Power Quality Data Mining Using Improved Probabilistic Neural Network

K. Manimala[1] *K. Selvi*[2] *R. Ahila*[3]

ABSTRACT

In this paper, we propose a new technique to enhance the learning capabilities and reduce the computation intensity of probabilistic neural network using the kmeans clustering algorithm for recognizing and classifying power quality disturbances. First, the Discrete Wavelet Transform is employed to extract the energy distribution features of the distorted signal at different resolution levels. Then, kmeans algorithm is applied to the training dataset to reduce the amount of samples to be presented to the neural network, by automatically selecting an optimal set of samples. Finally, the Probabilistic Neural Network classifies these extracted features to identify the disturbance type according to the energy features and disturbance duration. The obtained results demonstrate that the proposed technique performs exceptionally in terms of both accuracy and computation time when applied to the power quality disturbances data set compared to a standard learning schema that uses the full data set. The method is described and demonstrated using 6 different sets of power disturbances.

Keywords : data mining, Power quality, wavelet transform, Probabilistic neural network, kmeans clustering.

[1]Dr. Sivanthi Aditanar College of Engineering, Tiruchendur, India. Ph.no : 9942172538. Email : s_monimala@yahoo.com.

[2]Thiagarajar College of Engineering , Madurai, Email : kseee@tce.edu.

[3]Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, India. Email : r.ahila@yahoo.co.in.

1. INTRODUCTION

Electrical power of good quality is essential for proper operation of electronic equipments such as computers and Programmable Logic Controllers. Malfunction of such equipment may lead to loss of production or disruption of critical services resulting in huge financial and other losses. It is therefore necessary that critical loads be supplied with electricity of acceptable quality. Recognition of the presence of any disturbance and classifying any existing disturbance into a particular type is the first step in combating the problem. In particular, when the disturbance type has been classified accurately, the power quality engineers can define the major effects of the disturbance at the load and analyze the source of the disturbances so that an appropriate solution can be formulated [1], [2].

From earlier research, we have found that there exists a behavioral pattern in the disturbances that can be learned. That is why an artificial neural network is so successful in detecting power disturbances; it is also capable of identifying new disturbances to some degree of resemblance to the learned ones. The neural networks are widely considered as an efficient approach to adaptively classify patterns, but their high computation intensity and the long training cycles greatly hinder their applications, especially for the power disturbances problem, where the amount of treated data is very important. Neural networks have been identified since the beginning as a very promising technique of addressing the power disturbances problem. Many researches have

been performed to this end, and the results varied from inconclusive to extremely promising. The primary premise of neural networks that initially made it attractive was its generalization property, which makes it suitable to identify power disturbances. In addition neural networks also posses the ability to classify patterns, and this property can be used for power disturbances classification, which is a cross disciplinary problem combining signal processing, soft computing and power engineering. The Probabilistic Neural Network (PNN) [2,6,7] can function as a classifier and has the advantage of being a fast learning process as it requires only a single-pass network training stage without any iteration for adjusting weights. Further, it can itself adapt to architectural changes. As the structure of PNN is simple and learning efficiency is very fast it is suitable for signal classification problems. Hence PNN is considered as the best neural network for power disturbances classification. But the major disadvantage of PNN is that all training vectors must be stored and used to classify new vectors, thus requiring large memories for many practical problems [14]. A solution to this problem is to introduce a clustering operation to reduce the number of training vectors without affecting the performance. In this work, an attempt is made to improve the learning capabilities of a probabilistic neural network and reduce the amount of time and resource required by the learning process by sampling the input dataset to be learnt using the K-means algorithm. This paper is organized as follows: section 1 gives the signal processing technique for extracting features from raw signals; section 2 presents k-means clustering technique and the proposedtechnique of samples reduction. The section 3 presents the architecture of the probabilistic neural network with the different used parameters. Section 4 summarizes the obtained results with comparison and discussions. The paper is finally concluded with the most essential points and possible future works.

Fig1 illustrates the principle block diagram of the proposed classifier. The features that distinguish the distorted signals are extracted using discrete wavelet transform in the feature extraction block. The disturbances are classified by applying data mining technique using Probabilistic neural network (PNN) on the relevant features and the reduced training samples obtained from data selection phase using kmeans clustering.

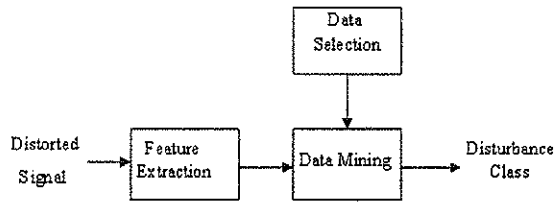

Figure 1 : Diagram Of Proposed Classifier

2. Feature Extraction

2.1 Wavelet Transform

Wavelet analysis is a technique for carving up function or data into multiple components corresponding to different frequency bands. This allows one to study each component separately. Wavelet analysis is a form of time-frequency technique as it evaluates signal simultaneously in the time and frequency domains [3]. It uses wavelets, "small waves", which are functions with limited energy and zero average,

