

ANN for node localization in Wireless Sensor Network

Shikha Bhardwaj*

Abstract

With the increasing use and application of Wireless Sensor Networks (WSN), need has arisen to make them more efficient in a cost effective way. An important area which can bring efficiency to WSNs is the localisation process by which the sensor nodes in the network can identify their own location in the overall network. The objective of this paper is to study and test the use of Artificial Neural Networks (ANNs) as a method of localisation in WSNs. This is achieved by using Backpropagation algorithm (BPN) based on multilayer perceptron (MLP) neural networks to carry out the localization process. The network consisting of sensor nodes is initially trained using training algorithms namely Levenberg-Marquardt and Resilient Backpropagation. The network is then tested with a new independent set of data to prove the effectiveness of proposed model. In the paper, other variables like number of anchor nodes, neurons in hidden layers etc. which impact the efficiency of the network in localisation process are also analysed.

Keywords: Wireless Sensor Network, Localization, Backpropagation algorithm, Resilient Backpropagation, Beacon nodes

Introduction

A Wireless Sensor Network is an interconnected network of multiple sensor nodes, which communicate with each other to create a smart sensor network which performs various functions [Akyildiz, I. F. (2002)]. The role of sensor nodes is to convert the physical conditions (e.g. temperature, atmospheric pressure, humidity etc.) surrounding them to electric signals. These signals are then transmitted in the entire network in a fashion that each sensor node co-operates and communicates with each other. The data being so transmitted through signals is centrally collated and analysed by few special nodes (often known as base station nodes) [Rashmi Agrawal, Brajesh Patel (2012)]. These anchor nodes provide the useful information about the area covered by WSN. These WSNs are now increasingly being used to provide critical and sometimes lifesaving information about the environment e.g. providing early warning of floods, cyclones, Tsunami etc. Apart from these, such networks also have practical usage in other areas like industrial, commercial, military operations etc. [Akyildiz, I. F. (2002)]. Many of these applications require the precise information about the location of sensor node in the network for tasks such as network routing, communication and to state geographical source of events such as fire detection, earthquake detection, flood detection, temperature monitoring etc. [Azzedine Boukerch, Horacio A.B.F., Edurado F. Nakamura, Antonio A.F. Loureiro

*Research Scholar, Guru Gobind Singh Indraprastha University, New Delhi, India.

(2007)]. This phenomenon is achieved by a process called Localization, wherein the node placed in a network determines its own location in relation to the anchor nodes. The general approach used to facilitate the localisation is to deploy certain nodes which are aware of their location in the overall network and are called anchor nodes (or Beacon nodes). The other nodes in the system calculate the distance between their location and the anchor nodes [Azzedine Boukerch, Horacio A.B.F., Edurado F. Nakamura, Antonio A.F. Loureiro (2007)].

There are quite a few techniques to enable the process of localisation which can be broadly classified into a) range based; and b) range free techniques. [He T., Huang C., Blum B. M., Stankovic J. A., and Abdelzاهر T. (2003)]. In a range based technique, absolute distance or absolute angle is measured between the two nodes using methodologies like ToA (Time of Arrival), TDoA (Time-Difference of Arrival), AoA (Angle of arrival), and RSS (Received signal strength) [He T., Huang C., Blum B. M., Stankovic J. A., and Abdelzاهر T. (2003)] [Wang J. et al (2010)]. On the contrary, range free techniques rely on the connectivity and proximity of nodes viz-a-viz the anchor nodes to estimate the distance between them. These estimates are based on certain parameters like amount of packets transferred during routing etc. Range based techniques are more precise and accurate than range free techniques however the latter are more simple and cost effective [He T., Huang C., Blum B. M., Stankovic J. A., and Abdelzاهر T. (2003)] [Wang J. et al (2010)]. These days, the range-free techniques are gaining more popularity than the range-based methods [Nazish Irfan, Miodrag Bolic, Mustapha C. E. Yagoub and Venkataraman Narasimha (2010)]. Application of Artificial neural network (ANN) as a range free technique is one of the forthcoming methods that can be used for localization in WSNs [Tian J. and Shi H. (2007)] [Syahrulanuar NGAH, Hui ZHU, Kui-Ting CHEN, Yuji TANABE and Takaaki BABA (2009)]. In this paper, the focus is to understand and validate the use of ANNs as a technique of localisation. The paper is divided into 4 sections explaining overview of Artificial Neural Networks including Multi-layer perceptron model, implementation of the proposed model, and analysis of the results.

