A Survey On Energy Efficient Neural Network Based Clustering Models In Wireless Sensor Networks

C.P. Subha[1], Dr. S. Malarkan[2], K.Vaithinathan[3]
[1,2,3]Manakula Vinayagar Institute of technology, Puducherry
[1]kvecers@gmail.com

Abstract- **The performance of wireless sensor networks strongly depends on their network lifetime. As a result, Dynamic Power Management approaches with the purpose of reduction of energy consumption in sensor node, after deployment and designing of the network, have drawn attentions of many research studies. Recently, there have been a strong interest to use the intelligent tools especially neural networks in energy efficient approach of Wireless sensor networks, due to their simple parallel distributed computation, distributed storage, data robustness, auto-classification off sensor nodes and sensor reading. Dimensionality reduction and prediction of classification of sensor data obtained simply from the outputs of the neural-networks algorithms can lead to lower communication costs and energy conservation. All these characteristics are well considered in the neural network based algorithms such as ART, ART1, FUZZY ART, IVEBF and EBCS. These algorithms and their performance in improving the lifetime of the WSN are discussed in this paper.**

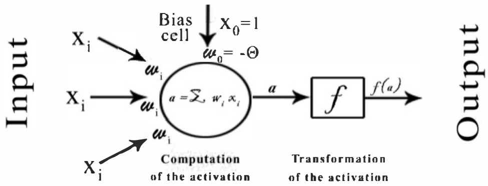
Keywords- Artificial Neural Networks (ANN), ART, ART1, Fuzzy ART, Improved Versatile elliptical basis function EBCS and SOM.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are an emerging technology which holds the potential to revolutionize everyday life. Continuous advances in semiconductor technology have enabled miniaturization of radios and mechanical structures and deployment of very cost-efficient wireless sensor nodes. A sensor node basically consists of a microcontroller (processor and memory), sensors, analog-to-digital converter (ADC), transceiver (sender and receiver) and power supply. The provision of sensors is to gather information about the physical world, for instance, temperature, light, acceleration, velocity, vibrations, pressure, magnetism and humidity. Those are analogous to organs such as eyes and ears of biological organisms. The individual sensor nodes are inherently resource constrained due to size limitation (form factor), price and power consumption. Hence, the sensor node itself is not very powerful at all but combined with others, essentially building the WSN, leads to a powerful entity as a result of synergy effects.In general Artificial Neural Networks (ANN) show characteristics such as distributed representation and processing, massive parallelism, learning and generalization ability, adaptivity, inherent contextual information processing, fault tolerance and low computation. Many of those characteristics are either inherent or desirable for WSNs as well. ANNs exchange information between neurons frequently and can be divided into two classes based on their connectivity: feedforward and feedback networks. Especially feedback networks exchange lots of information due to their iterative nature. In terms of WSNs this means that sensor nodes need to communicate with each other frequently. As a rule of thumb, the energy consumption of the transmission of one packet equals 1000 arithmetic operations. Hence, sensor nodes will be depleted faster which will reduce network lifetime significantly. Feedforward networks are better applicable since their connections are directed from input layer to output layer. It is seen that the whole sensor network as a neural network and within each sensor node there could run a neural network to decide on the output. Thus, it is possible to envision two-level ANN architecture for WSNs.Hence, the implementation of full ANNs on each single sensor node is beneficial since the discussed inherent characteristics such as parallelism and low computation are valid. Therefore, efficient neural network implementations using simple computations can replace traditional signal processing algorithms to enable sensor nodes to process data by using less resources.

II. BUILDING BLOCKS OF NEURAL NETWORKS

Neural networks are made of basic units arranged in layers. A unit collects information provided by other units (or by the external world) to which it is connected with weighted connections called synapses. These weights, called synaptic weights multiply (i.e., amplify or attenuate) the input information. A positive weight is considered excitatory, a negative weight inhibitory.



