Fuzzy ART Neural Network Algorithm for Classifying the Power System Faults

Slavko Vasilic, *Student Member, IEEE,* and Mladen Kezunovic, *Fellow, IEEE*

Abstract—This paper introduces advanced pattern recognition algorithm for classifying the transmission line faults, based on combined use of neural network and fuzzy logic. The approach utilizes self-organized, supervised Adaptive Resonance Theory (ART) neural network with fuzzy decision rule applied on neural network outputs to improve algorithm selectivity for a variety of real events not necessarily anticipated during training. Tuning of input signal preprocessing steps and enhanced supervised learning are implemented, and their influence on the algorithm classification capability is investigated. Simulation results show improved algorithm recognition capabilities when compared to a previous version of ART algorithm for each of the implemented scenarios.

Index Terms—Adaptive resonance theory, clustering methods, fuzzy logic, learning systems, neural networks, pattern classification, power system faults, protective relaying, testing, training.

I. INTRODUCTION

THE issue of detecting and classifying the transmission line faults based on three-phase voltage and current signals has been known for a long time. It was addressed some time ago by introducing the traditional relaying principles such as overcurrent, distance, under/over voltage, differential, etc. [1]. All of these principles are based on predetermined network configuration taking into account worst-case fault conditions. The settings determined by the classical approach have inherent limitations in classifying certain faults if the network actual configuration deviates from the anticipated one. In such instances, the existing relays may miss operate [2]. Consequently, a more dependable and secure relaying principle is needed for classifying the faults under a variety of time-varying network configurations and events.

The idea of using neural networks in protective relaying is not new. Various applications of neural networks were used in the past to improve some of the standard functions used in protection of transmission lines. They have been related to fault classification [3]–[9], fault direction discrimination [10]–[12], fault section estimation [8], [13], [14], adaptive relaying [15], [16], autoreclosing [17], [18], and fault diagnosis [19], [20]. The applications are mainly based on widely used Multilayer Perceptron (MLP) feed-forward networks [1]–[8], [11], [13], [15], [17]–[19], [21], and in recent years on Radial Basis Function (RBF) [5], [6], [9], [14], [22], Self-Organizing Maps (SOM) [5], [16], [23], Learning Vector Quantization (LVQ) [5], [23], Adaptive Resonance Theory (ART) [20], [24], Recurrent [12], [25], Counterpropagation [5], [26], and Finite Impulse Response neural networks [10], [27].

The new concept proposed in this paper is based on special type of self-organized, competitive neural network, called Adaptive Resonance Theory (ART), ideally suited for classifying large, highly-dimensional and time-varying set of input data [24], [28]. The new classification approach can reliably conclude, in a rather short time, whether, where and which type of the fault occurs under varying operating conditions [29]. This type of neural network algorithm for relay protection has already been used in its original form and with simplified assumptions about the network and operating conditions [30]–[33]. This paper introduces several enhancements of the mentioned original version of the algorithm [34]–[38]. They include: a.) Improved preprocessing of neural network inputs affecting the algorithm sensitivity; b.) Redefined concept of supervised learning which now allows improved neural network generalization capabilities; c.) Attuned fuzzy decision rule allowing an interpolation of neural network outputs; d.) Results of extensive solution evaluation, which cover a variety of power system operating conditions and events. The new version of the ART neural network algorithm is compared to the original version using elaborate modeling and simulation set up that represents a segment of an actual 345 kV network from CenterPoint Energy in Houston.

The paper is organized as follows. Section II provides the description of the neural network algorithm. A new technique for fuzzyfication of neural network outputs is introduced in Section III. Proposed relaying solution based on pattern recognition is outlined in Section IV. Power system modeling and simulation, pattern generation, design of the ART neural network based algorithm, as well as corresponding simulation results are given in Section V. The conclusions are summarized in Section VI. An elaborate list of relevant references is given at the end.

II. NEURAL NETWORK ALGORITHM

A. The Adaptive Neural Network Structure

The ART neural network is a typical representative of competitive networks. It tries to identify natural groupings of patterns from large data set through clustering. Groups of similar input patterns are allocated into clusters, defined as hyper-

spheres in multidimensional space, where the length of input pattern determines the space dimension. ART neural network discovers the most representative positions of cluster centers which represent pattern prototypes [24]. Similarly to SOM and LVQ networks, the prototype positions are dynamically updated during presentation of input patterns [23]. Contrary to SOM and LVQ the initial number of clusters and cluster centers are not specified in advance, but the clusters are allocated incrementally.

A diagram of the complete procedure for neural network training is shown in Fig. 1. The training consists of numerous iterations of alternating unsupervised and supervised learning stages [30]. The neural network firstly uses unsupervised learning with unlabeled patterns to form fugitive clusters. The category labels are then assigned to the clusters during the supervised learning stage.

The tuning parameter, called threshold parameter, controls the size and hence the number of generated clusters. It is being consecutively decreased during iterations. If threshold parameter is high, many different patterns can then be incorporated into one cluster. This leads to formation of a small number of coarse clusters. If threshold parameter is low, only very similar patterns activate the same cluster. This leads to creation of a large number of fine clusters.

After training, the structure of prototypes solely depends on density of input patterns. A category label that symbolizes a group of clusters with a common symbolic characteristic is assigned to each cluster, meaning that each cluster belongs to one of existing categories. The number of categories corresponds to the desired number of neural network outputs, determined by the given classification task.

B. Unsupervised Learning

The initial data set, containing all the patterns, is firstly processed using unsupervised learning, realized as modified ISO-DATA clustering algorithm [39], [40]. During this stage patterns are presented without their category labels. Neither the initial guess of the number of cluster nor their position is specified in advance, but only a strong distance measure between cluster prototypes needs to be defined. Unsupervised learning consists of two steps: initialization and stabilization.

The *initialization phase* incrementally iterates all the patterns and establishes initial cluster structure based on similarity between patterns. The entire pattern set is presented only once. Training starts by forming the first cluster with only the first input pattern assigned. New clusters are formed incrementally whenever a new pattern, sufficiently dissimilar to all previously presented patterns, appears. Otherwise, the pattern is allocated into the cluster with the most similar patterns. The similarity is measured by calculating the Euclidean distance between a pattern and existing prototypes. This phase does not reiterate the patterns, and although the clusters change their positions during incremental presentation of the patterns, patterns already presented are not able to change clusters. Consequently, the output is an initial set of unstable clusters, and stabilization phase is needed to refine the number and positions of the clusters.

