© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Localization Process for WSNs with Various Grid-Based Topology Using Artificial Neural Network

Batoul Sulaiman[1], Emad Natsheh[2*], Mohammad Sharaf[3], Mai Abusair[3], Naji Qatanani[1]

[1]Department of Computerized Mathematics,
An-Najah National University, PALESTINE

[2]Department of Computer Engineering,
An-Najah National University, PALESTINE

[3]Department of Computer Science,
An-Najah National University, PALESTINE

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2023.15.07.021
Received 28 January 2023; Accepted 06 November 2023; Available online 31 December 2023

Abstract: Wireless Sensor Network (WSN) is a technology that can aid human life by providing ubiquitous communication, sensing, and computing capabilities. It allows people to be more able to interact with the environment. The environment contains many nodes to monitor and collect data. Localizing nodes distributed in different locations covering different regions is a challenge in WSN. Localization of accurate and low-cost sensors is an urgent need to deploy WSN in various applications. In this paper, we propose an artificial automatic neural network method for sensor node localization. The proposed method in WSN is implemented with network-based topology in different regions. To demonstrate the accuracy of the proposed method, we compared the estimated locations of the proposed feedforward neural network (FFNN) with the estimated locations of the deep feedforward neural network (DFF) and the weighted centroid localization (WCL) algorithm based on the strength of the received signal index. The proposed FFNN model outperformed alternative methods in terms of its lower average localization error which is 0.056m. Furthermore, it demonstrated its capability to predict sensor locations in wireless sensor networks (WSNs) across various grid-based topologies.

Keywords: Feedforward neural network, deep neural network, wireless sensor networks, weighted centroid localization algorithm

1. Introduction

Recently, wireless sensor networks (WSN) receive a wide world interest. A WSN is a network of sensor nodes that are spatially distributed in different locations and can work collaboratively to transmit data collected from an investigation area. The WSN technology comes with various advantages over traditional networking solutions, such as flexibility, scalability, inexpensive costs, and ease of deployment in an enormous range of applications [1].

Many IoT applications, that deploy wireless sensor nodes, need the nodes to be aware of their positions relative to the sensor network [2]. The data collected from different nodes can be coupled with their locations in order to give more significant information. For example, the readings for soil humidity sensors, collected from a smart agriculture field, should be referred to the relative positions from where these readings were gathered. Thus, the smart system can decide the field locations that require irrigation [3].

**Corresponding author: e.natsheh@najah.edu*

Traditional ways are used for identifying sensors locations, such as global positioning system (GPS) [4]. Using GPS is not feasible for wireless sensor for several reasons. GPS has some limitations on location determination. It is able to determine the locations for outdoor sensors, while it cannot determine the specific locations for indoor sensors since the line of sight from the GPS satellite is not available. Moreover, the sensor nodes require external hardware to be equipped with GPS.

Many researchers propose various methods, techniques and algorithms based on different properties of sensor nodes to solve position estimation and localization problem. The researches mostly consider the computational complexity of the localization technique and the localization accuracy [5-7].

The analytical methods have been used to address the sensor node localization process. These methods are subdivided into two categories [8]:

i. The first category encompasses the range-based position algorithms [9]. They are used to compute the absolute distance between two nodes and, then, estimate the location for the unknown node. Moreover, they require using special hardware for estimating the location of the unknown nodes, and, thus, the design of the sensor node becomes more complex and leads to an increase in the associated costs. Examples on range-based position algorithms are time of arrival (TOA) [10] and angle of arrival (AOA) [11].

ii. The second category encompasses the range-free position algorithms [12]. They use the connectivity information to estimate the distance between the anchor node and the unknown node. Moreover, they do not need additional hardware for the localization process, and, thus, they are cost-effective and very attractive schemes for WSNs. Examples on range-free position algorithms are distance vector-hop (DV-Hop) [13], amorphous [14], and the centroid algorithm [15].

Emerging approaches have adopted artificial neural networks (ANN) in machine learning for the sake of node localization during the data collection phase. In [16], the proposed system use the WCL algorithm combined with a neural network to improve positioning accuracy. In [17], the author proposed to use weighted centroid methods and received signal strength indicators as primary factors achieving 97% localization accuracy. In [18], the RSSI-based weighted centroid method was used and the localization accuracy was used by iterative methods to gain the maximum possible accuracy level. In [19], a flexible location estimation algorithm is proposed using a generalized regression neural network (GRNN) and weighted centroid localization and the simulation and experimental results indicate that the location accuracy is satisfactory. In [20], a positioning system for the indoor wireless sensor networks based on the artificial neural network was introduced. The positioning systems use the received signal strength indicator (RSSI) as a substitute for the distance for localizing sensor nodes. ANN used to find the relation between the RSSI and the sensor node locations in the indoor environment. In [21], a feedforward neural network was used in the fingerprint methodology for indoor localization, where WSN sensor nodes were placed on fixed positions in an experimental room. This work demonstrates a solution for indoor localization based on created database RSSI measurements and trained neural network. In [22], Kumar proposed a methodology that uses an FF neural network with three hidden layers 12-12-2 for the node localization in WSNs. The input to the proposed methodology is the received signal strength indicator values of the signal received from the anchor nodes. In addition, Battiti et al. [23] proposed a method based on neural networks for reducing the errors in determining the locations of mobile nodes. This method used a training algorithm based on second-order information to evolve flexible models for the relationship between the location data and raw signal measurements. The paper in [24] proposed a multi-layer neural network that adopted TOA method for estimating the distance between nodes. This neural network model is trained to handle different noise measurements. Eventually, each of the aforementioned methods has a trained neural network capable of performing localization. The methods showed various performances, efficiencies, and accuracies. The restriction of these methods is that their neural networks are not able to deal with different areas for different network topologies.

In this paper, a low-cost localization procedure is realized utilizing a suggested FFNN model with good accuracy and adaptivity for varied network topologies. A comparison analysis is performed between two different types of neural networks, FFNN and DFF. In addition, an investigation of the adaptability of the FFNN model in the localization process of multiple dimensions of WSN is presented. As a result, the produced model investigates how well the FFNN model scales to different network sizes, from small-scale deployments to large-scale sensor networks covering huge areas and achieving good localization accuracy.