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \qquad (1)$$

The functions are typically normalized, $\|\psi\| = 1$ and centered in the neighborhood of $t = 0$. It plays the same role as the sine and cosine functions in the Fourier analysis. In wavelet transform, a specific wavelet is first selected as the basis function commonly referred to as the mother wavelet. Dilated (stretched) and translated

(shifted in time) versions of the mother wavelet are then generated [2]. Dilation is denoted by the scale parameter a while translation is adjusted through b

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \quad (2)$$

where a is a positive real number and b is a real number. The wavelet transform of a signal $f(t)$ at a scale a and time translation b is the dot product of the signal $f(t)$ and the particular version of the mother wavelet, $\psi_{a,b}(t)$. It is computed by circular convolution of the signal with the wavelet function

$$w\{f(a,b)\} = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{+\infty} f(t) \cdot \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \quad (3)$$

A contracted version of the mother wavelet would correspond to high frequency and is typically used in temporal analysis of signals, while a dilated version corresponds to low frequency and is used for frequency analysis. With wavelet functions, only information of scale $a < 1$ corresponding to high frequencies is obtained. In order to obtain the low-frequency information necessary for full representation of the original signal $f(t)$, it is necessary to determine the wavelet coefficients for scale $a > 1$. This is achieved by introducing a scaling function $\varphi(t)$ which is an aggregation of the mother wavelets $\psi(t)$ at scales greater than 1.The scaling function can also be scaled and translated as the wavelet function,

$$\phi_{a,b}(t) = \frac{1}{\sqrt{a}}\phi\left(\frac{t-a}{b}\right) \quad (4)$$

With scaling function, the low-frequency approximation of $f(t)$ at a scale a is the dot product of the signal and the particular scaling function [3], and can be computed by circular convolution given by (5).

$$L\{f(a,b)\} = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{a}}\phi\left(\frac{t-a}{b}\right) dt \quad (5)$$

Implementation of these two transforms (3) and (5) can be done smoothly in continuous wavelet transform (CWT) or discretely in discrete wavelet transform (DWT).

Successful application of wavelet transform depends heavily on the mother wavelet. The most appropriate one to use is generally the one that resembles the form of the signal. Among the several wavelet functions that were mentioned in the literature, the Daubechies families of wavelets are the most widely used. Among the different dbN (N-order) wavelets, db4 is the most widely adopted wavelet in power quality applications. It has sufficient number of vanishing moments to bring out the transients while maintaining a relatively short support to avoid having too many high-valued coefficients.

2.2. Multiresolution Analysis

One important trait of wavelet transform is that its nonuniform time and frequency spreads across the frequency plane. They vary with scale a but in the opposite manner, with the time spread being directly proportional to a while frequency spread to $1/a$. The resolutions of DWT vary across the planes. At low frequency when the variation is slow, the time resolution is coarse while the frequency resolution is fine. This enables accurate tracking of the frequency while allowing sufficient time for the slow variation to transpire before analysis. On the contrary, in the high-frequency range, it is important to pinpoint when the fast changes occur. The time resolution is therefore small, but the frequency resolution is compromised. This adjustment of the resolutions is inherent in wavelet transform as the wavelet basis is stretched or compressed during the transform [4]. This ability to expand function or signal with different resolutions is termed as multiresolution analysis,

which forms the cornerstone of many wavelet applications. In this sense, a recorder-digitized function $a_0(n)$, which is a sampled signal of $f(t)$, is decomposed into its smoothed version $a_1(n)$ (containing low-frequency components), and detailed version (containing higher-frequency components) $d_1(n)$, using filters $h(n)$ and $g(n)$, respectively. This is first-scale decomposition. The next higher scale decomposition is now based on signal $a_1(n)$ and so on (Fig. 1).

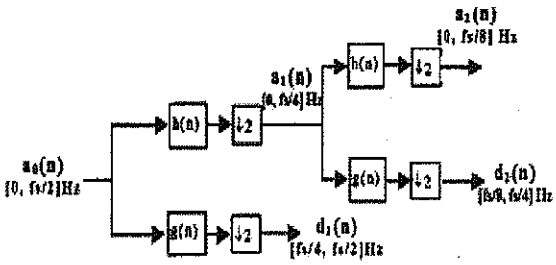

Figure 2: Multiresolution Signal Decomposition

Parseval's Theorem

The energy disturbance features are obtained using Parseval's theorem which is given by

$$\frac{1}{N}\sum_{t}\left|x[t]^2\right| = \frac{1}{N^j}\sum_{k}\left|w_{j,k}\right|^2 + \sum_{j=1}^{J}\left(\frac{1}{N}\sum_{k}\left|w_{j,k}\right|^2\right) \quad (6)$$

The first term on the right of (6) denotes the average power of the approximated version of the decomposed signal, while the second term denotes that of the detailed version of the decomposed signal. The second term giving the energy distribution features of the detailed version of distorted signal will be employed to extract the features of power disturbance.

2.3 Types of Disturbances

The Power Quality (PQ) of electrical power is usually attributed to power-line disturbances, such as sag, swell, interruption, harmonics, flicker, capacitor switching transients and impulse transients.

