An Improved Neural Network Algorithm for Classifying the Transmission Line Faults

S. Vasilic, *Student Member, IEEE,* M. Kezunovic, *Fellow, IEEE*

***Abstract*--This study introduces a new concept of artificial intelligence based algorithm for classifying the faults in power system networks. This classification identifies the exact type and zone of the fault. The algorithm is based on unique type of neural network specially developed to deal with large set of highly dimensional input data. An improvement of the algorithm is proposed by implementing various steps of input signal preprocessing, through the selection of parameters for analog filtering, and values for the data window and sampling frequency. In addition, an advanced technique for classification of the test patterns is discussed and the main advantages comparing to previously used nearest neighbor classifier are shown.**

***Index Terms*--clustering methods, electromagnetic transients, neural networks, pattern classification, power system faults, protective relaying, testing, training.**

I. Introduction

THIS paper deals with neural network based technique for protective relaying in power system networks. The problem of detecting and classifying the transmission line faults has been known for a long time. Traditional relaying principles are based on predetermined setting and take into account only common and easily anticipated fault 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, voltage levels, and varsity of other events, determine the corresponding transient current and voltage waveforms detected by the relays at line ends. All these effects influence the relays tuned only to perform well during anticipated fault conditions. The new classification approach has to reliably conclude, in a very short time (around one cycle), whether, where and which type of fault occurs under a variety of time-varying operating conditions [1].

Neural networks can be used to solve power system protection problems, particularly those where traditional approaches have difficulty in achieving the desired speed, accuracy and selectivity. Neural networks are convenient for these tasks because they use individual examples to capture general, always complex and nonlinear, relationships among the data. They learn from the environment and adapt their recognition capabilities.

Various applications of neural networks were used in the past to improve recognition of the impedance used in distance relaying of transmission lines [2]. These applications are mainly based on widely used multilayer feed-forward networks. Whenever input patterns with large dimensionality are present (as in this particular case), training of these networks is very slow, needs much larger training sets, and very easily converges on local minima, usually very far from the global minimum. On-line training of this type of network for any new training data requires presenting the entire set of training patterns again.

Instead of using multilayer neural network, a unique type of neural network may be used for fault classification [3]-[6]. The network is applied directly to the samples of voltages and currents, and produce the fault type and zone classification in real time. This network is based on ISODATA clustering algorithm [7] and belongs to a group of special neural networks named Self-Organizing Maps [8]. The adaptive behavior of the neural network is described by Adaptive Resonance Theory [9]. The main aim of this work is to provide some insights into possible approaches to significantly improve existing neural network algorithm, through appropriate conditioning of input patterns as well as optimizing the classification of new patterns that have not been presented in the training process.

Electromagnetic transient program ATP [10] is used for simulating a specially developed power network model. Simulation outputs of a large number of scenarios, including various fault and normal operating states, are used as a signal generator for the neural network algorithm design, implemented in MATLAB [11]-[13].

The paper is organized as follows. Section II presents a brief description of the neural network classification algorithm. Section III, divided into subsections, demonstrates generation and preprocessing of neural network input signals, and defines classification task. An advanced classification technique is introduced in section IV. The conclusion is given at the end.

II. Neural Network Classification Algorithm

Neural networks try to produce a concise representation of system's behavior through identifying natural groupings of data

This study was supported by an Army/EPRI contract # WO 8333-05 between EPRI and Carnegie Mellon University, and has been carried out by Texas A&M University under the subcontract # 542995-42590 titled "Self-Evolving Agents for Monitoring, Control and Protection of Large, Complex Dynamic Systems".

S. Vasilic and M. Kezunovic are with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128 USA (e-mails respectively: svasilic@ee.tamu.edu, kezunov@ee.tamu.edu).

from large data sets. The aim of this procedure, called clustering, is to partition a given set of input data (patterns) into several groups or clusters, so that each pattern is assigned to a unique cluster. Patterns that belong to the same cluster should be as similar as possible, while patterns that belong to different clusters should be as different as possible. Class label is assigned to each cluster, where class symbolizes a group of patterns with a common characteristic.

