Vol 8. No. 1 – March, 2015
African Journal of Computing & ICT

www.ajocict.net

# Energy Efficient Homogeneous Wireless Sensor Network Using Self-Organizing Map (SOM) Neural Networks

**M. Mittal & K. Kumar**
Department of Computer Science
Gurukul Kangri University
Haridwar , India
mittal.mohit02@gmail.com, kumar_krishana@yahoo.com
Phones: +918394832967 & +919319777952

**ABSTRACT**

Today, Wireless Sensor Network (WSN) is becoming an interesting research area for wireless communication in very harsh or hostile environment. In WSN, limited battery power is considered as the main constraint; due to which the network lifetime is very low. To overcome this problem, many types of improvement have been carried out in both hardware and software levels. But still, there is a much more need to improve. Data clustering or classification using artificial neural networks (ANN), an emerging area of artificial intelligence is a step forward to enhance the network's lifetime by means of optimizing some of its parameters like power battery backup, data traffic, end-to-end delay. Now-a-days, ANN has become one of the most popular techniques for solving real time optimization problems. In this paper, Kohonen's Self-Organization Map (SOM) neural network algorithm has been efficiently used for data clustering; that learns to classify data without any supervision i.e. in unsupervised learning mode. We have analyzed and reduced the real data to make the network less bulky, communication gets faster as due to lager volume of data is get reduced, and end-to-end delay and power consumption of communication network also gets lowered.

**Keywords** - Wireless sensor network (WSN); artificial neural network (ANN); self-organizing map (SOM); energy efficiency.

**African Journal of Computing & ICT Reference Format:**
M. Mittal & K. Kumar (2015) Energy Efficient Homogeneous Wireless Sensor Network Using Self-Organizing Map (SOM) Neural Networks. Afr J. of Comp & ICTs. Vol 8, No. 1. Pp 179-184.

## 1. INTRODUCTION

Wireless Sensor Network was developed for various real time applications such as battlefield surveillance, industrial process monitoring and control, health monitoring, animal tracking, environment and habitat monitoring, home automation etc. It consists of large number of distributed homogeneous tiny sensor nodes to cooperatively monitor physical or environmental conditions such as humidity, temperature, sound, pressure, pollution etc; their implementation became possible due to recent developments in micro-electro-mechanical systems (MEMS) and state of the art communication electronics [2]. These environmental parameters are sensed by these tiny nodes in the form of sensed data and forwarded to base station (BS). The base stations are also acts as gateway, which allows the user to access the collected data through an infrastructure network, such as internet [3].

There are many limitations battery life, bandwidth, end-to-end delay, data traffic, latency etc which restricts the communication. Among these limitations battery life is one of the major constraints because these sensor networks are mostly deployed in very harsh or hostile environment to which it is very difficult to change their batteries. So, to overcome this critical issue, we have to work on routing techniques to prolong the overall network lifetime.

The data processing consumes large amount battery power of the sensor nodes. Artificial neural network works efficiently on this problem. Self Organizing Map neural network create clusters of data given to the input. Clustering helps in reducing the large amount of data. So, by this clustering technique the network lifetime is improved.

There are some technical requirements of a WSN applied to environmental monitoring include:

- Autonomy: batteries must be able to power the nodes during the whole network lifetime. As the radio transceiver accounts for most of the power consumption in a node, the network has to reduce data traffic as much as possible, as well as the number of hops required to send a message.
- Robustness: In this kind of application, human maintenance is usually difficult because of the hardness of the terrain. Therefore it is important to design robust software and hardware that can be adapted to any incident.
- Flexibility: the network must be able to add, move or remove nodes to meet the applications requirements. The network must automatically detect the changes, organizing the communication in consequence.