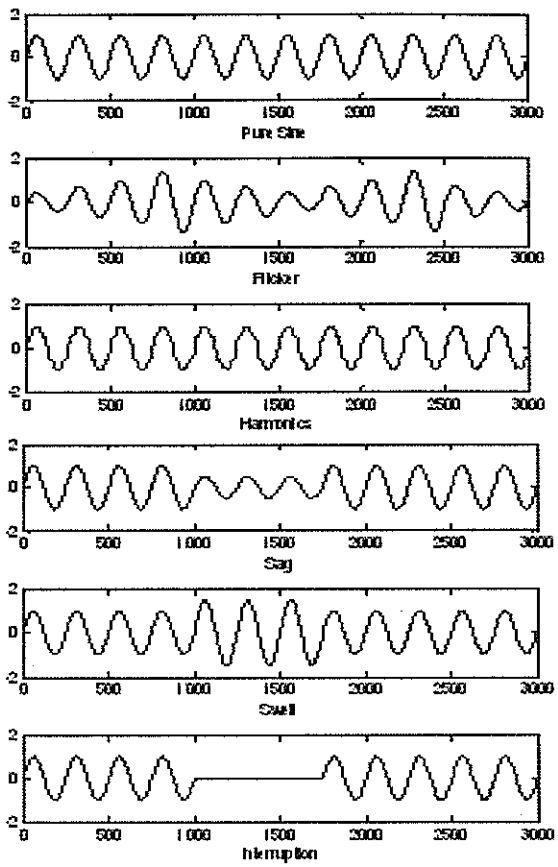

Figure 3: Types of Disturbances

Sag refers to decrease in amplitude and swell refers to increase in amplitude. Interruption refers to the absence of voltage for a short period. Thus, electromagnetic transients, which are momentary voltage surges powerful enough to shatter a generator shaft, can cause sudden catastrophic damage. Harmonics, sometimes referred to as electrical pollution, are distortions of the normal voltage waveforms found in ac transmission, which can arise at virtually any point in a power system. The types of disturbances considered in this paper for classification, including pure sine wave are shown in figure 3.

2.4 Detailed Energy Distribution

The energy of the distorted signal can be partitioned at different resolution levels in different ways depending on the power quality problem. The coefficient w of the

detailed version at each resolution level is used to extract the features of distorted signal for classifying different power quality problems [2]. The process can be represented mathematically by (7)

$$P_j = \frac{1}{N_j}\sum_k \left|w_{j,k}\right|^2 = \frac{\left\|w_j\right\|^2}{N_j} \qquad (7)$$

Normalize the energy features obtained from (7)

$$P_j^D = \left(P_j\right)^{1/2} \qquad (8)$$

Figure 4 Shows The Daubauchie "Db4" Wavelet Function Adopted To Perform The DWT.

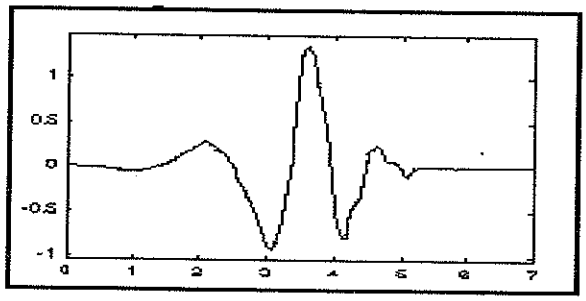

Figure 4 : Db 4 Mother Wavelet

The properties of energy disturbance features are

- The energy distribution remains unaffected by the time of disturbance occurrence.
- The outline of energy distribution remains the same despite variations in the amplitude of the same disturbance type.
- The low-level energy distribution will show obvious variations when the distorted signal contains high-frequency elements. On the contrary, the high- level energy distribution will show obvious variations when the distorted signal contains low-frequency elements.

Figures 5-8 shows that the energy distribution pattern remains the same for the event sag despite occurring at different instants and with different amplitudes.

