OPTIMIZING PERFORMANCE OF A TRANSMISSION LINE RELAYING ALGORITHM IMPLEMENTED USING AN ADAPTIVE, SELF-ORGANIZED NEURAL NETWORK

Slavko Vasilic
Texas A&M University
College Station, U.S.A.
svasilic@ee.tamu.edu

Mladen Kezunovic
Texas A&M University
College Station, U.S.A.
kezunov@ee.tamu.edu

Dejan Sobajic
Electric Power Research Institute
Palo Alto, U.S.A.
dsobajic@msm.epri.com

Abstract – This paper deals with neural network protection algorithm aimed at classifying transmission line faults. The approach utilizes self-organized adaptive type of neural network specially developed to deal with large amount of data. Various procedures of preprocessing neural network inputs are extensively investigated. The classification performance for different values of data window length and sampling frequency is demonstrated and illustrated. Simulation results show satisfactory algorithm responses in each of the implemented cases.

Keywords: Power System Faults, Neural Networks, Learning Systems, Protective Relaying, Pattern Clustering Methods

1 INTRODUCTION

This study discusses a new protective relaying principle for transmission lines based on the artificial neural network based pattern recognition. The approach is to apply the neural network directly to the samples of voltage and current measurements instead of using traditional concept of impedance measurement. The main role of the relaying principle is detecting and classifying the faults, based on three-phase voltage and current samples. Transmission line faults happen randomly, and they are outcome of varying conditions. Several varying parameters (type of fault, fault location, fault impedance, and fault incident time) as well as many other conditions imposed by actual network configuration (source/load variations, and variety of switching events), determine the corresponding transient current and voltage waveforms detected by the relays at line ends. All these effects influence relay's classification capability. Traditional concept is based on computing the settings ahead of time taking into account only easily anticipated fault conditions, and not the prevailing real conditions.

The new concept is based on special type of neural network ideally suited for protection tasks [1-4]. The current and voltage samples of the three transmission line phases are recognized as features of the events in the power network. They identify whether, where and which type of fault occurs under a variety of time-varying operating conditions [5-6]. Whenever any change in the prevailing conditions at the relay location occurs, this self-adaptive algorithm is capable of adjusting its response to deal with similar situations that may happen in the future [7-8].

Special attention is placed on the sensitivity of the neural network protection algorithm to the variations in the input signal preprocessing steps, including the data window size and choice of sampling frequency. Conditioning of input signals may adversely affect the algorithm behavior during the training as well as the performance during the testing. Mentioned parameters determine the number of neural network inputs, and influence the trade-off between performing the fault characterization more accurately and making the decision in real time. Requirements for real time fault classification impose the amount of calculation allowed to produce the final outcome, while the requirements for accuracy also need to be met.

For algorithm design and evaluation, one typical model of an actual power network was implemented in a program for simulating power network transients [9]. This model is used to simulate various fault and no-fault events in the network. Simulation output data are used for forming the training and testing patterns submitted as inputs to the neural network.

The paper is organized as follows. Description of a self-organized neural network algorithm is given in section II. Section III shows the selected model of an actual power network. Simulation of scenarios and preprocessing of neural network input signals are explained in Section IV. Simulation results obtained through algorithm training and testing are provided in Section V. The conclusions are given at the end.

2 SELF-ORGANIZED NEURAL NETWORK CLUSTERING ALGORITHM

2.1 Neural Networks for Protective Relaying

Various applications of neural networks were used in the past to improve recognition of the impedance utilized in distance relaying of transmission lines [10]. These applications are mainly based on widely used multilayer feed-forward networks with back-propagation learning rule [11]. Implementation of these networks for patterns with large dimensionality imposes slow training procedure, very easily converges to local minimum and on-line learning requires presenting the entire set of training patterns again. This type of network gives an output in a continuous range, and there is a difficulty in how to interpret such values for protection tasks.

