# ANN based Multi Classifier System for Prediction of High Energy Shower Primary Energy and Core Location

Gitanjali Devi, Kandarpa Kumar Sarma, Pranayee Datta, and Anjana Kakoti Mahanta

*Abstract*—Cosmic showers, during the transit through space, produce sub - products as a result of interactions with the intergalactic or interstellar medium which after entering earth generate secondary particles called Extensive Air Shower (EAS). Detection and analysis of High Energy Particle Showers involve a plethora of theoretical and experimental works with a host of constraints resulting in inaccuracies in measurements. Therefore, there exist a necessity to develop a readily available system based on soft-computational approaches which can be used for EAS analysis. This is due to the fact that soft computational tools such as Artificial Neural Network (ANN)s can be trained as classifiers to adapt and learn the surrounding variations. But single classifiers fail to reach optimality of decision making in many situations for which Multiple Classifier System (MCS) are preferred to enhance the ability of the system to make decisions adjusting to finer variations. This work describes the formation of an MCS using Multi Layer Perceptron (MLP), Recurrent Neural Network (RNN) and Probabilistic Neural Network (PNN) with data inputs from correlation mapping Self Organizing Map (SOM) blocks and the output optimized by another SOM. The results show that the set-up can be adopted for real time practical applications for prediction of primary energy and location of EAS from density values captured using detectors in a circular grid.

*Keywords*—EAS, Shower, Core, ANN, Location.

Cosmic showers are the generators of Extensive Air Showers (EAS) [1] and travels from the emission source until it reaches the Earth. During its passage through space, there are interactions with the intergalactic or interstellar medium for which sub-products of the original cosmic particles are produced. When these particles enter the Earth they have extremely high energy and produce complicated processes due to the interactions with atmospheric nuclei. As a result, certain secondary particles are generated which are called Extensive Air Shower or EAS [2]. The study of these EAS involves the measurement of the position, size, primary energy, time extent of the events and other factors. Detection and analysis of High Energy Particle Showers involve a plethora of theoretical and experimental works comprising of complex measurement and detection equipments. There are several constraints related to experimental works involving the showers some of which are due to partial knowledge regarding interactions of shower particles and primary energies [2]. As a result, inaccuracies become a part of measurements of EAS position, size, primary energy, time extent of the events etc. These difficulties make the analysis of showers a tedious work and requires constant support from expensive experimental set-ups which is a limitation for visualization, conceptualization and monitoring of EAS events. Therefore, there exist a necessity to develop a readily available system based on soft-computational approaches which can be used to predict primary energy and locations of shower events. This is due to the fact that soft computational tools like Artificial Neural Network (ANN)s can be trained to adapt to situations and learn the variations taking place. This knowledge can be utilized for predicting future events or other related aspects which can be an aid to expand the knowledge of EAS. Also the outcomes of experiments involving such tools can be made a part of physical experiential apparatus to facilitate adaptive orientation of monitoring and analysis of EAS phenomena that too in real time.

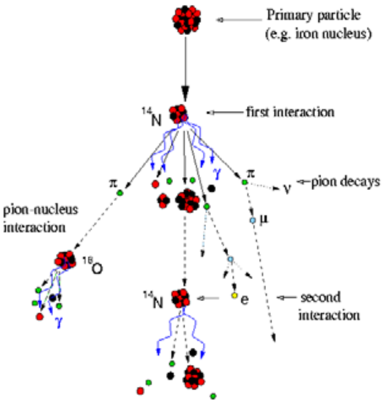

Fig. 1. Depiction of Extensive Air Shower generation

Several works exist which have used multiple approaches to analyze EASs and thereby develop applications suitable for shower events. A work by D. Hanna [3] reports application of ANNs for EAS. Another work by J C Perrett and J T P M van Stekelenborg [4] describes the implementation of an ANN to estimate the core position and energy of EASs recorded by the South Pole Air Shower Experiment (SPASE) [5]. Another work of similar nature is [7]. This work discusses the possibilities of using ANNs for individual EAS data evaluation. A work as cited in [6] uses ANN for

Gitanjali Devi is with the Deptt. of Physics, Lalit Chandra Bharali College, Guwahati - 781011, Assam, India. e-mail: (gitanjalilcb@gmail.com); Kandarpa Kumar Sarma and Pranayee Datta are with Deptt. of Electronics and Communication Technology, Gauhati University, Guwahati - 781014, Assam, India e-mail: (kandarpaks@gmail.com)

Anjana Kakoti Mahanta is with Deptt. of Computer Science, Gauhati University, Guwahati -781014, Assam, India.

providing a mass likelihood distribution for each measured shower, based on its multi-parameter training with simulated showers. Another work by A. Chilingarian . et. al [8] is based on ANN models to recognize the experimental EAS without known primary energy.

Application of soft computational methods to EAS analysis is related to the configuration of tools like ANNs as classifiers. ANN as a classifier generates a number of classification zones demarcated by decision boundaries which determines the clustering of input samples as per the decision rule. But single classifiers fail to reach optimality of decision making in many situations for which Multiple Classifier System (MCS)s are preferred to enhance the ability of the system to make decisions adjusting to finer variations. Such an aspect is more relevant to the study of EAS because of the volume of correlated data involved. This work considers the formulation a MCS for application in high energy shower analysis with special emphasis on prediction of EAS primary energy and core positions. The MCS is constituted using Multi Layer Perceptron (MLP), Recurrent Neural Network (RNN) and Probabilistic Neural Network (PNN) with data inputs from correlation mapping Self Organizing Map (SOM) blocks and the output optimized by another SOM. The complete work is related to the formation of a MCS with heterogeneous ANNs and a Committee Machine built using MLP blocks for prediction of primary energy and core locations of EAS from density values provided by detectors distributed in a circular arrangement of 100 meters. The EAS events are assumed to be taking place inside the arc with the detectors recording the phenomena from all the locations within the arrangement. The requirement is to train the MCS and the Committee Machine with density values from the detectors to enable them predict primary energies and core locations and produce a comparative performance measure as demonstrated by the two approaches.

Multiple Classifier System (MCS)s have become an area of research for application in pattern recognition including human computer interaction (HCI) due to their ability to provide improved discrimination capacity compared to single classifiers. The challenge is however becoming increasingly intense to reach the point of perfection as dictated by the theoretical optimality of decision making. Single classifiers, regardless of their capability to make satisfactory recognition performance, are limited by just one estimate of the optimal decision rule. In this context MCSs become relevant as these enable the creation of more than one optimal decision boundary and improve performance.