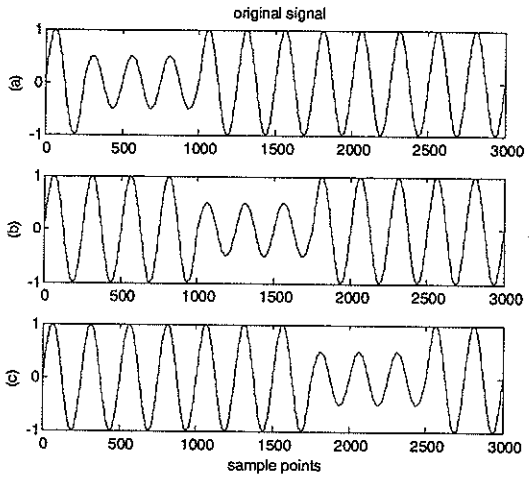

Figure 5: Sag at Different Instants

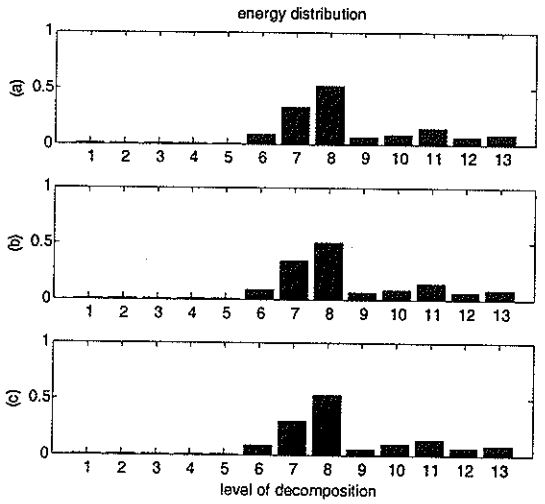

Figure 6 : Energy Distribution Diagram For Sag At Different Instants

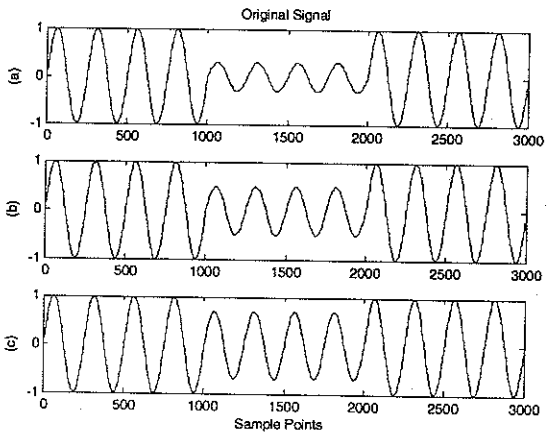

Figure 7 : Sag With Different Amplitudes

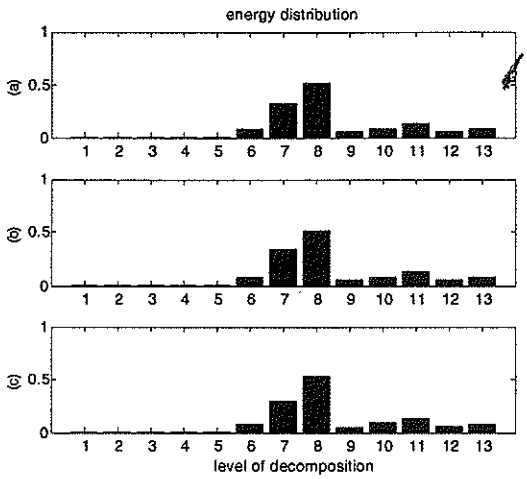

Figure 8: Energy Distribution Diagram For Sag With Different Amplitudes

2.5 Duration of Transients

Employing the DWT technique to analyze the distorted signal through one level decomposition of the MRA will cause the wavelet coefficients at the start and end points of the disturbance to generate severe variation. Therefore, we can easily obtain the start time and end time of the disturbance from the variations in absolute wavelet coefficients and calculate the disturbance duration. Fig 9, 10 shows the plot of level 1 and level 2 DWT coefficients for the disturbance interruption and pure sine wave. The coefficients show variation for interruption from which the disturbance duration can be determined but there is no variation for pure sine wave because the signal is smooth.

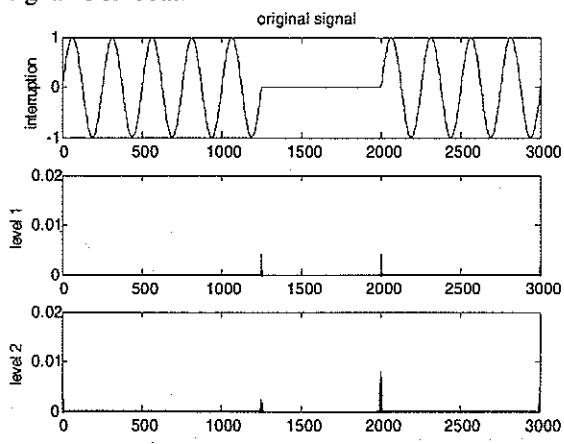

Figure 9 : Plot Of DWT Coefficients For Interruption

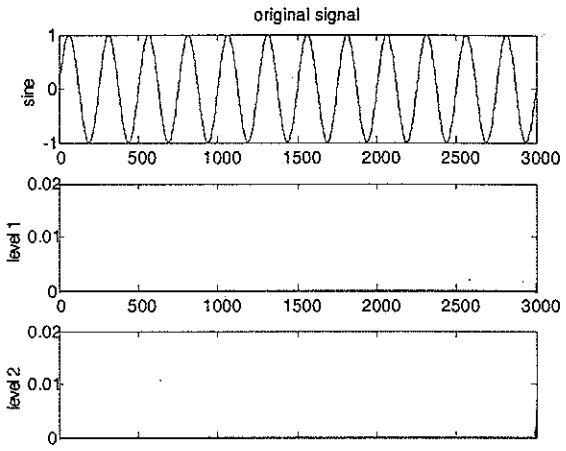

Figure 10 : Plot Of DWT Coefficients For Pure Sine Wave

3. Data Reduction Phase

3.1 Data Clustering

Clustering of data is a method by which large sets of data are grouped into clusters of smaller sets of similar data. A clustering algorithm attempts to find natural groups of components (or data) based on some similarities. The clustering algorithm also finds the centroid of a group of data sets. To determine cluster membership, most algorithms evaluate the distance between a point and the cluster centroids. The output from a clustering algorithm is basically a statistical description of the cluster centroids with the number of components in each cluster. The centroid of a cluster is a point whose parameter values are the mean of the parameter values of all the points in the clusters. The k-means algorithm used in this work is one of the most non-hierarchical methods used for data clustering.