The paper proceeds as follows: section 2 presents the proposed methodology. Section 3 presents the simulation for the neural network and the evaluation. Finally, section 4 concludes the work.

2. Proposed Methodology

In this section, we show the methodology we followed for building our neural network. Our neural network uses the locations of the unknown node estimated from the WCL algorithm, which is described in section 2.1. Moreover, in section 2.2, we show the proposed neural networks topologies.

2.1 Weighted Centroid Localization Algorithm Based on RSSI

The weighted centroid algorithm technique depends on the information from the received signal strength indicator between the unknown node and the anchor nodes. Its basic idea is to calculate the distance between the anchor and the unknown node. If there is no communication between the anchor and unknown node, then the value of RSSI is zero. If the unknown node is within the communication range, then more influence is given to the anchors that are nearer to the unknown node and use the data received from them for the localization process. Thus, WCL introduces the weights of the beacons depending on their RSSI towards the unknown node.

Assuming that we have n number of anchor nodes in wireless sensor network, and their coordinates (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), …, (x_n, y_n). By using WCL algorithm, the estimated coordinates (x_{est}, y_{est}) for the unknown node computed through the following formulas:

$$X_{est} = \frac{w_1 x_1 + w_2 x_2 + \ldots + w_n x_n}{w_1 + w_2 + \ldots + w_n} \tag{1}$$

$$Y_{est} = \frac{w_1 y_1 + w_2 y_2 + \ldots + w_n y_n}{w_1 + w_2 + \ldots + w_n} \tag{2}$$

$$w_i = \frac{RSSI_i}{RSSI_1 + RSSI_2 + \ldots + RSSI_n} \tag{3}$$

Where, (x_{est}, y_{est}) is the estimated coordinate for the unknown node, wi is the weight for the anchor node, and RSSI is the received signal strength from the anchor node.

Consequently, we can conclude that, from the previous formulas, the signal transmission distance is an important factor in the localization process. For instance, the greater distance we have the smaller signal strength we receive. Thus, the value of RSSI reflects the distance between nodes.

The localization error (E) [25], which is the difference between the actual location (X, Y) and the estimated location (x_{est}, y_{est}) for the unknown nodes, and that is impacted by communication radius [26], can be computed as in Eq. (4):

$$E = \frac{\sum_{i}^{n} \frac{1}{CR} \sqrt{\left(X_{est} - X\right)^2 + \left(Y_{est} - Y\right)^2}}{n} \tag{4}$$

Where CR is the communication range, (X, Y) is the actual location for the sensor nodes, and n is the number of anchor nodes. Finally, the localization precision is influenced by wireless signal propagation path loss that has an important role in the localization process. Accordingly, in this paper, we adopted a realistic model, log-normal shadowing path loss model [27], that can be applied in an indoor and outdoor environment, and defined as in Eq. (5):

$$PL(d) = PL(d_0) + 10n\log\frac{d}{d_0} + X_\sigma \tag{5}$$

Where PL(d) is the path loss after distance d, n is the path loss exponent, typically n = 2 ~ 6, x_σ is zero means Gaussian distributed random variable whose mean value is 0 and it reflects the change of the received signal power in a certain distance, d_0 is reference distance and usually equals 1 meter, and PL(d_0) is a known reference power value at a reference distance d_0 from the transmitter.

According to the aforementioned factors and formulas, the WCL algorithm can be computed according to the described pseudo-code in Fig. 1.

2.2 Artificial Neural Network Topology

In this work, two promising neural network topologies have been developed for estimating sensor node locations in wireless sensor networks. The first model is the feedforward neural network (FFNN), and the second is the deep feedforward neural network (DFF).

The FFNN is a type of artificial neural network in which the data is introduced in the forward direction. This network consists of interconnection neurons with their activation functions. The neurons are arranged into layers, the first layer is called the input layer, followed by the hidden layer and the last layer is called the output layer.

The input signals are propagated from the input layer, on the left, to the output layer, to the right, passing through the hidden layer. The hidden layer detects the input patterns hidden features and then uses these features in the output layer to compute the final output. The structure of the proposed FFNN is shown in Fig. 2.

Inputs: length and width of the simulation area, number of unknown nodes, number of anchor nodes, communication range (CR), transmitted power (P_t)

Algorithm:

Step1: Distribute the anchors in grid form in the simulation area

Step2: Distribute the unknown nodes randomly

Step3: Compute the distance (d) between the anchors and unknowns

Step4: IF (d<=CR) THEN

Step5: Compute the value of RSSI using log-normal shadowing model

$$PL(d) = PL(d_0) + 10n\log\frac{d}{d_0} + X_\sigma$$

Step6: Compute the anchors weights using

$$w_i = \frac{RSSI_i}{RSSI_1 + RSSI_2 + \ldots + RSSI_n}$$

Step7: Compute the estimated coordinate of the unknown node

$$X_{est} = \frac{w_1x_1 + w_2x_2 + \ldots + w_nx_n}{w_1 + w_2 + \ldots + w_n}$$

$$Y_{est} = \frac{w_1y_1 + w_2y_2 + \ldots + w_ny_n}{w_1 + w_2 + \ldots + w_n}$$

Step8: Compute the localization error

$$E = \frac{\sum_i^n \frac{1}{CR}\sqrt{(X_{est} - X)^2 + (Y_{est} - Y)^2}}{n}$$

Step9: END IF

Fig. 1 - Weighted centroid localization algorithm

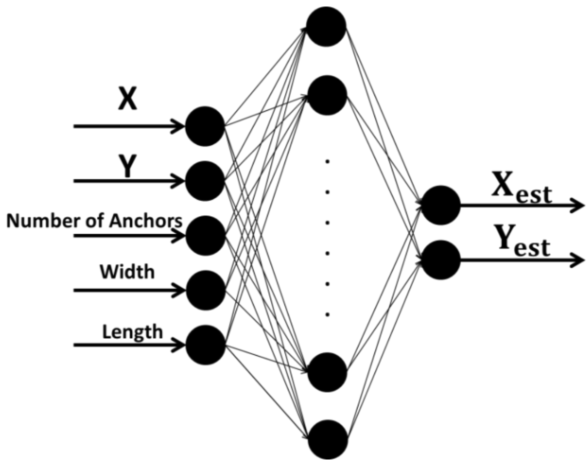

Fig. 2 - Structure of the proposed FF neural network

As illustrated in Fig. 2, the developed FF model is composed of three layers. The input layer consists of five nodes, which are the X and Y coordinates for the actual locations of the sensor node, the number of used anchors, the width and length of the network area. The hidden layer consists of 20 nodes. The output layer consists of 2 nodes that produce the (Xest, Yest) coordinates for the estimated sensor node location.