## 2. ARTIFICIALNEURAL NETWORK

The human brain, which possesses an extraordinary ability to learn, memorize and generalize, is a dense network of over 10 billion neurons, each connected on average to about 10,000 other neurons which is called synapses. Each neuron receives signals through synapses, which control the effects of the signals on the neuron. These synaptic connections play an important role in the behavior of the brain. These findings have inspired modeling of biological neural systems by means of NNs [15]. From the human brain, the computer scientist's creates artificial neural network which is used to solve real time problems. Learning is the process in which the weights of a NN are updated in order to discover patterns or features in the input data. Learning methods are generally classified into the two types: i) supervised learning and ii) unsupervised learning. In supervised learning, a teacher presents an input pattern and the corresponding target output. Network weights are adapted in such a way that the error is minimized. The objective of unsupervised learning is to discover patterns in the input data with no help from a teacher [14].

### 2.1 Self Organizing Map Neural Network

Kohonen Self-Organizing Maps (or just Self-Organizing Maps, or or SOMs), are a type of neural network. They were developed in 1982 by Tuevo Kohonen, a professor emeritus of the Academy of Finland. Self-Organizing Maps are aptly named. "Self-Organizing" is because no supervision is required. SOMs learn by their own unsupervised competitive learning. "Maps" is because they attempt to *map* their weights to conform to the given input data. The nodes in a SOM network attempt to become like the inputs presented to them. In this sense, this is how they learn. They can also be called "Feature Maps", as in Self-Organizing Feature Maps. Retaining principle 'features' of the input data is a fundamental principle of SOMs, and one of the things that makes them so valuable. Specifically, the topological relationships between input data are preserved when mapped to a SOM network [7].

### 2.2 Architecture

Each node in the SOM is mapped to neuron in the neural network. The architecture of SOM is shown in the "Fig.1". The neighborhood of the radii R=2, 1 and 0 are shown in the "Fig.2" for a rectangular grid and in "Fig. 3" for hexagonal grid. In each illustration, the winning unit is indicated by the symbol "#" and the other units are denoted by "*". Note that each unit has eight nearest neighbors in the rectangular grid, but only six in the hexagonal grid. Winning units that are close to the edge of the grid will have some neighborhoods that have fewer units than that shown in the respective figure [8].

Alternative structures are possible for reducing R and α.The learning rate α is a slowly decreasing function of time or training epochs. Kohonen indicates that a linearly decreasing function is satisfactory for practical computations; a geometric decrease would produce similar results. The radius of the neighborhood around the cluster unit also decreases as the clustering process progresses [8].

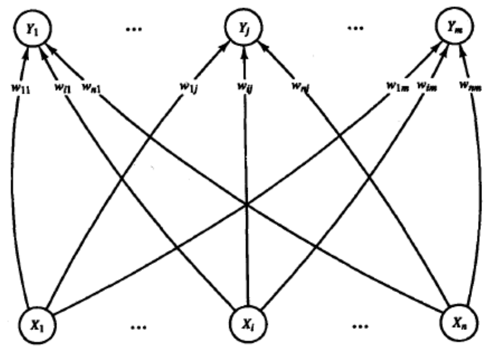


**Figure 1. Kohonen self-organizing map [8]**

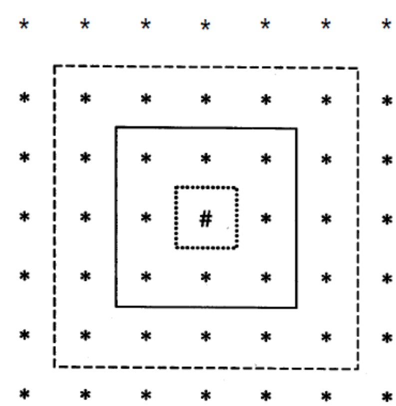

**Figure. 2. Neighborhood for rectangular grid [8]**

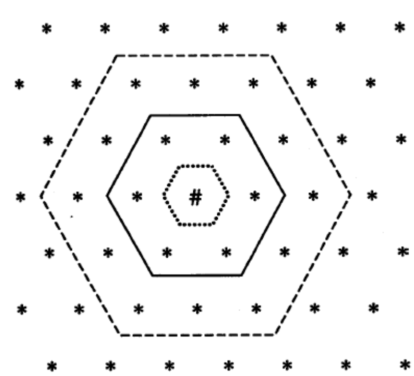

**Figure. 3. Neighborhoods for hexagonal grid [8]**

www.ajocict.net

### 2.3 Algorithm

Step 0. Initialize weights $w_{ij}$.(possible choices are discussed below). Set topological neighboring parameters. Set learning rate parameters.