Instead of using multilayer neural network, an idea is to apply neural network directly to the samples of voltages and currents, and produce the fault type and zone classification in real time. The problem which impose special requirements on this task is the large number of training and testing patterns due to a variety of power network fault and no-fault events. Also, a large dimensionality of patterns due to frequent taking of samples of 3-phase voltages and currents in given data window may pose a problem. Consequently, a very large amount of data may be expected due to the specific nature of the process and selected classification task, which usually includes detection of the fault type and zone. To be able to produce the fault detection and classification in real time, a special type of neural network is used.

2.2 The Structure of Used Neural Network

This neural network does not have a typical predetermined layered structure but rather an adaptive and self-organized structure, typical for Adaptive Resonance Theory (ART) networks [8]. Network structure depends only on the characteristics and presentation order of the input data set. It is a clustering algorithm which allocates input patterns into groups called clusters, depending on mutual similarity between the patterns. The cluster centers are recognized as pattern prototypes. The algorithm, based on modified ISODATA clustering algorithm, discovers the most representative positions of prototypes in the pattern space [12]. Similarly to Self-Organizing Maps (SOM) and Radial Basis Function (RBF) networks, the cluster (prototype) positions are dynamically updated during presentation of input patterns [7,13]. But contrary to SOM, RBF and ISODATA, the initial number of clusters and their positions are not specified in advance. The training starts by forming the first cluster with only the first input pattern assigned. A new cluster is formed incrementally whenever a new pattern, dissimilar to all previously presented patterns, appears. Otherwise, the new pattern is allocated into cluster with the most similar patterns. Using presented technique, the on-line training due to non-stationary inputs may be easily implemented. The similarity between patterns is measured by calculating the Euclidean distance between their features (input vector components). After training, the centers of self-organized clusters represent typical prototypes of input patterns so that each pattern is assigned to a unique cluster. However, each cluster contains one or more similar input patterns. The cluster center (pattern prototype) represents the center of gravity of all patterns allocated to that cluster. A class symbolizes a group of clusters with a common characteristic (in this particular case a class is a specific type of fault in a specific zone of relay protection), and each cluster belongs to one of possible classes. The number of classes corresponds to the desired number of neural network outputs, determined by the given classification task. The neural network training consists usually of few hundreds of iterations with consecutively alternating unsupervised and supervised learning phases (Fig. 1).

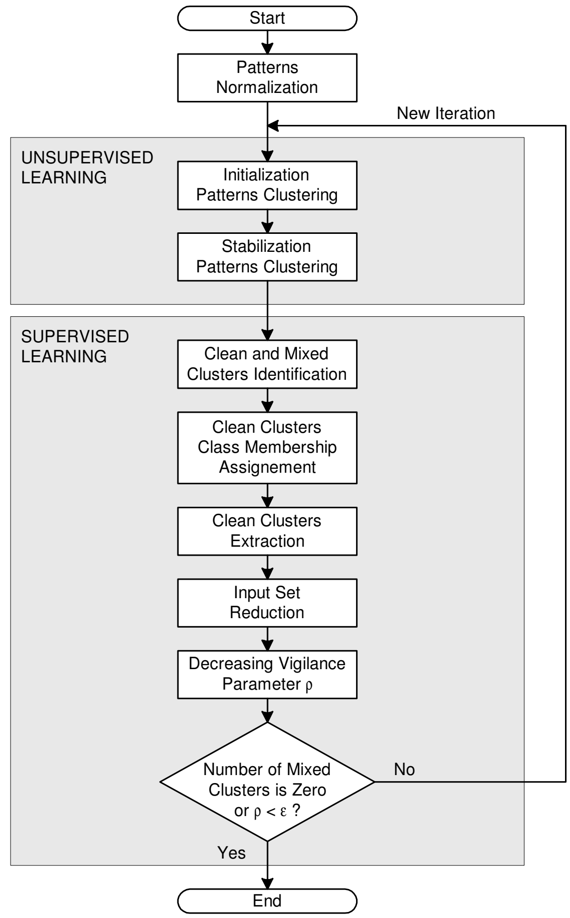

Figure 1: Neural Network Clustering Algorithm.