The other proposed neural network topology is the DFF, shown in Fig. 3. It consists of the input layer with 5 neurons, two hidden layer with 20 neurons in each layer, and the output layer. The input and output layers are specified in the same manner as the proposed FF neural network.

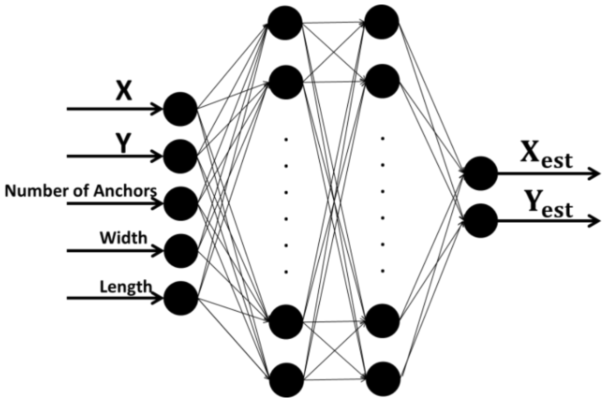

Fig. 3 - Structure of the proposed DFF

In the both proposed structures, the hidden neurons use the hyperbolic tangent activation function, and the output neurons use the linear activation function. The actual output of the hidden neurons and output neurons [28] can be computed as in Eqs. (6) and (7):

$$Y_h = \frac{2}{1+e^{-2X_h}} - 1 \tag{6}$$

$$Y_o = X_o \tag{7}$$

Where x_h is the net weighted input for neuron in the hidden layer, x_0 is the net weighted input for neuron in the output layer Y_h is the node output in the hidden layers, Y_0 is the node output in the output layer. The net weighted input is calculated as in Eq. (8).

$$X = \sum_{i=1}^{n} x_i w_i - \theta \tag{8}$$

Where n is the number of neural inputs, Ө is the threshold applied to the neuron, x_i is the value of input i, and w_i is the weight of input i.

Many training algorithms are used in the training process, such as the gradient descent, and levenberg-marquardt (LM) algorithms. In our two proposed neural networks, the LM algorithm has been chosen as training algorithm. The LM training algorithm locates the minimum of the loss function which is the mean squared error (MSE). It is considered as the most efficient training algorithm for median size ANN; it makes a balance between the stability and training speed [29]. The LM algorithm was developed to approach second-order training speed with no need to compute the hessian matrix (H), which requires high computational effort. It requires calculations for the approximated hessian matrix (H), as given in Eq. (9):

$$H \approx J^T J \tag{9}$$

Where J is the Jacobian matrix. To guarantee that H is invertible, the LM algorithm uses the modified H, as shown in Eq. (10):

$$H \approx J^T J + \mu I \tag{10}$$

Where µ is always positive, called combination coefficient, I is the identity matrix. Eq. (11) is used to calculate the loss function gradient vector when the performance function resembles the MSE.

$$\nabla f = J^T e \tag{11}$$

Consequently, Eqs. (11) and (10) can be combined to express the update rule for LM algorithm as in Eq.(12) [30] :

$$w_{p+1} = w_p - (J^T J + \mu I)^{-1} J^T e \tag{12}$$

The pseudo-code for LM algorithm is given in Fig. 4.

Input: A training input vector **(I_N),** desired output vector **(O_N)**
Output: A vector of modified weights and thresholds **(W)**
Algorithm:
Step1: Set all the weights and threshold levels of the network to random numbers uniformly distributed inside a small range $[{-2.4}/{fi}, {+2.4}/{fi}]$
Step2: Compute $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y}_i)^2$
Step3: WHILE (not stop-criterion) Do
Step4: Compute $\Delta w = (JJ^t + \mu I_m)^{-1} J^t e$
Step5: Update the network weights (w)using Δw
Step6: Recalculate MSE using the updated weights
Step7: IF MSE descrease THEN μ=0.1μ
Step 8: ELSE discard the new weights and μ=10μ
Step 9: END IF
Step 10: END WHILE

Fig. 4 - Levenberg Marquardt algorithm

In FFNN model, before starting the training process, the input data should be introduced to the input layer in order to train the network to recognize the relationship between the input and output data. In addition, the data preparation should be done to transfer the data into better form in order to use it through the training process. This can be done through the normalization process. There are many normalization techniques used to get more reliable networks [31]. In this paper, the Min-Max normalization method [32] is used as shown in Eq.13.

$$I_N = (I - I_{\min})\left[\frac{N_{\max} - N_{\min}}{I_{\max} - I_{\min}}\right] + N_{\min} \tag{13}$$

Where I is the non-normalized input value for the training process, I_N is the normalized input value, I_{min} is the minimum value of the input vector, and I_{max} is the maximum value for the input vector.

After performing the training process, the output predicting results are observed. The normalized values must become normal values, as given by Eq.14.

$$O = (O_N - N_{\min})\left[\frac{O_{\max} - O_{\min}}{N_{\max} - N_{\min}}\right] + O_{\min} \tag{14}$$

Where O is the non-normalized output value for the training process, O_N is the normalized input value, O_{min} is the minimum value of the output vector, and O_{max} is the maximum value for the output vector.

In ANN, to find the optimal weights through the training process, the weights and thresholds should be initialized. Thus, weights and thresholds are commonly set to small values that are uniformly distributed in the range of $(-\frac{2.4}{f_i}, +\frac{2.4}{f_i})$ [33], where fi represents the number of neuron inputs.