The *stabilization phase* is being reiterated numerous times until the initial unstable cluster structure becomes stable and
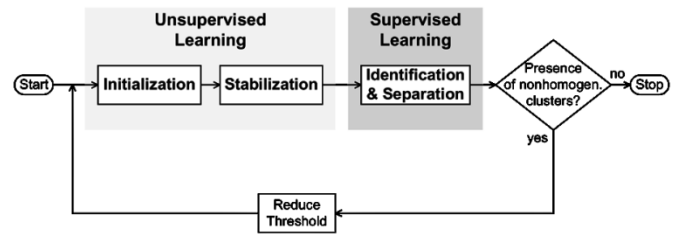

Fig. 1. Neural network training using combined unsupervised and supervised learning.

clusters retain all their patterns after single iteration. This enables more consistent matching of input pattern density.

Unsupervised learning produces a set of stable clusters, including homogenous clusters containing patterns of the identical category, and nonhomogenous clusters containing patterns of two or more categories.

C. Supervised Learning

During supervised learning, the category label is associated with each input pattern allowing identification and separation of homogenous and nonhomogenous clusters. Category labels are assigned to the homogeneous clusters. They are being added to the memory containing categorized clusters, including their characteristics like prototype position, size, and category. The patterns from homogeneous clusters are removed from further unsupervised-supervised learning iterations. The set of remaining patterns, present in nonhomogenous clusters, is transformed into new, reduced input data set and used in next iteration. The convergence of learning process is efficiently controlled by threshold parameter. This parameter is slightly decreased in each algorithm iteration. The learning is completed when all the patterns are grouped into homogeneous clusters.

An interesting phenomenon has been observed during supervised learning stage. Whenever clusters allocated to different categories mutually overlap, certain number of their patterns may fall in overlapping regions. In the previous ART version (ART^1), although each of such patterns has been nominally assigned to the nearest cluster, their presence in clusters of other category leads to questionable validity of those clusters. One typical example of a small set of two-dimensional training patterns and obtained clusters is given in Fig. 2. The training patterns that belong to different categories are noted with different symbols. Many of the patterns are at the same time members of two or more clusters with different categories. The clusters are shown with different colors depending on their category. The ambiguity can be resolved by introducing restricted condition for identification of homogeneous clusters.

The improved algorithm (ART^2) revises supervised learning by requiring the homogeneous cluster to encompass patterns of exactly one category. Both subsets of patterns, one assigned to the cluster, and other encompassed by the cluster but assigned to other cluster, will be taken into account during supervised learning. Consequently, ART^2 allows overlapping between the clusters of different categories only if there are no patterns in the overlapping regions. For the same set of training patterns, ART^2 produces finer graining of cluster structure, noticeable in Fig. 2.
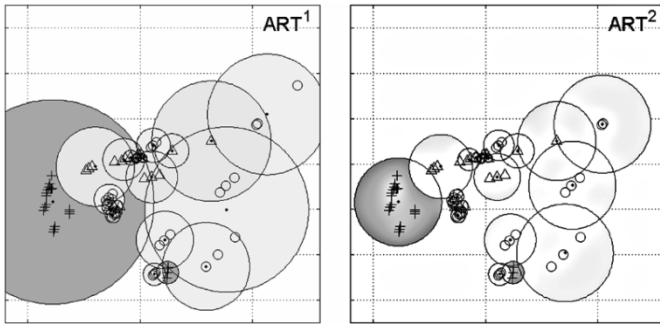
Fig. 2.   Comparison of cluster structures generated by ART^1 and ART^2.

D. Training Demo

The transformation of input patterns into clusters during unsupervised and supervised learning procedures is demonstrated using a *Training Demo*. The *Demo* is applied on a simplified real problem, but with retention of all its main characteristics. Simplification is based on reduced number of training patterns, which belong to only three categories. The pattern length is selected to be only two allowing pattern and cluster representation in two dimensional graph. Training patterns, shown in Fig. 3, are generated by simulating three types of ground faults, combined with six values of fault distance, three values of fault impedance and fifteen values of fault inception angle. Total number of presented patterns is $3 \times 6 \times 3 \times 15 = 810$. Patterns are built by using one sample per phase of two-phase currents, few sampling time steps after fault has occurred. Different symbols are used for showing patterns corresponding to various types of the fault.

After numerous unsupervised-supervised learning iterations, the final structure of homogeneous clusters and their prototypes is generated as shown in Fig. 4. The clusters have different size, location, and category. During implementation of a trained neural network, the cluster structure is used as an initial abstraction tool for facilitating classification of new patterns.

E. Implementation

During implementation or testing, new patterns are classified according to their similarity to the pattern prototypes generated during training. The classification is performed by interpreting the outputs of a trained neural network through K-Nearest Neighbor (K-NN) classifier [41]. The K-NN classifier determines the category of a new pattern based on the majority of categories represented in a pre-specified small number of nearest clusters retrieved from the cluster structure established during training. It requires only the number K that determines how many neighbors have to be taken into account. K-NN classifier seems to be very straightforward and is efficiently employed since the number of prototypes is significantly smaller then the number of training patterns.

III. FUZZYFICATION OF NEURAL NETWORK OUTPUTS

A. The Necessity for Applying Fuzzy Logic

The main advantage of the K-NN classifier is its computational simplicity. Substantial disadvantage is that each of the K clusters in the neighborhood is considered equally important in
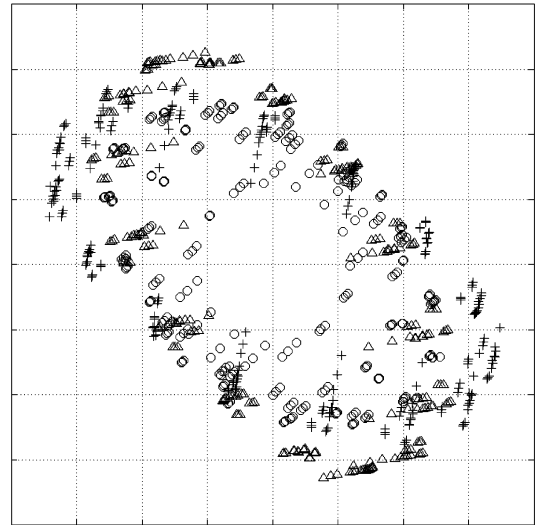

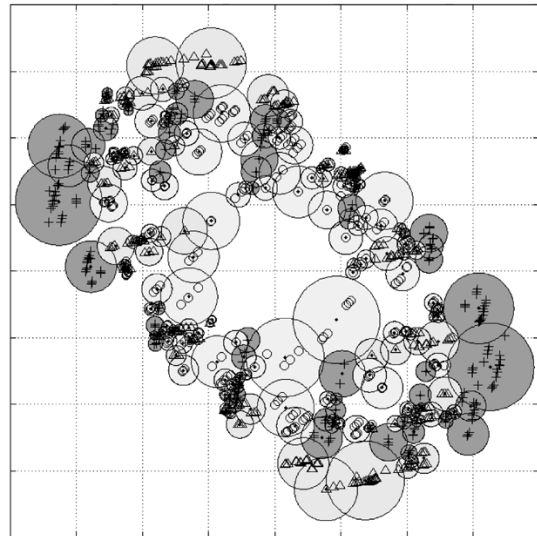Fig. 3.   Training Demo: set of labeled training patterns.