Artificial Neural Network

Introduction to ANN

Artificial Neural Network (ANN) is a computational tool modeled on the structure and interconnection of the neurons in the nervous system of the human brain. These are massive parallel computing systems consisting of an extremely large number of simple processors with many interconnections [Jain A.K., Mao J., Mohiuddin K.M. (1996)]. The basic processing unit of ANN is artificial neuron. The model of an artificial neuron was propounded by McCulloch and Pitts in 1943 [Jain A.K., Mao J., Mohiuddin K.M. (1996)] which consists of a) input weights, b) processing element; and c) output function as shown in the following figure:

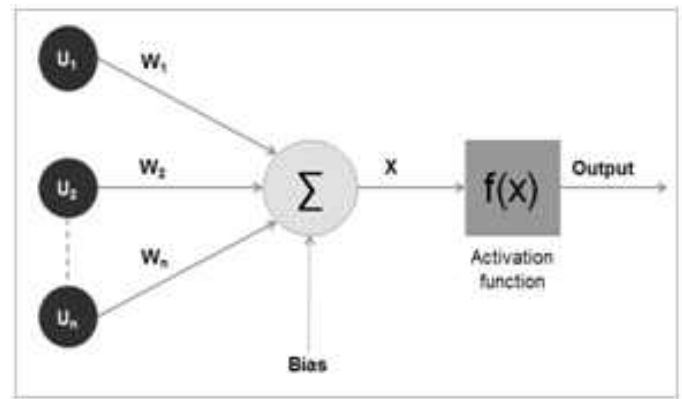


Figure 1: Mathematical model of artificial neuron

In the above figure, U_1, U_2, \dots, U_n are the inputs to the neuron; W_1, W_2, \dots, W_n are weights of the inputs and $X = \sum_{i=1}^n U_i * W_i$. The processing element in the neuron is represented by Σ in the figure, which processes the weighted inputs along with the bias element resulting in X . The output is then generated through the activation function by limiting the amplitude of the output within a specific range.

The most important property of an artificial neural network is its ability to learn by examples and generalize itself. To make ANN learn, it is trained to perform the desired function by inputting a particular input data which leads to a specific output data. Once the network is so trained, it can predict the output of any new dataset. The training techniques are broadly categorised into two:

- i) Supervised learning techniques [Jain A.K., Mao J., Mohiuddin K.M. (1996)]- These techniques use both input and target output dataset. The target output dataset is used as a milestone which the network is supposed to achieve by processing the input data; and
- ii) Unsupervised learning [Jain A.K., Mao J., Mohiuddin K.M. (1996)] - These techniques uses only input data set. The network is not provided with the target data set.

Multilayer Perceptron

There are various categories of neural networks and most commonly used is the Multilayer Perceptron (MLP) [Leonardo Noriega (2005)]. It is an enhancement of single layer perceptron consisting of only input and output layer, as explained above. In MLP, apart from the input and output layers, there are multiple layers of the hidden neurons. These hidden layers are not visible to the users and are used to provide high level of processing power combined with flexibility to the system. An diagrammatical representation of MLP is shown in Fig. 2:

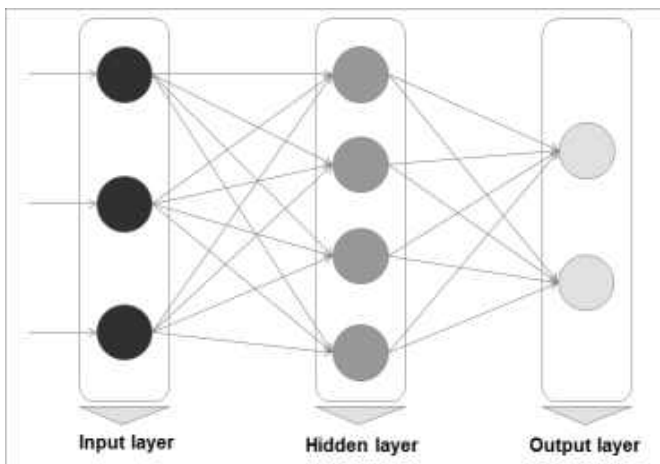


Figure 2: Model of Multilayer Perceptron

Given the complexity involved in the MLP, their training is generally performed in a supervised manner. The training is normally based on the minimization of errors between the output generated by the network and the desired output. This is achieved by backward propagating the errors calculated in the previous iterations through the network hence this training algorithm is known as Error Backpropagation [11]. The idea of the backpropagation algorithm is to reduce the error until the ANN learns the training data. The training begins with random weights, and the goal is to adjust them so that the error will be minimal. The performance function used for BPN is mean square error (MSE) where, MSE is the difference between desired output and estimated output. The objective of BPN is to minimize MSE till the required goal is met.

In the next sections of the paper, application of ANN in localisation process is explained, using BPN training algorithm for providing supervised training to MLP. The trained network is then tested to determine the effectiveness of the proposed model.

Implementation of Proposed Model

Wireless Sensor Network

A Wireless Sensor Network was created over a 100 x 100 grid as shown in Figure 3. This network will be first trained to locate the sensor nodes placed on the intersection points of the grid for eg. over the co-ordinates (20,20), (40,40), (40,60) etc. Then as a next step, the trained network will be tested to locate the nodes placed on the arbitrary points on the network. As mentioned earlier, the location of the nodes will be estimated with respect to the anchor nodes. In order to analyse the performance of the network, the training and testing of the network was done using different numbers of anchor nodes (In the paper, analysis is performed using 2, 3 and 4 anchor nodes).

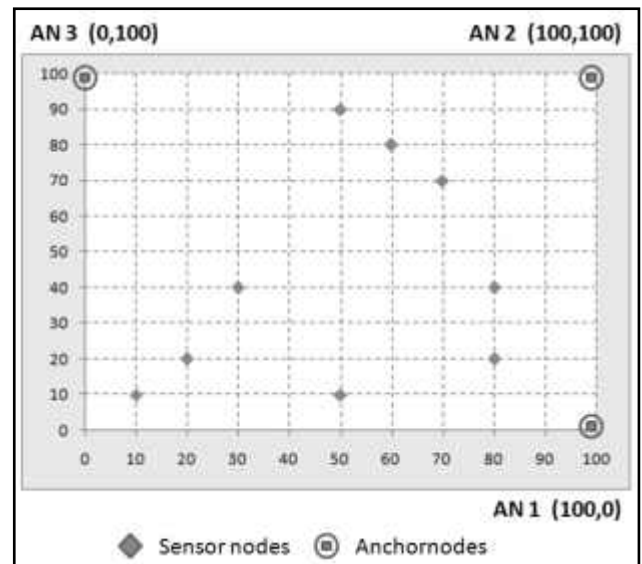


Figure 2: Grid of sensor and anchor nodes in WSN

Training of ANN

The ANN was trained using MATLAB 7.9 neural network toolbox [URL: www.mathworks.com/products/neuralnet/] to locate each sensor node in the grid by estimating the distance with respect to each defined anchor node. The Multilayer Perceptron network was trained by supervised training technique of error backpropagation method using the following two training algorithms:

- Levenberg-Marquardt (LM), and
- Resilient Backpropagation (RP)

The network was trained using tan-sigmoid/log-sigmoid as transfer functions in hidden layer with 10/15 neurons and linear function in the output layer. Also, the numbers of anchor nodes in the network were increased successively. The training dataset is generated by calculating Euclidean distance between 3 anchor nodes and each sensor node. This data is fed as input data along with the target data representing the actual location of the sensors in the co-ordinate grid. The snapshot of the data is presented in Table 1.