Step 1. While stopping condition is false, do step 2 to 8.

Step 2. For each input vector x, do Steps 3 to 5.

Step 3. For each j, compute: $D(j) = \sum(w_{ij} - x_i)^2$

Step 4. Find index J such that D (J) is a minimum.

Step 5. For all units j within a specified neighborhood of J, and for all I: $W_{ij}(new) = w_{ij}(old) + \alpha [x_i - w_{ij}(old)]$.

Step 6. Update learning rate.

Step 7. Reduce radius of topological neighborhood at specified times.

Step8. Usually the stopping criterion is a fixed number of iterations or till radius becomes zero or the weight matrix reduces to a very negligible value.

## 3. RESULT ANALYSIS

The simulation is done in MATLAB and Viscovery SOMine5. Humidity data values are taken as parameter for analysis. We are considering as 15 sensor nodes as input in SOM neural network. This data contain 100 values by each sensor nodes. Some of the input values are shown in TABLE I. We have created 4 clusters in the output layer. In Figure 4, Different clusters are shown cluster 1, cluster 2, cluster 3 and cluster 4 (simulation after 20 iterations). Figure 4. Represents clusters after 100 iterations. Correlation between sensor nodes is shown in TABLE II. PCA values and eigen values are shown in TABLE III.

**Table 1: Sensors Input Values**

| s_1 | s_2 | s_3 | s_4 | s_5 | s_6 | s_7 | s_8 | s_9 | s_10 | s_11 | s_12 | s_13 | s_14 | s_15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 45.93 | 48.09 | 35.3 | 37.16 | 40.93 | 32.09 | 30.3 | 33.16 | 49.93 | 32.09 | 35.3 | 37.16 | 33.16 | 46.1 | 33.09 |
| 45.9 | 48.55 | 35.33 | 37.16 | 40.9 | 32.55 | 30.33 | 33.16 | 50.9 | 32.55 | 35.33 | 37.16 | 33.16 | 46.2 | 33.02 |
| 45.9 | 48.61 | 35.23 | 37.09 | 40.9 | 32.61 | 30.23 | 33.09 | 50.9 | 32.61 | 35.23 | 37.09 | 33.09 | 46.26 | 33.89 |
| 45.93 | 48.71 | 35.16 | 37.02 | 40.93 | 32.71 | 30.16 | 33.02 | 50.93 | 32.71 | 35.16 | 37.02 | 33.02 | 46.3 | 33.82 |
| 45.93 | 48.71 | 35.09 | 36.89 | 40.93 | 32.71 | 30.09 | 33.89 | 50.93 | 32.71 | 35.09 | 36.89 | 33.89 | 46.23 | 33.82 |
| 45.9 | 48.58 | 35.09 | 36.82 | 40.9 | 32.58 | 30.09 | 33.82 | 50.9 | 32.58 | 35.09 | 36.82 | 33.82 | 46.16 | 33.78 |
| 45.9 | 48.38 | 35.06 | 36.82 | 40.9 | 32.38 | 30.06 | 33.82 | 50.9 | 32.38 | 35.06 | 36.82 | 33.82 | 46.13 | 33.78 |
| 45.97 | 48.25 | 35.02 | 36.78 | 40.97 | 32.25 | 30.02 | 33.78 | 50.97 | 32.25 | 35.02 | 36.78 | 33.78 | 46.1 | 33.68 |
| 46 | 48.16 | 35.02 | 36.78 | 40 | 32.16 | 30.02 | 33.78 | 49 | 32.16 | 35.02 | 36.78 | 33.78 | 46.1 | 33.58 |