Fig. 4.   Training Demo: final outcome of consecutive unsupervised-supervised learning stages.

determining the category of the pattern being classified, regardless of their size and distances to the pattern. Using such classifier, smooth and reliable boundaries between the categories cannot be established.

An unambiguous situation exists whenever a new pattern is very close to only one of the prototypes and intuitively has to be classified to the category of that prototype. One such example is shown in Fig. 5. A small portion of a previously obtained cluster structure (in Fig. 4) is enlarged. In reality the events are quite diverse. The corresponding patterns might appear in unlabeled space between the clusters or in their overlapping regions, and be more or less similar to several prototypes located nearby, and possibly labeled with different categories.

Obviously, K-Nearest Neighbor classifier used in the past needs to be improved to achieve better generalization of the patterns that correspond to a new set of events, previously unseen during training, and conceivably dissimilar to any of the existing prototypes. The classification of a new pattern may be redefined
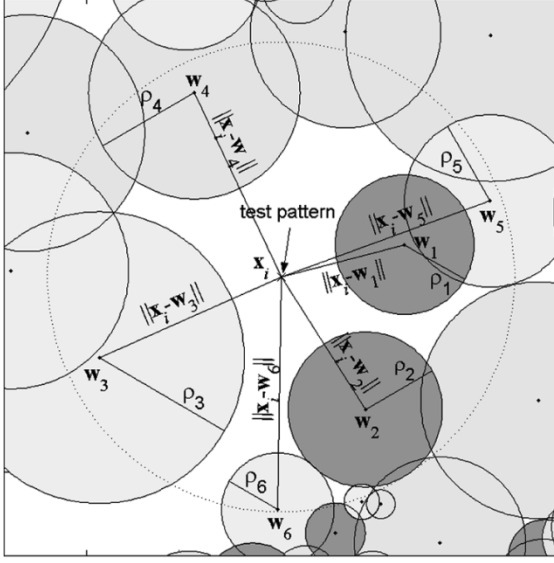
Fig. 5. Test pattern and the nearest clusters with different category label, radius, and distance to the pattern.

not only to be a simple function of categories of nearest clusters, but a complex function composed of their category, size, and distance to the pattern. A new approach can be proposed by introducing the theory of fuzzy sets into the classifier concept to develop its fuzzy version [42]. New classifier is supposed to provide more realistic classification of new patterns.

B. Crisp K-Nearest Neighbor Classifier

Given a set of categorized clusters, crisp or nonfuzzy K-NN classifier determines the category of a new pattern $\mathbf{x}_i$ based only on the categories of the K nearest clusters

$$\mu_c(\mathbf{x}_i) = \frac{1}{K} \sum_{k=1}^{K} \mu_c(\mathbf{w}_k) \qquad (1)$$

where $\mathbf{w}_k$ is prototype of cluster k, $\mu_c(\mathbf{w}_k)$ is membership degree of cluster k belonging to category c, $\mu_c(\mathbf{x}_i)$ is membership degree of pattern $\mathbf{x}_i$ belonging to category c. $i = 1, \ldots, I; c = 1, \ldots, C; k = 1, \ldots, K$; where I, K, and C are the number of patterns, nearest neighbors, and categories, respectively. Given classifier allows $\mu_c(\mathbf{w}_k)$ having only crisp values 0 or 1, depending on whether or not cluster k belongs to category c

$$\mu_c(\mathbf{w}_k) = \begin{cases} 1 & \text{if cluster } k \text{ belongs to category } c \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

If two or more K nearest clusters have the same category, then they add membership degree to the cumulative membership degree of that category. Finally, when contributions of all neighbors are encountered, the most representative category is assigned to the pattern

$$\mathrm{g}(\mathbf{x}_i) = \max_c \mu_c(\mathbf{x}_i) \qquad (3)$$

where $\mathrm{g}(\mathbf{x}_i)$ is category assigned to pattern $\mathbf{x}_i$.

C. The Effect of Weighted Distances

Initial enhancement of K-NN classifier is the introduction of weighted contribution of each of the K neighbors according to their distance to a pattern, giving greater weight to the closer neighbors. The distance $d_k(\mathbf{x}_i)$ is generally selected to be the weighted Euclidean distance between pattern $\mathbf{x}_i$ and prototype $\mathbf{w}_k$

$$d_k(\mathbf{x}_i) = \|\mathbf{x}_i - \mathbf{w}_k\|^{\frac{2}{m-1}} \qquad (4)$$

where the parameter $m \geq 1$ is fuzzyfication variable and determines how heavily the distance is weighted when calculating each neighbors' contribution to the pattern category membership. For choice of $m = 2$, calculated distance is identical to Euclidean distance. Moreover, as m increases toward infinity, the term $\|\mathbf{x}_i - \mathbf{w}_k\|^{2/(m-1)}$ approaches one regardless of the distance, and neighbors are more evenly weighted. However, as m decreases toward one, the closer neighbors are weighted more heavily than those further away. If $m \approx 1$, the algorithm will behave like crisp K-NN classifier for $K = 1$.

D. The Effect of Cluster Size

Next improvement of K-NN classifier is to insert fuzzy membership degree as a measure of a cluster belonging to its own category. There is no meaningful reason why membership value $\mu_c(\mathbf{w}_k)$ must retain only crisp value. We propose an extension of a crisp K-NN by considering size of generated clusters in an original way [36]. Since each cluster belongs exactly to one of the categories, membership value $\mu_c(\mathbf{w}_k)$ may be redefined to reflect the relative size of the actual cluster k

$$\mu_c(\mathbf{w}_k) = \begin{cases} \rho_k & \text{if cluster } k \text{ belongs to category } c \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

where $\mu_c(\mathbf{w}_k)$ is the membership degree of cluster k belonging to category c, and is selected to be proportional to the radius of cluster k. The outcome is that the larger neighbors would contribute more than the smaller ones.

