.

Thus, increasing the number of neurons from 10 to 20 adversely affects the performance of RP. The SCG, LM and BFG Quasi newton algorithms exhibit rather random behavior. In these algorithms, an increase in the number of neurons from 5 to 10 strengthens the efficiency of the given machine learning algorithm. A further increase in neurons from 10 to 15 overshoots the error values for RMSE, NRMSE, and MAPE and decreases the regression value, R. The behavior of the BR backpropagation algorithm is quite distinct from the above-stated algorithms where an increase in the number of neurons boosts the performance of ANN. Overall, it can be concluded that increasing the number of neurons to a certain limit has a healing effect on the GD algorithm and BR backpropagation algorithm. For the rest of the algorithms, the number of neurons must be chosen as the optimum, and a general trend should not be followed.

The above-mentioned results are based on a single hidden layer, and only the number of neurons in the hidden layer was varied. Increasing the number of hidden layers also effects the performance of the neural network. In this paper, three hidden layers with different numbers of neurons were considered. The results obtained are shown in Figure 11, where [x, y, z] in the legend represents the number of neurons in each hidden layer. It was noted that by increasing the number of hidden layers, the computational complexity of the proposed neural network increased; however, the computational performance of proposed algorithms was not tested in this work. Comparing the results shown in Figure 11 to that of Figure 10, it can be noted that the performance of some algorithms improved with the increased number of hidden layers, while others deteriorated at the same time. The BR backpropagation algorithm which performed better for a single hidden layer worsened when increasing the number of hidden layers and neurons. In other words, increasing the number of hidden layers

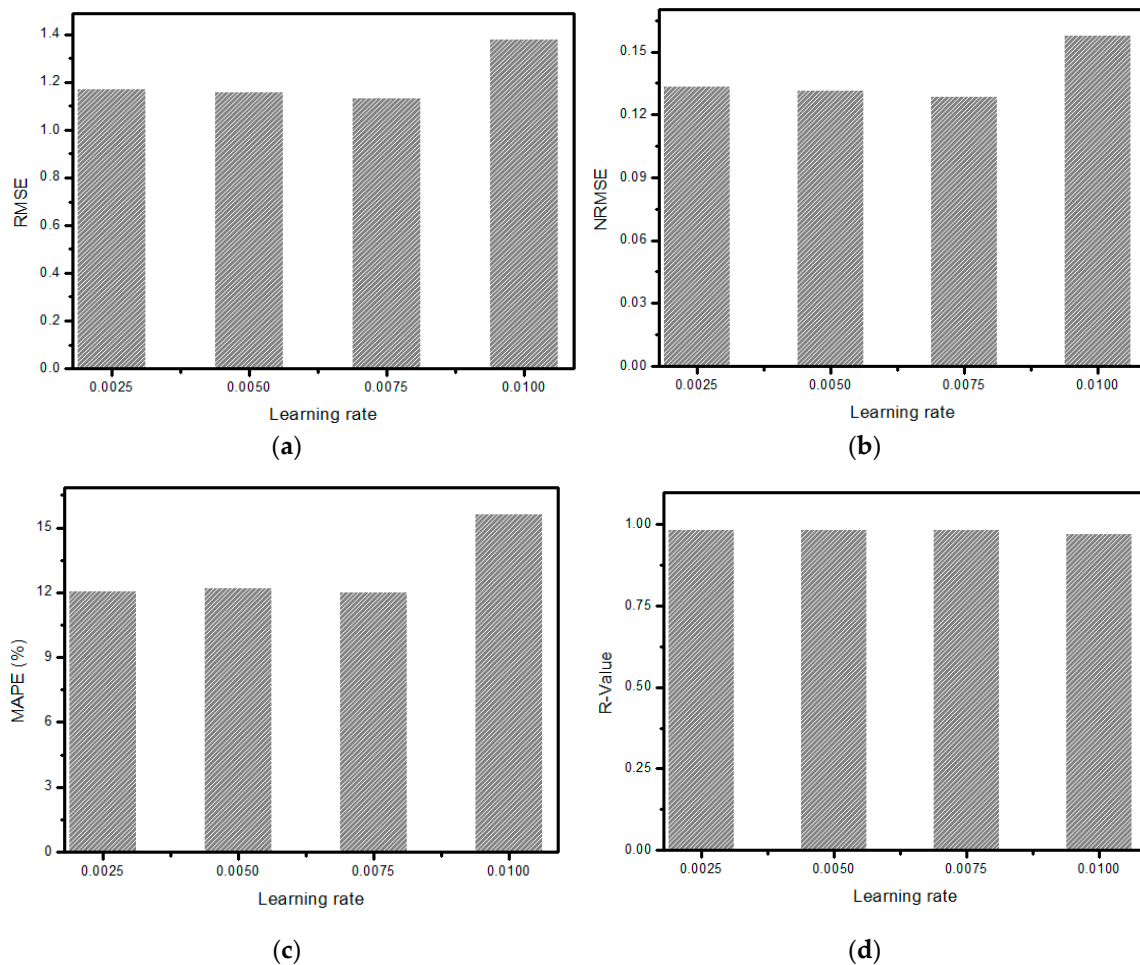
caused overfitting of the given data. Similarly, the performance of the RP algorithm is also adversely affected. On the other hand, the performance of the remaining algorithms has improved as indicated by their error values. It is very important to note here that the performance of any algorithm is also dependent on the number of neurons in that layer. All these algorithms exhibit random behavior. For example, in the case of the SCG algorithm, increasing the number of neurons in the hidden layer from [20, 10, 5] to [30, 20, 10] reduces the RMSE from 1.22 to 0.59, NRMSE from 0.19 to 0.069, and MAPE from 10.93 to 5.09%.