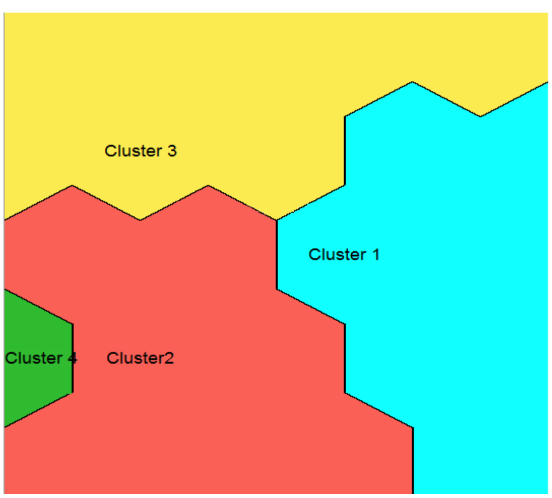


**Figure. 4. Clustering after 20 iterations**

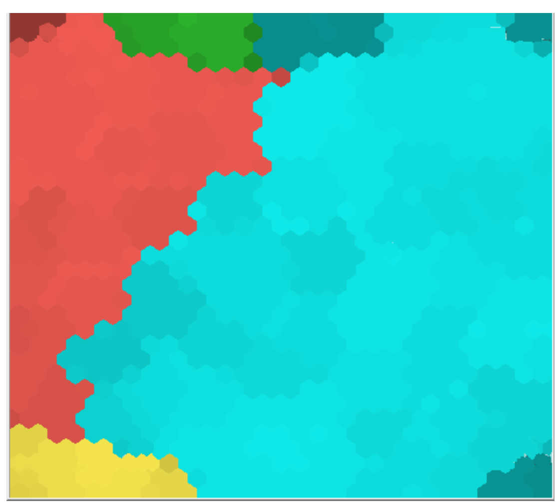

**Figure. 5. Clustering after 100 iterations**

African Journal of Computing & ICT



www.ajocict.net

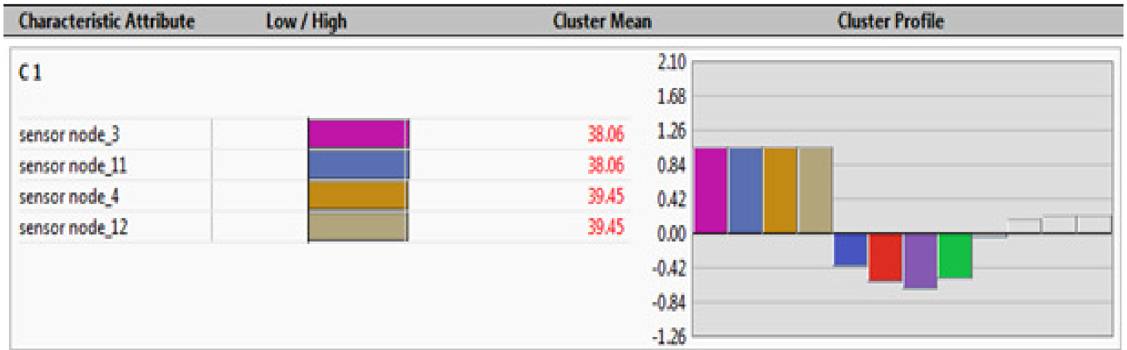


**Figure. 6 a. Cluster 1 (c1) consists 4 sensor nodes (sensor node_3, sensor node_11, sensor node_4, sensor node_12).