A few works related to the use of MCS for data analysis, pattern recognition and clustering is included here as an indicator of the effectiveness of such approaches. A work [9] shows that the approach using MCS is effective as well as robust for the classification of time series data. Another work [10] focuses on the role of Artificial Neural Network (ANN)s in MCSs for remote sensing applications. A work referred in [11] is related to a method of reduction of the data set with the use of multiple classifiers. A tutorial on the basic considerations of MCS and its applications is included in [12]. Formation of a tree classifiers using ANNs for speech processing is reported by [13]. Application of MCS for Arabic work recognition is available in [14].

The description is organized as below: Section I provides a brief account of the proposed system for prediction of priamry energy and location of EAS using a MCS and Committee Machine. This section has several subsections of which Section I-A deals with MLP training. Section I-B provides an account of the RNN block used and Section I-C describes the considerations related to the PNN. Section I-D shows how the Committee Machine architecture can be formulated for the work. The experiential details are provided by Section II where Section II-A and Section II-B include the outcomes derived using MLP, RNN and PNN blocks separately combined by SOM optimizer. The summary results from the two approaches are included at the end of the section. Section III concludes the description.

## I. System Model

The system consists of a conceptual arrangement of detectors in a circle of radius 100 meters with density values of the shower events considered in groups taken from each of the four quadrants. The core positions are considered to be placed inside the 50 meter radius. The experimental set - up consists of a group of Self Organizing Map (SOM)s used for data mapping / or data size reducing and three ANNs blocks. These ANN blocks are Multi Layer Perceptron (MLP), Recurrent Neural Network (RNN) and Probabilistic Neural Network (PNN) structures. The MLP, RNN and PNN forms the MCS system. The density of the detector size is taken to be 100 per quadrant to obtain more detailed description for the EAS but since the data shall be correlated, SOM blocks are used to map the most relevant portion for use with the MCS system. The complete system is depicted in Figure 2.

Let $Dlr_i$ and $Cr_i$ be the sets of density values and core positions respectively captured from a detector array placed in the circular arrangement. During the shower event a core may be placed within the 50 meter arc during which the sensors placed inside and outside the 50 meter circle act as detectors. The data captured from the detectors are density values of the showers and are applied to Self Organizing Map (SOM) blocks. These reduce the density values to one fourth of the applied size which is equally contributed by each of the four quadrants of the circular arrangement.

The output of the first SOM is

$$y_{11}[n] = \sum_{i=1}^{4} Dlr_i[n] \times y_1[n] \tag{1}$$

where $y_1[.]$ represents the system function of the SOM. This $y_{11}[n]$ represents the optimized density values contributed by the four quadrants.

Similarly, the second SOM produces an output as given by

$$y_{21}[n] = \sum_{i=1}^{4} Cr_i[n] \times y_2[n] \tag{2}$$

with $y_2(.)$ representing the system function of the SOM. The set $y_{21}[n]$ represents the optimized core position coordinates distributed along the four quadrants. The processes represented

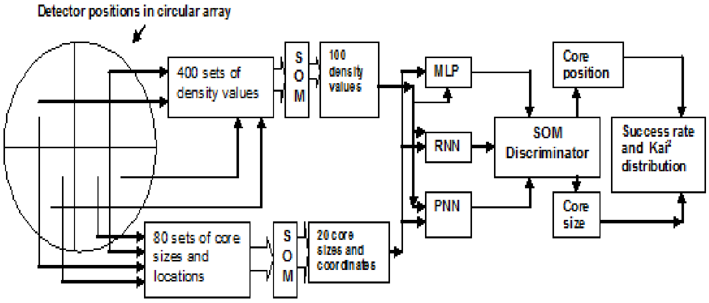
Detector positions in circular array
400 sets of density values
S O M
100 density values
MLP
RNN
PNN
SOM Discriminator
Core position
Core size
Success rate and Kai² distribution
80 sets of core sizes and locations
S O M
20 core sizes and coordinates

Fig. 2. MCS formed by heterogeneous ANN blocks for EAS size and location prediction

by Eq.s 1 and 2 are akin to a mapping process where the correlation between data sets are reduced to around 25% of the original.

The training process of the SOM is linked with the creation of a code-book which provides the account of finding the best match among the set of input patterns given to it. There are two ways in which the best matching code-book vector can be found [15] [16]. The first way is to employ an inner product criterion: "select the best reference vector by choosing the neuron in the competitive layer that receives the maximum activation". This means that for the current input vector $X_k$, all neuron activations are computed

$$y_j = X_k^T W_j \qquad j = 1, ......, m \tag{3}$$

and the winning neuron index, J, satisfies

$$y_J = max_j\{X_k^T W_j\} \tag{4}$$

Alternatively one might select the winner based on a Euclidean distance measure. Here the distance is measured between the present input $X_k$ and the weight vectors $W_j$, and the winning neuron index J, satisfies

$$\|X_k - W_J\| = min_j\{\|X_k - W_j\|\} \tag{5}$$

Competitive learning requires that the weight vector of the winning neuron be made to correlate more with the input vector. This is done by perturbation of only the winning weight vector $W_J = (w_{1J}, ..............., w_{nJ})^T$ towards the input vector. The scalar form of this learning law in difference form is presented below:

$$w_{iJ}^{k+1} = w_{iJ}^k + \eta x_i^k \qquad i = 1, ........, n \tag{6}$$

However, the weight can grow without bound for which some form of normalization is required [15] [16].

$$w_{iJ}^{k+1} = w_{iJ}^k + \eta(\frac{x_i^k}{\sum_j x_j^k} - w_{iJ}^k) \qquad i = 1, ........, n \tag{7}$$

This equation leads to total weight normalization.

$$\tilde{W}_{J(k+1)} = \tilde{W}_{J(k)} + \eta(1 - \tilde{W}_{J(k)}) \tag{8}$$

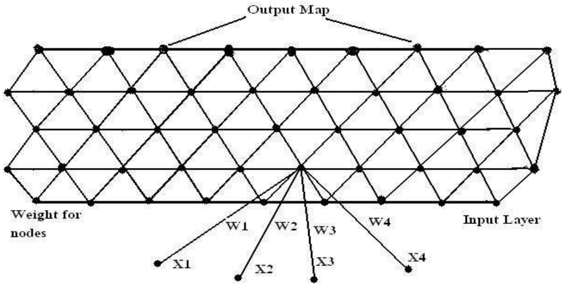

Fig. 3. Self Organizing Feature Map

where $\tilde{W}_{J(k)} = \sum_{i=1}^n w_{iJ}^k$.