2.3 Unsupervised Learning

The initial data set, containing all the patterns, is firstly processed using unsupervised learning, applied as the clustering algorithm. During unsupervised learning, patterns are presented without their class labels. This procedure tries to identify characteristic (typical) patterns or prototypes that can serve as cluster centers. The number of clusters is not specified, but a strong interclass distance measure is specified. It does not require either the initial guess of the number of cluster, or the initial cluster center coordinates. The outcome of unsupervised learning is a stable family of clusters, defined as spheres in an n-dimensional space, where the space dimension is determined with the length of input vector (i.e. number of features). Unsupervised learning forms both "clean" (having patterns with the same class label) and "mixed" (having patterns with two or more class labels) clusters. It consists of the initialization and stabilization phases.

The initialization phase establishes initial cluster structure based on similarity between the patterns, and by presenting each pattern only ones. During stabilization phase the clustering algorithm is reiterated many times until a stable cluster structure occurs when no pattern changes its cluster membership during the iterations.

2.4 Supervised Learning

In the supervised learning the class label is associated with each data point (pattern). Supervised learning separates "mixed" clusters from the "clean" ones. "Clean" clusters create a reference set of labeled clusters, because they are recognized as characteristic prototypes of presented input data set. Vigilance parameter is a tuning parameter and is being decreased after each iteration. It controls the number and size of generated clusters. The whole procedure, including unsupervised and supervised learning, is repeated many times until only "clean" clusters exist. In this case the problem of convergence during learning does not exist as in the standard supervised learning, because the learning is controlled by only one tuning parameter.

One illustrative example of a reference set of clusters related to fault classification requirements of transmission line protective relaying is shown in Fig. 2. It relates to classification of fault type and fault location (zone of relay protection where fault has happened). It is significantly simplified and given in only two dimensions.

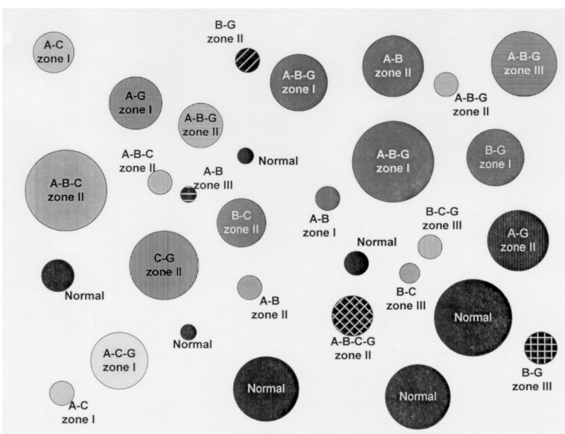

Figure 2: Example of cluster structure established during training.

2.5 Testing

Test patterns are classified according to their similarity to prototypes adopted during training. Classification is performed by applying the K-nearest neighbor classifier (decision rule) to the cluster structure established during training [14]. Given a set of classified data, the K-nearest neighbor classifier determines the classification of a new pattern based on the most represented class label amongst the K nearest clusters, retrieved from the cluster structure adopted during training. This classifier is efficiently implemented since the number of optimized prototypes is significantly smaller than the number of training patterns. Consequently, input parameter for algorithm testing is only the number which determines how many nearest neighbors have to be taken into account. The outcome of the testing is a class label assigned to each test pattern. Thus, output of this neural network is in the discrete form inherently reflecting different types of faults common in protective relaying.

