DETERMINATION OF POSITIONING ACCURACIES BY USING FINGERPRINT LOCALISATION AND ARTIFICIAL NEURAL NETWORKS

Hakan KOYUNCU

Computer Engineering Dept., Istanbul Gelisim University, Istanbul. Turkey

E-mail: hkoyuncu@gelisim.edu.tr

Abstract: *Fingerprint localization technique is an effective positioning technique to determine the object locations by using radio signal strength (RSSI), values in indoors. The technique is subject to big positioning errors due to challenging environmental conditions. In this paper, initially, a fingerprint localization technique is deployed by using classical k-NN method to determine the unknown object locations. Additionally, several artificial neural networks, (ANN), are employed, using fingerprint data, such as SFFNN, MFFNN, MBPNN, GRNN and DNN to determine the same unknown object locations. Fingerprint database is built by RSSI measurement signatures across the grid locations. The construction and the adapted approach of different neural networks using the fingerprint data are described. The results of them are compared with the classical k-NN method and it was found that DNN was the best neural network technique providing the maximum positioning accuracies.*

Key words: *Received Signal Strength indicator (RSSI), k-Nearest Neighborhood (k-NN), Artificial Neural Networks (ANN), Single-Layer Feed Forward Neural Network (SFFNN) Multi-Layer Feed Forward Neural Network (MFFNN), Multi-Layer Back Propagation Neural Network (MBPNN), General Regression Neural Network (GRNN), Deep Neural Network (DNN)*

1. Introduction

Indoor position detection is an important topic to detect the positions of unknown objects [1], [2] in recent years. An ideal positioning should achieve an accurate position detection, a robust construction, fast training stage and a low price. It should give accurate positioning results under difficult environmental conditions where the RSSI propagation is affected by multipath components and signal noise.

Fingerprint localization technique identifies the object location by relying on the RSSI recordings from a test region. k number of closest RSSI values with respect to RSSI values received at object location is selected in that region and their weighted mean value of their coordinates is taken as the object location. Fingerprinting localisation system has two stages: training stage and localisation stage [3], [4]. During the training stage, a database of RSSI values against grid coordinates across the test area is obtained and stored. In the localisation stage, this database is used to locate the unknown object coordinates. One of the most popular algorithms to determine the object locations is the k-nearest neighbourhood algorithm [5]. RSSI values received from unknown object coordinates are incorporated with the RSSI values in the fingerprint database through Euclidean signal distances in this technique. The coordinates which give the smallest Euclidean distances are employed to calculate the unknown object locations [6].

Artificial Neural Networks, [7], are deployed on the fingerprint databases to predict the unknown object coordinates. There are several artificial neural network techniques such as *SFFNN*, *MFFNN* [8], *GRNN* [9], *MBPNN*, [10], and *DNN*, [11], to calculate the unknown object positions.

Presently, positioning accuracies of physical objects are calculated by using RF waves in indoor areas and several NN techniques. In this study an overview of NN techniques are employed. Similarly, these NN techniques can also be deployed in thermal sciences [12]. Thermal data of indoor environments can be measured with the help of wireless sensor nodes and used to calculate real time object locations by employing neural networks. Furthermore, static and dynamic thermal problems, dynamic thermal control, prediction of thermal conductivity and dynamic viscosity are a few examples of using Neural Networks [13].

Initially, fingerprint localization method with k-NN is employed in this work and unknown object locations are determined. Previously mentioned neural network techniques are also deployed to determine the same unknown object locations. Finally, Positioning accuracies are calculated and compared among all these techniques to find the most accurate of them.

In section 1, a general introduction is given. Starting with section 2, Fingerprint technique and construction of fingerprint database is presented. In section 3, artificial neural network techniques are described. Their architectures and the algorithms used during the learning processes are given. In section 4, experimental methodology and results are presented. Finally, the results are analysed, and discussions are given in section 5.

2. Fingerprint Localization Technique

2.1. Fingerprint theory

Fingerprint localization technique determines the unknown object locations by using a set of RSSI values at object locations and comparing them with a pre-built RSSI fingerprint map. A general map of example fingerprint area is presented in Fig. 1.