E. Fuzzy K-Nearest Neighbor Classifier

The extensions proposed in (4) and (5) can now be used to define Fuzzy K-NN that generalizes crisp K-NN given in (1). A new pattern has to be classified based on the categories of K nearest clusters, their relative size and weighted distances to the pattern. The Fuzzy K-NN classifier calculates, using superposition, a set of membership values $\mu_c(\mathbf{x}_i)$ of input pattern $\mathbf{x}_i$ belonging to all categories present in the K nearest clusters based on the following formula

$$\mu_c(\mathbf{x}_i) = \frac{\sum_{k=1}^{K} \mu_c(\mathbf{w}_k) \frac{1}{\|\mathbf{x}_i - \mathbf{w}_k\|^{\frac{2}{m-1}}}}{\sum_{k=1}^{K} \rho_k \frac{1}{\|\mathbf{x}_i - \mathbf{w}_k\|^{\frac{2}{m-1}}}} \qquad (6)$$

where $\mu_c(\mathbf{w}_k)$ is given in (5). The pattern is assigned to the category with the highest membership degree according to (3). The denominator of (6) uses ρ_k to allow all K neighbors to
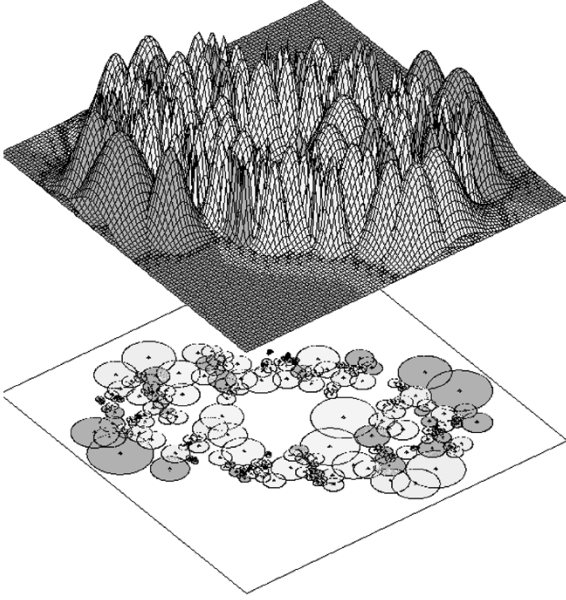
Fig. 6. Fuzzyfication Demo: Fuzzy K-Nearest Neighbor classifier with contribution of cluster size and weighted distance included.
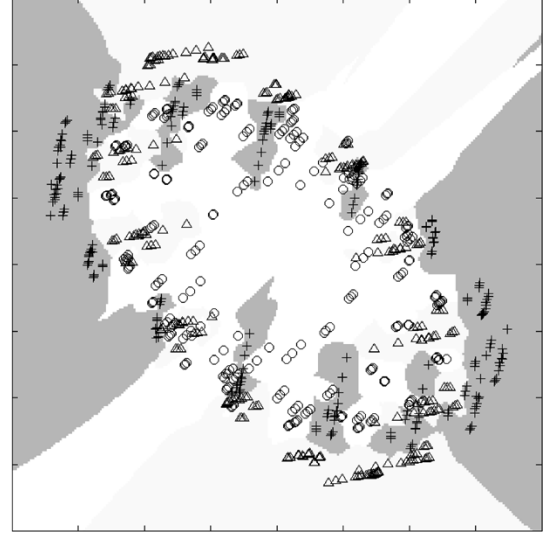

Fig. 7. Fuzzyfication Demo: Category decision regions established by Fuzzy K-Nearest Neighbor classifier.

Fig. 8. Proposed solution for protective relaying based on neural network and fuzzy logic.

normalize total membership value, which must be equal to one. For $K = 1$, both Fuzzy and crisp K-NN become identical.

The idea of fuzzyfication of neural network outputs, by their interpretation through fuzzy decision rule, is an advanced concept that classifies new patterns according to weighted contribution of few similar prototypes. Playing with parameters K and m, and selecting the best pair of values for a particular implementation can attain adaptive behavior of the classifier. Fuzzy classifier with properly tuned parameters may result in very precise category decision boundaries and will improve generalization capabilities accordingly.

F. Fuzzyfication Demo

Fuzzy membership degree, introduced for clusters obtained by the *Training Demo*, is given in Fig. 6. The clusters are shown in their original two-dimensional space, while their impact modeled through membership function is represented in a third dimension. Merged effects of cluster's size and distances to the points in a pattern space are shown for $m > 1$ and $\mu_c(\mathbf{w}_k) = \rho_k$.

Fig. 7 shows the decision regions between the categories obtained by superposition of cluster's membership functions. The solution for $K = 4$ and $m = 1.50$ is presented. As expected, cluster's impact is substantial in their proximity, having bigger clusters broadly involved in creating decision boundaries, while smaller clusters react only in a narrow area nearby.

IV. PROPOSED RELAYING SOLUTION

A functional block diagram of a proposed protective relaying solution, based on combined neural network/fuzzy logic approach, is given in Fig. 8.

Transmission line current and voltage signals at relay location are measured with instrument transformers and converted into lower signal levels. Attenuated signals are filtered with low-pass analog filter to remove noise and higher frequency components, according to the sampling theorem requirement. Signals are sampled with specified sampling frequency, and samples are extracted in a dynamic data window, normalized and scaled. Finally pattern is constructed, and used for either neural network training and testing or implementation.
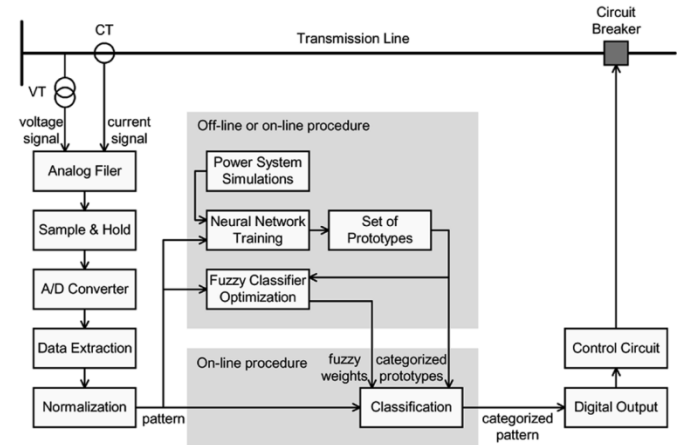
The training of neural network can be performed either using measurement data directly taken from the process or accessing historical record of fault and disturbance cases. Since interesting cases do not happen frequently, sufficient data are provided using simulation of relevant power system scenarios. The main function assigned to the relay is detection of the fault and classification of the fault type. Supplemental function is the recognition of the transmission line faulted section.