**

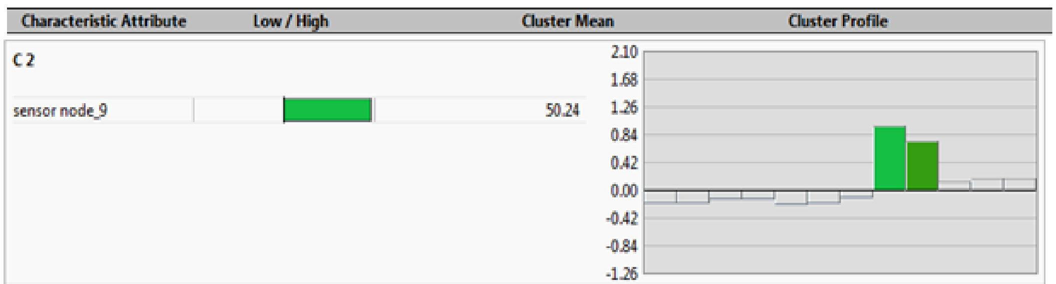


**Figure. 6 b. Cluster 2 (c2) consists 1 sensor node (sensor node_9)**

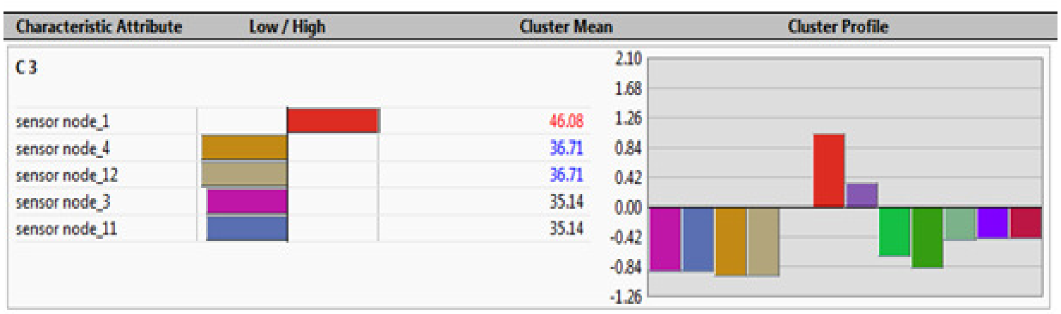


**Figure. 6 c. Cluster 3 (c3) consists 5 sensor nodes (sensor node_1, sensor node_4, sensor node_12, sensor node_3 and sensor node_11).**