The output of the SOMs are applied to the three ANNs blocks which are MLP, RNN and PNN structures. The output of these three blocks can be given as

$$y_{31} = y_{11} \times [ANN_1]|_{y_{21}} \tag{9}$$

$$y_{32} = y_{11} \times [ANN_2]|_{y_{21}} \tag{10}$$

$$y_{33} = y_{11} \times [ANN_3]|_{y_{21}} \tag{11}$$

where $ANN_1$ is and MLP, $ANN_2$ is a RNN and $ANN_3$ is a PNN and are trained as per the considerations mentioned in Sections I-A to I-C.

The final segment of the system is a SOM block used as an optimizer of the outputs generated by the three ANNs. At a given instant the SOM retains the best output among the three ANNs. The selection is made by resorting to "Winner Takes All" approach of training and an Euclidean distance based cost function [16]. The optimization rule can be expressed as

$$Y_{out} = Best \quad of\{y_{31}, y_{32}, y_{33}\} \tag{12}$$

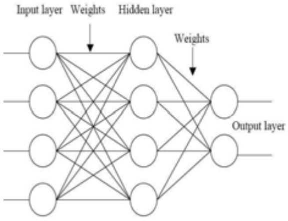

Fig. 4. Multi Layer Perceptron

### A. *Multi Layer Perceptron (MLP) training using (error) Back Propagation Algorithm:*

A MLP (Figure 4) consists of several layers of neurons. The equation for output in a MLP with one hidden layer is given as:

$$O_x = \sum_{i=1}^{N} \beta_i g[w]_i.[x] + b_i \tag{13}$$

where [x] is the input vector, [w] is the associated weight vector, b is a bias value and g(.) is the activation function and $\beta_i$ is the weight value between the $i^{th}$ hidden neuron. The process of adjusting the weights and biases of a MLP is known as *training* carried out using (error) back-propagation algorithm. It entails a backward propagation of the error correction through each neuron in the network. Classification by an ANN involves training it so as to provide an optimum decision rule for grouping all the outputs of the network such that the training minimizes the risk functional [15] [16] :

$$R = \frac{1}{2N} \sum (d_j - F(x_j))^2 \tag{14}$$

where $d_j$ is the desired output pattern for the prototype vector $x_j$, $((.))$ is the Euclidean norm of the enclosed vector and N is the total number of samples presented to the network in training. The decision rule therefore can be given by the output of the network:

$$y_{kj} = F_k(x_j) \tag{15}$$

for the $j^{th}$ input vector $x_j$.

The steps involved during adaptive updating of the MLP are as below:

1) **Initialization and Samples**: Let input be $p_m = [p_{m1}, p_{m2}.....p_{mL}]$, the desired output be $d_m$=$[d_{m1}, d_{m2}......d_{mL}]$ and **W** be a matrix of $C$x$P$ where $P$ is the length of the input vector used for each of the $C$ classes holding random weight values for the connectionsit links between the constituent layers of the MLP.
   - Compute the values of the hidden nodes as:

$$net^h_{mj} = \sum_{i=1}^{L} w^h_{ji} p^{mi} + \emptyset^h_j \tag{16}$$

   - Calculate the output from the hidden layer as

$$o^h_{mj} = f^h_j(net^h_{mj}) \tag{17}$$

   where
   f(x)=$\frac{1}{e^x}$
   or
   f(x)=$\frac{e^x - e^{-x}}{e^x + e^{-x}}$

   depending upon the choice of the activation function.
   - Calculate the values of the output node as:

$$o^o_{mk} = f^o_k(net^o_{mj}) \tag{18}$$

2) **Forward Computation**: Compute the errors:

$$e_{jn} = d_{jn} - o_{jn} \tag{19}$$

   Calculate the mean square error(MSE) as :

$$MSE = \frac{\sum_{j=1}^{M} \sum_{n=1}^{L} e^2_{jn}}{2M} \tag{20}$$

   Error terms for the output layer is:

$$\delta^o_{mk} = o^o_{mk}(1 - o^o_{mk}) e_{mn} \tag{21}$$

   Error terms for the hidden layer:

$$\delta^h_{mk} = o^h_{mk}(1 - o^h_{mk}) \sum_j \delta^o_{mj} w^o_{jk} \tag{22}$$

3) **Weight Update**: A generalized weight update expression with a momentum term is:

$$w^h_{ji}(t+1) = w^h_{ji}(t) + \eta \delta^h_{mj} p_i + \alpha(w^o_{ji}(t+1) - w_{ji} \tag{23}$$

One cycle through the complete training set forms one epoch. The above is repeated till MSE meets the performance criteria. While repeating the above the number of epoch elapsed is counted. A few methods used for MLP training includes:

- Gradient Descent (GDBP)
- Gradient Descent with Momentum BP (GDMBP)
- Gradient Descent with Adaptive Learning Rate BP (GDALRBP) and
- Gradient Descent with Adaptive Learning Rate and Momentum BP (GDALMBP).

### B. *Recurrent Neural Network (RNN) training using Real time Recurrent Learning:*

A RNN is an ANN with one or more feedback loops. The feedback can be of a local or global kind. The RNN maybe considered to be an MLP having a local or global feedback in a variety of forms. It may have feedback from the output neurons of the MLP to the input layer. Yet another possible form of global feedback is from the hidden neurons of the ANN to the input layer [15] [16]. RNN learning takes place by following a method referred to as *Real-Time Recurrent Learning* (RTRL). The name is related to the fact that adjustments are made to the synaptic weights of a fully connected RNN in real time. It means that while the network continues to perform its function, the training continues. Figure 5 shows the the layout of such a RNN. The network has two distinct layers: a concatenated input-feedback layer and a processing layer of computation nodes. Correspondingly, the synaptic connections

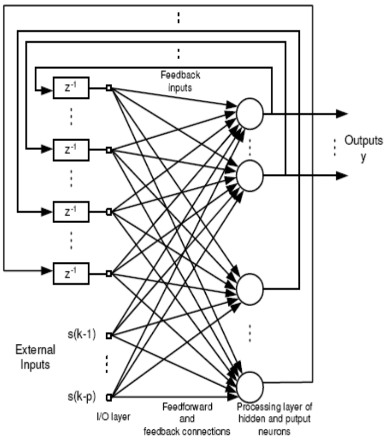

Fig. 5. Fully connected RNN

of the network are made up of feedforward and feedback connections.