Self-organizing maps are special type of neural networks that map input patterns with similar features into adjacent clusters after enough input patterns have been presented [8]. The similarity between patterns is usually measured by calculating the Euclidean distance between two *n*-dimensional input vectors. After training, self-organized clusters represent prototypes of classes of input patterns. ISODATA clustering algorithm discovers the most representative positions of prototypes in the pattern space [7]. Adaptive Resonance Theory is characterized by its ability to form clusters incrementally, whenever a pattern, sufficiently different from all previously presented patterns, appears [9]. Incremental clustering capability can handle an infinite set of input data, because their cluster prototype units contain implicit representation of all the inputs previously encountered. Using this technique, the on-line training due to non-stationary inputs may be easily implemented.

This neural network is without hidden layers and its self-organized structure depends only on the presented input data set. The neural network training consists of unsupervised and supervised learning phases (Fig. 1) [4].

During *unsupervised learning*, patterns are presented without their class labels. This procedure tries to identify characteristic patterns or prototypes that can serve as cluster centers. The outcome of unsupervised learning is a stable family of clusters, defined as hyperspheres in an *n*-dimensional space, where *n* denotes the number of features, i.e. the length of input vector. Unsupervised learning forms stable family of both "clean" (homogenous, having patterns with the same class label) and "mixed" (non-homogenous, having patterns with two or more class labels) clusters. It does not require either the initial guess of the number of cluster, or the initial cluster center coordinates. It consists of two steps: initialization and stabilization. The *Initialization* establishes initial cluster structure based on similarity between the patterns, and by presenting each pattern only ones. During the *stabilization* all patterns are presented again until a stable cluster structure occurs and there are no patterns changing their cluster membership during the iterations. Stabilization phase is repeated many times until no pattern changes its cluster membership.

In the *supervised learning* the class label is associated with each data point. *Supervised learning* separates "mixed" clusters from the "clean" ones. It assigns class labels to the "clean" clusters. The tuning parameter ρ, called vigilance parameter, controls the number and size of generated clusters and is being consecutively decreased during iterations.

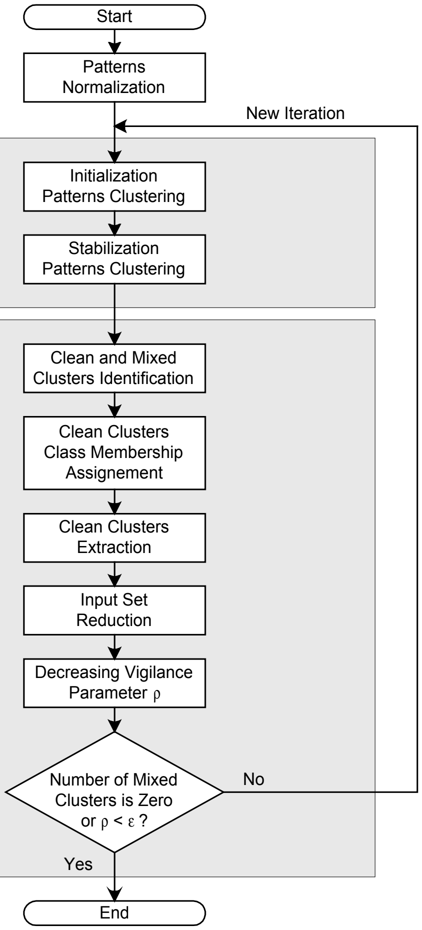
Start
Patterns Normalization
New Iteration
UNSUPERVISED LEARNING
Initialization Patterns Clustering
Stabilization Patterns Clustering
SUPERVISED LEARNING
Clean and Mixed Clusters Identification
Clean Clusters Class Membership Assignement
Clean Clusters Extraction
Input Set Reduction
Decreasing Vigilance Parameter ρ
Number of Mixed Clusters is Zero or ρ < ε ?
No
Yes
End

Fig. 1. Neural Network Clustering Algorithm

Unsupervised and supervised learning procedures are repeated unless only "clean" clusters exist, or current value of vigilance parameter is less then specified value.

During the testing procedure, distances between each test pattern and established clusters are calculated, and the nearest neighbors classifier [14] is efficiently implemented since the number of optimized prototypes is significantly smaller then the number of training patterns. The outcome of the testing are class labels assigned to test patterns according to the most common value among the *K* nearest prototypes.