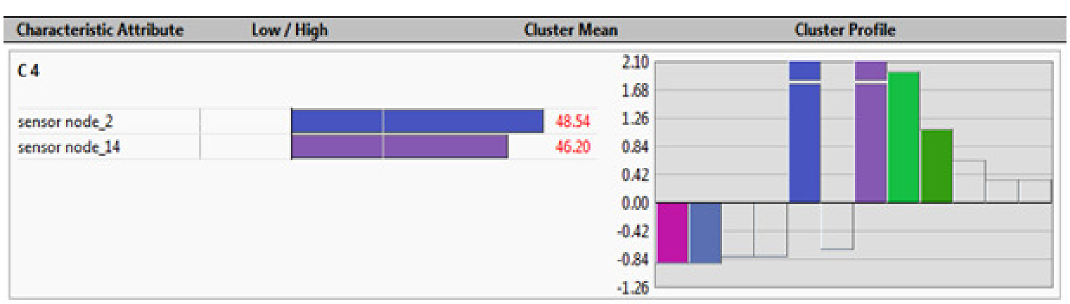


**Figure. 6 d. Cluster 4 (c4) consists 2 sensor node (sensor node_2, sensor node_14)**

Vol 8. No. 1 – March, 2015
African Journal of Computing & ICT

www.ajocict.net

**Table 2: Correlation Matrix Of Sensor Nodes.**

| | S_1 | S_2 | S_3 | S_4 | S_5 | S_6 | S_7 | S_8 | S_9 | S_10 | S_11 | S_12 | S_13 | S_14 | S_15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **S_1** | 0 | -0.0661 | -0.6722 | -0.6816 | 0.7648 | -0.2046 | 0.4439 | 0.3016 | 0.3111 | -0.2046 | -0.6722 | -0.6816 | 0.3016 | 0.1766 | 0.3183 |
| **S_2** | -0.0661 | 0 | 0.1987 | 0.2039 | -0.0205 | -0.1653 | 0.1149 | -0.0726 | -0.0318 | -0.1653 | 0.1987 | 0.2039 | -0.0726 | -0.283 | 0.1808 |
| **S_3** | -0.6722 | 0.1987 | 0 | 0.9958 | -0.0918 | 0.2274 | -0.3261 | -0.3093 | 0.419 | 0.2274 | 1 | 0.9958 | -0.3093 | -0.2699 | -0.3062 |
| **S_4** | -0.6816 | 0.2039 | 0.9958 | 0 | -0.1229 | 0.1968 | -0.3214 | -0.278 | 0.3912 | 0.1968 | 0.9958 | 1 | -0.278 | -0.2142 | -0.2992 |
| **S_5** | 0.7648 | -0.0205 | -0.0918 | -0.1229 | 0 | -0.0169 | 0.2402 | 0.099 | 0.8038 | -0.0169 | -0.0918 | -0.1229 | 0.099 | -0.0229 | 0.0754 |
| **S _6** | -0.2046 | -0.1653 | 0.2274 | 0.1968 | -0.0169 | 0 | -0.2709 | 0.0439 | 0.1719 | 1 | 0.2274 | 0.1968 | 0.0439 | -0.2044 | 0.0294 |
| **S _7** | 0.4439 | 0.1149 | -0.3261 | -0.3214 | 0.2402 | -0.2709 | 0 | -0.1915 | -0.0951 | -0.2709 | -0.3261 | -0.3214 | -0.1915 | 0.2082 | 0.3366 |
| **S _8** | 0.3016 | -0.0726 | -0.3093 | -0.278 | 0.099 | 0.0439 | -0.1915 | 0 | 0.0465 | 0.0439 | -0.3093 | -0.278 | 1 | 0.2244 | 0.0232 |
| **S _9** | 0.3111 | -0.0318 | 0.419 | 0.3912 | 0.8038 | 0.1719 | -0.0951 | 0.0465 | 0 | 0.1719 | 0.419 | 0.3912 | 0.0465 | -0.1723 | -0.2037 |
| **S _10** | -0.2046 | -0.1653 | 0.2274 | 0.1968 | -0.0169 | 1 | -0.2709 | 0.0439 | 0.1719 | 0 | 0.2274 | 0.1968 | 0.0439 | -0.2044 | 0.0294 |
| **S _11** | -0.6722 | 0.1987 | 1 | 0.9958 | -0.0918 | 0.2274 | -0.3261 | -0.3093 | 0.419 | 0.2274 | 0 | 0.9958 | -0.3093 | -0.2699 | -0.3062 |
| **S _12** | -0.6816 | 0.2039 | 0.9958 | 1 | -0.1229 | 0.1968 | -0.3214 | -0.278 | 0.3912 | 0.1968 | 0.9958 | 0 | -0.278 | -0.2142 | -0.2992 |
| **S _13** | 0.3016 | -0.0726 | -0.3093 | -0.278 | 0.099 | 0.0439 | -0.1915 | 1 | 0.0465 | 0.0439 | -0.3093 | -0.278 | 0 | 0.2244 | 0.0232 |
| **S _14** | 0.1766 | -0.283 | -0.2699 | -0.2142 | -0.0229 | -0.2044 | 0.2082 | 0.2244 | -0.1723 | -0.2044 | -0.2699 | -0.2142 | 0.2244 | 0 | -0.0435 |
| **S _15** | 0.3183 | 0.1808 | -0.3062 | -0.2992 | 0.0754 | 0.0294 | 0.3366 | 0.0232 | -0.2037 | 0.0294 | -0.3062 | -0.2992 | 0.0232 | -0.0435 | 0 |