Each of these units is a simplified model of a neuron and transforms its input information into an output response. This transformation involves two steps: First, the activation of the neuron is computed as the weighted sum of it inputs, and second this activation is transformed into a response by using a transfer function. Formally, if each input is denoted xi, and each weight wi, then the activation is equal to $a = \sum X\, iWi$and the output denoted O= f(a).Any function whose domain is the real numbers can be used as a transfer function. The most popular ones are the linear function (O α a), the step function (activation values less than a given threshold are set to

0 or to −1 and the other values are set to +1), the logistic function

$$f(x) = \left[\frac{1}{1+exp\{-x\}}\right] \text{........} (1)$$

which maps the real numbers into the interval [−1 + 1] and whose derivative, needed for learning, is easily computed *{f'(x) = f(x) [1 − f(x)]}*, and the normal or Gaussian function.

$$f(x) = \left(\sigma\sqrt{2\pi}\right)^{-1} X\ exp\left\{-\frac{1}{2}(a|\sigma)^2\right\} \text{.....} (2)$$

Some of these functions can include probabilistic variations; for example, a neuron can transform its activation into the response +1 with a probability of ½ when the activation is larger than a given threshold. The architecture (i.e., the pattern of connectivity) of the network, along with the transfer functions used by the neurons and the synaptic weights, completely specify the behavior of the network [5].

III.ARTIFICIAL NEURAL NETWORKS PARADIGM IN WIRELESS SENSOR NETWORKS

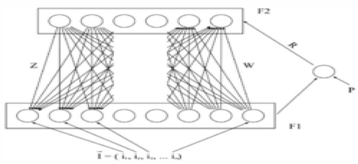
Some of the energy efficient techniques such as clustering and classification can be very well accomplished by using some of the algorithms like ART, ART1, Fuzzy ART, IVEBF and SOM developed within the artificial neural network paradigm which can be easily adopted to Wireless Sensor Networks.

ART can learn arbitrary input patterns in a stable, fast and self organizing way, thus overcoming the effect of learning in stability that plagues many other competitive networks. ART is not, as is popularly imagined, a neural network architecture. It is a learning theory, that resonance in neural circuits can trigger fast learning. As such it subsumes a large family of current and future neural networks architectures, with many variants. ART1 is the first member, which only deals with binary input pattern, although it can be extended to arbitrary input patterns by a variety of coding mechanisms. ART2 and Fuzzy ART extend the applications to analog input patterns. Fuzzy ART (FA) benefits the incorporation of fuzzy set theory and ART. Fuzzy ART maintains similar operations to ART1 and uses the fuzzy set operators, so that it can work for all real data sets. Fuzzy ART exhibits many desirable characteristics such as fast and stable learning and typical pattern detection [6].

1. ART ALGORITHM

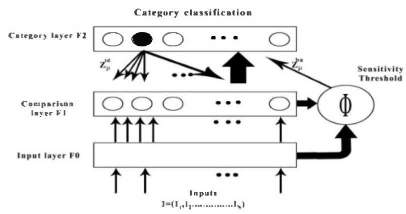
The basic ART system is an unsupervised learning model. It typically consists of a comparison field and a recognition field composed of neurons, a vigilance parameter, and a reset module. The vigilance parameter has considerable influence on the system: higher vigilance produces highly detailed memories, while lower vigilance results in more general memories. The comparison field takes an input vector and transfers it to its best match in the recognition field. Its best match is the single neuron whose set of weights (weight vector) most closely matches the input vector.

Each recognition field neuron outputs a negative signal to each of the other recognition field neurons and inhibits their output accordingly. In this way the recognition field exhibits lateral inhibition, allowing each neuron in it to represent a category to which input vectors are classified. After the input vector is classified, the reset module compares the strength of the recognition match to the vigilance parameter. If the vigilance threshold is met, training commences. Otherwise, if the match level does not meet the vigilance parameter, the firing recognition neuron is inhibited until a new input vector is applied; training commences only upon completion of a search procedure. In the search procedure, recognition neurons are disabled one by one by the reset function until the vigilance parameter is satisfied by a recognition match. If no committed recognition neuron's match meets the vigilance threshold, then an uncommitted neuron is committed and adjusted towards matching the input vector.



2. ART 1 ALGORITHM

The ART1 model is described in Fig.1. Each Ft node xi is connected to all F2 nodes yi through bottom up connection Weights Z_{ij}^{bi} , so that the input received by each F2 node yi is given by



$$Tj = \textstyle\sum_{i=1} Z_{ij}^{bi} I_i \text{......}(3)$$

Bottom up weights Z_{ij}^{bi} any real value in the interval [0,K], where

$$K = \frac{L}{L-1+N}$$

and L> 1. Layer F2 acts as Winner-Take-All network, i.e. a competitive layer for the outputs, so that all nodes yi will stay inactive, except the one that receives the largest bottom up input T. Once an F2 winning node arises a top-down template is activated through the top-down weights.Let us call this top-down template X = (X1, X2,,......XN) . The resulting vector X is given by the equation