3. Simulation and Results

3.1 Evaluation of the WCL Algorithm

To evaluate the performance of the weighted centroid localization algorithm, we implemented it using MATLAB R2018a software. We had set up the following conditions for the simulation environment:

1) The anchor nodes are regularly distributed in a network area of 100m x 100m and the number of these anchors is set to be 100.
2) 37 unknown sensor nodes are deployed randomly in our experiment area.
3) The communication range of sensor node (CR) is set to 20m.

The network was generated using the parameters given in Table 1.

Table 1 - Parameters for weighted centroid algorithm [23]

Parameters	Values
Number of anchor nodes	100
Number of unknown nodes	37
Communication range	20m
Deployment area	100m × 100m
Transmitted power (P_t)	10watt
Path loss exponent (n)	1.8
Reference distance (d_0)	1

Estimated locations for the 37 unknown nodes using the weighted centroid algorithm and the localization error for each one of them are illustrated in Fig. 5. The dots represent the anchor nodes. The actual and estimated positions for sensor nodes using the WCL algorithm are indicated by stars and squares, respectively.

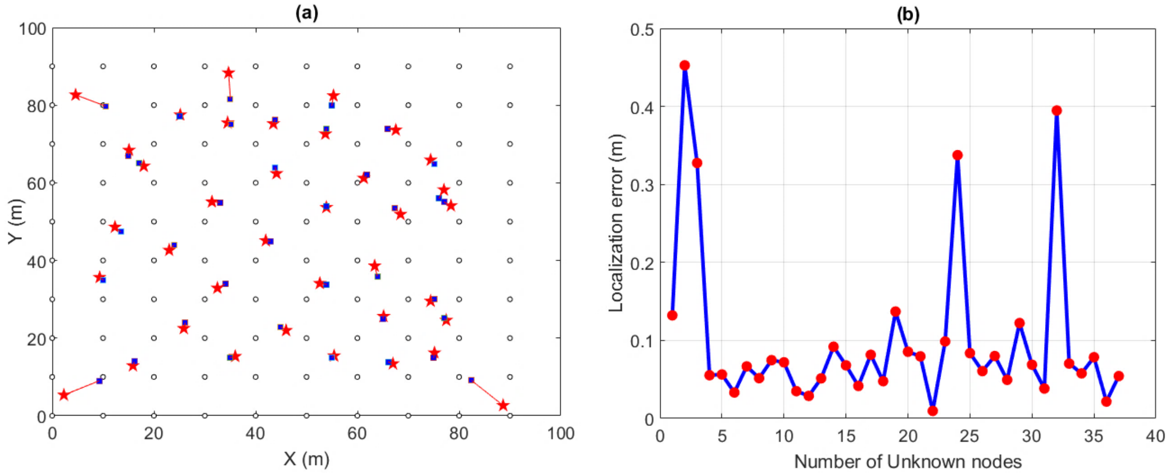

Fig. 5 - Estimated and real positions for the testing data set and localization error using WCL algorithm

Therefore Fig.5 demonstrate the result obtained from the WCL algorithm such that, Fig.5 (a) illustrate the actual and estimated locations for the random nodes. Moreover, Fig.5 (b), shows the error for each node separately. As a result, the average location error is equal to 0.1002. The second node has the maximum error, which means it has the maximum displacement from its actual position. In addition, node number 22 has the minimum error.

3.2 Multi-Layer Network

- **Data collection**

Data collection for the FFNN is a critical issue. Therefore, our data-set consists of the input data that represents the random locations for the unknown nodes. The target output (X_{est}, Y_{est}) is generated from an applied MATLAB code for the WCL algorithm. The RSSI measurements used in the WCL are explained in the previous section. The RSSI is the most important factor that is used to apply WCL algorithm for getting required data.

The data used for the training process were obtained from sensor networks with varying dimensions. Examples of the training data used as part of the training set are illustrated in Table 2.

Table 2 - Examples of the training set

X	Y	Anchors	Width	Length	X_{est}	Y_{est}
63.2421	45.9368	100	100	100	63.8928	46.1567
17.1165	44.4277	100	100	100	16.0529	43.7939
89.8132	44.511	100	100	100	84.7544	45.0956
16.7709	52.9684	225	150	150	16.0913	53.9389

124.7502	35.2376	225	150	150	125.0299	34.9931
47.2888	96.1498	225	150	150	46.0366	96.1385
68.5729	127.8781	400	200	200	67.3496	126.5028
35.1601	34.7811	400	200	200	34.9888	35.0222
28.3134	129.9582	400	200	200	27.8921	130.0209
96.1611	62.3364	625	250	250	96.1209	64.027
190.5943	4.1689	625	250	250	192.1931	9.3643
154.826	91.0939	625	250	250	155.0135	93.025
181.6974	122.3443	900	300	300	184.0838	124.0403
178.9024	131.9046	900	300	300	178.0624	131.8682
171.3831	150.7318	900	300	300	171.9142	151.989

- **Neural network structure**

A series of trails were performed to get the best neural network structure. Table 3 shows our trails for the proposed FF neural network.

Table 3 - Summary of different structure trails FF neural network

Number of Nodes	Training Algorithm	MSE Train	MSE Test	R
10	LM	2.03107	2.16265	0.99779
12	LM	2.10060	2.10978	0.999771
15	LM	1.96859	2.12699	0.999783
20	LM	1.44416	1.42572	0.999841
25	LM	1.44641	1.62184	0.999842
30	LM	1.53435	1.70304	0.999833
20	GD	7.85491	8.08337	0.999144

From Table 3, we can observe that the network with 20 hidden neurons using LM training algorithm is the best obtained structure. It has the least MSE with greatest correlation coefficient (R), and therefore, it has the best performance.

Our trails for DFF neural network are shown in Table 4.

Table 4 - Summary of different structure trails for DFF neural network

Number of Nodes/ Neurons	Training Algorithm	MSE Validation	R
10-10	LM	2.0539	0.99979
20-20	LM	1.7592	0.99984

According to the observation in Table 4, we realize that the FF neural network outperforms the DFF neural network. Moreover, a network with 20-20 neurons in two hidden layers needs a large computational complexity and a processing time.