The Real Time Recurrent Learning Algorithm can be summarized as follows [15] [16]:

Let

1) $\Lambda_j(n)$ be a $q - by - (q + m + 1)$ matrix holding the partial derivative of the state vector $x(n)$ with respect to the weight vector $w_j$;
2) $U_j(n)$ be a $q - by - (q + m + 1)$ matrix with zeros in all rows, except for the $j^{th}$ row that is equal to the transpose of vector $\xi(n)$:

$$U_j(n) = \begin{pmatrix} 0 \\ \xi_n^T \\ 0 \end{pmatrix} \longleftarrow j^{th}, \qquad j = 1, 2......., q \quad (24)$$

3) $\Phi(n)$ be a q-by-q diagonal matrix whose $k^{th}$ diagonal elements is the partial derivative of the activation function with respect to its argument, evaluated at $w_j^T\xi(n)$:

$$\Phi(n) = diag\ \left(\varphi'(w_1^T\xi(n)), ......\varphi'(w_j^T\xi(n)), ....., \varphi'(w_q^T\xi(n))\right) \quad (25)$$

Also, let, m be the dimensionality of input space, q the dimensionality of state space, p the dimensionality of output space and $w_j$ be the synaptic weight vector of the neurons. With synaptic weight **w** taking small random values, state vector $x(0) = 0$ and $\Lambda_j(0) = 0$ for j = 1, 2,...q.

Compute for $n = 0, 1, 2.....,$

$$\Lambda_j(n+1) = \Phi(n)[W_a(n)\Lambda_j(n) + U_j(n)]$$
$$e(n) = d(n) - Cx(n)$$
$$\Delta w_j(n) = \eta C \Lambda_j(n) e(n)$$

### C. Probabilistic Neural Network (PNN) training:

A PNN has three layers of nodes and can be extended to classify any number of patterns with a probabilistic linkage. The PNN is a Bayes Parzen classifier (Masters, 1995) and was first introduced by Specht (1990), who showed how the Bayes Parzen classifier could be broken up into a large number of simple processes implemented in a MLP each of which could be run independently in parallel [17] [18]. Because of ease of training and a sound statistical foundation in Bayesian estimation theory, PNN has become an effective tool for solving many classification problems [19] [20] [21] [22] [23]. However there are issues related to the size, locations of pattern layer neurons as well as the value of the smoothing parameter of the PNN. With size of the PNN, higher computational capability is related. There are several solutions suggested for these problems [24] [25] [26] [27] [28]. These methods can roughly be considered to be unsupervised methods for PNN training. A method using supervised approach is suggested in [29] which is applied for pattern recognition purposes.

A PNN structure is shown in Figure 6. The input layer contains N nodes have one value for each of the N input of a feature vector. No computation is carried out by the input layer. It simply distributes the input to the neurons in the pattern layer. The responses of the hidden nodes are added into one of K classes each of which is related to a Gaussian function. All of the Gaussian values for Class k are summed and the sum forms a probability density function (pdf).

If the input layer receives an input x, the neuron $x_{ij}$ of the pattern layer computes the output as

$$\phi_{ij}(x) = \frac{1}{(2\pi)^{\frac{d}{2}}\sigma^d} exp[-\frac{(x - x_{ij})^T(x - x_{ij})}{2\sigma^2}] \qquad (26)$$

where d denotes the dimension of the pattern vector x, $\sigma$ is the smoothing parameter and $x_{ij}$ is the neuron vector. The summation layer neurons compute the maximum likelihood of pattern x being classified into class $C_i$ the response of which can be expressed as

$$p_i(x) = \frac{1}{(2\pi)^{\frac{d}{2}}\sigma^d} \frac{1}{N_i} \sum_{j=1}^{N_i} exp[-\frac{(x - x_{ij})^T(x - x_{ij})}{2\sigma^2}] \quad (27)$$

where $N_i$ denotes the total number of samples in class $C_i$. The decision layer classifies the input pattern in accordance with the Bayess decision rule for estimated class $\hat{C}(x)$ of pattern x as

$$\hat{C}(x) = arg \quad max\{p_i(x)\}, \qquad i = 1, 2......., m \qquad (28)$$

### D. Committee Machines using ANN:

The fundamental considerations governing the working and parameter selection of the cooperative ANNs or committee machines can be explained using the following analysis [16] [30]:

Let a training set of m input - output pairs be $(x^1, t_1), (x^2, t_2), ...(x^m, t_m)$ be given and N networks are trained using this set of data. For simplicity, let for n-dimensional input there be a single output. Let for network functions $f_i$ for a number of networks represented by indices $i = 1, 2, ...N$, the cooperative or committee network formed

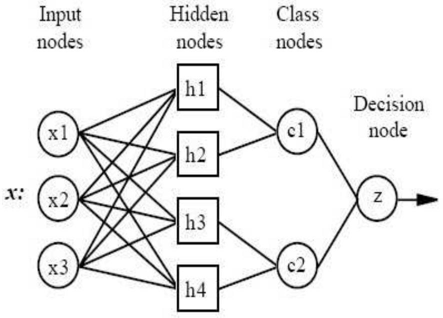

Fig. 6. Probabilistic Neural Network

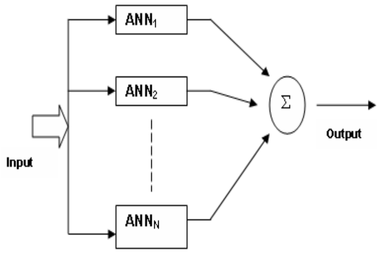

Fig. 7. ANN Committee Machines

generates as output given as

$$f = \frac{1}{N}\sum_{i=1}^{N} f_i \tag{29}$$

The rationale behind the use of the averaging in the output of the cooperative or committee network as given by Eq. 29 is the fact that if one of the constituent networks in the ensemble is biased to some part of the input samples, the ensemble average can scale down the prediction error considerably [30]. A quadratic error function can be computed from each of the error vectors $e_i$ using the ensemble function f as

$$Q = \sum_{i=1}^{m}[t_i - \frac{1}{N}\sum_{i=1}^{N} f_i]^2 \tag{30}$$

By simplifying the quadratic error function, optimal weight set can be calculated as

$$w = \frac{1(EE^T)^{-1}}{1(EE^T)^{-1}1^T} \tag{31}$$

assuming that the denominator does not vanish. This method, however, is dependent on the constraint that $EE^T$ is not ill-conditioned.