3 POWER NETWORK MODEL AND FAULT SCENARIOS

A typical 345kV power system section, from Reliant Energy (RE) HL&P, was modeled for relay testing and simulation studies (Fig. 3). The modeling involved two major steps: first, obtaining reduced Thevenin equivalent sources for all the boundary buses; second, detailed modeling of all the elements of the studied section (STP-SKY). The reduced network equivalent was obtained by using the load flow and short circuit data, and verified using both the steady state and transient state results (recordings captured during actual fault events in the system). This reduced system is to be used for simulation of various fault events and operating states. Appropriate transient signals from this system will be utilized for performing evaluation and testing of the relaying algorithm.

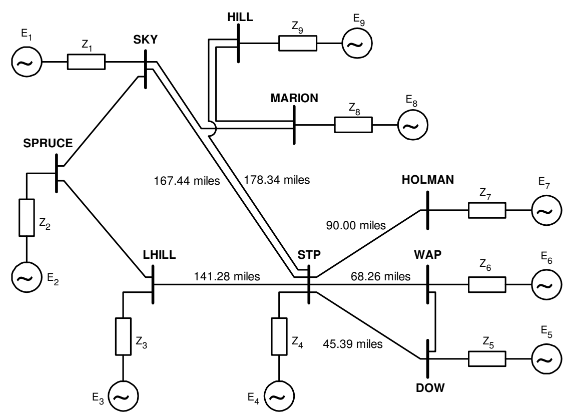

Figure 3: RE HL&P STP-SKY Power Network Model.

4 IMPLEMENTATION

4.1 Simulation of Scenario Cases

Model of the given power network is implemented in the Alternative Transient Program (ATP) program [15]. This model is used for simulating various fault scenarios on one of the transmission lines (STP-SKY), by varying fault parameters. Neural network based protection algorithm takes voltage and current measurements from the SKY end of the line and has to be trained to recognize the characteristics of any fault on that line. The characteristics include fault type and corresponding fault location in either zone I or II of relay protection. Specially developed custom interface enables running simulations for a large number of different scenarios by changing network topologies and parameters. Current and voltage samples obtained through simulations are used for forming training and testing patterns for the neural network algorithm design, implemented in MATLAB [16].

Scenarios for neural network training are generated by specifying several values for each of four fault parameters and combining such values to cover diversity of fault cases. Parameters used for generation of the training patterns are: all 11 types of fault (AG, BG, CG, AB, BC, CA, ABG, BCG, CAG, ABC, ABCG) and normal state; fault distances at 5 to 95 % of the line length in increments of 10%; fault resistance (for ground faults) of 0, 10, 20 Ohms; fault angle between 0 to 330 degrees, in increments of 30 degrees. Total number of training patterns by combining all the parameters is 2652.

A set of test patterns for algorithm evaluation in previously unseen situations was generated by random setting of all fault parameters. Total number of testing patterns is 1000.

4.2 Preprocessing of Neural Network Input Signals

The example of the simulation output data for one specific case, phase-A-to-B-to-ground fault (ABG), is shown in Fig. 4.

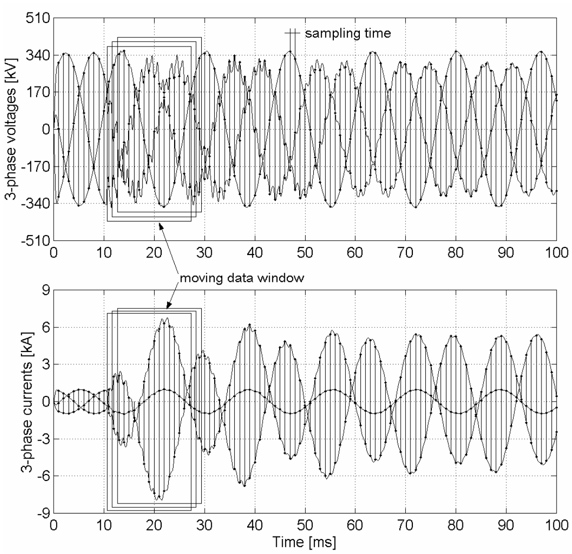

Figure 4: Moving data window for voltage and current samples.