- **Evaluation of the proposed network models**

The performance of the FF neural network with 20 hidden neurons is analyzed in terms of the number of epochs versus MSE. This structure achieves the best performance compared to the other structures, such that, the best validation performance in terms of MSE is 1.5448 at epoch 300. See Fig. 6, which shows the FFNN performance.

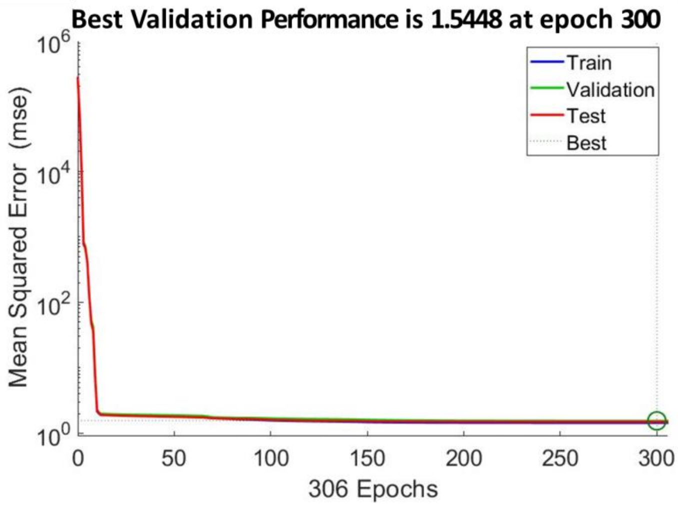

Fig. 6 - The performance of FF neural network (Epochs vs. MSE)

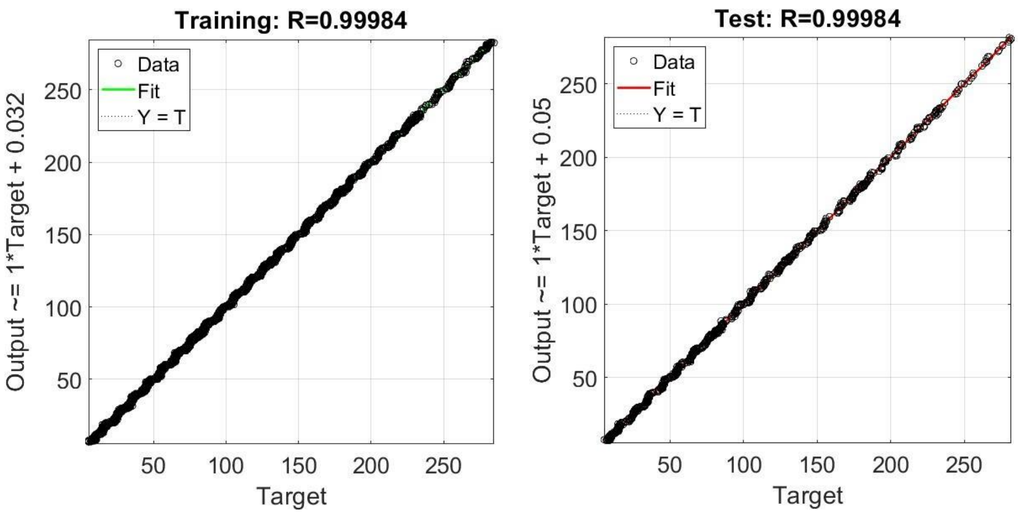

Fig. 7 - Regression of error for FF neural network

For the training process, the data set was divided into 450 testing samples, 300 validation samples and 2250 training samples. Fig. 7, shows the results for the training and validation stage. As illustrated in Fig. 7 the measure error is very low.

To compare our proposed FF neural network topology with the WCL algorithm, we implemented it on the same simulation environment using WCL algorithm. Fig. 8 show the estimated positions for 37 unknown nodes using FF neural network in 100m x 100m network area and the localization error for each unknown node. The dots represent the anchor nodes. The actual and estimated positions (using the FF neural network for sensor nodes) indicated by the stars and squares, respectively.

As illustrated in Fig. 8, the average localization error for the 37 unknown nodes is equal to 0.056m, and the second node has the maximum error. By comparing the results obtained from the FF neural network with the WCL algorithm based on RSSI, we realized that our network model gives more accurate results. It achieves a less average localization error.

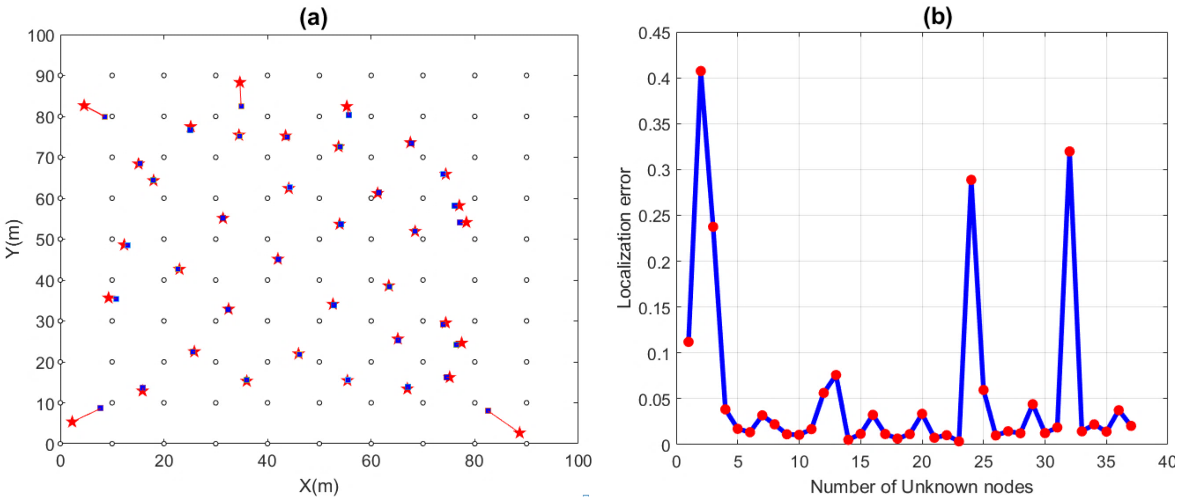

Fig. 8 - Estimated locations for 37 unknown nodes and the localization error for each node using FF neural network

In addition, our FF neural network model has the ability to predict the locations for the sensors in different WSNs areas. The average localization error for all the unknown sensor nodes is used to evaluate the performance of the localization schemes. Summary of performance comparison for 37 unknown nodes is shown in Table 5.