**Figure 11.** Performance parameters comparison of the different learning algorithms based on changes in the number of neurons and using three hidden layers. (a) RMSE; (b) NRMSE; (c) MAPE (%); (d) R value.

Choosing a certain learning rate for a neural network algorithm is also very important for improved performance. The learning rate is considered a hyperparameter in neural networks, and it accounts for alterations that should be made in the current model in response to calculated errors. A small value of the learning rate requires a large number of training epochs, whereas a large learning rate value may cause convergence of the algorithm rapidly to the local minima or maxima. Figure 12 shows the performance comparison of the GD algorithm for different learning rates. It can be observed from these plots that increasing the learning rate from 0.0025 to 0.0075 apparently does not have any significant effect on error values obtained through RMS, N-RMS, and MAPE. However, a further increase in the learning rate value depicts the dominant increment in the value of these matrices, therefore indicating

a drastic deterioration of the GD algorithm. On the other hand, the R-value did not account for any variation in the learning rate value, and it remains constant.



**Figure 12.** Performance parameters comparison of the GD algorithm for different learning rates. (a) RMSE; (b) NRMSE; (c) MAPE (%); (d) R value.

#### 4. Conclusions

In this paper, different training algorithms of ANN were applied for the prediction of critical flashover voltage of insulators. These learning algorithms were applied by varying various parameters like the number of neurons, hidden layers, and learning rate. It was found that increasing the number of neurons to a certain limit can boost the performance of the machine learning algorithm for accurate prediction of flashover voltage, but after crossing a certain threshold, any further increase deteriorates the performance. Similarly, increasing the number of hidden layers had a positive influence on machine learning algorithms, except BR backpropagation, whose performance was affected badly with increasing hidden layers. The performance of the GD algorithm changed with the changing learning rate. Any inappropriate value may lead to large prediction errors of deployed algorithms. Therefore, it is important to choose the optimum values of the learning rate, number of neurons, and hidden layers for better performance of the machine learning algorithm. Additionally, the performance of the ANN algorithm is related to the type of learning algorithm utilized. These results can help scientists and engineers choose the best learning algorithm and associated parameters while predicting the critical flashover voltage of outdoor polymeric insulators.