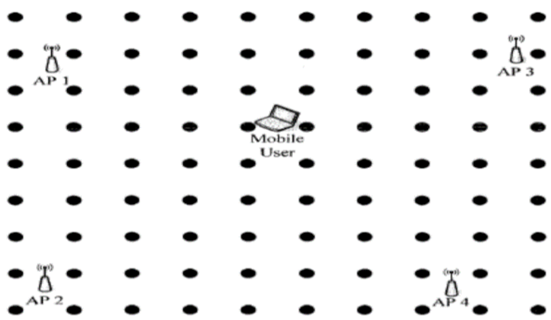

Figure 1: **A test area of 4 transmitters, a grid of RSSI measurement points and a mobile receiver**

Measurement points in a grid formation depending on the area topology is organized. WSN transmitters stationed at several points provide RF propagation over the entire grid area. During the training stage, RSSI values are received from all the transmitters at each grid point and recorded against the grid coordinates in a database.

RSSI values at measurement points contain random behaviour due to environmental conditions. Therefore, average value of multiple RSSI measurements is deployed as the RSSI fingerprint at that point. RSSI measurement values are received at each grid point from T_i transmitters. There are N number of transmitters. These measurement values are written in a database and defined as "*location fingerprint RSSI vector"* F_j. There are M number of grid points and F_j at j^{th} grid point is given as:

$$F_j = RSS_{T_1}^{j}, RSS_{T_2}^{j}, RSS_{T_3}^{j}, \ldots . RSS_{T_N}^{j} \tag{1}$$

A sample of RSSI measurements are recorded on the object receiver and identified as "*sample RSSI vector* "R_k. There are Z number of object points and R_k at k^{th} object point is given as:

$$R_k = RSS_{T_1}^{k}, RSS_{T_2}^{k}, RSS_{T_3}^{k}, \ldots . RSS_{T_N}^{k} \tag{2}$$

The difference between the "*location fingerprint RSSI vector"* F_j and *"sample RSSI vector" R_k* are used to calculate the nearest grid points to the object point. Euclidean distance, E_k , is identified as the signal distance and it is defined in Eq. (3).

$$E_k = \sqrt{\sum_{j=1}^{M} (F_j - R_k)^2} \tag{3}$$

Euclidean distance defines the distance between two locations with respect to radio signal strengths. Therefore, weakest signal strength difference refers to the smallest physical distance between two locations [14]. Minimum Euclidean distances corresponding to minimum physical distance between grid and object locations are selected for object location calculations.

2.2. k-NN Algorithm

k number of grid points are determined by k-NN algorithm in Euclidean database. Mean value of these grid coordinates are calculated to give the estimated object location.

Furthermore, a weighting scheme is introduced to increase the accuracy of object localization with k-NN algorithm. Hence weight functions, [15], are deployed to compensate the RSSI variations. Different weight functions can be utilized with respect to changing indoor conditions during positioning calculations.

Weights and distances are inversely proportional in many applications. Distant transmitters can have a dominant effect during localisation. Estimated positions move towards the nearest transmitter locations and the positioning errors increase. Finally, estimated object coordinates are calculated by using weight function w_j, and grid coordinates (x_j,y_j), defined by k-NN algorithm. See Eq. (4).

$$(x,y)_{estimated} = \sum_{j=1}^{k} w_j (x_j, y_j) \tag{4}$$

w_j is the weight function of the j^{th} grid point in k-nearest neighbourhood. The selection of weight functions effects the positioning accuracy. Hence w_j is formulated to express this dependency with D_j Euclidean distances of k-nearest neighbourhood values. Empirical weight function in Eq. (5) is deployed in order to estimate the unknown object coordinates [16].

$$w_j = \frac{1}{D_j^2} \Bigg/ \sum_{j=1}^{k} \frac{1}{D_j^2} \tag{5}$$

3. Artificial Neural Networks

Neural network is a numerical modelling technique that mimics biological brains. A Neural network can initially be trained on a sample of known dataset. It determines the future similar patterns for unknown data which it has never seen before. Neural network can change its structure depending on internal and external information during their flow through the network in training phase. Neural networks can be considered as non-statistical data modelling techniques [17], [18]. They are employed to show the relation between input and output data.