Table 1: Input and target output data set with 3 anchor nodes

Target data		Input data		
X_i	Y_i	AN1 (100,0)	AN2 (100,100)	AN3 (0,100)
0	0	100.00	141.42	100.00
10	0	90.00	134.54	100.50
20	0	80.00	128.06	101.98
30	0	70.00	122.07	104.40
40	0	60.00	116.62	107.70
....
90	100	100.50	10.00	90.00
100	100	100.00	0	100.00

Based on the above data, the network provides the estimated location of each sensor node and localisation error is computed using the formula:

$$\sqrt{(Xe - Xi)^2 + (Ye - Yi)^2}$$

Where Xe and Ye are estimated co-ordinates and Xi and Yi are actual co-ordinates of the sensor nodes.

These localization errors are analysed to determine the best and most effective model. Below is the quick comparative analysis of the localization error obtained from different training algorithms with 3 anchor nodes:

Table 2: Statistical analysis of localization errors with 3 anchor nodes

Transfer functions		Algori thm	Localisation error				
Hidden layer	Output layer		Max	Min	Avg	Std. Dev	Med ian
With 10 nodes in hidden layer							
Tansig	Linear	LM	5.91	0.15	1.89	1.38	1.44
Logsig	Linear	LM	5.85	0.22	1.88	1.16	1.63
Tansig	Linear	RP	13.21	0.27	4.09	2.64	3.26
Logsig	Linear	RP	16.18	0.40	3.44	2.19	3.00
With 15 nodes in hidden layer							
Tansig	Linear	LM	5.58	0.07	1.45	1.19	1.11
Logsig	Linear	LM	7.39	0.02	1.04	1.19	0.73
Tansig	Linear	RP	10.80	0.32	3.13	1.98	2.70
Logsig	Linear	RP	6.42	0.13	2.67	1.63	2.29

The above analysis shows that the LM algorithm is resulting in the lowest average localization error. The localization errors obtained using LM algorithm with logsigmoid function is presented in the following graph. The graph shows the maximum and minimum localization errors also which are obtained for different nodes.

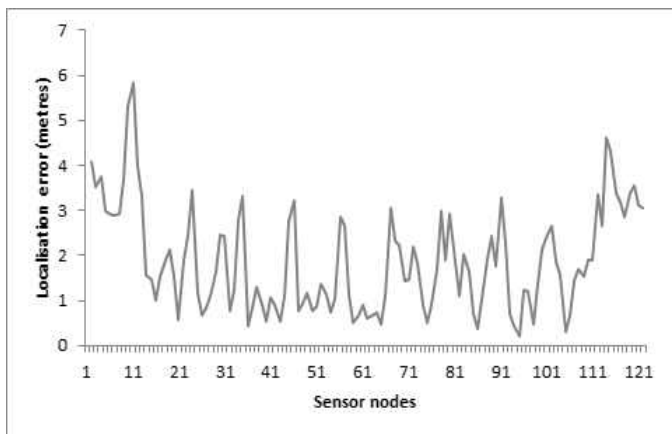


Figure 4 (a): Localisation error using LM algorithm with Logsig function using 3 anchor nodes and 10 hidden layer.

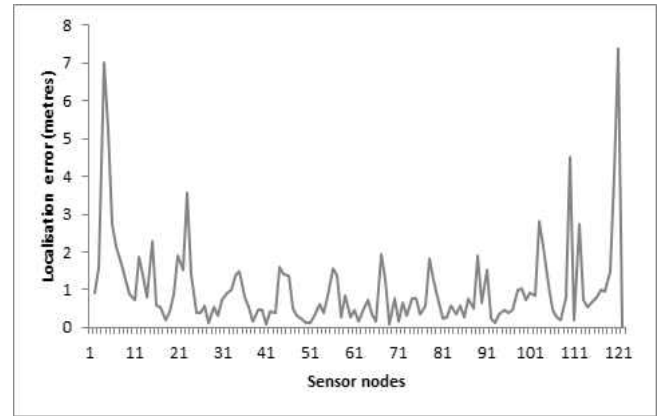


Figure 4 (b): Localisation error using LM algorithm with Logsig function using 3 anchor nodes and 15 hidden layer neurons.