**Author Contributions:** Data curation, J.A., F.A., F.A.B. and F.A.-A.; Investigation, A.; Methodology, M.T.K.N., A. and J.A.; Project administration, F.A., F.A.B., and F.A.-A.; Software, J.A.; Validation, J.A., A., F.A.-A.; Visualization,

F.A. and F.A.B.; Writing—original draft, M.T.K.N., and A.; Writing—review & editing, A., F.A.B. and F.A.-A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** In this section, you can acknowledge any support given, which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

This section describes the details of the learning algorithms used to train the neural network.

### Appendix A.1. Gradient Descent

The GD method is usually applied for maximization or minimization of any n-dimensional function. It is described in the form of a gradient vector 'g' that points towards the steepest point of the given n-dimensional function  $f(x_n)$ , given that 'g' is differentiable on that point. Mathematically, it can be written as:

$$g(x_1, x_2, x_3, \dots, x_n) = \nabla f(x_1, x_2, x_3, \dots, x_n) \quad (A1)$$

where '∇' is the gradient operator. The gradient is equivalent to the derivative of that point, therefore, a negative gradient will always point to the steepest point (minima) of the given function ' $f(x_n)$ '. In physical terms, gradient descent means moving downwards in steps proportional to the magnitude of gradient vector '|g|'. There are many shortcomings associated with this algorithm, like convergence to local minima instead of global and a low convergence rate. But due to low memory requirements, it is still considered a good algorithm for processing large data sets.

### Appendix A.2. Conjugate Gradient Descent (CGD)

For quadratic functions, the GD algorithm exhibits slow convergence and involves many iterations. Therefore, to overcome this shortcoming, the CDG method was introduced in Reference [35]. This algorithm detects the minimum of any quadratic function 'f' by searching in orthogonal directions.

Let  $f(x)$  be the function to be minimized, with 'x' being a vector of 'N' variables. The CGD algorithm consists of the following steps:

---

#### The CGD Algorithm

---

1. Start with initial set point of ' $x_0$ ' (iteration =  $k = 0$ ).
2. In the second step, direction is computed as given below:

$$g(k) = \frac{\partial f(w(k))}{\partial w(k)}$$

- i- If  $g(k) = 0$ , then  $x(k)$  is already present at optimal minimum point.
- ii- If  $g(k) \neq 0$  and  $k = 0$ , then  $r(k) = -g(k)$ , move to step 3.
- iii- If  $g(k) \neq 0$  and  $k > 0$ , then  $r(k)$  can be calculated as:

$$r(k) = -g(k) + \frac{g^H(k) \cdot g(k)}{g^H(k-1) \cdot g(k-1)} \times r(k-1)$$

3. In this step,  $w(k+1)$  is calculated, which steers to the minima of function 'f' in the direction  $w(k) + \alpha \times r(k)$ .
4.  $k = k + 1$ , move back to step 2.

Optimal point is obtained in  $K$  iterations where  $K \leq N$

---

Appendix A.3. Quasi Newton Method

The Quasi-Newton method is an improved form of Newton’s method. Newton’s method requires a lot of computational space as it involves calculation of the Hessian matrix. In the Quasi-Newton method, an approximation of the inverse Hessian matrix is made at each iteration step. Newton’s method initiates by finding out first derivative ‘ $\nabla_x f(x)$ ’, with an initial estimate of ‘ $x^k$ ’. This nonlinear function ‘ $\nabla_x f(x^k + u)$ ’ can be expanded by applying the Taylor series, up to two terms only:

$$\nabla_x f(x^k + u) = g(x^k + u) = g(x^k) + \nabla_x g(x^k)u \tag{A2}$$

Further, setting it equal to zero and assuming  $u = u^k$

$$g(x^k) + \nabla_x g(x^k)u = 0 \tag{A3}$$

$$\nabla_x g(x^k)u = -g(x^k) \tag{A4}$$

where ‘ $\nabla_x g(x^k)$ ’ is Hessian matrix. As stated above, the Quasi-Newton calculates only the approximate value for the Hessian matrix. This approximation is made possible by making use of Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. By applying the Quasi-Newton BFGS algorithm, the following secant condition is obtained:

$$\nabla_x^2 f(x^{k+1})(x^{k+1} - x^k) \approx \nabla_x f(x^{k+1}) - \nabla_x f(x^k) \tag{A5}$$