3.1. FFNN

Feed-Forward Neural Network is basically a straight forward network which relates the input and output functions [19]. In this technique, the signals travel through the network between input and output layers without any feedback. The performance of the neural network is related to its input-output transfer function. Weights of the neurons reduce the errors between estimated and actual outputs. Sigmoid function is employed for activation during the training stage to transfer the input data to output layer [20].

FFNN consist of neurons which are constructed in layers. It generally has 1 input layer, 1 output layer. Neurons in the input layers are defined as input data and Neurons in the output layer are defined as output data. Hidden layers, between the input and output layers, transform the input data to generate the target output data. Neural networks calculate the error derivative of the weights and back-propagation algorithm, [21], is the best method to derive these error derivatives.

A description of the back-propagation multilayer network can be given in Fig. 2. There are ***m*** layers in the multilayer network. K_j is the number of neurons in j^{th} layer. Network has P number of patterns of training set with K_0 dimensional X inputs. K_m dimensional T targets are used in training of the network. The network O output response to the input and target training patterns is given as

$$O_{P1}, O_{P2}, O_{P3} ...... O_{PKm}$$

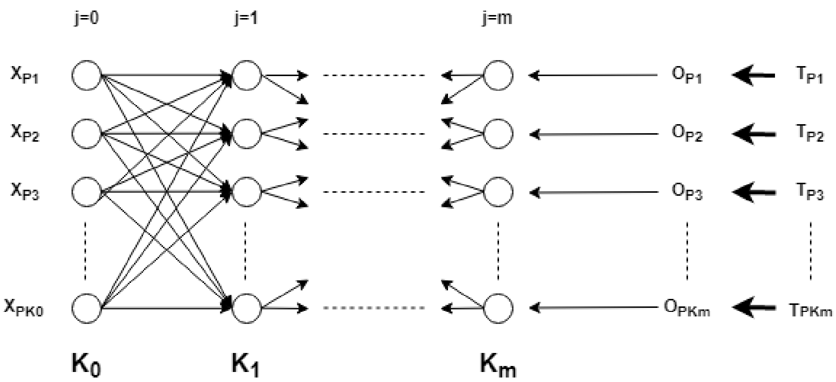

Figure 2: General Feed Forward and Back Propagation Neural Networks

Finally, different neural network techniques are employed, and the most accurate technique is determined. Firstly, SFFNN technique is used. Its neural network structures is shown in Fig. 3**.** The neurons in the input and output layers represent the input and output data respectively. The hidden layer between input and output layer carries out the transformation of input data to generate the target output data.

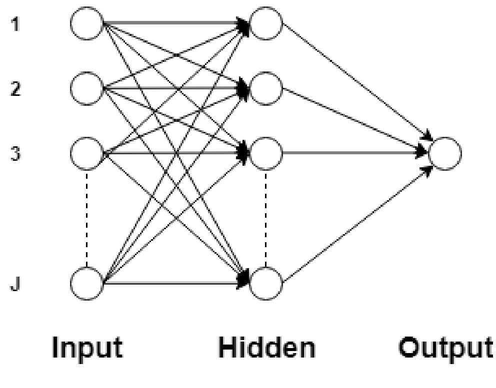

Figure 3: SFFNN Architecture

Target Output data is the (x, y) position coordinates of grid points across the test area. There are j fingerprint signatures which are defined as the RSSI values received from j transmitters at each grid location. Each neuron in a layer is connected to all the neurons in the next layer. Each of these connections is related by its weight, w_{ij}^{k} , where index *i* is the neuron number of the (k-1) layer and j is the neuron number of k^{th} layers. The output value of a neuron is multiplied by the related weight and added to the signal value of the neuron in the next layer. Generally, the number of neurons in hidden layer is less than or equal to the size of the input layer and more than the size of the output layer. In this study, the size of hidden layer is taken as equal to the size of input layer as shown in Fig. 3. Discrete back-propagation algorithm can be applied to SFFNN if required. It is deployed to determine the weights and adjust them for each neuron in the hidden layer. This is a supervised learning method. The weights are adjusted several times in each epoch during the training.