**Table 3: Calculation Of Pca Value And Eigenvalue**

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | PC8 | PC9 | PC10 | PC11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **S_1** | 0.7703 | 0.3615 | 0.5021 | -0.0742 | 0.0005 | 0.0169 | -0.009 | -0.0594 | -0.0333 | 0.1156 | 0.0058 |
| **S_2** | -0.1698 | -0.2326 | 0.2692 | 0.0565 | 0.7884 | 0.002 | -0.3959 | -0.2508 | 0.0144 | -0.0056 | 0.0002 |
| **S_3** | -0.9706 | 0.0513 | 0.1385 | 0.1223 | 0.0403 | -0.1192 | 0.0477 | 0.0423 | -0.0269 | 0.0027 | 0.0214 |
| **S_4** | -0.9591 | 0.0337 | 0.1177 | 0.1719 | 0.0478 | -0.1669 | 0.0537 | 0.0325 | -0.0156 | 0.0297 | -0.0195 |
| **S_5** | 0.2166 | 0.5765 | 0.7645 | -0.0581 | -0.089 | 0.0457 | 0.0063 | -0.086 | -0.1036 | -0.0681 | -0.0059 |
| **S_6** | -0.3323 | 0.5688 | -0.3998 | -0.6067 | -0.0173 | -0.1196 | -0.1509 | -0.027 | -0.0033 | 0.0039 | -0.0004 |
| **S_7** | 0.4214 | -0.2926 | 0.4767 | -0.2388 | 0.0137 | -0.4623 | -0.2494 | 0.4198 | 0.0131 | -0.0098 | -0.0003 |
| **S_8** | 0.3947 | 0.6 | -0.343 | 0.5032 | 0.2954 | -0.0938 | 0.0045 | 0.1313 | -0.0053 | -0.0052 | 0.0004 |
| **S_9** | -0.3134 | 0.6674 | 0.6382 | 0.0732 | -0.1259 | 0.0735 | 0.0388 | -0.0265 | 0.1417 | -0.0122 | 0.0007 |
| **S_10** | -0.3323 | 0.5688 | -0.3998 | -0.6067 | -0.0173 | -0.1196 | -0.1509 | -0.027 | -0.0033 | 0.0039 | -0.0004 |
| **S_11** | -0.9706 | 0.0513 | 0.1385 | 0.1223 | 0.0403 | -0.1192 | 0.0477 | 0.0423 | -0.0269 | 0.0027 | 0.0214 |
| **S_12** | -0.9591 | 0.0337 | 0.1177 | 0.1719 | 0.0478 | -0.1669 | 0.0537 | 0.0325 | -0.0156 | 0.0297 | -0.0195 |
| **S_13** | 0.3947 | 0.6 | -0.343 | 0.5032 | 0.2954 | -0.0938 | 0.0045 | 0.1313 | -0.0053 | -0.0052 | 0.0004 |
| **S_14** | 0.3489 | -0.0431 | -0.1142 | 0.3613 | -0.4327 | -0.6587 | -0.0978 | -0.3192 | 0.0101 | -0.0061 | 0.002 |
| **S_15** | 0.3569 | -0.0744 | 0.0901 | -0.4379 | 0.5147 | -0.3695 | 0.5086 | -0.0819 | 0.0159 | -0.0109 | 0.0004 |
| **Eigenvalue** | **5.4504** | **2.4302** | **2.1859** | **1.7282** | **1.2806** | **0.9222** | **0.5447** | **0.4009** | **0.0347** | **0.0203** | **0.0018** |
| **% variance** | **36.34** | **16.2** | **14.57** | **11.52** | **8.54** | **6.15** | **3.63** | **2.67** | **0.23** | **0.14** | **0.01** |
| **Cumulative % variance** | **36.34** | **52.54** | **67.11** | **78.63** | **87.17** | **93.32** | **96.95** | **99.62** | **99.85** | **99.99** | **100** |

African Journal of Computing & ICT

www.ajocict.net

## 4. CONCLUSION

In this paper, Kohonen's Self-Organization Map neural network is implemented & clustered using MATLAB and Viscovery SOMine5. Simulation is done using clustering. In simulation, we have processed 1500 values and send it to base station. In this process a lot of power is wasted. To overcome this problem, we reduced these values by doing clustering and choosing the best suitable cluster. Therefore only selected clustered values are able to process and send it to base station.