Hessian matrix ‘ $\nabla_x^2 f(x^{k+1})$ ’ is replaced by approximation ‘ $H^{k+1}$ ’.

$$H^{k+1}d^k = y \tag{A6}$$

where,  $d^k = (x^{k+1} - x^k)$ , and  $y = \nabla_x f(x^{k+1}) - \nabla_x f(x^k) = g^{k+1} - g^k$ .

Hessian matrix ‘ $H^{k+1}$ ’ can be calculated by an earlier computed Hessian matrix ‘ $H^k$ ’ as follows:

$$H^{k+1} = H^k + \frac{g^k g^{kT}}{d^k g^{kT}} + \frac{y^k y^{kT}}{d^k y^{kT}} \tag{A7}$$

Moreover, further simplification to the given problem is made by assuming  $A^k = H^{-k}$

And  $A^{k+1}$  can be computed as:

$$A^{k+1} = \left( I - \frac{d^k y^{kT}}{d^{kT} y^k} \right) A^k \left( I - \frac{y^k d^{kT}}{d^{kT} y^k} \right) + \frac{d^k d^{kT}}{d^{kT} y^k} \tag{A8}$$

Appendix A.4. Levenberg-Marquardt (LM)

The LM algorithm is widely used for solving nonlinear least-squares problems. It is also termed as the damped least-square method. This algorithm does not involve the computation of the Hessian matrix. However, it incorporates the Jacobean matrix and gradient vector for obtaining the optimal point of any function ‘ $f$ ’. The optimal point is calculated using the following steps:

---

**Calculate Steps**


---

- Let  $j_k$  denotes Jacobian matrix,  $d_k$  indicates search direction, the initial iteration parameter value is set to greater than zero ( $a_1 > 0$ ):

$$0 \leq p_0 \leq p_1 \leq p_2 \leq 1$$

- If  $\|j_k^T F_k\| \leq \varepsilon$ , terminate the criteria here. Otherwise

$$\mu_k = \alpha_k(\theta \|F_k\| + (1 + \theta) \|j_k^T F_k\|)$$

compute  $d_k$  as:

$$(j_k^T j_k + \mu_k I) d = -j_k^T F_k$$

- Then solve the following equation

$$r_k = \frac{\|F_k\|^2 - \|F(x_k + d_k)\|^2}{\|F_k\|^2 - \|F_k + j_k d_k\|^2}$$

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } r_k > p_0 \\ x_k & \text{else} \end{cases}$$

$$\alpha_{k+1} = \begin{cases} 4\alpha_k & \text{if } r_k < p_1 \\ \alpha_k & \text{if } r_k \in [p_1, p_2] \\ \max\{\frac{\alpha_k}{4}, m\} & \text{else} \end{cases}$$

$k = k + 1$ , move back to step 2.

---

#### Appendix A.5. Resilient Backpropagation

The RBP algorithm is one of the most widely deployed learning algorithms in neural networks. In this algorithm, the magnitude of the partial derivative is ignored, and only its sign is used as an indication for introducing any alterations in weights. An update in the weight is made only if the sign of partial derivative changes. The work of this algorithm can be summarized as [36]:



---

**The RBP Algorithm**


---

1. If the sign of derivative of the given function does not change in the next succeeding iterations, then an update in the weight is made as indicated below.

$$\text{If } \left( \frac{\partial E}{\partial w_k} (t-1) \times \frac{\partial E}{\partial w_k} (t) \right) > 0$$