TABLE I
IMPROVEMENT OF MCS PERFORMANCE DUE TO THE USE OF SOM DATA MAPPING BLOCKS AT THE INPUT

| Case | Epochs | Time in Sec.s | Success in Rate in % | Difference in time in sec.s | Difference in success rate in % |
|---|---|---|---|---|---|
| Without | 5000 | 89.3 | 92.1 | - | - |
| SOM at | 10000 | 108.1 | 93.2 | - | - |
| Input | 15000 | 134.5 | 93.2 | - | - |
| With | 5000 | 72.3 | 93.1 | 19.0 | 1.1 |
| SOM at | 10000 | 93.1 | 95.1 | 13.9 | 2.03 |
| Input | 15000 | 112.3 | 95.1 | 16.5 | 2.03 |

## II. EXPERIMENTAL DETAILS AND RESULTS

Experiments are carried out using density values taken from detector readings spread around a radius of 100 meters. The cores are assumed to be concentrated in an arc of 50 meter radius thus providing a set-up for derivation of density values using the NKG function [3]. These values are related to shower size, primary energy and the coordinates of the location where the event is assumed to have occurred.

The SOM data mappers at the input perform a process through which only 25% of the samples supplied by the detectors are retained. These are data values which are least correlated and can be considered to be provided by 100 detectors spread in a circular arc. In the true sense, the actual training data size comes from about 400 detectors which have high correlation. The highly correlated data can lead to inefficiency, hence the four SOM data mappers are used which reduce the size of the input samples. The presence of the four SOM data mappers helps in improving efficiency of the system as depicted by Table I. Experiments are carried out in several phases. The first phase deals with the training of each of the ANNs forming the MCS. The next phase is related to the optimization by the SOM of the results provided by the MLP, RNN and the PNN. The subsequent stage is to train the Committee Machine architecture formed by ten MLP blocks. The results derived are compared and $\chi^2$ - distributions generated for the success rates for the prediction of primary energy and core location with increasing epochs. The experimental stages are explained below and the results depicted in the following sections.

### *A. Configuring and training Multi Layer Perceptron for Prediction of Primary Energy and Core Location:*

The application of the MLP considers two aspects. First is the choice of the hidden layer and second is the combination of activation functions. A MLP is constituted with one hidden and one each of input and output layers. Trial and error method is used to find the best suitable hidden layer configuration. For this case several sizes of the hidden layer lengths have been considered. Table II shows the performance obtained during training by varying the size of the hidden layer. Two and three hidden layer MLPs are also tested but these provide no superior performance at the cost of higher computational complexity which was observed during the experiment. The case where the size of the hidden layer taken to be 1.5 times to that of the input layer is found to be computationally efficient. Its MSE convergence rate and learning ability is found to be superior to the rest of the cases. Hence, the size of the hidden

TABLE II
PERFORMANCE VARIATION AFTER 2500 EPOCHS DURING TRAINING OF A MLP WITH VARIATION OF SIZE OF THE HIDDEN LAYER

| Case | Size of hidden layer (x input layer) | MSE Attained | Precision attained in % |
|---|---|---|---|
| 1 | 0.75 | 1.2 x $10^{-3}$ | 87.1 |
| 2 | 1.0 | 0.56 x $10^{-3}$ | 87.8 |
| 3 | 1.25 | 0.8 x $10^{-4}$ | 87.1 |
| 4 | 1.5 | 0.3 x $10^{-4}$ | 90.1 |
| 5 | 1.75 | 0.6 x $10^{-4}$ | 89.2 |
| 6 | 2 | 0.7 x $10^{-4}$ | 89.8 |

TABLE III
EFFECT ON AVERAGE MSE CONVERGENCE AFTER 2500 EPOCHS WITH VARIATION OF ACTIVATION FUNCTIONS AT INPUT, HIDDEN AND OUTPUT LAYERS

| Case | Input layer | Hidden Layer | Output Layer | MSE x $10^{-4}$ |
|---|---|---|---|---|
| 1 | log-sigmoid | log-sigmoid | log-sigmoid | 1.45 |
| 2 | tan-sigmoid | tan-sigmoid | tan-sigmoid | 1.32 |
| 3 | tan-sigmoid | log-sigmoid | tan-sigmoid | 1.05 |
| 4 | log-sigmoid | tan-sigmoid | log-sigmoid | 1.02 |
| 5 | log-sigmoid | log-sigmoid | tan-sigmoid | 1.15 |
| 6 | log-sigmoid | tan-sigmoid | log-sigmoid | 1.19 |

layer of the ANNs considered is 1.5 times to that of the input layer.

The selection of the activation functions of the input, hidden and output layers plays an important part in the performance of the system. A common practice can be to use a similar type of activation function in all layers. But certain combinations and alterations of activation function types carried out during training provide a way to attain better performance. Two types of MLP configurations are considered- the first type constituted by a set of similar activation functions in all layers and the other with a varied combination of activation functions in different layers. Both these two configurations are trained with gradient descend with variable learning rate and momentum back propagation (GDMALBP) algorithm as a measure of training performance standardization. The outcome of the MLP blocks vary depending upon the number of training sessions and the data used. Mean Square Error (MSE) convergence and prediction precision are used to ascertain the performance of the MLP blocks. Samples used for training includes data samples of density values with several types of noise between -3 to 30 dB.

Experimental results to determine the best training method of the MLP is based on the results shown in Table IV. From the Table IV it is seen that a three layered MLP trained with *traingdm* provides the best success rate within 12000 epochs. This set-up is taken as the MLP block forming the MCS and Committee Machines for prediction of primary energy and core location. The MLP block receives density values from 80 to 100 detectors with particle content between $10^{10.5}$ to $10^{20.5}$ eV with Moliere radius of 70 m. The cores are considered to be evenly distributed within a circle of radius 50 m centered on the middle of the array. This restriction is adopted to avoid edge effect.