Illustration of all the steps of mapping the input space into the decision region categories is shown in Fig. 9. Using unsupervised/supervised learning, the space of training patterns is transferred into an initial abstraction level containing set of clusters with corresponding prototypes, sizes and categories. Moreover, the clusters are fuzzyfied and transformed into an intermediate abstraction level. Final abstraction level is attained when, using
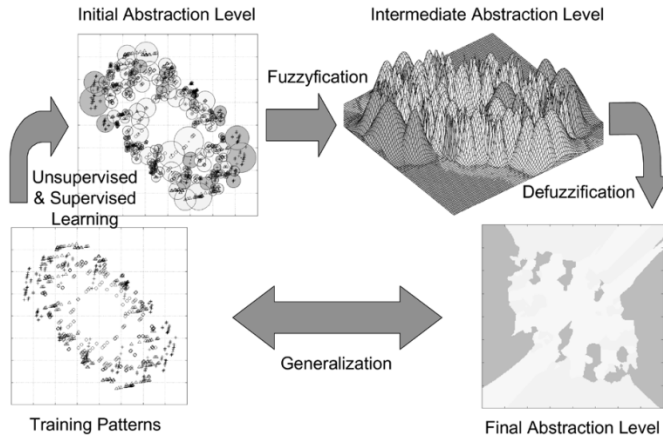
Fig. 9. Mapping of the pattern space into category decision regions using unsupervised and supervised learning, fuzzification and defuzzification.

Fig. 10. CenterPoint Energy STP-SKY power network model.

defuzzification, category decision regions with smooth boundaries are established.

During real-time (on-line) implementation, the trained neural network possesses generalization ability and is expected to successfully classify new patterns that have not been presented during the training process. If a new pattern is classified to the category of unfaulted state then the input data window is shifted for one sample, pattern vector is updated and the comparison is performed again until fault is detected and classified. Depending on the recognized event and selected logic, circuit breaker either trips the faulted phase(s) within proposed time period, or stays unaffected.

V. MODEL IMPLEMENTATION AND SIMULATION RESULTS

A. Power Network Model

A model of an actual 345 kV power system section shown in Fig. 10, with data obtained from CenterPoint Energy, was developed for relay testing and simulation studies [37]. The network model is unique since it has the configuration that can exercise many normal as well as fault conditions that are rather difficult for the traditional relaying to deal with. The reduced network equivalent was obtained by using calculations based on load flow and short circuit data. The model was verified using both steady state and faults using recordings captured during actual events in the system. The STP-SKY transmission line is of particular interest. Transients obtained at the bus SKY were utilized for performing protective algorithm design and evaluation.

B. Simulation of Training and Test Cases

Alternate Transient Program (ATP) is used for detailed modeling of power network and simulation of electromagnetic transients [43]. Setting procedure for the network simulation scenarios and relaying algorithm parameters is implemented in the MATLAB software package, which is interfaced with the network model implemented in ATP [44].

Simulation of transmission faults depends on four main fault parameters. Combining these values, diversity of fault events is created as follows:

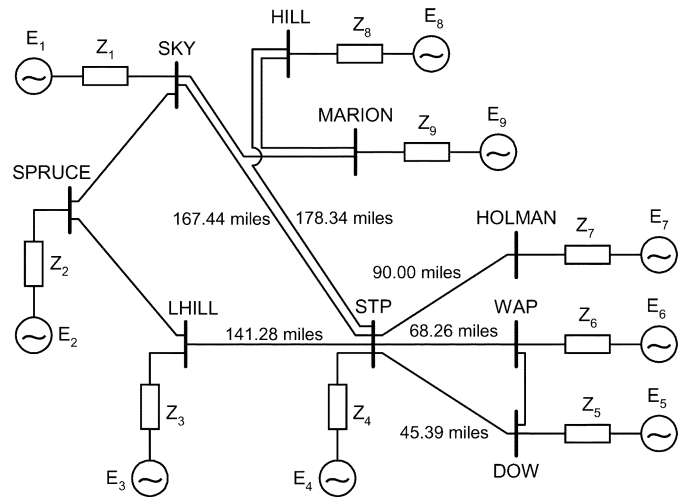- Fault type: all eleven types of fault and no-fault state;

- Fault distance: from 5 up to 95% of the line length, in increments of 10%;
- Fault resistance for ground faults: 3, 11, and 19 Ω;
- Fault angle: from 0 up to 336°, in increments of 24°.

Total number of patterns in the training set, obtained by combining all the selected parameters, is 3315.

Test scenarios are selected to be statistically independent from the training scenarios. Fault parameters used for generation of test scenarios are randomly selected from an uniform distribution of: fault type, fault distance between 0 and 100% of the line length, fault angle between 0 and 360°, and normal distribution of fault resistance, with mean 0 Ω and standard deviation 10 Ω, taking into account only positive values. There are four sets of test cases:

- Nominal system—5000 patterns;
- Weak infeed in the network, due to disconnected sources E1 and E9—5000 patterns;
- Off-nominal voltage source E1 with phase angle shift of −30°—5000 patterns;
- Off-nominal system frequency of 59 Hz—5000 patterns.

Another set of scenarios, statistically independent from test scenarios, is used to tune the parameters of the fuzzy classifier. Total number of patterns in that set is 4000.