Then

$$\Delta k_j(t) = \text{minimum} \left( \Delta k_j(t-1) \times \eta_+, \Delta \text{max} \right)$$

$$\Delta w_k_j(t) = -\text{sgn} \left( \frac{\partial E}{\partial w_k} (t) \right) \times \Delta k_j(t)$$

$$\Delta w_k_j(t+1) = w_k_j(t) + \Delta w_k_j(t)$$

2. But, if the sign of derivative changes in the next iteration then weight decreases as shown below.

$$\text{If } \left( \frac{\partial E}{\partial w_k} (t-1) \times \frac{\partial E}{\partial w_k} (t) \right) < 0, \text{ Then}$$

$$\Delta k_j(t) = \text{maximum} \left( \Delta k_j(t-1) \times \eta_-, \Delta \text{min} \right)$$

$$\Delta w_k_j(t+1) = w_k_j(t) - \Delta w_k_j(t-1)$$

$$\frac{\partial E}{\partial w_k} (t) = 0$$

3. In case, derivative is equal to zero then no changes are made to the weight value:

$$\text{If } \left( \frac{\partial E}{\partial w_k} (t-1) \times \frac{\partial E}{\partial w_k} (t) \right) = 0, \text{ Then}$$

$$\Delta w_k_j(t) = -\text{sgn} \left( \frac{\partial E}{\partial w_k} (t) \right) \times \Delta k_j(t)$$

$$\Delta w_k_j(t+1) = w_k_j(t) + \Delta w_k_j(t)$$

Where  $\Delta k_j$  = size of update

---

### Appendix A.6. Bayesian Regularization Backpropagation

Traditional backpropagation method performs the task of minimization of given function

$$F = E_d, \tag{A9}$$

where

$$E_d = \sum_{i=1}^n (t_i - a_i)^2 \tag{A10}$$

In this equation, 'n' denotes the number of training inputs, 't<sub>i</sub>' indicates anticipated output and 'a<sub>i</sub>' is the i<sup>th</sup> output obtained as a result of neural network operation.

In regularization problems, the objective function is described as.

$$F = \alpha E_w + \beta E_d \tag{A11}$$

$$E_w = \sum_{i=1}^n w_i^2 \tag{A12}$$

Here, 'E<sub>w</sub>' is the penalty factor and is equivalent to the addition of the squares of all network weights and 'α and β' are regularization parameters. It is very important to obtain optimal values for these regularization parameters. Generally, smaller weights for these parameters are preferred as it

enhances the generalization capability of the given network. Too large a value of  $\alpha$  ( $\alpha \gg \beta$ ) results in tolerance to higher errors. The converse condition ( $\alpha \ll \beta$ ) may lead to overfitting. In Reference [37], David Mackay presented a methodology for obtaining optimum weights of regularization parameters, commonly known as Bayesian regularization.

In the Bayesian regularization algorithm, a network’s weights are considered as random variables. Let ‘ $D$ ’ indicate a training data set for a particular neural network model ‘ $M$ ’, then the posterior distribution for network’s weights can be written as;

$$P(w|D, \alpha, \beta, M) = \frac{P(D|w, \beta, M)P(w|\alpha, M)}{P(D|\alpha, \beta, M)} \tag{A13}$$

Here, ‘ $w$ ’ is the vector containing network’s weights,  $P(w|\alpha, M)$  is prior distribution,  $P(D|w, \beta, M)$  is likelihood function and  $P(D|\alpha, \beta, M)$  is a normalization term. Normalization factor  $P(D|\alpha, \beta, M)$  can be expressed as;

$$P(D|\alpha, \beta, M) = \int P(D|w, \beta, M)P(w|\alpha, M)dw \tag{A14}$$

By considering the nature of noise in training data and prior distribution to be Gaussian in nature, we can write then,

$$P(D|w, \beta, M) = \frac{1}{Z_D(\beta)} \exp(-\beta E_d) \tag{A15}$$

$$P(w|\alpha, M) = \frac{1}{Z_w(\alpha)} \exp(-\alpha E_w) \tag{A16}$$

where

$$Z_D(\beta) = \left(\frac{\pi}{\beta}\right)^{\frac{n}{2}} \tag{A17}$$

$$Z_w(\alpha) = \left(\frac{\pi}{\alpha}\right)^{\frac{N}{2}} \tag{A18}$$

$$P(w|D, \alpha, \beta, M) = \frac{\exp(-F(w))}{Z_F(\alpha, \beta)} \tag{A19}$$

$$Z_F(\alpha, \beta) = Z_D(\beta) + Z_w(\alpha) \cdot P(D|\alpha, \beta, M) \tag{A20}$$