3.2 Algorithm Description

The K-means algorithm [11] is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a

priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early grouping is done. At this point we need to recalculate k new centroids as barycentre of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated; as a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function:

$$J=\sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^j - c_j\right\|^2 \qquad (9)$$

where $\left\|x_i^j - c_j\right\|^2$ is a chosen distance measure between a data point and the cluster centre cj, is an indicator of the distance of the n data points from their respective cluster centers.

The general algorithm is composed of the following steps:

Input:

k: the number of clusters,

D: a data set containing n objects.

Output: A set of k clusters

Method:

(1) Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.

(2) Assign each object to the group that has the closest centroid.

(3) When all objects have been assigned, recalculate the positions of the K centroids.

(4) Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect. K-means is a simple algorithm that has been adapted to many problem domains. The proposed procedure is a simple version of the k-means clustering. Unfortunately there is no general theoretical solution to find the optimal number of clusters for any given data set. A simple approach is to compare the results of multiple runs with different k classes and choose the best one according to a given criterion, but we need to be careful because increasing k results in smaller error function values by definition, but also an increasing risk of over-fitting.

3.3 Data Selection Algorithm

In the present work, the role of the k-means algorithm is to reduce the computation intensity and size of the neural network, by reducing the input set of samples to be learned. This can be achieved by clustering the input dataset using the k-means algorithm, and then take only discriminant samples from the resulting clustering schema to perform the learning process [11]. By doing so, we are trying to select a set of samples that cover at maximum the region of each class in the N-dimensional

space (N is the size of the training vectors). The input classes are clustered separately in such a way to produce a new dataset composed with the centroid of each cluster, and a set of boundary samples selected according to their distance from the centroid. Reducing the number of used samples will enhance significantly the learning performances, and reduce the training time and space requirement, without great loss of the information handled by the resulting set, due to its specific distribution.

Algorithm

Let **A** be the input class;

k: the number of cluster; **S**: the number of samples to be selected ($S \geq k$); **Sam(i)**: the resulting selected set of samples for the cluster i;

Out_ Sam: the output set of samples selected from the class A;

Candidates: a temporary array that contain the cluster points and their respective distance from the centroid.

i,j,min, x: intermediates variables;

ε : Neighborhood parameter

1) Cluster the class A using the k-means algorithm into k cluster.

2) For each cluster cli (i: 1...k) **do**

{Sam (i) :={ centroid (cli)};

j:=1;

For each x **from** cli **do**

{Candidates [j].point:=x;

Candidates [j].location:=dist(x, centroid (cli));

j:=j+1 ;};

Sort the array Candidates in descending order with respect to the values of location field;

j: =1;

While ((card (Sam (i)))<Num_samples(cli))

and (j<card (cli)) **do**{min:=100000;

For each x **from** Sam (i) **do**

{ **if** dist(Candidates[j].point,x)<min

then min:= dist(Candidates[j].point,x) ;

}

if (min > ε) **then**

Sam (i):=Sam (i) $\bigcup$ Candidates[j].point};

j: =j+1; }

if card(Sam(i)) < Num_samples(cli) **then**

repeat {Sam(i):=Sam(i) $\bigcup$ Candidates [random].point

} **until** (card (Sam (i)) = Num_samples (cli));

3) For i=1 **to** k **do**

Out_sam:=Out_sam $\bigcup$ Sam (i);

3.4 Clustering and Selection Parameters

As described above, the sampling algorithm has two parameters to be defined as inputs: the cluster number k of each class, and the number of samples to be extracted. Different possible values were tested during the experiments to find a good compromise between the size of the resulting dataset and its coverage of the input classes' space. The initial centroid values are chosen randomly from the training data set. Table1 lists the final chosen parameters for each class. It is a small portion of the initial dataset (40% only).The proposed algorithm reduces the training data size from 600 samples to 240 samples.

Table 1: Samples Selected Using Clustering Technique

Class	Initial Samples	Number Of clusters	Selected Samples
Sag	150	6	60
Swell	100	5	50
Interruption	150	6	60
Harmonics	100	4	30
Flicker	100	4	40

4. Probabilistic Neural Network (PNN)

Specht developed the Probabilistic Neural Network (PNN), a 3-layer, feed forward, one pass training algorithm for classification. In its standard form a PNN is not trained. Training vectors simply become the weight vectors in first layer. The major difference from a feed forward network trained by the back propagation is that it can be constructed after only a single pass of the training exemplars.

The probabilistic neural-network (PNN) model is one of the supervised learning networks, and has the following features distinct from those of other networks in the learning processes.

- It is implemented using the probabilistic model, such as Bayesian classifiers.
- A PNN is guaranteed to converge to a Bayesian classifier provided that it is given enough training data.
- No learning processes are required
- No need to set the initial weights
- No relationship between learning processes and recalling processes.
- The differences between the inference vector and the target vector are not used to modify the weights of the network

The input layer has m units to which the m dimensional input vector is applied. The first hidden layer has one pattern unit for each exemplar. The second hidden layer contains one summation unit for each class. The output layer is the decision layer which contains one neuron to specify the class.