C. Pre-Processing of the Training and Test Patterns

Quantities selected for forming the patterns are either three phase currents, or three phase voltages, or both three phase currents and voltages. The antialiasing filter is the second order low-pass Butterworth filters with the cut-off frequency equal a half of the selected sampling frequency. Sampling frequencies of 1.92 kHz (32 samples/cycle) and 3.84 kHz (64 samples/cycle), and dynamic data windows of 8.33 ms (half cycle) and 16.67 ms (one cycle) are implemented. The patterns are normalized by scaling their features to have zero mean, and be in the range $[-1, 1]$. Identical scaling ratio is used later on for normalization of test patterns. Whenever patterns with both current and voltage samples are involved, additional heuristic rescaling of only the current portion of the patterns is performed several times for each individual training case until the best scaling ratio

TABLE I
SUMMARY OF ART^1 AND ART^2 TEST RESULTS FOR
NOMINAL POWER SYSTEM

	Patterns					Training						Testing				
	Data		Extraction			Classif.		Time [h:m]		Clusters		Classifier			Error [%]	
case	currents	voltages	data window [cyc.]	samp. freq. [s./cyc.]	length	fault type	fault type and sect.	ART^1	ART^2	ART^1	ART^2	ART^1 K	ART^2 K m		ART^1	ART^2
1			1.0	32	96			0:09	0:36	131	242	1	1	-	0.02	0.00
2			1.0	32	96			0:06	0:14	284	538	1	3	1.20	7.36	3.80
3			0.5	32	96			0:18	0:41	265	343	1	10	1.45	0.46	0.00
4			0.5	64	192			0:28	1:20	339	354	1	7	1.50	0.38	0.00
5			1.0	32	192			0:21	1:11	202	309	1	5	1.50	1.08	0.00
6			1.0	64	384			0:43	2:25	205	310	1	3	1.55	0.82	0.04
7			1.0	32	96			0:19	0:49	563	877	1	5	1.75	4.30	2.90
8			1.0	32	96			0:12	0:17	579	826	1	1	-	10.30	8.08
9			0.5	32	96			0:27	0:49	644	750	1	4	1.30	2.84	1.78
10			0.5	64	192			0:53	1:15	648	834	1	1	-	2.26	1.16
11			1.0	32	192			0:45	1:51	725	916	1	4	1.40	2.34	1.52
12			1.0	64	384			1:46	3:28	716	934	1	1	-	2.14	1.14

offering lowest predictive error is discovered and subsequently adopted.

D. Simulation Results

Two versions of neural network algorithms, ART^1 and ART^2 are evaluated. ART^1 is a version that has simplified supervised learning and nonfuzzy classifier. ART^2 is with improved supervised learning and fuzzy classifier. Table I provides summary of ART^1 and ART^2 training and test results for nominal power system.

Patterns are generated using different measurement quantities, sampling frequencies and data windows. Classification tasks are aimed at recognition of fault type or both fault type and section. Boundary distance between the first and second protection zone is set to be 80% of the line length. The number of pattern prototypes, classifier parameters and classification error for nominal system are given for each training case. Moreover, the time required for training of each neural network configuration on a Pentium 4 PC with CPU 2.4 GHz and RAM 512 Mb is shown as well.

First observation is related to the classification error of ART^2 for nominal system with respect to varying data window and sampling frequency. ART^2 provides very good classification of fault type since errors are zero or almost zero in all cases. For fault type and section classification, error is obviously lower with longer data window and especially with higher sampling frequency. Comparing to ART^1, ART^2 required longer training time and produces more prototypes due to stronger condition for supervised learning, which generally accepts lower number of clusters per single iteration. For fault type and section classification, the number of prototypes is few times higher than for fault type classification case because proper classification of faults around the boundary between the transmission line sections requires many small, fine-grained clusters.

The comparison of the classification error of ART^1 and ART^2 for different type of measurements is given in Fig. 11.

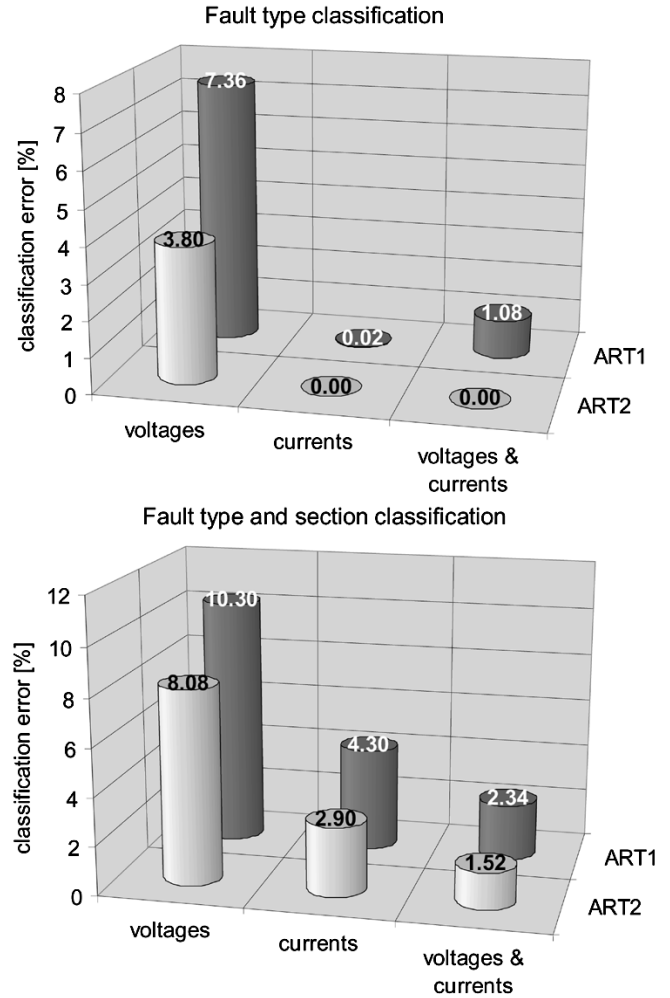Fig. 11. Classification error versus pattern type, algorithm and classification task.

Results show the advantage of ART^2 comparing to ART^1. Generally, patterns based only on the voltage samples are not sufficient for successful classification. ART^2 algorithm with patterns based on combination of currents and voltages guarantees very low classification error for both classification tasks.

Fig. 12 shows comparative evaluation of ART^1 and ART^2 for faults during nominal operating conditions, weak infeed, voltage source variation and system frequency variation. ART^2 classifies fault type almost perfectly for all nominal and off-nominal network conditions, and comparing to ART^1, extraordinary improvements have been observed. Repeatedly, ART^2 is in some degree better than ART^1 for all network conditions, reducing the classification error to 16–35% interval.