Variation in the number of anchor nodes in the network also impact the level of localisation error. The graph presented in Figure 5 represents the changes in the average localisation error with using different number of anchor nodes for training the network:

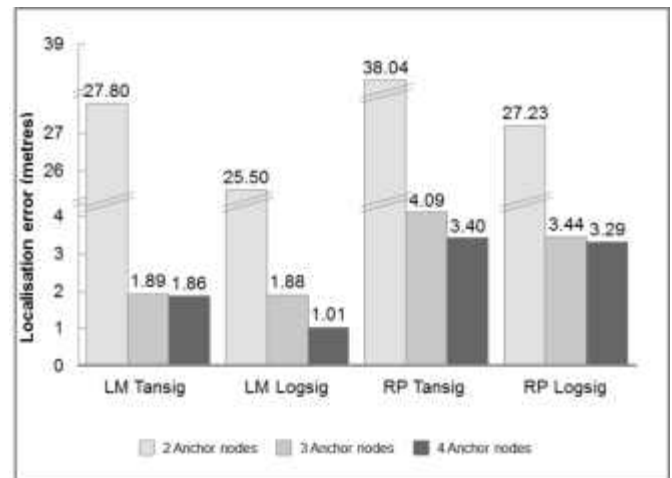


Figure 5: Comparative analysis of average localisation error with different anchor nodes

Figure 5 shows that the level of average localisation error is high, when only 2 anchor nodes were used. However, the level of error decreases significantly for all algorithms when the number of anchor nodes are increased from 2 to 3. Further, the incremental change in the number of anchor nodes from 3 to 4 is not that significant as compared to change from 2 to 3.

Testing of the proposed model

The trained network was further tested to predict the location of around 1000 sensor nodes on the same network grid. For testing purpose, the sensor nodes were deployed in a scattered manner throughout the grid and not necessarily on the intersection points. This is illustrated through the grid shown in figure 6.

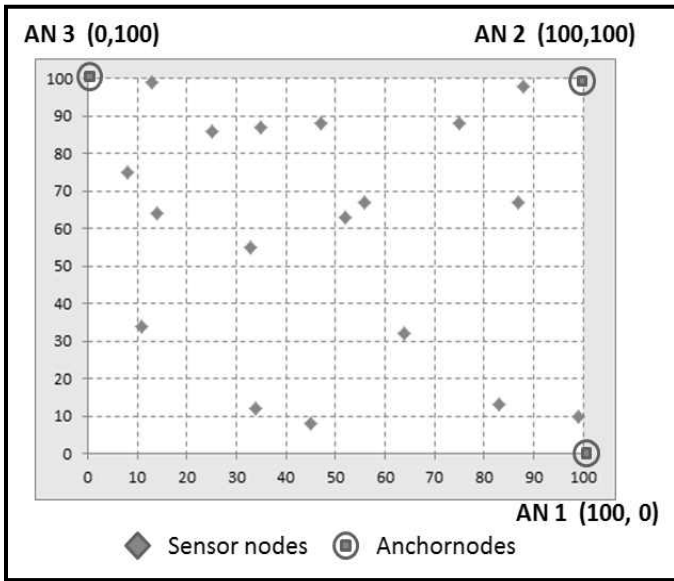


Figure 6: Grid of the network showing sensor nodes and anchor nodes

In order to test the network, only input values in the form of distance of those sensor nodes to the pre-defined anchor nodes were used. Unlike in training, no target values were fed into the system. The following is the snapshot of the input values uploaded into the system:

Table 3: Table showing the input data

Location of sensor nodes			Input data representing distance from sensor node		
S.No.	X	Y	AN1 (100,0)	AN2 (100,100)	AN3 (0,100)
1	54	18	49.40	94.02	98.18
2	66	69	76.92	46.01	72.92
3	31	34	76.92	95.48	72.92
4	70	95	99.62	30.41	70.18
....			
1000	5	56	110.28	104.69	44.28

The network provides the output as an estimated location of each sensor node in the form of X and Y co-ordinates. The output data was then compared with the actual data set to compute the localisation error using the same formula as mentioned above. The localisation error so obtained was analysed to compare the results of different algorithms to determine the most accurate algorithm. The statistical results of localisation error of test data with network having 3 anchor node are presented in the table below:

Table 4: Statistical analysis of localization error

Transfer functions		Algorithm	Localisation error				
Hidden layer	Output layer		Max	Min	Avg	Std. Dev	Median
With 10 nodes in hidden layer							
Tansig	Linear	LM	11.86	0.03	2.39	1.87	1.89
Logsig	Linear	LM	11.77	0.06	2.35	1.69	1.80
Tansig	Linear	RP	16.94	0.03	3.80	2.72	3.07
Logsig	Linear	RP	14.91	0.13	3.48	2.12	2.99
With 15 nodes in hidden layer							
Tansig	Linear	LM	11.69	0.13	2.21	1.89	1.58
Logsig	Linear	LM	10.12	0.04	1.79	1.56	1.33
Tansig	Linear	RP	16.62	0.06	3.06	2.38	2.40
Logsig	Linear	RP	11.35	0.06	2.81	2.13	2.36

The testing statistical analysis also highlights that the average localization error is minimum with LM training algorithm and logsigmoid function. It can also be observed that the performance of the network improved with increase in number of hidden neurons. The localization error obtained for the algorithm is presented in the graphs below:

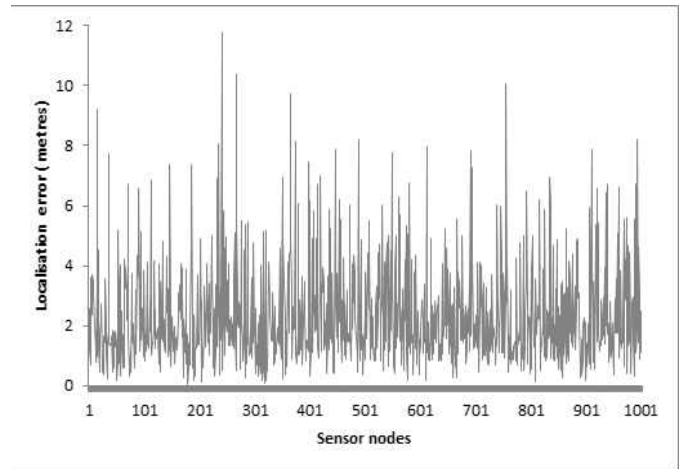


Figure 7a: Localisation error of LM logsig with 10 hidden neurons and 3 anchor nodes

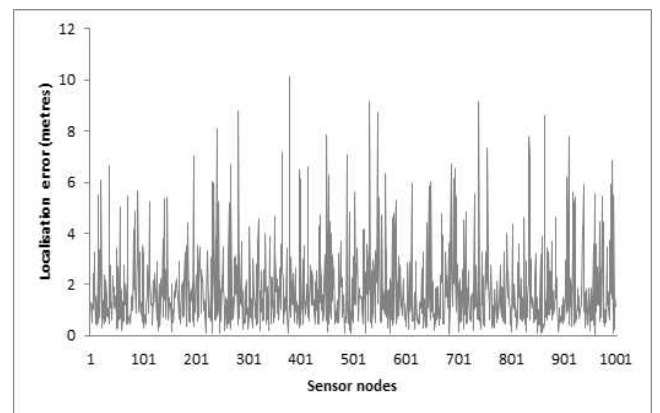


Figure 7b: Localisation error of LM logsig with 15 hidden neurons and 3 anchor nodes

The testing was also performed with 2, 3 and 4 anchor nodes to analyse the performance of the network. The graph below shows the comparative result.

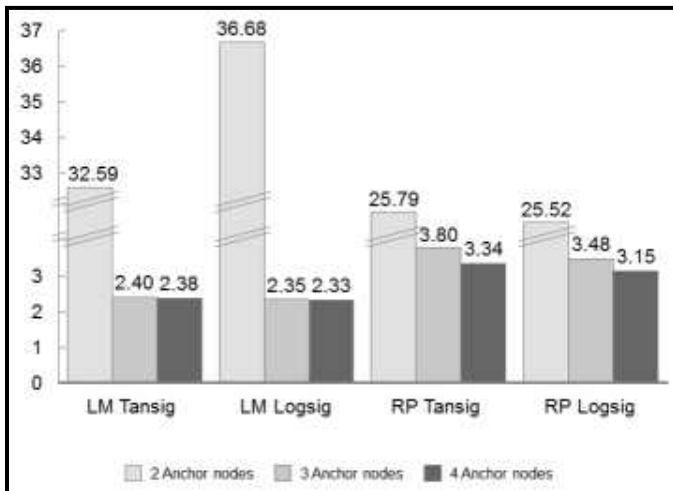


Figure 8: Comparative analysis of average localization errors with different anchor nodes