The main purpose is to find out the values or weights that will cause minimization of ‘ $F(w)$ ’. In other words, this is analogous to maximization of  $P(w|D, \alpha, \beta, M)$ . So, by Baye’s rule:

$$P(\alpha, \beta|D, M) = \frac{P(D|\alpha, \beta, M)P(\alpha, \beta|M)}{P(D|M)} \tag{A21}$$

By considering the prior density  $P(D|\alpha, \beta, M)$  to be uniform, then maximization of posterior  $P(\alpha, \beta|D, M)$  will be equal to the maximization of  $P(D|\alpha, \beta, M)$ .

$$P(D|\alpha, \beta, M) = \frac{\left(\frac{1}{Z_D(\beta)}\right) \exp(-\beta E_D) \left(\frac{1}{Z_w(\alpha)}\right) \exp(-\alpha E_w)}{\left(\frac{1}{Z_F(\alpha, \beta)}\right) \exp(-F(w))} \tag{A22}$$

$$P(D|\alpha, \beta, M) = \frac{Z_F(\alpha, \beta)}{Z_D(\beta)Z_w(\alpha)} \tag{A23}$$

' $Z_w(a)$ ' and ' $Z_D(\beta)$ ' are already known values. ' $Z_{F(\alpha,\beta)}$ ' can be estimated by Taylor expansion. For normalization constant, we can solve it as:

$$Z_F = 2\pi^{\frac{N}{2}} \left( \det(H^{MP})^{-1} \right)^{\frac{1}{2}} \exp(-F(w)^{MP}) \quad (A24)$$

Here, ' $H$ ' is Hessian matrix and can be calculated as

$$H = \beta \nabla^2 E_D + \alpha \nabla^2 E_w \quad (A25)$$

Putting the value of ' $Z_F$ ' and further solving it gives us the optimum weight of ' $\alpha$ ' and ' $\beta$ ' at ' $w_{MP}$ '. So,