A summary of the results can be provided through several remarks. Classification of fault type and section has been far more difficult task than classifying only the fault type. ART^2 has better recognition capabilities than ART^1 for all anticipated scenarios. Therefore, ART^2 with the fuzzy classifier and improved supervised learning provides remarkable recognition capabilities, and is very robust for off-nominal network operating conditions. The classification results are satisfactory for all values of sampling frequency and data window, which correspond to typical values used in protective relaying.
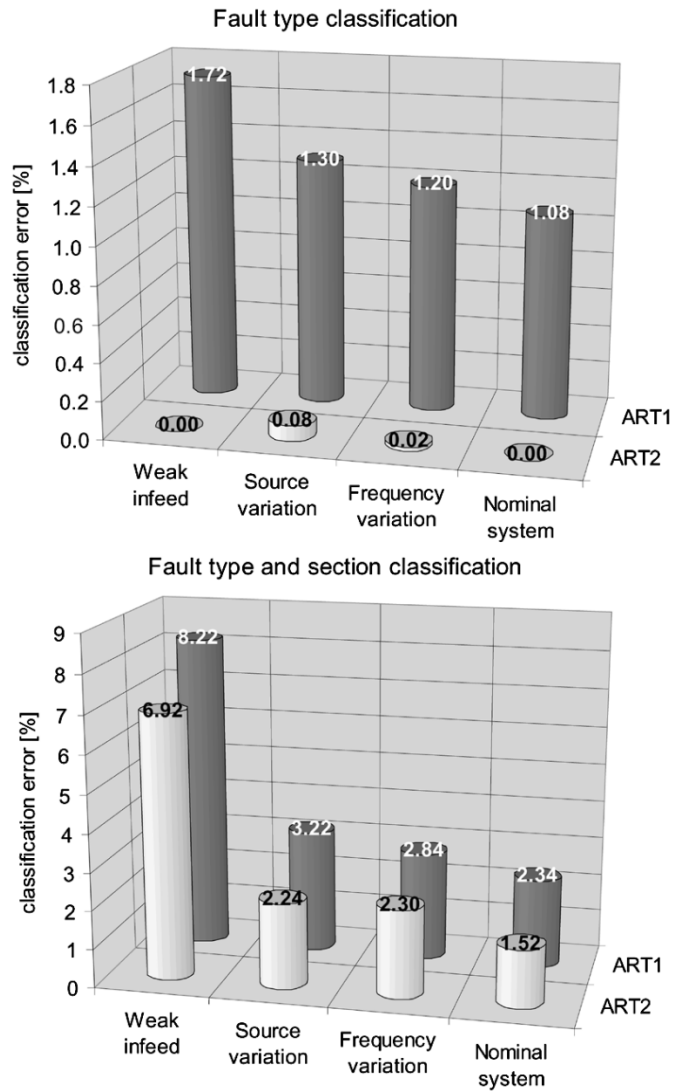
Fault type classification

Fault type and section classification

Fig. 12. Classification error versus algorithm for faults in the nominal system and system with frequency deviation.

VI. CONCLUSION

This paper has introduced an advanced transmission line protection algorithm with several important improvements when compared to the previous solution:

- Optimized input signal preprocessing steps enable better selection of the most representative set of input signals
- Redefined supervised learning allows improved cluster generation;
- Fuzzy approach provides an interpolation technique for interpreting neural network outputs and is robust for untypical and noisy inputs;
- Extensive sets of training and test patterns have been utilized, by simulating appropriate power network model and a variety of fault and disturbance scenarios;
- Combined use of neural and fuzzy techniques in the same algorithm leads to complex reasoning that improves the event classification ability;
- Simulation results show superior behavior of the novel ART^2 algorithm when compared to the original ART^1 algorithm.

REFERENCES

[1] W. A. Lewis and S. Tippet, "Fundamental basis for distance relaying on 3-phase systems," in *AIEE Trans.*, vol. 66, Feb. 1947, pp. 694–708.
[2] *IEEE Guide for Protective Relay Applications to Transmission Lines*, Feb. 2000.
[3] T. Dalstein and B. Kulicke, "Neural network approach to fault classification for high speed protective relaying," *IEEE Trans. Power Del.*, vol. 10, pp. 1002–1011, Apr. 1995.
[4] W. W. L. Keerthipala, C. T. Wai, and W. Huisheng, "Neural network based classifier for power system protection," *Elect. Power Syst. Res.*, vol. 42, no. 2, pp. 109–114, Aug. 1997.
[5] Y. H. Song, Q. X. Xuan, and A. T. Johns, "Comparison studies of five neural network based fault classifiers for complex transmission lines," *Elect. Power Syst. Res.*, vol. 43, no. 2, pp. 125–132, Nov. 1997.
[6] W.-M. Lin, C.-D. Yang, J.-H. Lin, and M.-T. Tsay, "A fault classification method by RBF neural network with OLS learning procedure," *IEEE Trans. Power Del.*, vol. 16, pp. 473–477, Oct. 2001.
[7] A. Poeltl and K. Frohlich, "Two new methods for very fast fault type detection by means of parameter fitting and artificial neural networks," *IEEE Trans. Power Delivery*, vol. 14, pp. 1269–1275, Oct. 1999.
[8] H. Wang and W. W. L. Keerthipala, "Fuzzy-neuro approach to fault classification for transmission line protection," *IEEE Trans. Power Del.*, vol. 13, pp. 1093–1104, Oct. 1998.
[9] A. K. Pradhan, P. K. Dash, and G. Panda, "A fast and accurate distance relaying scheme using an efficient radial basis function neural network," *Elect. Power Syst. Res.*, vol. 60, no. 1, pp. 1–8, Nov. 2001.
[10] A. L. O. Fernandez and N. K. I. Ghonaim, "A novel approach using a FIRANN for fault detection and direction estimation for high-voltage transmission lines," *IEEE Trans. Power Del.*, vol. 17, pp. 894–900, Oct. 2002.
[11] T. S. Sidhu, H. Singh, and M. S. Sachdev, "Design, implementation and testing of an artificial neural network based fault direction discriminator for protecting transmission lines," *IEEE Trans. Power Del.*, vol. 10, pp. 697–706, Apr. 1995.
[12] M. Sanaye-Pasand and O. P. Malik, "Implementation and laboratory test results of an Elman network-based transmission line directional relay," *IEEE Trans. Power Del.*, vol. 14, pp. 782–788, Jul. 1999.
[13] T. Dalstein, T. Friedrich, B. Kulicke, and D. Sobajic, "Multi neural network based fault area estimation for high speed protective relaying," *IEEE Trans. Power Del.*, vol. 11, pp. 740–747, Apr. 1996.
[14] P. K. Dash, A. K. Pradhan, and G. Panda, "Application to minimal radial basis function neural network to distance protection," *IEEE Trans. Power Del.*, vol. 16, pp. 68–74, Jan. 2001.
[15] A. G. Jongepier and L. van der Sluis, "Adaptive distance protection of double-circuit lines using artificial neural networks," *IEEE Trans. Power Del.*, vol. 12, pp. 97–105, Jan. 1997.
[16] R. K. Aggarwal, Q. Y. Xuan, R. W. Dunn, A. T. Johns, and A. Bennett, "A novel fault classification technique for double-circuit lines based on a combined unsupervised/supervised neural network," *IEEE Trans. Power Del.*, vol. 14, pp. 1250–1256, Oct. 1999.
[17] Z. Q. Bo, R. K. Aggarwal, A. T. Johns, H. Y. Li, and Y. H. Song, "A new approach to phase selection using fault generated high frequency noise and neural networks," *IEEE Trans. Power Del.*, vol. 12, pp. 106–115, Jan. 1997.
[18] D. S. Fitton, R. W. Dunn, R. K. Aggarwal, A. T. Johns, and A. Bennett, "Design and implementation of an adaptive single pole autoreclosure technique for transmission lines using artificial neural networks," *IEEE Trans. Power Del.*, vol. 11, pp. 748–756, Apr. 1996.
[19] S. Websper, R. W. Dunn, R. K. Aggarwal, A. T. Johns, and A. Bennett, "Feature extraction methods for neural network-based transmission line fault discrimination," in *Proc. Inst. Elect. Eng., Gen., Transm. Distrib.*, vol. 146, May 1999, pp. 209–216.
[20] K. L. Butler, J. A. Momoh, and D. J. Sobajic, "Fault studies using a neural-net-based approach for fault diagnosis in distribution networks," in *Proc. Inst. Elect. Eng., Gen. Transm. Distrib.*, vol. 144, Sept. 1997, pp. 429–436.
[21] B. Widrow and M. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1442, Sept. 1990.
[22] M. J. D. Powell, "Radial basis functions for multivariable interpolation: A review," in *Algorithms for the Approximations of Functions and Data*, J. C. Mason and M. G. Cox, Eds, Oxford, U.K.: Clarendon Press, 1987, pp. 143–167.
[23] T. Kohonen, *Self-Organizing Maps*. New York: Springer, 2001, p. 501.
[24] 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.