Ultimately we reduce the data traffic by which we minimize bandwidth usage and enhance the battery power. In this way, communication gets faster due to lager volume of data is get reduced. Hence end-to-end delay and power consumption of communication network gets lowered. Finally we can get an energy efficient wireless network.

## REFERENCES

[1] Chiranjib Patra, Matangiri Chattppadhyay , Parama Bhaumik and Anjan Guha Roy , "Using Self Organizing Map in Wireless Sensor Network for Designing Energy Efficient Topologies," 2nd International Conference on Wireless Communication, Vechicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), pp. 1-6, March 2011.

[2] Abdul Mannan and Haroon A. Babri, "Categorizing WSN's Sensory Data Using Self Organizing Maps," Third International Conference on Electrical Engineering (ICEE),pp. 1-6, April 2009.

[3] D. F. Larios, J. Barbancho, G. Rodriguez, J.L. Sevillano, F. J Molina, C. Leon, "Energy efficient wireless sensor network communications based on computational intelligent data fusion for environmental monitoring," IET Communication, pp. 2189-2197 ,September 2012.

[4] Luca Paladina, Antonino Biundo, Marco Scarpa, Antonio Puliafito, "Self Organizing Maps for Synchronization in Wireless Sensor Networks," ESR groups France, 2008.

[5] Andrea Kulakov, Danco Davcev and Goran Trajkovski, "Implementation artificial neural-network in wireless sensor networks,"

[6] Neda Enami, Reza Askari Moghadam and Kourosh Dasashtabar Ahmadi, "A New Network Based Energy Efficient Clustering Protocol For Wireless Sensor Network,"

[7] T. Kohonen, "Generalizations of the self-organizing map," Neural Networks. IJCNN '93-Nagoya. Proceedings of 1993 International Joint Conference on, On page(s): 457 - 462 vol.1 Volume: 1, 25-29 Oct. 1993.

[8] Laurene Fausett, Fundamentals of Neural networks: Architecture, Algorithm and Applications, Pearson Education, ISBN 978-81-317-0053, 1994**.**

[9] K. Kumar, R. Singh, Z. Khan, "Air Traffic Enroute Conflict Detection Using Adaptive Resonance Theory Map Neural Networks (ART1)," Ubiquitous Computing and Communication International Journal, Korea, Vol. 3, No 3, July, 2008, pp. 28-35.

[10] K. Kumar, R. Singh, Z. Khan, A. Indian, "Air Traffic Runway Allocation Problem ARTMAP (ART1)," Ubiquitous Computing and Communication International Journal, Korea, Vol. 3, No 3, July, 2008, pp. 130-136.

[11] Krishan Kumar, "ART1 neural networks for air space sectoring," International Journal of Computer Applications, USA, 2012, pp. 20 – 24.

[12] Mohit Mittal and Krishan Kumar, "Network lifetime enhancement of homogeneous sensor network using ART1 neural network", Sixth International conference on computational intelligence and communication networks, IEEE, 2014, pp. 472-475.

[13] Self organizing Map (SOM) Neural Networks for Air Space Sectoring", Sixth International conference on computational intelligence and communication networks, IEEE, 2014, pp. 1096-1100.

[14] Kulkarni, R.V, Forster, A. ; Venayagamoorthy, G.K.(2011), 'Computational Intelligence in Wireless Sensor Networks: A Survey', *IEEE Communications Surveys & Tutorials*, VOL. 13, NO. 1, FIRST QUARTER 2011, pp. 68 – 96.

[15] S. Haykin (1994),'Neural Networks: A Comprehensive Foundation', *Prentice Hall*, Prentice Hall PTR Upper Saddle River, NJ, USA