$$Xi = Ii \sum_j Z_{ji}^{td} y_i \quad .... (4)$$

Since only one yi is active, let us call this winning F2 node Yj, so that Yj =o if j≠0 and Yj =1. In this case we can state

$$Xi = IiZ_{ji}^{td} \ or \ \ X = I \cap Z_{ji}^{td} \ \ ....(5)$$

note that only the weights of the connections touching the F2 winning node yJ are updated. This algorithm along with flowchart is described in [7].The advantage about adaptive resonance theory is that it gives the user more control over the degree of relative similarity of patterns placed on the same cluster. ART1 net achieves stability when it cannot return any patterns to previous clusters.

3. FUZZY ART NEURAL NETWORK MODEL

The general structure of the Fuzzy ART neural network is shown in Figure. It consists of two layers of neurons that are fully connected: a 2M-neuron input or comparison layer (F1) and an N neuron output or competitive layer (F2). A weight value *Zji* is associated with each connection, where the indices i and j denote the neurons that belong to the layer Ft and F2 respectively. The set of weight Z = {*zji*: i = 1,2,...,2M; j = 1,2,...,N} encodes information that defines the categories learned by the network. These can be modified dynamically during network operation. For each neuron j of F2, the vector adaptive weights zj = (zj1, zj2,….., zj2M) correspond to the subset of weights (zj - Z) connected to neuron j. This vector Z is named prototype vector or template, and it represents the set of characteristics defining the category j. Each prototype vector Z is formed by the characteristics of the input patterns to which category j has previously been assigned through winner-take-all competition.

FUZZY ART ALGORITHM

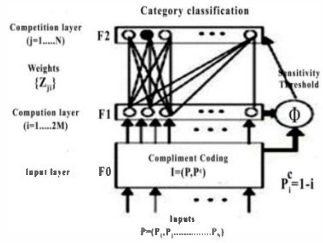
This algorithm can be described in five execution steps:

3.1 Weights and parameter initialization

Initially, all the neurons of F2 are uncommitted, and all weight values *Zji* are initialized to 1. An F2 neuron becomes committed when it is selected for an input P. Then, the corresponding weights *zji* can take values expressed by real number in the interval[0,1]. The Fuzzy ART weight vector *Zji* subsumes both the bottom-up and top-down weight vectors of ART1 [10, 11].

3.2 Input Vector Coding

When a new input vector P (P1, P2, ….., PN) of N elements. This coding is used to prevent a categoryproliferation when the weights erode.



\

3.3. Category Choice:

With each presentation of an input I to Fl, the choice function Tj(I) is calculated for each neuron j in F2:

$$Tj = \frac{|I \wedge z_{ji}|}{\alpha + |z_{ji}|} ....(6)$$

I ^ zj = (min(I1, z jl, min(I2 , zj2).... min(I2N , I j2N)) and α is a user defined choice parameter such that α >0. F2 is a winner-taking-all competitive layer, where the winner is the neuron j=J with the greatest value of activation Tj for the input I, Tj = max {Tj : j = 1,...., N}. If the same Tj value is obtained by two or more neurons, the one with the smallest index j wins. The winning neuron J is retained for steps D and E [11].

3.4 Vigilance Test

This step serves to compare the similarity between the prototype vector of the winning neuron zji and input I, against a user defined vigilance parameter ρ, through the following test:

$$\frac{|I \wedge z_J|}{|I|} \geq \rho ....(7)$$

where ρ =[0,1]. This comparison is carried out on layer Fl: the winning neuron J transmits its learned expectancy, zj, to Fl for comparison with I. If the vigilance test (Equ. 7) is passed, then neuron J becomes selected and is allowed to adapt its prototype vector as per step E. Otherwise, neuron J is deactivated for the current input I: TJ is set equal to -1for the duration of the current input presentation. The algorithm searches through the remaining F2 layerneurons (steps C and D), until some other neuron J passes the vigilance test. If no committed neuron from the F2layer can pass this test, an uncommitted neuron is selected and undergoes prototype vector update (i.e. the new class is assigned for the input)

3.5 Prototype vector update

The prototype vector of the winning neuron J is updated according to:

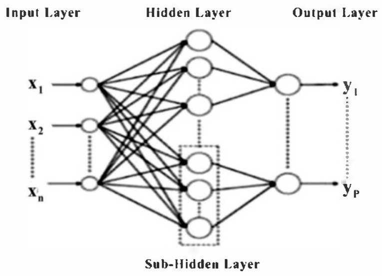
$$Z_J^{new} = \beta(I \wedge Z_j^{old}) + (1-\beta)Z_j^{old} .....(8)$$