Fig.11 shows the architecture of the proposed PNN classifier that is composed of the radial basis layer and the competitive layer. The learning speed of the PNN model is very fast, making it suitable for fault diagnosis and signal classification problems in real time [2].

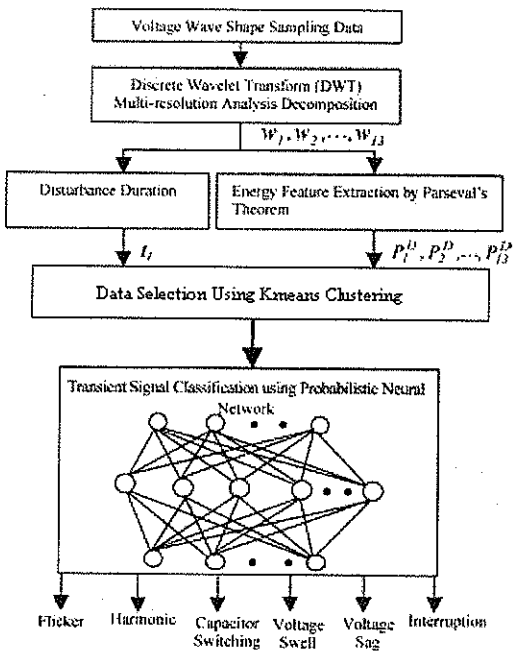

Figure 11: Diagram of the Proposed PNN Classifier

The PNN can function as a classifier and has the advantage of being a fast learning process as it requires only a single pass network training stage without any iteration for adjusting weights. Further, it can itself adapt to architectural changes. As the structure of PNN is simple and learning efficiency is very fast it is suitable for signal classification problems. Its major disadvantage is that all training vectors must be stored and used to classify new vectors, thus requiring large memories for many practical problems. A solution to this problem is to introduce a clustering operation to reduce the number of training vectors without affecting the performance [14]. The reduced training samples obtained from kmeans clustering technique and the sample selection algorithm described above improves the performance of PNN classifier.

In the signal classification application, the training examples are classified according to their distribution values of probabilistic density function (PDF), which is the basic principle of the PNN. A simple PDF is defined as follows:

$$f_k(X) = \frac{1}{N_k}\sum_{j=1}^{Nk} \exp\left(-\frac{\|X - X_{kj}\|}{2\sigma^2}\right) \quad (10)$$

Modifying and applying (10) to the output vector H of the hidden layer in the PNN is as below

$$H_h = \exp\left(\frac{-\sum\left(X_i - W_{ih}^{xh}\right)^2}{2\sigma^2}\right) \quad (11)$$

The algorithm of the inference output vector Y in the PNN is as follows:

$$net_j = \frac{1}{N_j}\sum_h W_{hj}^{hy} \cdot H_h \ and \ N_j = \sum_h W_{hj}^{hy} \quad (12)$$

$$if \ net_j = \max_k(net_k) \ then \ Y_j = 1, else \ Y_j = 0 \quad (13)$$

where

i number of input layers;

h number of hidden layers;

j number of output layers;

k number of training examples;

N_k number of classifications (clusters);

ó smoothing parameter or spread factor

X input vector;

$\|X - X_{kj}\|$ Euclidean distance between the vectors X and X_{kj}, i.e. $\|X - X_{kj}\| = \sum_i (X_i - X_{kj})^2$;

W_{ih}^{xh} is the connection weight between the input layer X and the hidden layer H;

W_{hj}^{hy} is the connection weight between the hidden layer H and the output layer Y .

4.1 Network Architecture Used With the Standard Method

The input layer contains 14 neurons (13 energy features from DWT, time of duration), the first hidden layer contains 600 neurons (600 training exemplars) and the second hidden layer contains 6 neurons (6 classes) and the final output layer contains one neuron. The spread factor is chosen as 0.01.

4.2 Network Architecture Used With the Proposed Method

Since the proposed schema use a reduced set of samples from the original data set because of clustering technique, the network architecture can be more trivial. We use an input layer of 14 neurons two hidden layers with 240(selected training exemplars from clustering) and 6 neurons respectively, and the final output layer with one neuron.

5. Application And Results

5.1 Signal Modeling

Signal modeling by parametric equations for classifiers tests was advantageous in some aspects. It was possible to change testing and training signal parameters in a wide range and in a controlled manner. Signals simulated that way were very close to reality. On the other hand, different signals belonging to the same class gave the opportunity to estimate the generalization ability of classifiers based on Neural Networks. Signals belonging to six main groups of disturbances [9], were simulated. The classes and respective parametric equations for simulation of signals are summarized in Table 2 As a special group, the nondisturbed signals were chosen (pure sinusoid).The ranges of signals parameter variation are shown in Table 3.The variation range corresponds to values measured in real power systems.

In order to obtain representative signals for the most common power quality disturbances to serve the purpose of training, as well as the testing of the PNN classifier, power quality disturbance signals are simulated using Matlab. Six categories of disturbances are simulated, namely, undisturbed sinusoid, sudden swell, sudden sag, interruption, harmonics, and voltage flicker. The disturbances are based on ten cycles of voltage waveform. These waveforms are generated at a sampling rate of 256 samples / cycle for a total of 3001 points.