$$\alpha^{MP} = \frac{\gamma}{2E_w(w^{MP})} \quad (A26)$$

And

$$\beta^{MP} = \frac{n - \gamma}{2E_D(w^{MP})} \quad (A27)$$

## References

1. Arshad; Nekahi, A.; McMeekin, S.G.; Farzaneh, M. Flashover characteristics of silicone rubber sheets under various environmental conditions. *Energies* **2016**, *9*, 683. [CrossRef]
2. Hamza, A.-S.H.; Abdelgawad, N.M.; Arafa, B.A. Effect of desert environmental conditions on the flashover voltage of insulators. *Energy Convers. Manag.* **2002**, *43*, 2437–2442. [CrossRef]
3. Farzaneh, M.; Chisholm, W.A. *Insulators for Icing and Polluted Environments*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 47.
4. Asimakopoulou, G.E.; Kontargyri, V.T.; Tsekouras, G.J.; Elias, C.N.; Asimakopoulou, F.E.; Stathopoulos, I.A. A fuzzy logic optimization methodology for the estimation of the critical flashover voltage on insulators. *Electr. Power Syst. Res.* **2011**, *81*, 580–588. [CrossRef]
5. Mahdjoubi, A.; Zegnini, B.; Belkheiri, M.; Seghier, T. Fixed least squares support vector machines for flashover modelling of outdoor insulators. *Electr. Power Syst. Res.* **2019**, *173*, 29–37. [CrossRef]
6. Samakosh, J.D.; Mirzaie, M. Flash-over voltage prediction of silicone rubber insulators under longitudinal and fan-shaped non-uniform pollution conditions. *Comput. Electr. Eng.* **2019**, *78*, 50–62. [CrossRef]
7. Belhouchet, K.; Bayadi, A.; Bendib, M.E. Artificial neural networks and genetic algorithm modelling and identification of arc parameter in insulators flashover voltage and leakage current. *Int. J. Comput. Aided Eng. Technol.* **2019**, *11*, 1–13. [CrossRef]
8. Kamarudin, M.S.; Othman, N.A.; Jamail, N.A.M. Artificial Intelligence Techniques for Predicting the Flashover Voltage on Polluted Cup-Pin Insulators. *Emerg. Trends Intell. Comput. Inform. Data Sci. Intell. Inf. Syst. Smart Comput.* **2019**, *1073*, 362.
9. Lu, S.; Lin, G.; Liu, H.; Ye, C.; Que, H.; Ding, Y. A weekly load data mining approach based on hidden Markov model. *IEEE Access* **2019**, *7*, 34609–34619. [CrossRef]
10. Farshad, M. Detection and classification of internal faults in bipolar HVDC transmission lines based on K-means data description method. *Int. J. Electr. Power Energy Syst.* **2019**, *104*, 615–625. [CrossRef]
11. Narayanan, V.J.; Sivakumar, M.; Karpagavani, K.; Chandrasekar, S. Prediction of Flashover and Pollution Severity of High Voltage Transmission Line Insulators Using Wavelet Transform and Fuzzy C-Means Approach. *J. Electr. Eng. Technol.* **2014**, *9*, 1677–1685. [CrossRef]
12. Natarajan, A.; Narayanan, S. S-transform based time-frequency analysis of leakage current signals of transmission line insulators under polluted conditions. *J. Electr. Eng. Technol.* **2015**, *10*, 616–624. [CrossRef]
13. Prasad, P.S.; Rao, B.P. Review on Machine Vision based Insulator Inspection Systems for Power Distribution System. *J. Eng. Sci. Technol. Rev.* **2016**, *9*, 135–141. [CrossRef]
14. Lan, L.; Gorur, R.S. Computation of AC wet flashover voltage of ceramic and composite insulators. *IEEE Trans. Dielectr. Electr. Insul.* **2008**, *15*, 1346–1352. [CrossRef]
15. Venkataraman, S.; Gorur, R.S. Prediction of flashover voltage of non-ceramic insulators under contaminated conditions. *IEEE Trans. Dielectr. Electr. Insul.* **2006**, *13*, 862–869. [CrossRef]