Where β is a user defined learning rate parameter such that β = [0,1]. The algorithm can be set to slow learning, with 0< β <1. For fast learning β =1, the new weight can be updated as $Z_J^{new} = (I \wedge Z_j^{old})$

Once this update step is accomplished, the network can process a new input vector from step B. Neuro-fuzzy system based on an underlying fuzzy system is trained by means of a data-driven learning method derived from neural network theory. This heuristic only takes into account local information to cause local changes in the fundamental fuzzy system. It can be represented as a set of fuzzy rules at any time of the learning process, i.e., before, during and after.

IV. IMPROVED VERSATILE ELLIPTIC BASIS FUNCTION

The network consists of an input layer, a hidden layer, and an output layer. The structure of the IVEBF neural network is shown in Fig.



IVEBF is based on a hyper-ellipsoidal function that can be translated and rotated to cover the data set during learning process. The structure is flexible and can be adjusted during the training process. Neural network have very fast training algorithm to learn a data set in only one pass. Once a datum is learned, it is discarded. There is no need to use old data again for the future learning with new incoming data [14].

Steps for IVEBF Algorithm:

1. Create the trn/val/tst input/target subsets [trn =training/ val =validation/ tst =testing input/target] [MSE=mean square error, I=input layer, O= output, H=hidden]
2. For a each class classifier, the columns of the target matrices are all columns of the c-dimensional unit matrix eye [matlab] with the single 1 indicating the corresponding class. Inputs will be assigned to the class corresponding to the largest output.
3.Standardize the trn input and normalize the val and trn inputs with the means and co-variances from trn.
4. Now, take I-H-O(O=c) architecture and output by using single hidden layer.
5. Calculate the trn/val/tst MSEs for No. of equations (Neq), No. of weights.
6. For training to convergence, a necessary criterion for a unique solution is that either no. of equation is greater than or equal to no. of weights or hidden layers are less than or equal to how many hidden layers are sufficient (Hub) for elliptical function. Therefore, the decision function is designed such that Hub = (Neq-O)/(I+O+1).
7. To avoid non-optimal local min solutions, choose the best of many designs obtained by using multiple random weigh initializations for each candidate value of hidden layer.
8. Determine a search grid for hidden layer and the number of random weight initialization trials or test folds for each hidden layer.
9. Find the result based on the value of decision function.

EBC-SOM:

The motivation of creating EBC-S was inattention of previous clustering algorithms to energy level of the nodes as a key parameter to cluster formation of the networks. We tried to develop the classic idea for topological clustering and incorporate a topology energy based clustering method in order to approach to our main goal in WSNs, extending life time of the network with enough network coverage[15,16].

A. Algorithm Assumptions

The proposed algorithm is more like LEACH-C and LEA2C protocols. Thus the assumption about BS cluster formation tasks and energy consumptions models of normal and cluster head nodes are the same as previous. The operation of the algorithm is divided into rounds in a similar way to LEACH-C. Each round begins with a cluster setup phase, in which cluster organization takes place, followed by a data transmission phase, throughout which data from the simple nodes is transferred to the cluster heads. Each cluster head aggregates/fuses the data received from other nodes within its cluster and relays the packet to the base station.

In every cluster setup phase, Base Station has to cluster the nodes and assign appropriate roles to them. After determining the cluster heads of current round, BS sends a message containing cluster head ID for each node. If a node's cluster head ID matches its own ID, the node is a cluster head otherwise it is a normal node. BS also creates a Time Division Multiple Access (TDMA) table for each cluster and affects this table to CHs [16].