Table 2 : Parametric Equations For Simulation Of Disturbed Signals

Event	Equation
Pure Sinusoid	$v(t)=\sin(wt)$
Sag	$v(t)=1-\alpha_{ss}((1(t-t_1)-1(t-t_2)))\sin(wt)$
Swell	$v(t)=1+\alpha_{sw}((1(t-t_1)-1(t-t_2)))\sin(wt)$
Interruption	$v(t)=1-((1(t-t_1)-1(t-t_2)))\sin(wt)$
Harmonics	$v(t)=\begin{pmatrix}\alpha_{k1}\sin(wt)+\alpha_{k3}\sin(3wt)+\\ \alpha_{k5}\sin(5wt)+\alpha_{k7}\sin(7wt)\end{pmatrix}$
Flicker	$v(t)=(1+\alpha_f \sin(\beta_f wt))\sin(wt)$

The described experiments were implemented using the MATLAB7 environment, on a Pentium IV, 2.88 GHz, with 256 Mb of memory.

5.2 Result of the Proposed Classification Method

Using the output set of samples obtained from the clustering phase, we construct a new training set and the resulting set is presented to the probabilistic neural network described above. Table5 summarizes the obtained performances results. Because the goal of this work is to enhance the learning capabilities of the probabilistic neural network techniques for classification

Table 3 : Parameters Variation In Simulated Signals

Event	Parameters Variation
Pure Sinusoid	*Amplitude* : 1 *Frequency* : 50 *Hz*
Sag	*Duration* : $(t_2 - t_1) = (0 - 9)T$ *Amplitude* : $\alpha_{ss} = 0.3 - 0.8$
Swell	*Duration* : $(t_2 - t_1) = (0 - 8)T$ *Amplitude* : $\alpha_{ss} = 0.3 - 0.7$
Interruption	——
Harmonics	*Order* : 3,5,7 *Amplitude* : 0 – 0.9
Flicker	*Frequency* : $(5 - 10)Hz$ $\alpha_f = 0.1 - 0.2$

of power quality disturbances, the proposed method is compared to a classic probabilistic neural networks implementation that use the full set of samples. The 10-fold cross validation evaluation result of the PNN classifier for the six data sets is shown in Table 4.The diagonal element (shown by bold letters) represent correctly classified PQ events. In Table 5, the 10-fold cross validation evaluation result of the PNN classifier with clustering technique for the six data sets is shown. Cross-validation, is the practice of partitioning a sample

Table 4 : 10-Fold Cross Validation Evaluation Result Of The PNN Classifier

Signal	Sine	Sag	Swell	Inter	Har	Flick
Sine	**100**	0	0	0	0	0
Sag	0	**88**	0	12	0	0
Swell	0	0	**100**	0	0	0
Interr	0	17	0	**83**	0	0
Har	0	0	0	0	**100**	0
Flick	0	0	0	0	0	**100**
Overall	**94**					

of data into subsets such that the analysis is initially performed on a single subset, while the other subset(s) are retained for subsequent use in confirming and validating the initial analysis. The proposed PNN classifier with clustering technique achieves 97% accuracy.

Table 5 : 10-Fold Cross Validation Evaluation Result Of PNN With Clustering

Signal	Sine	Sag	Swell	Inter	Flick	Har
Sine	**100**	0	0	0	0	0
Sag	0	**94**	0	6	0	0
Swell	0	0	**100**	0	0	0
Interr	0	10	0	**90**	0	0
Har	0	0	0	0	**100**	0
Flick	0	0	0	0	0	**100**
Overall	97					

Parameters	PNN	PNN with Clustering
Training time	0.02	0.01
Testing time	0.005	0.005
Training Samples	600	240
Overall Accuracy	94	97

The test result shows that the PNN classifier with clustering technique attains better recognition rates with less number of training data when compared with the conventional PNN classifier. The problem of over fitting does not likely happen in our PNN model since only one parameter (smoothing factor) needed to be adjusted. The tuning of the smoothing parameter is stopped whenever the trained PNN classifier generates the fewest errors in cross validation tests. It is chosen to be 0.01.

6. Conclusion and Future Work

Since the high computation intensity and the long training cycles are the main obstacle to any neural networks, we propose a new learning schema to reduce the amount of used samples using a k-means clustering algorithm. The proposed method can reduce the quantity of training data without losing its property, thus requiring less memory space and computing time for proper classification of disturbance types. The experimental results showed that the proposed method has the ability of recognizing and classifying different power disturbance types efficiently, and it has the potential to enhance the performance of the power transient recorder with real time processing capability. The proposed classifier has provided significant improvement in classifying the PQ events, as compared to the conventional PNN classifiers.

More work must be performed to find an optimal way to determine the number of used clusters and selected samples of each class. This work use only heuristics and trays to determine these parameters. Because the distorted signals in this study were generated by simulation, employing real distorted signals measured by the digital recorder to improve the proposed method for more number of disturbances is one of our future works.