TABLE IV
RESULTS DERIVED DURING TRAINING- ANN TRAINED WITH *traingd*, *traingdm*, *traingdx* AND *traingda*

| SL Num | Epochs | Success rate in % | Time in sec.s |
|---|---|---|---|
| *traingd* | 5000 | 93.8 | 25.6 |
| | 10000 | 92.2 | 35.6 |
| | 15000 | 93.9 | 68.3 |
| | 20000 | 94.1 | 200.5 |
| *traingdm* | 5000 | 93.13 | 25.1 |
| | 10000 | 93.9 | 36.8 |
| | 15000 | 93.4 | 69.1 |
| | 20000 | 94.1 | 201.8 |
| *traingdx* | 5000 | 92.8 | 26.6 |
| | 10000 | 92.2 | 38.6 |
| | 15000 | 93.9 | 68.5 |
| | 20000 | 94.4 | 207.1 |
| *traingda* | 5000 | 93.1 | 24.6 |
| | 10000 | 88.9 | 35.8 |
| | 15000 | 93.5 | 74.5 |
| | 20000 | 89.1 | 209.6 |

TABLE V
VARIATION OF THE AVERAGE TRAINING TIME OF A RNN

| Sl Num | Epoch | Time is sec.s | Success rate in % |
|---|---|---|---|
| 1 | 5000 | 22.2 | 93.4 |
| 2 | 10000 | 33.1 | 94.9 |
| 3 | 15000 | 55.3 | 94.9 |
| 4 | 20000 | 124.2 | 94.7 |

### B. RNN and PNN Configuration and Training:

The RNNs are faster but take different times to reach the desired goal. The average time taken by the RNNs to train upto 2000 epochs is shown in Table V. The RNNs formed with one hidden layer containing a mix of tan-sigmoid and log-sigmoid activation functions are trained with RTRL algorithm with Levenberg-Marquardt optimization. The RNN is designed to accept density values from 80 to 100 detectors for 20 shower events.

A PNN can be considered to be combination of Radial Basis Function (RBF) ANN and a SOM [16]. RBS-ANNs are approximating functions and can be used in combination of a SOM to create a PNN. With density values from 80 to 100 detectors the PNN block provides greater accuracy at less training session than MLP and the RNN. After the MLP, RNN and PNN blocks are ready, the MCS is set-up and the prediction performed using density values from 80 to 100 detectors. The SOM at on given instant receives three inputs from the three ANN blocks and produces the optimized output. The process is repeated for density values obtained from all the detectors for 20 shower events. The SOM, on an average, is found to provide better results between 1500 to

TABLE VI
VARIATION OF THE AVERAGE TRAINING TIME OF A PNN

| Sl Num | Epoch | Time is sec.s | Success rate in % |
|---|---|---|---|
| 1 | 5000 | 20.1 | 93.6 |
| 2 | 10000 | 24.2 | 95.1 |
| 3 | 15000 | 47.1 | 95.2 |
| 4 | 20000 | 69.4 | 95.1 |

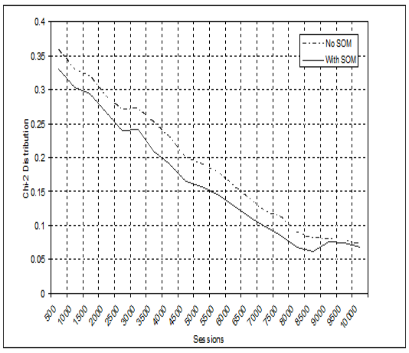

Fig. 8. Effect of the SOM optimization block

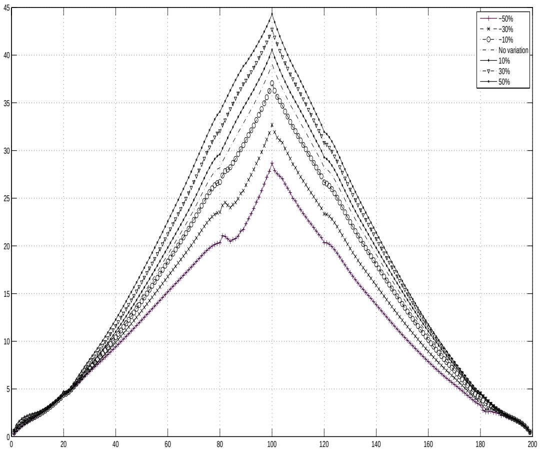

Fig. 10. Correlation plot of predicted primary energy values with variation of ±50% with ideal input

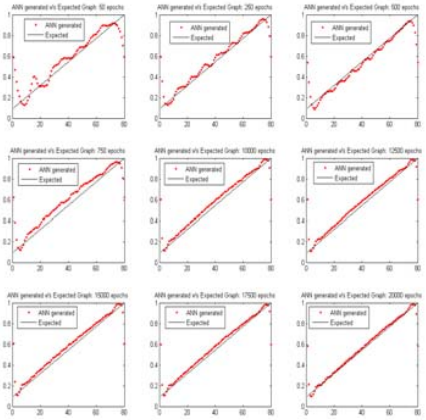

Fig. 9. Expected versus ANN generated results after 20000 epochs

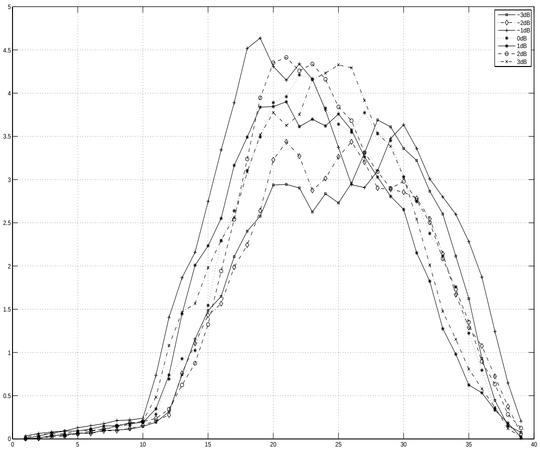

Fig. 11. Correlation plot of predicted core locations with sample variation of ±50% with ideal input

8500 training sessions. The effect of the SOM optimization block can be summarized for average results recorded for primary energy prediction and location detection as shown by Figure 8. Without the use of the MCS block, the trained MLP can be used to provide a prediction of the shower sizes for density values from 80 to 100 detectors. The results derived by the trained ANN after training it for about 20000 epochs is depicted in Figure 9. Experiments are repeated with the trained predictor block having density samples with variations upto 50% . A correlation plot for predicted primary energies between ideal and corrupted input samples (upto ±50%) is generated which is shown in Figure 10. Similarly a plot of correlation of predicted core locations for ideal and corrupted input samples (upto ±50%) is obtained as shown in Figure 11.

Figure 12 shows the location of shower events as predicted by the ANN blocks with detector positions and core positions shown. The plot is for one event of which the density values are fed to the trained ANN set-up. After training with fifty sets of data the plotted values are generated as the average of twenty sets of inputs of which half are with noise variation in the mentioned range. The results show a success rate of around 95%. The above is repeated for another event and a similar success rate is obtained.