Secondly, MFFNN, multi-layer Feed-Forward artificial neural network technique is deployed. It includes one input layer, one output layer and several hidden layers as shown in Fig. 4**.** MFFNN technique is like SFFNN with the exception that the number of hidden layers is more than 1.

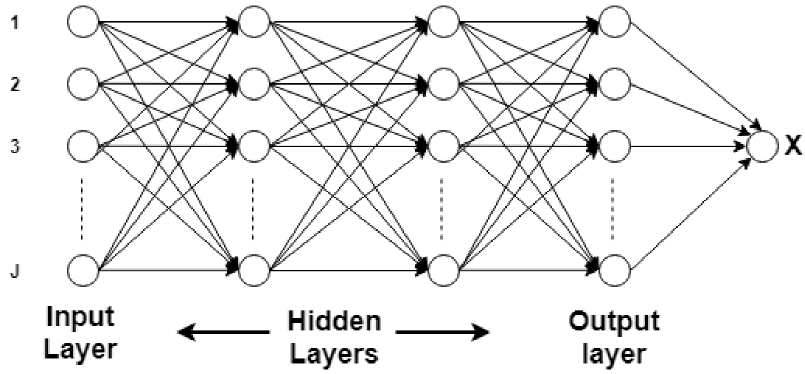

Figure 4: MFFNN architecture

Thirdly, back propagation method, MBPNN, can also be employed between hidden layers. This is identified as Multiple Back Propagation Neural Network technique, (MBPNN).

The i^{th} neuron output in k^{th} layer in the network can be determined as:

$$y_i^k = g(\sum_{i=1}^{n_{k-1}} w_{ij}^k y_i^{k-1} + T_i^k) \tag{6}$$

Where **g** is the sigmoid activation function. It is used to model the nonlinear relationship between input and output layers. T_i^k is the threshold value added to the input.

Back propagation method is used to calculate the final weights of neurons in the network. The error, calculated in the output layer neurons, is than back propagated to hidden layers by deploying the Gradient Descent method [22]. This minimizes the squared-error cost function in neurons. The sum of the squared-errors of cost function is given as:

$$E_p = \frac{1}{2}\sum_{k=1}^{m}(h_{kp} - O_{kp})^2 \tag{7}$$

Where **p** is the number of training pattern, **O** is the actual output; **h** is the predicted value with size **m**.

Similarly, j number of RSSI signatures is collected from j number of transmitters at each grid point. These signatures are employed as the input to the training network. The outputs of the network are defined as the (x, y) coordinates of the grid points and they are used as the 2 targets during the training. Once the training phase is completed, any j numbers of RSSI signatures received from an unknown object location are applied as input to MFFNN and the output (x, y) coordinates are calculated. The prediction errors between the actual object coordinates and the calculated object coordinates are determined with different ANN techniques. Average error calculations between the ANN techniques are carried out to find the most accurate technique.

3.2. GRNN

GRNN has a one pass learning algorithm, [23]. The regression equation is considered a parallel neural network like structure. If the joint probability density function (pdf) of x and y is known and z coordinate is omitted, the conditional pdf and the expected value can be computed. Parameters of structures can be determined from examples rather than iteratively. The structure learns and starts to generalize. It can estimate y values for any new x value during the propagation time. Suppose that f (x, y) represents the joint continuous probability density function of a vector random variable x and scalar random variable y. The regression of y on x or conditional mean of y given x is defined by:

$$R(y|x)y = \frac{\int_{-\infty}^{\infty} y f(x,y)dy}{\int_{-\infty}^{\infty} f(x,y)dy} \tag{8}$$