Before the neural network training starts, certain preprocessing of the network inputs is needed. Pattern extraction from obtained measurements depends on several tuning parameters and they significantly determine quality of subsequent algorithm training. Phase current measurements are filtered by an analog filter and sampled with desired sampling frequency. Each pattern is extracted from the samples obtained in a desired length of a moving data window, normalized, and arranged together to form a common vector with feature components.

The effect of selecting different data windows is illustrated in Fig. 5. for the window lengths of 0.5 and 1 cycles (8.33 and 16.67 ms). Taking longer window increases the number of features used for training, and gives better information about the signals. At the same time it means slower training and testing, and may cause difficulties in classifying the fault in real time.

Sampling frequency has similar effect on forming the patterns as the data window does. Pattern feature vectors for sampling frequencies of 2 kHz and 4 kHz are shown in Fig. 5. Increased sampling frequency offers improved signal detection but also may cause significant computational burden.

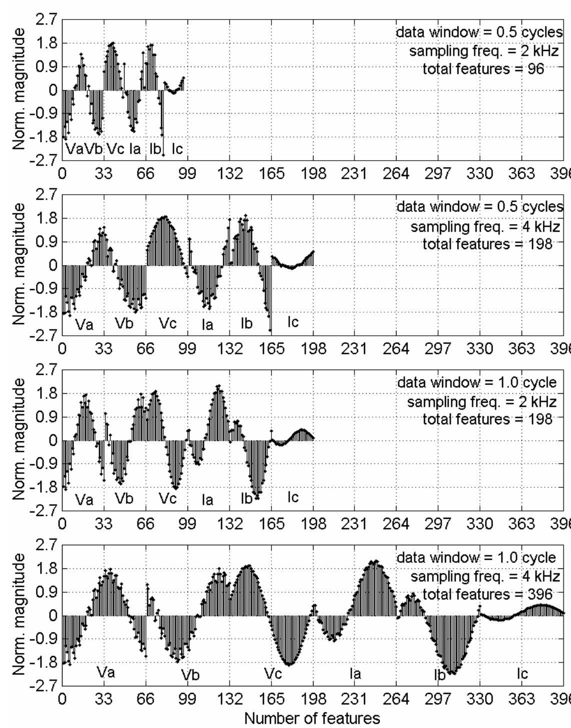

Figure 5: Example of the patterns for different values of the data window and sampling frequency.

The scaling ratio is a value used for multiplying only the voltage samples (during pattern extraction) to assign a higher or lower impact on the relay decision versus current samples. Using this value, the voltage contribution may be increased or decreased over the current contribution (Fig. 6). Optimal scaling ratio should be determined for each particular implementation whenever either the training set or classification task is changed.

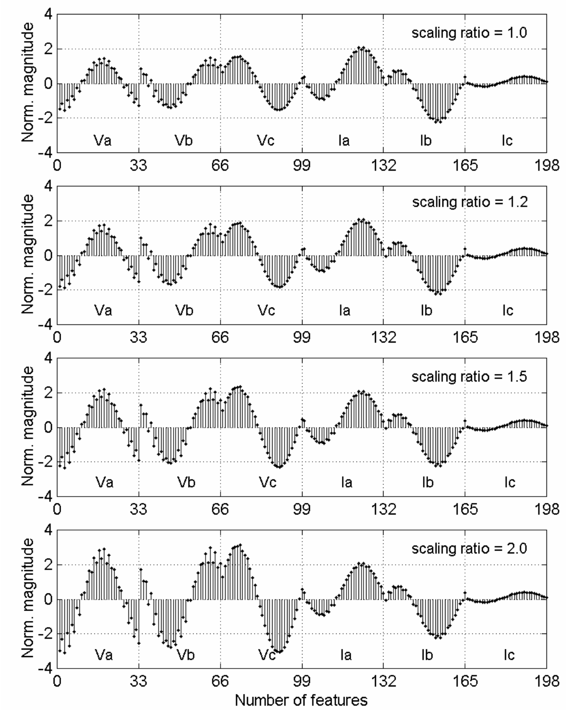

Figure 6: Example of the patterns established using different scaling ratios for multiplying voltage samples.