[25] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 4, pp. 179–211, Apr./Jun. 1990.

[26] R. Hecht-Neilsen, "Counterpropagation networks," *Appl. Opt.*, vol. 26, no. 23, pp. 4979–4984, Dec. 1987.

[27] E. A. Wan, "Temporal backpropagation for FIR neural networks," in *Proc. IEEE Intl. Joint Conf. Neural Networks*, vol. 1, San Diego, CA, Jun. 1990, pp. 575–580.

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

[29] Working Group D5 of the Line Protection Subcommittee of the IEEE Power System Relaying Committee, "Proposed statistical performance measures for microprocessor-based transmission line protective relays, Part I: Explanation of the statistics, Part II: Collection and uses of data," *IEEE Trans. Power Delivery*, vol. 12, pp. 134–156, Jan. 1997.

[30] Y. H. Pao and D. J. Sobajic, "Combined use of unsupervised and supervised learning for dynamic security assessment," *IEEE Trans. Power Syst.*, vol. 7, pp. 878–884, May 1992.

[31] M. Kezunovic, M. I. Rikalo, and D. Sobajic, "High-speed fault detection and classification with neural nets," *Elect. Power Syst. Res.*, vol. 34, pp. 109–116, Aug. 1995.

[32] M. Kezunovic and I. Rikalo, "Detect and classify faults using neural nets," *IEEE Comput. Applicat. Power*, vol. 9, pp. 42–47, Oct. 1996.

[33] M. Kezunovic, I. Rikalo, and D. Sobajic, "Real-time and off-line transmission line fault classification using neural networks," *Eng. Intell. Syst.*, vol. 4, no. 1, pp. 57–63, Mar. 1996.

[34] S. Vasilic, M. Kezunovic, and D. Sobajic, "Optimizing the performance of transmission line relaying algorithm implemented using an adaptive, self-organized neural network," in *Proc. 14th Power Syst. Computation Conf.*, vol. ses. 42, Seville, Spain, Jun. 2002, pp. 1–6.

[35] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[36] S. Vasilic and M. Kezunovic, "An improved neural network algorithm for classifying the transmission line faults," in *Proc. IEEE Power Eng. Soc. Power Winter Meeting*, vol. 2, New York, Jan. 2002, pp. 918–923.

[37] D. Ristanovic, S. Vasilic, and M. Kezunovic, "Design and implementation of scenarios for evaluating and testing distance relays," in *Proc. 33rd North American Power Symp.*, College Station, TX, Oct. 2001, pp. 470–475.

[38] S. Vasilic and M. Kezunovic, "New design of a neural network algorithm for detecting and classifying transmission line faults," in *Proc. IEEE Power Eng. Soc. Transm. Distrib. Conf.*, Atlanta, GA, Oct. 2001. presented.

[39] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–137, Mar. 1982.

[40] G. H. Ball and D. J. Hall, "A clustering technique for summarizing multivariate data," *Behav. Sci.*, vol. 12, pp. 153–155, Mar. 1967.

[41] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 21–27, Jan. 1967.

[42] J. Keller, M. R. Gary, and J. A. Givens, "A fuzzy K-nearest neighbor algorithm," *IEEE Trans. Syst., Man Cybern.*, vol. 15, pp. 580–585, Jul./Aug. 1985.

[43] *Alternative Transient Program (ATP) Rule Book*, Sep. 1992.

[44] *Using MATLAB*, Aug. 2002.

Slavko Vasilic (S'00) received the B.S. and M.S. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1993 and 1999, respectively. He is currently pursuing the Ph.D. degree in electrical engineering from Texas A&M University, College Station, working on artificial intelligence application for power system analysis, control, and protection.

His research interests include signal processing, modern control systems, pattern recognition, numerical modeling and optimization, parameter and state estimation, system identification, machine learning, neural networks, fuzzy logic, and statistical analysis.

Mladen Kezunovic (S'77–M'80–SM'85–F'99) received the Dipl. Ing. degree in electrical engineering from the University of Sarajevo in 1974, and the M.S. and Ph.D. degrees in electrical engineering from the University of Kansas, Lawrence, in 1977 and 1980, respectively.

He was with Texas A&M University, College Station, since 1987 where he is now the Eugene E. Webb Professor and Director of Electric Power and Power Electronics Institute. His main research interests are digital simulators and simulation methods for equipment evaluation and testing as well as application of intelligent methods to control, and protection and power quality monitoring.

Dr. Kezunovic is a registered professional engineer in Texas, a member of CIGRE, and a Fellow of the IEEE.