With the committee machines, a set of experiments are carried out to predict the shower positions. Each of the twenty units of the ANN cluster is formed by cascade feed-forward networks - a variation of the MLP trained with back-propagation. The average data size for each of the block is fifty sets of $20 \times 100$ where 20 represents the number of shower cores and 100 denotes the density values recorded by the detectors. Noise between -3 dB and 3 dB are mixed to make the ANN cluster robust enough to deal with variations found from experiential works. Initially as the training is limited to a few thousand session, the event cluster is spread randomly inside and outside the fifty meter radius. It reflects the inability of the ANN cluster to make appropriate classification due to insufficient training. The expected results are a grouping inside the fifty meter arc.

As training sessions are increased with more number of samples, the predicted results start to cluster inside the intended

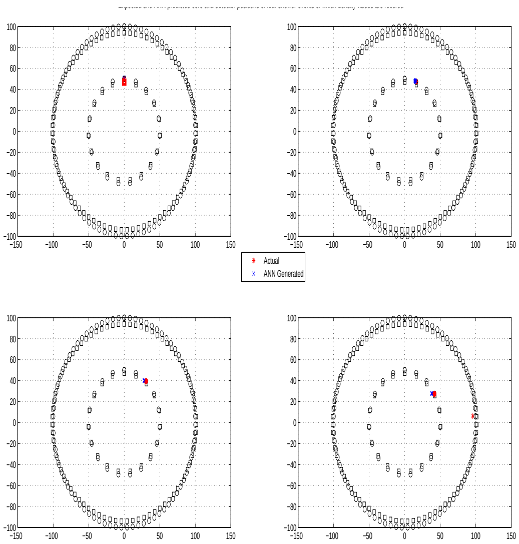

Fig. 12. Location of shower events with detector and core positions generated by the ANN set-up for one event

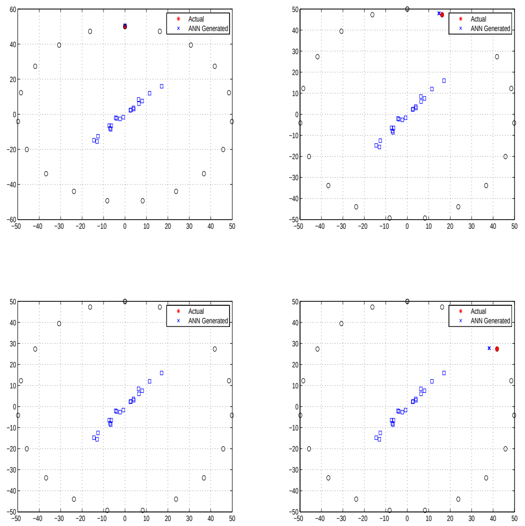

Fig. 13. Shower events of four cases predicted by ANN after 5000 sessions taking density values from 100 detectors

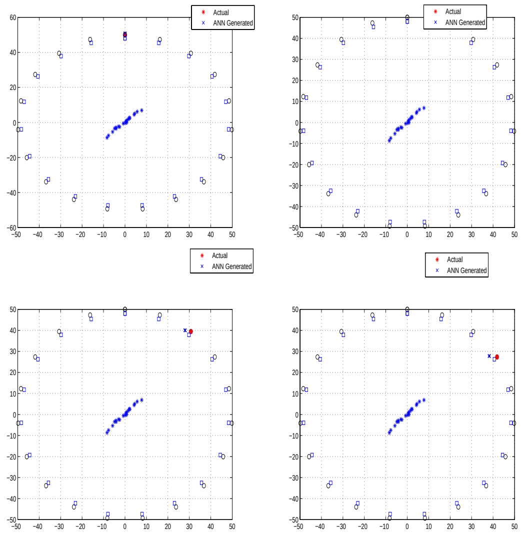

Fig. 14. Shower events of four cases predicted by ANN after 10000 sessions taking density values from 100 detectors

circle. This means that with increasing number of sessions the ANN cluster learns the given patterns more efficiently and develops the ability to make proper adjustments and predictions with a marked increase in the success rates. The consequence is better clustering of the detector and core positions as required both inside and outside the 50 meter arc. The shower events are assumed to be taking place inside the 50 meter circle with detectors recording the events from positions inside and outside the 50 meter arc. Figure 13 show a grouping generated by the ANN - cluster after 5000 sessions of training. The grouping clearly shows the location of shower events generated using density values placed inside the circle. A better clustering of the events recorded after 10,000 iterations is shown by Figure 14. The number of training sessions have been extended to 20,000 also but the best results are obtained around the 10,000 to 12,000 mark. Hence, testing results are derived from the ANN cluster trained upto this limit. The results derived for core size prediction and location detection performed by the MCS and the committee machine blocks can be summarized by the Figure 15. The advantage of the MCS system for prediction of primary energy and core location is obvious though the Committee Machines also provide satisfactory performance.

## III. Conclusion

The work shows the use of MCS and Committee Machines for a combined prediction of primary energy and core location of EAS. MCS with heterogeneous ANN blocks have proved to be effective for such applications. Committee Machines are also useful but the former proves to be superior at the cost of greater computational and implementation complexity.

Since the data involved is considerable, therefore effective means of reducing the size of the input samples with lower correlation is one of the considerations which needs to be looked into while configuring such systems. The work provides an insight into that aspect also and shows how SOM blocks can be used to reduce size of the input data. MLP, RNN and PNN blocks are ANN architectures with individual uniqueness which provides a heterogeneous cooperative environment for MCS training and testing for EAS related analysis. MLPs are simple and reliable non-parametric prediction tools suitable for pattern matching and prediction application which for primary energy prediction and core location determination proves to be effective. Its relatively slow speed is supplemented by the RNN which can even track time varying properties of the EAS events for which the proposed MCS can be modified for real time applications. The PNN strengthens and reinforces the decision making capacity of the MCS due to its ability to provide better success rate using its statistical foundations

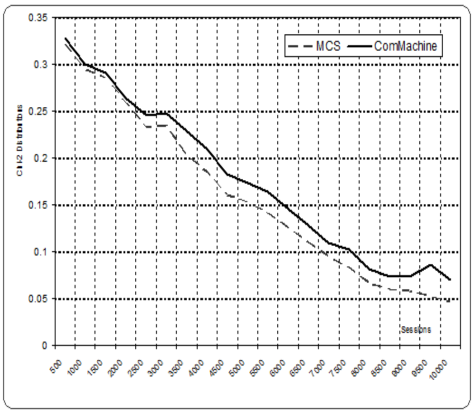

Fig. 15. Average $\chi^2$ - distribution for core size prediction and location detection performed by the MCS and the committee machine blocks

based on Bayesian estimation principles.
The ability and efficiency of the MCS is further enhanced by the use of the SOM optimization block at the output. The result is a system suitable for applications for primary energy and core location detection proving to be effective for large scale applications and demonstrating superior performance capacity than Committee Machines formed by homogeneous ANN blocks.