Parameters used for forming the patterns are: 3 phase voltages and currents selected as data source for training; second order analog Butterworth filter with selected crossover frequency of 1 kHz; time window for taking patterns of either 8.33 (1/2 cycles) or 16.67 ms (1 cycle); sampling frequency of either 2 kHz (33 samples per cycle) or 4 kHz (66 samples per cycle). Features of the training patterns are extracted by using simulation data obtained in desired time windows. The selected values for data window and sampling frequency give four possible combinations with 96, 198 (two times) and 396 components, as shown in Fig. 5. Also, data were normalized, and the same normalization factor has been used for normalizing the test patterns.

There are 4 sets of training patterns, based on simulation of the same cases. The waveforms are sampled with different sampling frequency and taken with different time length of the moving data window.

5 SIMULATION RESULTS

Training was performed for establishing the cluster structure capable of recognizing different types of fault (Normal, AG, BG, CG, AB/ABG, BC/BCG, CA/CAG, ABC/ABCG) and zones of fault (Normal, Zone I, Zone II). The boundary between the first and second zone is preset to be at 80% of the line length.

After numerous iterations, the training procedures for each set of training patterns terminated with various number of clusters (prototypes). For each cluster structure the training was repeated several times until optimal scaling ratio between voltage and current samples is identified. The outcome of training is the cluster structures given in Tab. 1.

cluster structure	patterns characteristic			number of generated clusters	classif. error [%]
	data window (cycles)	sampl. freq. (kHz)	dimension (number of features)		
I	0.5	2.0	96	530	3.1
II	0.5	4.0	198	534	2.3
III	1.0	2.0	198	589	2.0
IV	1.0	4.0	396	646	1.9

Table 1: Cluster structures as outcome of trained network for various values of data window and sampling frequency.

Optimal value for parameter K is 1 in each implemented case, hence the cluster identified as the nearest cluster always completely determines the classification of each test pattern.

Classification results obviously show that the case with longer data window and higher sampling frequency (with 396 vectors) gives the lowest error. Increasing data window and sampling frequency enables better classification. The results are still very good for smaller data window and lower sampling frequency, which corresponds to commonly used frame for standard distance relaying.

Propagation of classification error during testing for all cluster structures is shown in Fig. 7. In the beginning of testing, presented number of testing patterns is quite small and classification error considerably varies, but then slowly converges to the final stable value.

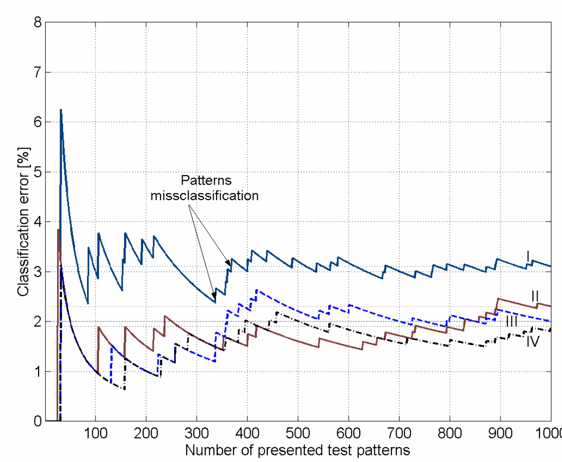

Figure 7: Propagation of classification error during testing.

It can be observed that the proposed approach assures satisfactory detection of the fault type and zone, even for the short data window and low sampling frequency. This is the most important requirement for proper action when protecting the network against transmission line faults.

6 CONCLUSION

This study discusses neural network based protective relaying approach implemented as adaptive, self-organized clustering algorithm based on combined use of unsupervised and supervised learning strategies. This algorithm guarantees well tuned pattern recognition capabilities for the prevailing operating conditions.