References

1. Mishra S. C, Bhende N, Panigrahi B. K, *"Detection and Classification of Power Quality Disturbances Using S-Transform and Probabilistic Neural Network"*, IEEE Transactions on Power Delivery, Vol. 23, PP.280-287, January 2008.
2. Gaing Zwe-Lee, *"Wavelet-Based Neural Network for Power Disturbance Recognition and Classification"*, IEEE transactions on Power delivery, Vol. 19, PP. 1560-1567 , October 2004.

3. S. Chen and H. Y. Zhu, *"Wavelet Transform for Processing Power Quality Disturbances"*, EURASIP Journal on Advances in Signal Processing Vol.2007, Article ID 47695, PP. 20 .

4. Borras Dolores, Castilla .M, Moreno Narciso and Montano J.C , *"Wavelet and Neural Structure: A New Tool for Diagnostic of Power System Disturbances"*, IEEE Transactions on Power Delivery, Vol.37, PP. 184-190, January 2001.

5. Santoso.S, Powers E. J, Grady. W. M and Hofmann.P, *"Power quality assessment via wavelet transform analysis"* , IEEE Trans. Power Delivery, Vol. 11, PP. 924–930, Apr. 1996.

6. Specht.D. F, *"Probabilistic neural networks"*, International Neural Network Society, Neural Networks, Vol. 3, PP. 109 -118, 1990.

7. Rutkowski Leszek, *"Adaptive Probabilistic Neural Networks for Pattern Classification in Time-Varying Environment"*, IEEE Transactions on Neural Networks, Vol. 15, PP.811-827, July 2004.

8. Gargoom Ameen, Ertugrul Nesimi, Soong Wen, *"Investigation of Effective Automatic Recognition Systems of Power-Quality Events"*, IEEE Trans. On Power Delivery, Vol. 22, PP.2319-2326, October 2007.

9. IEEE Recommended Practice for Monitoring Electric Power Quality, IEEE/Std. 1159-1995.

10. Janik Przemyslaw and Lobos Tadeusz, *"Automated Classification of Power- Quality Disturbances Using SVM and RBF Networks"*, IEEE Transactions on Power Delivery, Vol. 21, PP.1663-1669, July 2006.

11. Faraoun.K.M and Boukelif .A, *"Neural Networks Learning improvement using the K-Means Clustering Algorithm to Detect Network Intrusions"*, International Journal of Computational Intelligence, Vol.3, No. 2, 2005.

12. Reaz Ibne, Choong Florence, Sulaiman Mohd, Kamada Faisal, *"Expert System for Power Quality Disturbance Classifier"*, IEEE Transactions On Power Delivery, Vol. 22, PP.1979-1988, July 2007.

13. Youssef A. M, Abdel-Galil. T. K, Salama.M, *"Disturbance Classification Utilizing Dynamic Time Warping Classifier"*, IEEE Transactions On Power Delivery, Vol. 19, PP.272-278, January 2004.

14. Zaknich Anthony, *"A Vector Quantization Reduction Method for the Probabilistic Neural Network"*,(no date).

15. Yang Kuihe, Shan Ganlin, Zhao Lingling, *"Application of Wavelet Packet Analysis and Probabilistic Neural Networks in Fault Diagnosis"*, Proceedings of the 6th World Congress on Intelligent Control and automation, June 21 – 23, 2006.

16. Huang Chenn, Liao Wei, *"A comparative Study of Feature Selection Methods for Probabilistic Neural Networks in Cancer classification"*, Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03) 1082-3409.

17. Erman Jeffrey, Arlitt Martin, Mahanti Anirban, *"Traffic Classification Using Clustering Algorithms"*, SIGCOMM'06 Workshops September 11-15, 2006.

18. Chuang Cheng, Lu Yen, Tsong-Liang, Huang Ying, *"Recognition of Multiple PQ Disturbances Using Wavelet-based Neural Networks—art Implementation and Applications"*,IEEE/PES Transmission and Distribution Conference , 2005.

Author's Biography

Manimala K. received B.E. degree in Electrical and Electronics Engineering from Government College of Engineering, Tirunelveli, India. She earned a Master's degree in Computer science and Engineering from Manonmaniam Sundaranar University, Tirunelveli, India in 2004. She has 12 years teaching experience. She is currently working as an Assistant Professor at the Computer Science and Engineering Department of Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, Tamilnadu, India. She is pursuing PhD in the area of Data Mining.

Dr.Selvi K. Obtained B.E (EEE) with Honours, M.E (Power System) with Distin, from Madurai Kamaraj University in the year 1989 and 1995 respectively. She obtained Ph.D in Electricity Deregulation in June 2005 from Madurai Kamaraj University. She is currently working as Assistant Professor in Department of Electrical Engineering, in Thiagarajar college of Engineering, Madurai.,Tamilnadu, India. She has obtained Young Scientist Fellowship from Dept. of Science and Technology. Her research interests are Electricity deregulation and AI techniques. She has published many journal papers.

Ahila R. Received B.E. degree in Electrical and Electronics engineering in 1995 from Government College of Engineering, Tirunelveli and M.E. degree in Computer Science and Engineering in 2004 from Manonmaniam Sundaranar University, Tirunelveli, India. She has 13 years teaching experience. She is currently an Assistant Professor at the Computer Science and Engineering Department of Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, India. Her research interests are Data Mining, Pattern Recognition.