Table 5 - Estimated localization error using WCL and FF network

Method	Error in meters
Weighted centroid localization algorithm	0.1002
Feedforward Neural Network	0.056

Patterned topologies for WSNs [34] provide a longer network lifetime than the randomly deployed WSNs when they used the same number of sensors. Moreover, this type of WSNs can efficiently save energy. Therefore, we built our FF neural network model to be able to predict the sensor node locations in grid-based WSNs with different areas.

The estimated locations for the unknown nodes in different WSNs areas and different number of anchor nodes can be obtained using our model with high accuracy as shown in Fig.9 – Fig.12. The localization error for these sensor nodes in each area using the FF neural network is summarized in Table 6.

Table 6 - Estimated localization error

Network Area	Error in Meters
150m × 150m	0.0585
200m × 200m	0.0507
250m × 250m	0.0515
300m × 300m	0.0519

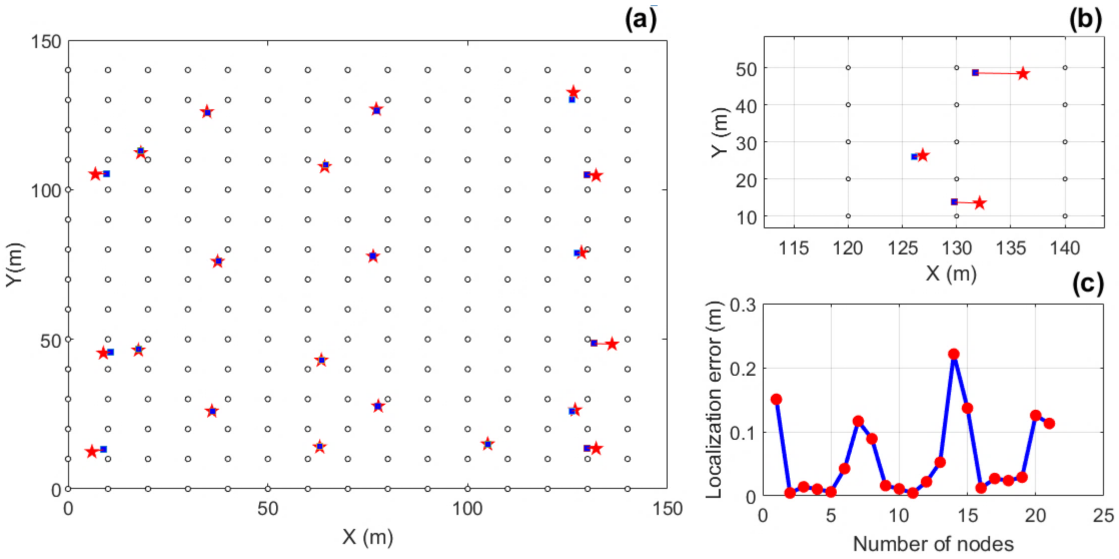

Fig. 9 - Estimated locations for 21 unknown nodes distributed in area 150m x 150m (a) zoom out; (b) zoom in; (c) localization error

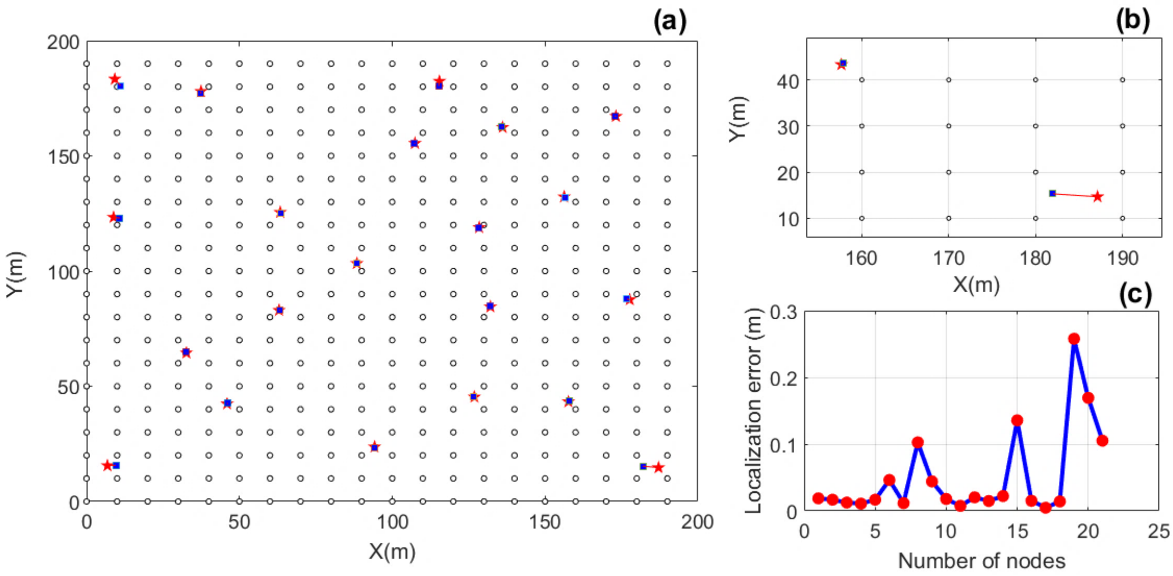

Fig. 10 - Estimated locations for 21 unknown nodes distributed in area 200m x 200m (a) zoom out; (b) zoom in; (c) localization error