B. Cluster Setup phase

The protocol uses a two phase clustering method SOM followed by K means algorithm which had been proposed in with an exact comparison between the results of direct clustering of data and clustering of the prototype vectors of the SOM. We selected SOM for clustering because it is able to reduce dimensions of multi-dimensional input data and visualize the clusters into a map. In our application, dimensions of input data relates to the number of variables (parameters) that we need to consider for clustering. The reason for using SOM as preliminary phase is to make use of data pretreatment (dimension reduction, regrouping, visualization...) gained by SOM. Therefore the data set is first clustered using the SOM, and then, the SOM is clustered by kmeans. The variables that we want to consider as SOM input dataset is x and y coordination of every node in network space and the energy level of them. So we will have a D matrix with N-3 dimensions. Since we are applying two different type variables, first we have to normalize all values.

$$V' = \frac{v - \min_a}{(\max_a - \min_a)} \quad \text{.......(9)}$$

So by means of above equation, our dataset matrix would be:

$$D = \begin{bmatrix} \frac{xd_1}{xd_{max}} & \frac{yd_1}{yd_{max}} & \frac{E_1}{E_{max}} \\ 0 & 0 & 0 \\ 0 & 0 & \bullet \\ \frac{xd_n}{yd_{max}} & \frac{yd_1}{yd_{max}} & \frac{E_n}{E_{max}} \end{bmatrix} \quad \text{.........(10)}$$

Where D is the data sample matrix or input vectors of SOM, XD=(xd1...xdn) are X coordinates, YD=(yd1...ydn) are Y coordinates, E=(E1...En) are energy levels of all sensor nodes of the networks, xd_{max} is the maximum value for x coordinate of the network space, yd_{max} is the maximum value for Y coordinate of network space and E_{max} is the remain energy of maximum energy node of the network(at the beginning it is equal to E initial).

C. Cluster Head selection phase

Different parameters can be considered for selecting a CH in a formed cluster. In three criterions have been considered for CH selection:

1) The sensor having the maximum energy level
2) The nearest sensor to the BS
3) The nearest sensor to gravity center (centroid) of the cluster.

When we select the nearest node to BS in a cluster as CH, we insure to consume least energy to transmit the messages to BS. Also the nearest sensor to gravity center (centroid) of the cluster insure least average energy consumption for intra cluster communications while the reduction of CH overhead is not guaranteed. The results from LEA2C showed that the selecting the nodes with maximum energy level (first factor) as cluster head, gives the best results. This profit over two other criterions might be cause of having CH rotation. Because in the case of two other criterions (nearest sensor to BS or cluster centroid) the selected CHs stay fixed during the transmission phase until next re-clustering phase which may last for several rounds and it will cause the rapid depletion of that CHs, while applying these two criterions showed a longer lifetime (last dead) results. After determining the cluster head nodes, BS assign appropriate roles to all nodes .

D. Transmission phase

After formation of clusters and selecting their related cluster heads, now it's time to send data packets sensed at normal nodes to their related cluster heads and after applying data aggregation functions to received packets by CHs, send messages on to the base station. The energy consumption of all nodes is computed. After every transmission phase, we count a new round and would have a cluster head rotation (in the case of using maximum energy criterion) The best time for reclustering can be when a relative reduction occurs in energy level of nodes. So the energy level of m selected highest energy nodes are checked regularly. 20 percent depletion of initial energy for first time reclustering phase and 5 percent depletion for next times are used. When the re-clustering threshold is satisfied, BS sends a reclustering message to whole network.

Probably the advantages about SOMs that they are very easy to understand. It's very simple, if they are close together and there is area connecting them, then they are similar. If there is some variationbetween them, then they are different.

V. COMPARISION

ART network can discover structure in the data by finding how the data is clustered. The ART network are capable of developing stable cluster of arbitrary sequence of input pattern by self organization.

ART1 is a first member which only deals with binary inputs patterns by a variety of coding mechanisms. ART may be used for hierarchical clustering.

Fuzzy ART model, because of its unique ability to solve the stability-plasticity dilemma, to learn in a short period of time in the fast-learning mode, and to continually learn from new events without forgetting what has already been learned .Nevertheless, ART Models cannot detect time-related changes.

The fuzzy ART neural network builds the prototype online according to observations collected. In other words, different prototypes are kept in the fuzzy ART neural network if the data is presented in a different order. The fuzzy ART neural network has lower accuracy if the time correlation is removed from the data presented to the neural network. And also on comparison with other algorithms, on the basis of Training no. of Neurons the IVBEF approach provides good accuracy of node parameters [14].

The important difference of a SOM training algorithm with other vector quantization algorithms is that not only the best matching units (the winner neuron) but also its topological neighbors would be updated. Close observations in input space would activate two close units of the SOM. The learning phase continues until the stabilization of weight vectors.

VI. CONCLUSION

It has been observed that the improvement in network varies according to the network topology. From the survey it has been observed that ART1 is better than ART and the improvement in lifetime in ART1 is consistently around 45%. The maximum network lifetime improvement is found to be 47%. This effectively improves the bandwidth of the communication channel and also reduces the energy consumption [8].