When $f(x,y)$ is unknown, it can be estimated from a sample of x and y data. A class of consistent estimators is proposed by Parzen, [24], which can be applicable in the multi-dimensional case. The probability estimator $f(x,y)$ is based on sample values of x^i and y^i of random variables of x and y and can be estimated from the training set by using the Parzen estimator below,

$$f(x,y) = \frac{1}{(2\pi)^{(p+1)/2}\sigma^{(p+1)}} \cdot \frac{1}{n}\sum_{i=1}^{n} \exp\left[-\frac{(x-x^i)^T(x-x^i)}{2\sigma^2}\right] . \exp\left[-\frac{(y-y^i)^2}{2\sigma^2}\right] \tag{9}$$

n is the number of sample observations and p is the dimension of the vector variable x. $f(x,y)$ assigns a sample probability of width σ for each sample of x^i and y^i. Probability estimate becomes the total sum of these sample probabilities. Placing the joint probability estimate $f(x,y)$ into conditional mean in Eq. (8) generates the desired conditional mean y for given x. See Eq. (10);

$$y(x) = \frac{\sum_{i=1}^{n} \exp\left[\frac{(x-x^i)^T(x-x^i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} y \exp\left[-\frac{(y-y^i)^2}{2\sigma^2}\right] dy}{\sum_{i=1}^{n} \exp\left[\frac{(x-x^i)^T(x-x^i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} \exp\left[-\frac{(y-y^i)^2}{2\sigma^2}\right] dy} \tag{10}$$

The term $(x-x^i)^T(x-x^i)$ in Eq. (10) can be defined as a scalar function W_i^2 and Eq. (10) becomes:

$$y(x) = \frac{\sum_{i=1}^{n} y^i \exp\left(-\frac{w_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{w_i^2}{2\sigma^2}\right)} \tag{11}$$

The resulting regression in Eq. (11) deals with the summations of the observations. The estimate y(x) can be considered as the weighted average of observed values, y^i. Each observed value is exponentially weighted according to its Euclidean distance from x.

3.3. DNN

A deep neural network, DNN, is an artificial neural network with multiple hidden layers between input and output layers, [25]. DNN is a feed forward network. Data flows from input layer to output layer without looping back. Deep learning methods target at learning feature hierarchies by using features from higher level hierarchy formed by the lower level features. They have learning methods of deep architectures with neural networks of many hidden layers.

There are two important issues with training DNNs. These are overfitting and computation time. Overfitting takes place due to added layers of abstraction which allow to model simple dependences in the training data. Large number of data is used in DNN and computation time is longer compared to shallow ANNs. DNNs use training parameters such as network size, learning rate and initial weights.

The most important advantage of deep learning is replacing handmade features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction. In this study 20 hidden layers with one input and one output layers are employed for DNN. Similarly, RSSI values received at grid points are used as the inputs and the grid point coordinates are the outputs of the neural network for training. "Neural Designer "software is deployed for DNN work.

4. Experiments

4.1. Fingerprint Localization

Fingerprint dataset which are used in this study is recorded in a sports hall with dimensions of 20 meters by 10 meters and a grid space of 1 meter. 8 RFCODE manufactured wireless sensor nodes are used as transmitters at the corners and the boundaries of the hall. An RFCODE wireless sensor receiver is employed to collect RSSI data by placing it at each grid point during training stage.

The collection of 8 RSSI values in dB form at each grid point is recorded in a dataset identified as fingerprint map in a server computer. There are total 200 fingerprint points and 1600 single recordings of RSSI values in the fingerprint map. Fingerprint map is extended by obtaining 100 sets of 8 RSSI recordings totalling 800 RSSI values at each grid point. Hence, fingerprint map contains 160000 RSSI values. A sample data set of single RSSI values with respect to (x, y) grid coordinates are presented for the reader in Tab. 1. Similarly, 800 RSSI values received at each unknown object location are also recorded across the test area as the target RSSI values during localisation stage. They are used to calculate the unknown object location coordinates by deploying the trained neural networks.