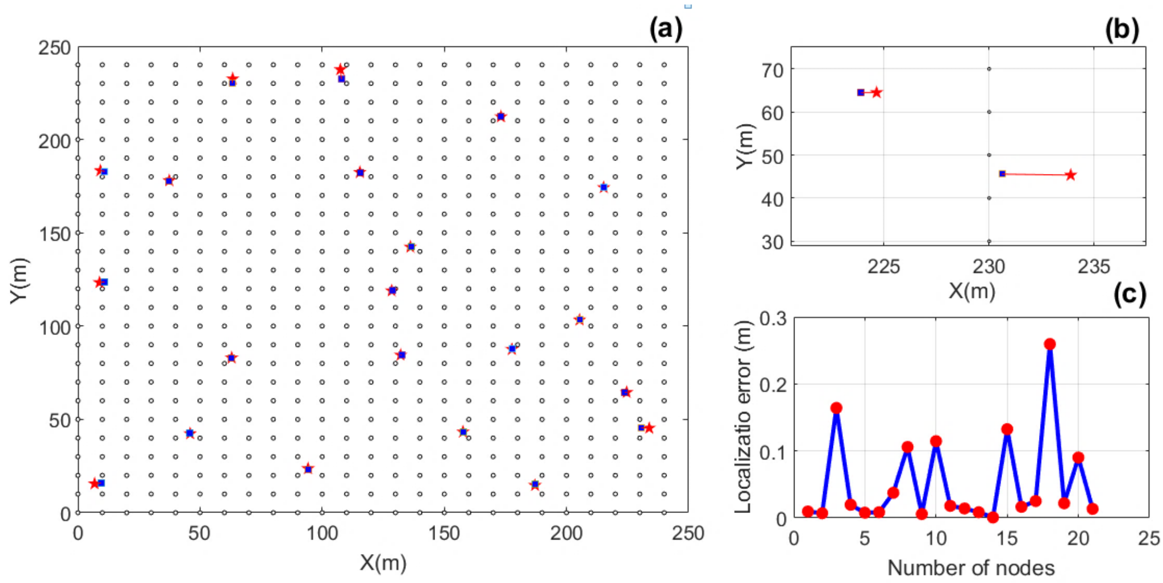

Fig. 11 - Estimated locations for 21 unknown nodes distributed in area 250m x 250m (a) zoom out; (b) zoom in; (c) localization error

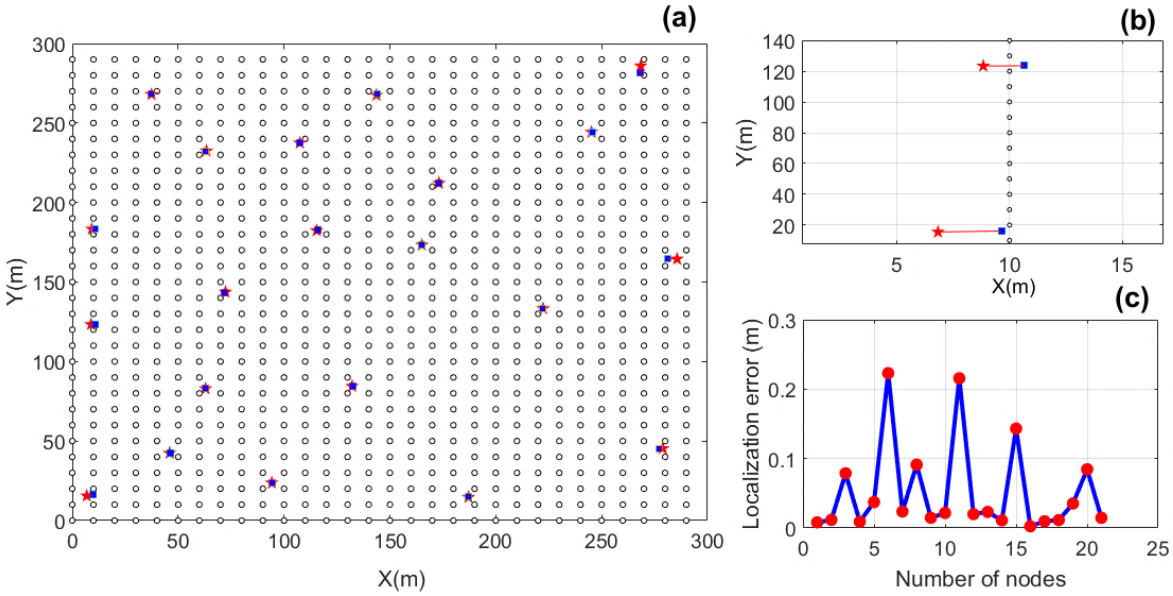

Fig. 12 - Estimated locations for 21 unknown nodes distributed in area 300 x 300m^2 (a) zoom out; (b) zoom in; (c) localization error

4. Conclusion

In this study, an artificial feedforward neural network based on the weighted centroid technique was presented for the localization of wireless sensor nodes. The neural network was constructed and trained to predict the positions of sensing nodes inside grid-based topologies wireless sensor networks with many diverse areas.

The estimated node locations obtained from the FFNN, the DFF neural network, and the WCL method were compared. The results reveal that FFNN surpasses both DFF and WCL in terms of localization accuracy and adaptation to diverse WSN situations. The acquired results show a reliable, accurate, and low-cost localization technique. In the future, the proposed model can be modified to be employed in practical applications in real-world environments.

Acknowledgement

The authors fully acknowledged An-Najah National University for supporting this work.

References

[1] Christin, D., Reinhardt, A., Mogre, P.S., Steinmetz, R., et al., Wireless sensor networks and the internet of things: selected challenges, *Proceedings of the 8th GI/ITG KuVS Fachgespräch Drahtlose sensornetze*, pages 31-34, 2009.

[2] Azam, F.M., and M. N. Ayyaz. Location and position estimation in wireless sensor networks, *In Wireless Sensor Networks: Current Status and Future Trends*, pages 179-214. CRC Press, 2016.

[3] Sharaf, M., Abusair, M., Eleiwi, R., Shana'a, Y., Saleh, I., and Muccini, H., Architecture description language for climate smart agriculture systems, *In Proceedings of the 13th European Conference on Software Architecture-Volume 2*, pages 152–155. ACM, 2019.

[4] Hofmann-Wellenhof, B., Lichtenegger, H., Collins, J., Global positioning system: theory and practice, *Springer Science & Business Media*, 2012.