III. Algorithm Implementation for Classifying Transmission Line Faults

A. Generation of Input Signals

Power system used for algorithm evaluation and testing is 345 kV power system section, provided by Reliant Energy (RE) HL&P company. The model of the given power network is implemented in Alternative Transient Program (ATP) program [10]. It is used for simulating various fault scenarios on one of the transmission lines, by varying fault parameters. 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. Neural network based algorithm takes voltage and current measurements from one end of the line. It has to be trained to recognize the type and zone of the fault. The classification is based only on direct use of samples without impedance computing. The example of the simulation output data, including three-phase line currents and voltages, for one specific case, phases A to B to ground fault, is shown in Fig. 2. Current and voltage samples obtained through simulations are used for forming training and test patterns for protective algorithm learning and evaluation.

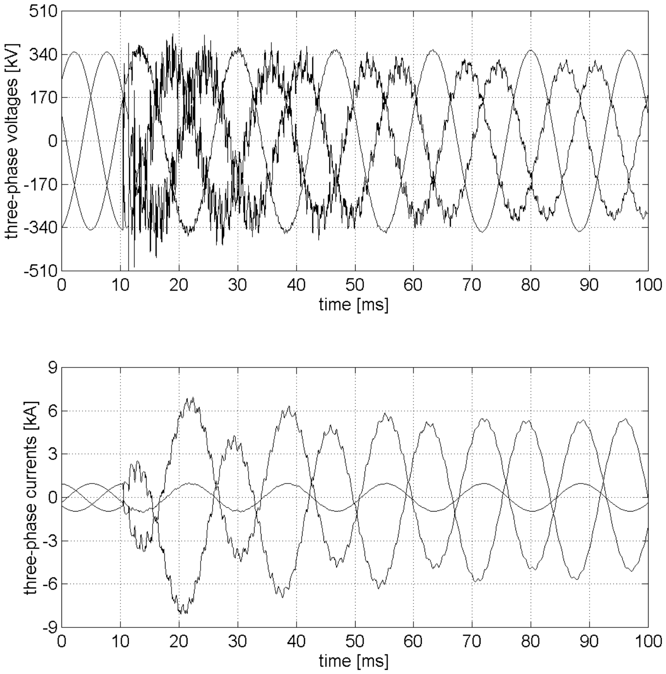

Fig. 2. Typical voltage and current measurements for ABG fault.

Training patterns are generated by specifying several values for each fault parameter, and combining these values to cover possibility space of fault cases [14]. Prototypes formed during training represent the most typical patterns obtained through simulations. Test patterns for algorithm evaluation in heuristic, previously unseen, situations are generated by random setting of all fault parameters. The test patterns might be very heterogeneous and quite different from the training patterns since there are many operating states and possible events in the power network. They are classified according to their similarity to prototypes adopted during training.

B. Preprocessing of Input Signals

Neural network training can be made more efficient if certain preprocessing steps are performed on the network inputs. Pattern extraction from obtained measurements depends on several algorithm tuning parameters. Those parameters significantly determine quality of algorithm training.

Input into the neural network is in the form of the moving data window containing samples of phase currents and voltages (Fig. 3). Selection of sampled data for training in desired data window may include: three phase currents, three phase voltages, or both the three phase currents and voltages. Comparison of algorithm performances in all three cases is a challenging task.

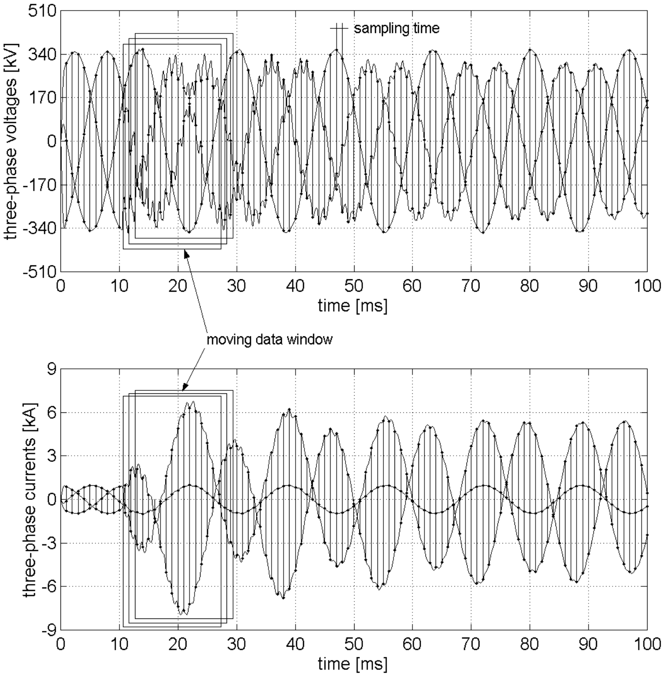

Fig. 3. Voltage and current samples and moving data window used for forming the patterns.