The above graph depicts that when only two anchor nodes were used, the network could not accurately estimate the location of the sensor nodes in the grid. However, with an increase in the number of anchor nodes the estimated are closer to the actuals and hence the average localization error decreased. It is also interesting to note that the similar proportionate decrease in the localization error could not be experienced when the number of anchor nodes increased from 3 to 4. This proves that increase in anchor nodes helps improving the performance of the network but only to a certain extent. After a point any increase in anchor nodes does not materially impact the performance of the network.

Conclusion

In this paper, the application of artificial neural network to localisation process was explained. The network has been trained and tested in a multilayer perceptron network using two different training algorithm viz. Levenberg-Marquardt (LM) and Resilient Backpropagation (RP) to predict the location of the sensor nodes in the network grid. The computations are made by varying parameters such as the transfer function, training algorithm, number of anchor nodes and number of hidden neurons. On analysis of all results obtained from these parameters, the Levenberg-Marquardt (LM) algorithm is found to be the better algorithm which provided results with minimum average localisation error. The Levenberg-Marquardt algorithm is the faster learning algorithm with fast convergence and is effective in goal achievement especially for function approximation problems. It is also observed that on increase in the number of neurons in hidden layer performance of the network also increased but to a certain extent. This makes it the efficient and effective localization algorithm in application in wireless sensor network. Further, it is also seen that variations in the number of anchor nodes and neurons in the hidden layer can improve the performance of network in accurately estimating the location of the sensor nodes.

References

1. Akyildiz I. F. et al (2002), "Wireless Sensor Networks: A Survey" *Computer Networks (Elsevier)*, vol. 38(4), pp 393-422.
2. Rashmi Agrawal, Brajesh Patel (2012), "Localization in Wireless Sensor Network Using MDS", *International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN)*, ISSN No. 2248-9738 Volume-1, Issue-3.
3. Azzedine Boukerch, Horacio A.B.F., Edurado F. Nakamura and Antonio A.F. Loureiro (2007), *University of Ottawa "Localization systems for Wireless Sensor Networks"*, *IEEE wireless Communications*, volume 14, Issue 6, pp 6-12, December.
4. He T., Huang C., Blusm B. M., Stankovic J. A. and Abdelzaher T. (2003), "Range-free localization schemes for large scale sensor networks," in *Proc. MobiCom '03*, 2003, p. 81.
5. WANG J. et al (2010), "A survey on sensor localization" *J Control Theory Appl*, vol. 8 (1), pp 2-11.
6. Nazish Irfan, Miodrag Bolic, Mustapha C. E. Yagoub and Venkataraman Narasimha (2010), "Neural-based approach for localization of sensors in indoor environment" *Telecommunication Systems*, Volume 44, Issue 1-2, pp 149-158.
7. Tian J. and Shi H. (2007), "Study of localization scheme based on neural network for wireless sensor networks," in *IET Conference on Wireless, Mobile and Sensor Networks*, pp. 64-67.
8. Syahrulanuar NGAH, Hui ZHU, Kui-Ting CHEN, Yuji TANABE and Takaaki BABA (2009), "Artificial Neural Network Based Model for Location Position Systems", *IJCSNS International Journal of Computer Science and Network Security*, VOL.9 No.7.
10. Jain A. K., Mao J. and Mohiuddin K. M. (1996), "Artificial Neural Network: a tutorial", <http://web.iitd.ac.in/~sumeet/jain.pdf>
11. Leonardo Noriega (2005), *School of Computing Staffordshire University, Beaconside Staffordshire ST18 0DG, "Multilayer Perceptron Tutorial"*, Nov 17, http://www.cs.sun.ac.za/~kroon/courses/machine_learning/mlp.pdf
12. Almeida L.B. (1997), "Handbook of Neural Computation", January release, <http://www.lx.it.pt/~lbalmeida/papers/AlmeidaHNC.pdf>
13. *Network Toolbox for MATLAB*, www.mathworks.com/products/neuralnet/