16. Zhang, D.; Zhang, Z.; Jiang, X.; Yang, Z.; Zhao, J.; Li, Y. Study on Insulator Flashover Voltage Gradient Correction Considering Soluble Pollution Constituents. *Energies* **2016**, *9*, 954. [[CrossRef](#)]
17. Li, Y.; Yang, H.; Zhang, Q.; Yang, X.; Yu, X.; Zhou, J. Pollution flashover calculation model based on characteristics of AC partial arc on top and bottom wet-polluted dielectric surfaces. *IEEE Trans. Dielectr. Electr. Insul.* **2014**, *21*, 1735–1746. [[CrossRef](#)]
18. Hadjrioua, F.; Mahi, D.; Slama, M.E.A. Application of the analytical arc parameters on the dynamic modeling of HVDC flashover of polluted insulators. In Proceedings of the 2014 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM), Tunis, Tunisia, 3–6 November 2014; pp. 1–5.
19. Badachi, C.; Dixit, P. Prediction of pollution flashover voltages of ceramic string insulators under uniform and non-uniform pollution conditions. *J. Electr. Syst. Inf. Technol.* **2016**, *3*, 270–281. [[CrossRef](#)]
20. Shahabi, S.; Gholami, A. Dynamic model to predict AC critical flashover voltage of nonuniformly polluted insulators under thermal ionization conditions. *Turk. J. Electr. Eng. Comput. Sci.* **2017**, *25*, 2322–2335. [[CrossRef](#)]
21. Palangar, M.F.; Mirzaie, M.; Mahmoudi, A. Improved flashover mathematical model of polluted insulators: A dynamic analysis of the electric arc parameters. *Electr. Power Syst. Res.* **2020**, *179*, 106083. [[CrossRef](#)]
22. Salem, A.A.; Abd Rahman, R.; Kamarudin, M.S.; Othman, N.A.; Jamail NA, M.; Hamid, H.A.; Ishak, M.T. An alternative approaches to predict flashover voltage on polluted outdoor insulators using artificial intelligence techniques. *Bull. Electr. Eng. Inform.* **2020**, *9*, 533–541. [[CrossRef](#)]
23. Salem, A.A.; Abd-Rahman, R.; Al-Gailani, S.A.; Kamarudin, M.S.; Othman, N.A.; Jamail, N.A.M. Artificial Intelligence Techniques for Predicting the Flashover Voltage on Polluted Cup-Pin Insulators. In *International Conference of Reliable Information and Communication Technology*; Springer: Cham, Switzerland, 2019; pp. 362–372.
24. Zegnini, B.; Mahdjoubi, A.H.; Belkheiri, M. A least squares support vector machines (LS-SVM) approach for predicting critical flashover voltage of polluted insulators. In Proceedings of the 2011 Annual Report Conference on Electrical Insulation and Dielectric Phenomena, Cancun, Mexico, 16–19 October 2011; pp. 403–406.
25. Cho, M.Y.; Lin, P.S. Using Support Vector Machine for Classifying Insulator Leakage Current. *Intl. J. Electr. Comput. Sci. IJECS-IJENS* **2015**, *15*, 30–38.
26. Gencoglu, M.T.; Uyar, M. Prediction of flashover voltage of insulators using least squares support vector machines. *Expert Syst. Appl.* **2009**, *36*, 10789–10798. [[CrossRef](#)]
27. Saranya, K.; Muniraj, C. A SVM Based Condition Monitoring of Transmission Line Insulators Using PMU for Smart Grid Environment. *J. Power Energy Eng.* **2016**, *4*, 47–60. [[CrossRef](#)]
28. Govindaraju, P.; Saranya, K.; Muniraj, C. Condition monitoring of transmission line insulators using PMU for smart grid environment. In Proceedings of the 2016 6th International Conference on Intelligent and Advanced Systems (ICIAS), Kuala Lumpur, Malaysia, 15–17 August 2016; pp. 1–6.
29. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning*; Springer: New York, NY, USA, 2013; Volume 112.
30. Khan, J.S.; Ahmad, J.; Khan, M.A. TD-ERCS map-based confusion and diffusion of autocorrelated data. *Nonlinear Dyn.* **2015**, *87*, 93–107. [[CrossRef](#)]
31. Jeyasurya, B. Application of artificial neural networks to power system transient energy margin evaluation. *Electr. Power Syst. Res.* **1993**, *26*, 71–78. [[CrossRef](#)]
32. Daut, M.A.M.; Hassan, M.Y.; Abdullah, H.; Rahman, H.A.; Abdullah, M.P.; Hussin, F. Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review. *Renew. Sustain. Energy Rev.* **2017**, *70*, 1108–1118. [[CrossRef](#)]
33. Nielsen, M.A. *Neural Networks and Deep Learning*; Determination Press: San Francisco, CA, USA, 2015; Volume 2018.
34. Arshad; Ahmad, J.; Tahir, A.; Stewart, B.G.; Nekahi, A. Forecasting Flashover Parameters of Polymeric Insulators under Contaminated Conditions Using the Machine Learning Technique. *Energies* **2020**, *13*, 3889. [[CrossRef](#)]
35. Cheney, E.W.; Kincaid, D.R. *Numerical Mathematics and Computing*; Cengage Learning: Boston, MA, USA, 2012.

36. Patil, S.; Naik, G.M.; Pai, K.R. An application of wavelet transform and artificial neural network for microarray gene expression based brain tumor sub-classification. *Int. J. Emerg. Technol. Adv. Eng.* **2015**, *5*, 410–414.
37. MacKay, D.J. Bayesian interpolation. *Neural Comput.* **1992**, *4*, 415–447. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).