Examples of extracted patterns for different values of algorithm parameters are shown in Figs. 4 to 6 for the choice of three-phase currents. Phase current measurements are filtered by an analog filter and sampled with desired sampling frequency. Patterns are extracted from these measurements during the desired data window (after the fault occurs), normalized, and placed together in one row to form feature vector components.

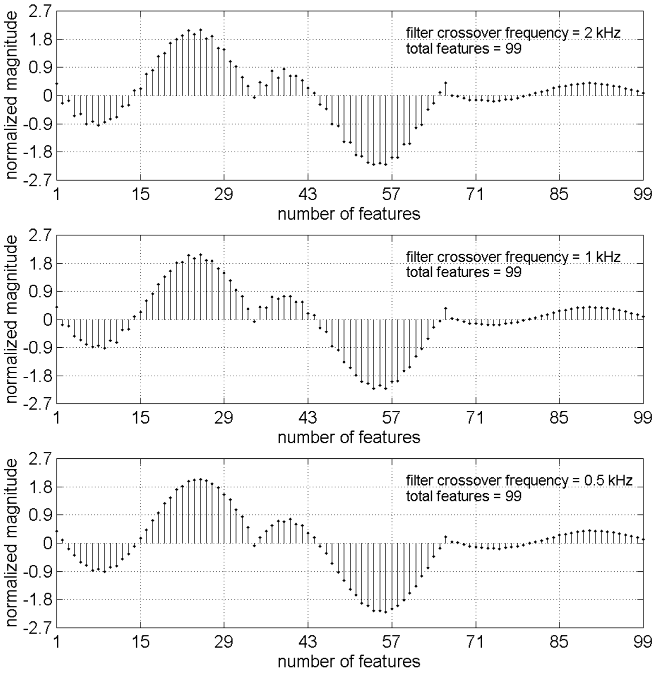

Fig. 4. Example of pattern feature vectors for different values of II order Butterworth analog filter crossover frequencies.

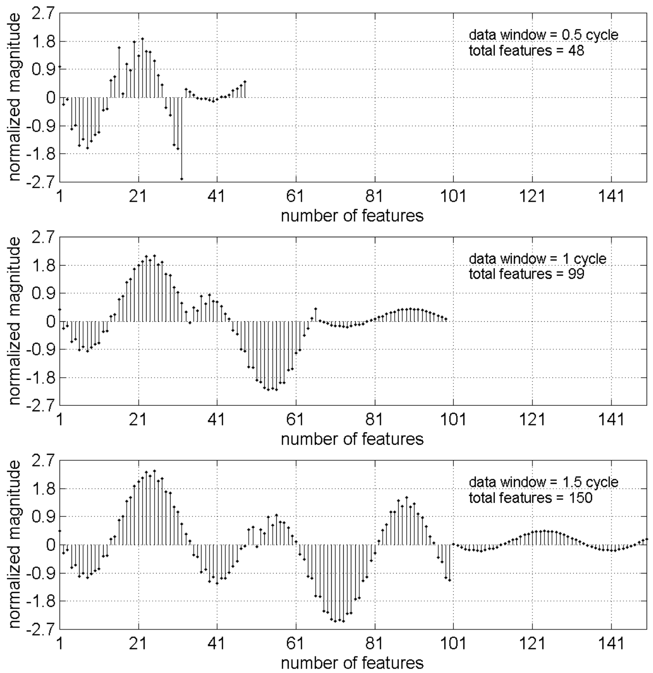

Fig. 5. Example of pattern feature vectors for different lengths of data window.