[5] Khan, H., Hayat, M. N., and Rehman, Z. U., Wireless sensor networks free-range base localization schemes: A comprehensive survey, *In 2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, pages 144–147. IEEE, 2017.

[6] Ijaz, F., Yang, H. K., Ahmad, A. W., and Lee, C., Indoor positioning: A review of indoor ultrasonic positioning systems, *In 2013 15th International Conference on Advanced Communications Technology (ICACT)*, pages 1146-1150. IEEE, 2013.

[7] Vo, Q. D., and De, P., A survey of fingerprint-based outdoor localization, *IEEE Communications Surveys & Tutorials*, 18(1):491-506, 2015.

[8] Rahman, M. S., Park, Y., and Kim, K.-D., RSS-based indoor localization algorithm for wireless sensor network using generalized regression neural network, *Arabian journal for science and engineering*, 37(4):1043-1053, 2012.

[9] Singh, P., Tripathi, B., and Singh, N. P., Node localization in wireless sensor networks, *International journal of computer science and information technologies*, 2(6):2568-2572, 2011.

[10] Meghani, S. K., Asif, M., and Amir, S., Localization of WSN node based on time of arrival using ultra-wide band spectrum, *In WAMICON 2012 IEEE Wireless & Microwave Technology Conference*, pages 1-4. IEEE, 2012.

[11] Bahl, P. Padmanabhan, V.N., Bahl, V., and Padmanabhan, V., Radar: An in-building RF-based user location and tracking system, 2000.

[12] Han, G., Xu, H., Duong, T. Q., Jiang, J., and Hara, T., Localization algorithms of wireless sensor networks: a survey, *Telecommunication Systems*, 52(4): 2419-2436, 2013.

[13] Ramazany, M., and Moussavi, Z., Localization of nodes in wireless sensor networks by mdv-hop algorithm, *ARPN Journal of Systems and Software*, 2(5):166-171, 2012.

[14] Nagpal, R., Shrobe, H., and Bachrach, J., Organizing a global coordinate system from local information on an ad hoc sensor network, *In Information processing in sensor networks*, pages 333-348. Springer, 2003.

[15] Wang, Z.-M., and Zheng, Y., The study of the weighted centroid localization algorithm based on RSSI, *In 2014 International Conference on Wireless Communication and Sensor Network*, pages 276-279. IEEE, 2014.

[16] Y. Li, "Weighted centroid localization algorithm based on mea-bp neural network and dbscan clustering," in Journal of Physics: Conference Series, vol. 2363, p. 012006, IOP Publishing, 2022.

[17] S. Sinha and S. Ashwini, "Rssi based improved weighted centroid localization algorithm in wsn," in 2021 2nd International Conference for Emerging Technology (INCET), pp. 1–4, IEEE, 2021.

[18] K. Akhil, K. Seethalakshmi, and S. Sinha, "Rssi based positioning system for wsn with improved accuracy," in 2021 3rd International Conference on Signal Processing and Communication (ICPSC), pp. 325–329, IEEE, 2021.

[19] M. S. Rahman, Y. Park, and K.-D. Kim, "Rss-based indoor localization algorithm for wireless sensor network using generalized regression neural network," Arabian journal for science and engineering, vol. 37, pp. 1043–1053, 2012.

[20] Stella, M., Russo, M., and Begusic, D., Location determination in indoor environment based on RSS fingerprinting and artificial neural network, *In 2007 9th International Conference on Telecommunications*, pages 301–306. IEEE, 2007.

[21] Gogolak, L., Pletl, S., and Kukolj. D., Neural network-based indoor localization in WSN environments, *Acta Polytechnica Hungarica*, 10(6): 221-235, 2013.

[22] Kumar, S., and Lee, S.-R., Localization with RSSI values for wireless sensor networks: An artificial neural network approach, *In International Electronic Conference on Sensors and Applications*, volume 1. Multidisciplinary Digital Publishing Institute, 2014.

[23] Battiti, R., Le, N.T., and Villani, A., Location-aware computing: a neural network model for determining location in wireless LANS, Technical report, University of Trento, 2002.

[24] Vaghefi, S. Y. M., and Vaghefi, R. M., A novel multilayer neural network model for to a based localization in wireless sensor networks, *In the 2011 International Joint Conference on Neural Networks*, pages 3079-3084. IEEE, 2011.

[25] Xu, L., Wang, K., Jiang, Y., Yang, F., Du, Y., and Li, Q., A study on 2d and 3d weighted centroid *localization* algorithm in wireless sensor networks, *In 2011 3rd International Conference on Advanced Computer Control*,

pages 155-159. IEEE, 2011.

[26] Kong, Q., Yang, X., and Dai, X., Research of an improved weighted centroid localization algorithm and anchor distribution, *In 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pages 400-405. IEEE, 2010.

[27] Zheng, J., Liu, Y., Fan, X., and Li, F., The study of RSSI in wireless sensor networks, *In 2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016),* Atlantis Press, 2016.

[28] Natsheh, E., and Samara, S., Toward better PV panel's output power prediction; a module based on nonlinear autoregressive neural network with exogenous inputs, *Applied Sciences*, 9(18):3670, 2019.

[29] Yu, H., and Wilamowski, B. M., Levenberg-marquardt training. *Industrial electronics handbook*, 5(12):1, 2011.

[30] Liu, H., On the levenberg-marquardt training method for feed-forward neural networks, *In 2010 sixth international conference on natural computation*, volume 1, pages 456–460. IEEE, 2010.

[31] Jayalakshmi, T., and Santhakumaran. A., Statistical normalization and back propagation for classification, *International Journal of Computer Theory and Engineering*, 3(1):1793-8201, 2011.

[32] Samara, S., and Natsheh, E., Intelligent real-time photovoltaic panel monitoring system using artificial neural networks. *IEEE Access*, 7:50287-50299, 2019.

[33] Negnevitsky, M., Artificial intelligence: a guide to intelligent systems. *Pearson education*, 2005.

[34] Tian, H., Shen, H., and Matsuzawa, T., Developing energy-efficient topologies and routing for wireless sensor networks, *In IFIP International Conference on Network and Parallel Computing*, pages 461-469. Springer, 2005.