## References

[1] R. A. Mewaldt: “Cosmic Rays”, Macmillan Encyclopedia of Physics- available at http://www.srl.caltech.edu/personnel/dick/cos-encyc.html, 1996.

[2] R. Engel: “Theory and Phenomenology of Extensive Air Showers”, Forschungszentrum Karlsruhe, Germany- available at http://moriond.in2p3.fr/J05/trans/sunday/engel1.pdf, 2005.

[3] D. Hanna, “Application of Neural Nets to Extensive Air-Showers”, Proceedings of the 22nd International Cosmic Ray Conference, IUPAP, Volume 4, p: 500, 1991.

[4] J. C. Perrett and J T P M van Stekelenborg, “The applications of neural networks in the core location analysis of extensive air showers”, Bartol Research Institute, University of Delaware, 217 Sharp Laboratory, Newark, DE 19716, USA J. Phys. G: Nucl. Pert. Phys. 17, pp:1291-1302, 1991.

[5] H. J. Mayer, “A neural network algorithm for core location analysis at large extended air shower arrays”, Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 317, Issues 1-2, pp. 339-345, June, 1992.

[6] P. Sommers: “Extensive Air Showers and Measurement Techniques”, Physics - Astrophysics, C. R. Acad. Sci. Paris, t. 4, Serie IV, pp. 1 - 12, 2004.

[7] T. Wibig, “The Artificial Neural Networks as a tool for analysis of the individual Extensive Air Showers data”, Preprint submitted to Elsevier Preprint (1 February 2008), available at http://arxiv.org/abs/hep-ph/9608227v1.

[8] A. Chilingarian, G. Gharagyozyan, S. Ghazaryan, G. Hovsepyan, E. Mamidjanyan, L. Melkumyan, V. Romakhin, A. Vardanyan and S. Sokhoyan: “Study of extensive air showers and primary energy spectra by MAKET-ANI detector on mountain Aragats”, Elsavier Journal of Astroparticle Physics, vol. 28, pp. 5871, 2007.

[9] L. Chen and M. S. Kamel: “Design of Multiple Classifier Systems for Time Series Data ”, Spinger Lecture Notes in Computer Science, 2005, vol. 3541, pp. 216-225, 2005.

[10] M. T. El-Melegy and S. M. Ahmed: “Neural Networks in Multiple Classifier Systems for Remote-Sensing Image Classification ”, Spinger Studies in Fuzziness and Soft Computing, vol. 210, pp. 65-94, 2007.

[11] Z. Suraj, N. E. Gayar, P. Delimata: “A Rough Set Approach to Multiple Classifier Systems”, Fundamenta Informaticae, vol. 72, no. 1-3, pp. 393-406, 2006.

[12] F. Roli: “Mini Tutorial on Multiple Classifier Systems”, University of Cagliari, Dept. of Electrical and Electronics Eng., Italy, School on the Analysis of Patterns, 2009.

[13] M. Sharma and R. Mammone: “Speech recognition using sub-word neural tree network models and multiple classifier fusion”, CAIP Center, Rutgers University, Piscataway, NJ, USA, 1995.

[14] M. Sharma and R. Mammone: “Artificial neural network fusion: Application to Arabic words recognition”, Proceedings of ESANN - 2005- European Symposium on Artificial Neural Networks Bruges, (pp. 151 - 156), Belgium, 27-29 April, 2005.

[15] B. Yegnanarayana, *Artificial Neural Networks*, $1^{st}$ Ed., PHI, New Delhi, 2003.

[16] S. Haykin, *Neural Networks A Comprehensive Foundation*, $2^{nd}$ Ed., Pearson Education, New Delhi, 2003.

[17] M. Hajmeer and I. Basheer: “A probabilistic neural network approach for modeling and classification of bacterial growth/no-growth data”, Elsavier Journal of Microbiological Methods, vol. 51, pp. 217 226, 2002.

[18] “Probabilistic Neural Network Tutorial”, available at http://www.cse.unr.edu/ looney/cs773b/PNNtutorial.pdf.

[19] C. Kramer, B. Mckay, and J. Belina: “Probabilistic neural network array architecture for ECG classification,” Proc. Annu. Int. Conf. IEEE Eng. Medicine Biol., vol. 17, pp. 807808, 1995.

[20] M. T. Musavi, K. H. Chan, D. M. Hummels, and K. Kalantri: “On the generalization ability of neural-network classifier,” IEEE Transactions on Pattern Anal. Machine Intell., vol. 16, no. 6, pp. 659663, 1994.

[21] R. D. Romero, D. S. Touretzky, and G. H. Thibadeau: “Optical Chinese character recognition using probabilistic neural networks,” Pattern Recognit., vol. 3, no. 8, pp. 12791292, 1997.

[22] D. F. Specht: “Probabilistic neural networks,” Neural Networks, vol. 3, no. 1, pp. 109118, 1990.

[23] Y. N. Sun, M. H. Horng, X. Z. Lin, and J. Y. Wang: “Ultrasonic image analysis for liver diagnosis-a-noninvasive alternative to determine liver disease,” IEEE Eng. Med. Biol. Mag., vol. 15, no. 1, pp. 93101, 1996.

[24] P. Burrascano: “Learning vector quantization for the probabilistic neural network,” IEEE Transactions on Neural Networks, vol. 2, pp. 458461, July 1991.

[25] P. P. Raghu and B. Yegnanarayana: “Supervised texture classification using a probabilistic neural network and constraint satisfaction model,” IEEE Transactions on Neural Networks, vol. 9, pp. 516522, May 1998.

[26] “Enhancements to the probabilistic neural networks,” Proc. of IEEE Int. Joint Conf. Neural Networks, Baltimore, MD, pp. 761768, 1992.

[27] H. G. C. Traven: “A neural-network approach to statistical pattern classification by semiparametric estimation of a probability density functions,” IEEE Transactions on Neural Networks, vol. 2, pp. 366377, 1991.

[28] A. Zaknich: “A vector quantization reduction method for the probabilistic neural network,” Proc. IEEE Int. Conf. Neural Networks, Piscataway, NJ, pp. 11171120, 1997.

[29] K. Z. Mao, K. C. Tan, and W. Ser: “Probabilistic Neural-Network Structure Determination for Pattern Classification,” IEEE Transactions on Neural Networks, vol. 11, no. 4, July, 2000.

[30] R. Rojas, *Neural Networks-A Systematic Introduction*, Springer, Berlin, 1996.