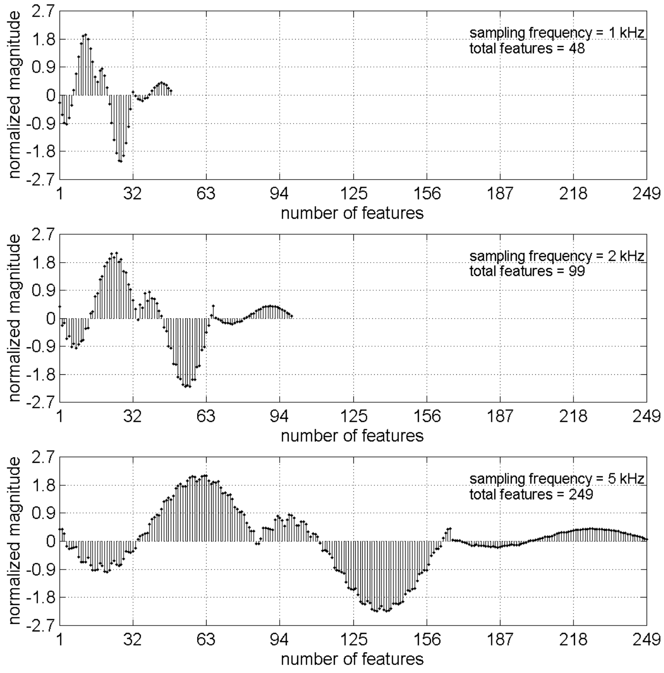

Fig. 6. Example of pattern feature vectors for different values of sampling frequency.

The applied analog filter is the second order Butterworth filter, and the effect of various selected crossover frequencies, 0.5 kHz, 1 kHz, and 2 kHz, is shown in Fig. 4. The filter crossover frequency has to be carefully adopted, such that the noise and other high frequency components are filtered, while the characteristic signal waveform is retained.

The effect of the selected data window is illustrated in Fig. 5. for the window lengths of 0.5, 1 and 1.5 cycles (1 cycle = 16.67 ms). 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 deficiencies in classifying the fault in a real time.

Sampling frequency has similar effect on forming the patterns as the data window does. Pattern feature vectors for sampling frequencies 1 kHz, 2 kHz, and 5 kHz are shown in Fig. 6. Increased sampling frequency offers improved signal detection but also may cause significant computational burden.

In each particular application, extensive set of simulations have to be performed to optimize all mentioned parameters for algorithm training. When this condition is satisfied, algorithm training may be started.

C. Classification Task

Type of classification might be based on detecting the fault type (Normal, AG, BG, CG, AB/ABG, BC/BCG, CA/CAG, ABC/ABCG), fault zone (Normal, Zone I, Zone II), fault resistance (Normal, Low, High), or any combination among them. Classification of testing patterns is performed by using the cluster structure established during training and applying the K-nearest neighbor rule. Clusters are formed as spheres in the *n*-dimensional space. Simplified example in two dimensions is given in Fig. 7. The cluster centers are identified as prototypes or characteristic patterns. Input parameter for algorithm testing is only the number *K* of the nearest neighbors for the rule. Test patterns are extracted and normalized from generated patterns using the same procedure as for training patterns and have equal number of features. For each test pattern, Euclidean distances to all clusters retrieved from reference set are computed and sorted in an increasing order.

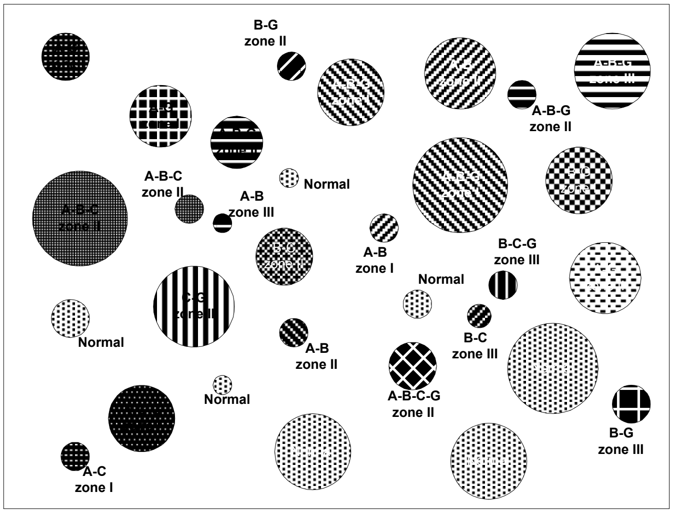

Fig. 7. Example of cluster structure established during training.