An example of actual power network was modeled and used to simulate various fault events in the network. Neural network based clustering algorithm is used to form pattern prototypes (homogenous structure of clusters representing various subsets of input data set) related to different events. Testing patterns are classified by combining the established cluster structure and K-nearest neighbor classifier.

The most important aspect of this research is to show how various preprocessing steps may influence the algorithm classification capability. Conditioning of input signals, i.e. selecting the values for data window and sampling frequency for taking the patterns, play significant role in the algorithm behavior during training and performance testing. Different aspects of these factors are illustrated through several examples. Furthermore, classification of the test patterns is analyzed through comparison of various cluster structures, generated through training, based on combination of different values for data window length and sampling frequency. Selecting longer data window and higher sampling frequency gives better information about the signals, but produces slower training and testing. This may cause difficulties in classifying the fault in a real-time application. Optimal scaling between voltage and current samples has also been applied in each particular case.

Proposed tunings of the neural network algorithm may enable better selection of the most representative set of samples that forms a neural network input. Simulation results show satisfactory behavior of the algorithm for each of the implemented cluster structures used in classifying the fault type and zone of the fault.

REFERENCES

[1] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Reading: Wesley, 1989, p. 309.

[2] Y. H. Pao and D. J. Sobajic, "Combined Use of Unsupervised and Supervised Learning for Dynamic Security Assessment", *IEEE Trans. Power Systems*, vol. 7, no 2, pp. 878-884, 1992.

[3] M. Kezunovic M., I. Rikalo, and D. Sobajic, "High-speed Fault Detection and Classification with Neural Nets", *Electric Power Systems Research*, vol. 34, pp. 109-116, 1995.

[4] M. Kezunovic and I. Rikalo, "Detect and Classify Faults Using Neural Nets", *IEEE Computer Applications in Power*, vol. 9, no. 4, pp. 42-47, 1996.

[5] S. Vasilic and M. Kezunovic, "New Design of a Neural Network Algorithm for Detecting and Classifying Transmission Line Faults", IEEE PES Transmission and Distribution Conf., Atlanta, GA, U.S.A., Oct. 2001.

[6] S. Vasilic and M. Kezunovic, "An Improved Neural Network Algorithm for Classifying the Transmission Line Faults", IEEE PES Power Winter Meeting, New York, NY, U.S.A., Jan. 2002.

[7] T. Kohonen, *Self-Organizing Maps*, 3rd ed., New York: Springer, 2001, p. 501.

[8] G. A. Carpenter and S. Grossberg, "ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns", *Applied Optics*, vol. 26, no. 23, pp. 4919-4930, Dec. 1987.

[9] M. Kezunovic and S. Vasilic, "Design and Evaluation of Context-Dependent Protective Relaying Approach", IEEE Porto Power Tech' Conf., Porto, Portugal, Sep. 2001.

[10] M. Kezunovic, "A Survey of Neural Net Applications to Protective Relaying and Fault Analysis", *Engineering Intelligent Systems*, vol. 5, no. 4, pp. 185-192, Dec. 1997.

[11] B. Widrow and M. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation", *IEEE Proceedings*, vol. 78, no. 9, pp. 1415-1442, Sep. 1990.

[12] G. H. Ball and D. J. Hall, "A Clustering Technique for Summarizing Multivariate Data", *Behavioral Science*, vol. 12, pp. 153-155, 1967.

[13] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks", *IEEE Transaction on Neural Networks*, vol. 2, no. 2, pp. 302-309, Mar. 1991.

[14] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification", *IEEE Trans. Information Theory*, vol. IT-13, pp. 21-27, Jan. 1967.

[15] CanAm EMTP User Group, *Alternative Transient Program (ATP) Rule Book*, Portland, 1992.

[16] The MathWorks, Inc., *Using MATLAB*, Natick, Jan. 1999.