The Fuzzy ART neural network is self-learning, processes any input sequentially, needs no buffering of samples, and adapts to both, changing environmental conditions and new evolving signals. Finally, the high compression rate lowers communications costs.

Self organized mapping which is trained on sample two dimensional data collected from various active nodes which results enormous energy saving which is around 48.5%.

This paper presented a classification for the most important applications of neural networks in energy efficiency of WSNs depend on different research studies have been done so far. The most important application of neural networks in WSNs can be summarized to sensor data prediction, sensor fusion, path discovery, sensor data classification and nodes clustering which all lead to less communication cost and energy conservation in WSNs. The classification for neural network based methods by Self Organizing Maps has been found to be providing good performance than the ART, ART1, FUZZY ART and IVBEF for the purpose of energy conservation of nodes, and also shows more applications in recent WSN platforms.

REFERENCES

[1] A Wendi B. Heinzelman, Ananth P. Chandrakasan, and HariBalakrishnan, An Application –Specific protocol Architecture for Wireless MicrosensorNetworks, IEEE Transaction on Wireless Communications, Vol.1 No. 4 October 2002.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE CommMag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] Teresa Serrano-Gotarrendona and BemabeLinaresbarranco, An ART1 Microchip and its use in Multi- ART1 systems, IEEE Transaction on Neural

[4] Networks, VOL 8, No. 5, Sep 1997.

[5] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. of National Academy of Sciences*, 2007

[6] Neuralnetworkbasedon energy efficiency in Wireless Sensor Network- A Survey , International Journal of Computer Science & Engineering Survey (IJCSES) Vol.1, No.1, August 2010 Selection of Cluster Head using Neural
Network in Wireless Sensor Network, European Journal of Scientific Research ISSN 1450-216X Vol.83 No.3 (2012), pp.320-373

[7] Sudhir G. Akojwar, R.M.Patrikar, Neural networks for Wireless Sensor Networks, IMAPS India National Conference on Microelectronics and VLSI (IINC 2005), IIT Bombay, INDIA, 19-21 Dec 2005.

[8] Gail A. Carpenter and Stephen Grossberg, A Self- Organizing Neural Network For Supervised Learning, Recognition, and Prediction, IEEE Communications Magazine, Sep 1992

[9] J.A.Freeman, David M. Skapura, "Neural networks Algorithms, applications, and programming techniques", Pearson Education Asia, 2001.

[10] G. Carpenter, S. Grossberg, D. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," Neural Networks, Vol. 4, pp. 759-771, 1991.

[11] Novel technique for signal classification of Neural networks, UACEE International Journal of Advances in Electronics Engineering Volume 2: Issue 2 ISSN 2278 - 215X

C.P. Subha was born in 1973 in Coimbatore, India. Currently working as Associate Professor in ManakuulaVinayagar Institute of Technology, Puducherry since 2008. She obtained her B.E. degree from Government College of Technology, Bharathiyar University, Coimbatore in 1995 and M.E. degree in Applied Electronics in Vinayaga Missions, Salem in 2007. She has over 15 years of experience in teaching. She has published and presented papers in various International and National Conferences. She is lifetime member of ISTE , IETE and IEEE. Her area of interest includes mobile communication and wireless sensor networks.

Dr.S. Malarkkanis working as principal in ManakulaVinayagar Institute of Technology, Puducherry . He obtained his B.E. degree from Thiagarajar College of Engineering, Madurai in 1988 and M.E in Optical Communication and Ph.D in Wireless & Mobile Communication from college of Engineering, Anna University, Chennai. He has over 23 years of experience in teaching. He has published and presented more than 55 papers in various International and national journals and conferences. He was the principal investigator for 2 AICTE funded projects worth Rs.17 lakhs and completed successfully. He is a life time member of ISTE, CSI and fellow of IETE. He was serving as a member in IETE Executive committee, Chennai zone from 2006 to 2008. He is the coordinator for the education sector of CII, Puducherry and making contribution towards strengthening of Industry and Institute interaction. His area of interest includes analog and digital communication, mobile communication and wireless networks.

K.Vaithinathan currently pursing Bachelors of Technology degree in Electronics and Communication Engineering in ManakulaVinayagar Institute of Technology, Pondicherry University, Pondicherry, India from 2009 to 2013. He has published5 papers in both International and National conferences. His areas of interest include Wireless sensor networks and Signal processing applications in various fields.