The most frequent class label of K nearest clusters is computed and assigned to actual pattern. If the input pattern belongs to any of the “normal-state” clusters then the input data window is "moved" for one sample and the comparison is performed again. If the input pattern does not belong to the “normal-state” clusters then the fault is detected and execution of the fault classification logic is initiated. The parameter used to force the neural network to make the final decision is the time. After decision time has expired, if pattern still does not come back to the normal state, the neural network will classify the fault event according to the fault type detected in that instance.

IV. Advanced K-Nearest Neighbors Technique

Standard version of the K-nearest neighbors rule may be improved to achieve better classification of the test set of patterns that corresponds to a new set of simulated events, previously unseen during training. Between established clusters there is an unlabeled space making the pattern recognition more difficult, because in real situation many new upcoming patterns appear in that space.

Given a set of classified clusters, the standard K-nearest neighbors rule [13] determines the classification of the input pattern x_i based only on the class labels of the *K* closest clusters in the cluster structure established during training

$$\mu_j(x_i) = f[K, \mu_j(v_l)] \tag{1}$$

where:

- $\mu_j(v_l)$ is membership value which determines the degree of belonging of cluster *l* to class *j*;
- $\mu_j(x_i)$ is membership value of pattern *i* belonging to class *j*;
- $i = 1,\ldots,P$; where *P* is number of patterns;
- $j = 1,\ldots C$; where *C* is number of classes;
- $l = 1,\ldots K$; where *K* is number of neighbors;
- $v_1, v_2, \ldots, v_K$ denotes the centers of *K* nearest neighbors of pattern x_i.

$\mu_j(v_l)$ has only crisp values 0 or 1, depending on whether or not a cluster v_l belongs to class *j*:

$$\mu_j(v_l) = \begin{cases} 1 & \text{if cluster } l \text{ belongs to class } j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In this rule all *K* nearest clusters have the equal importance, without taking into account their radii, and distances to the pattern that has to be classified.

The advanced K-nearest neighbors technique is a fuzzy classification technique that generalizes the K-nearest neighbors rule. New patterns are classified based on the weighted distances (d_l) to *K* nearest clusters, as well as on relative size (r_l) and class labels (c_l) of these clusters (Fig. 8). The advanced K-nearest neighbors technique calculates a vector of membership values $(\mu_1(x), \mu_2(x), \ldots \mu_C(x))$ of input pattern x_i in the existing classes. The class membership values are calculated based on the following formula:

$$\mu_j(x_i) = f[K, \mu_j(v_l), d_l(x_i)] \tag{3}$$

where now $\mu_j(v_l)$ may take any value between 0 and 1, representing the relative size of the actual cluster *l*. Each cluster belongs to one of the existing classes, with membership value defined by the following adopted relation:

$$\mu_j(v_l) = \begin{cases} r_l / r_{\max} & \text{if cluster } l \text{ belongs to class } j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The membership degree of cluster v_l belonging to class *j* is equal to the ratio between radius (r_l) of actual cluster *l* and radius $(r_{\max})$ of the largest cluster in the cluster structure. The outcome is that the larger clusters have more influences then the smaller ones, and the clusters with longest radius have $\mu_j(v_l) = 1$.

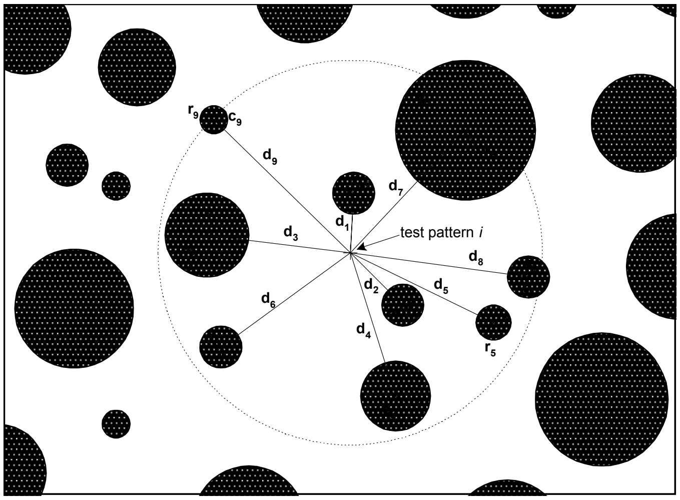

Fig. 8. *K* nearest clusters (*K*=9 in this example) to test pattern *i*, with their class labels c_j, radii r_j, and distances d_j.

Another improvement toward realistic classification is taking into account distances between pattern x_i and *K* nearest clusters. The distance $d_l(x_i)$ may be generally selected to be a weighted Euclidean distance between pattern x_i and cluster *l*

$$d_l(x_i) = \|x_i - v_l\|^m \tag{5}$$

where the parameter *m* determines how heavily the distance is weighted when calculating the class membership.

If two or more of *K* nearest clusters have the same class label, then they give cumulative membership value to that class. When values $\mu_j(x_i)$ for all *K* neighbors have been calculated, pattern x_i is classified to the class with the highest membership degree

$$\text{Class}(x_i) = \{ j \mid \mu_j(x_i) \geq \mu_m(x_i), \quad j, m = 1, \ldots, C \quad m \neq j \tag{6}$$

This idea of new fuzzyfied classification technique algorithm will help classify better a variety of test patterns, comparing to the previously used K-nearest neighbors algorithm. The optimal values, for number of neighbors *K* and parameter *m* which establishes weighted distances, have to be determined and applied in each particular implementation.

V. Conclusion

This study introduces possible directions to improve existing neural network algorithm for classifying the transmission line faults. This algorithm was used earlier in the efforts aimed at replacing distance relays with new relays not having traditional setting and based on pattern recognition capabilities. The algorithm is a specific type of clustering algorithm. It translates input patterns into pattern prototypes, the structure of clusters that represents various classes of input data sets. The algorithm is very flexible and easily enables further modifications and upgrading. Conditioning of input signals, as well as selection of values for analog filtering, data window for taking the patterns, and sampling frequency, play significant role in the algorithm behavior during training and performance during testing. Different aspects of these factors are illustrated through several examples. Furthermore, classification of the test patterns is analyzed through comparison of a simple K-nearest neighbors classifier used so far, and the new fuzzyified approach of that classifier. The advanced approach offers more realistic classification of the test patterns. This is done by taking into account weighted distances between a pattern and the K nearest clusters, as well as the relative size of those clusters.

Proposed tunings and improvements of the neural network algorithm may enable better classification of the fault type and zone. Combines use of neural and fuzzy techniques in the same algorithm leads to complex reasoning that improves the event classification ability for recognizing a variety of the events that could happen in power networks.

VI. References

[1] Power System Relaying Committee, Working Group D5 of the Line Protection Subcommittee, “Proposed statistical performance measures for microprocessor-based transmission line protective relays, Part I and II”, *IEEE Trans. Power Delivery*, vol. 12, no. 1, pp. 134-156, Jan. 1997.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] G. H. Ball and D. J. Hall, "A Clustering Technique for Summarizing Multivariate Data", *Behavioral Science*, vol. 12, pp. 345-370, 1967.

[8] T. Kohonen, *Self-Organizing Maps*, 1st ed., New York: Springer-Verlag, 1997, p. 426.

[9] 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.

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

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

[12] M. Kezunovic and S. Vasilic, "Advanced Software Environment for Evaluating Protection Performance During Power System Disturbances Using Relay Models", CIGRE SC 34 Colloquium, Sibiu, Romania, Sep. 2001.

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

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

VII. Biographies

Slavko Vasilic (S'00) received his B.S. and M.S. degrees in electrical engineering from University of Belgrade in 1993 and 1999, respectively, and currently is a Ph.D. candidate in electrical engineering at Texas A&M University. His research interests are neural networks, fuzzy logic, genetic algorithms, multivariable, robust and adaptive systems, and their implementation in process control and pattern recognition, especially in power systems control, protection and monitoring.

Mladen Kezunovic (S'77, M'80, SM'85, F'99) received his Dipl. Ing. degree from the University of Sarajevo, the M.S. and Ph.D. degrees from the University of Kansas, all in electrical engineering, in 1974, 1977 and 1980, respectively. He has been with Texas A&M University since 1987 where he is 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, protection and power quality monitoring. Dr. Kezunovic is a registered professional engineer in Texas, and a Fellow of IEEE.