Table 1: A sample of recorded single RSSI values in decibel form from 8 transmitters against grid coordinates (x, y).

(X,Y)	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)	(6,0)	(7,0)	(8,0)	(9,0)	10,0)	(11,0)	(12,0)	(13,0)
A	-59	-63	-45	-58	-59	-63	-45	-58	-55	-47	-60	-62	-44	-41
B	-59	-63	-46	-55	-57	-61	-48	-56	-54	-49	-56	-58	-45	-40
C	-58	-61	-50	-54	-56	-62	-49	-58	-54	-50	-56	54-	-46	-45
D	-57	-59	-49	-53	-57	-61	-47	-55	-52	-53	58-	-55	-45	-42
E	-58	-58	-47	-55	-58	-64	-42	-56	-54	-56	-59	-54	-47	-43
F	-59	-57	-44	-56	-59	-62	-44	-57	-58	-54	-56	-53	-48	-44
G	-57	-55	-45	-55	-58	-63	-47	-59	-57	-55	-57	-54	-49	-43
H	-59	-63	-47	-54	-59	-60	-45	-58	-57	-56	-58	-55	-48	-41

Initially, classical fingerprint localization technique is employed by using Euclidean distances between RSSI values at grid points and unknown object points. k number of minimum Euclidean distances are selected by using k-nearest neighbourhood algorithm (k-NN). Weighting mechanism in Eq. (6) is applied with the grid coordinates corresponding to k number of Euclidean distances. Finally, Eq. (5) is used to determine the final unknown object location coordinates. Error distance values are calculated between the actual and the estimated object coordinates. A sample of k-NN localization results is tabulated in **Tab. 2**. Average error distance with k-NN localization is approximately equal to 1 grid space.

4.2. SFFNN localization

Single Feed-Forward Neural Network model is employed to determine the unknown object locations. RSSI values arriving from 8 transmitters are recorded at each grid point and stored in a database. There are 100 RSSI values coming from each transmitter resulting a total of 800 RSSI values at each grid point.

Table 2: Samples of k-NN, SFFNN and MFFNN localization results

Unknown Object coordinates (m)		**k-NN** estimated object coordinates (m)			**SFFNN** estimated object coordinates (m)			**MFFNN** estimated object coordinates (m)		
X_0	Y_0	X_E	Y_E	Error	X_E	Y_E	Error	X_E	Y_E	Error
2	2	2.9	2.8	1.20	2.7	2.7	0.99	2.6	2.5	0.78
2	8	2.8	8.7	1.06	2.6	8.5	0.78	2.5	8.7	0.86
3	4	3.7	4.9	1.14	3.5	4.5	0.71	3.6	4.4	0.72
5	7	5.8	7.7	1.06	5.4	7.6	0.72	5.5	7.4	0.64
6	3	6.7	3.8	1.06	6.5	3.6	0.78	6.4	3.7	0.81
8	5	8.9	5.9	1.27	8.6	5.6	0.85	8.6	5.5	0.78
12	3	12.7	3.6	0.92	12.5	3.6	0.78	12.6	3.7	0.92
14	8	14.8	8.7	1.06	14.6	8.5	0.78	14.5	8.6	0.78
16	5	16.9	5.5	1.02	16.6	5.9	1.08	16.4	5.9	0.98
18	7	18.6	7.9	1.08	18.5	7.7	0.86	18.6	7.5	0.78
		Avg. error (m)		1.08	Avg. error (m)		0.83	Avg. error (m)		0.81

Hence these recordings for 200 grid points are used to train the neural network. Neural network is an adaptive network system. It can change its structure depending on external and internal information which flows through the network during the training phase. In SFFNN, the signals travel from input to output without any feedback [26]. Once the neural network has been trained on samples of the known fingerprint database it will be able to predict the similar unknown patterns of future input RSSI data. The behaviour of the neural network depends on the input-output transfer function. During the learning process, sigmoid activation function is employed to translate input signal to output signal. See Fig. 5**.**

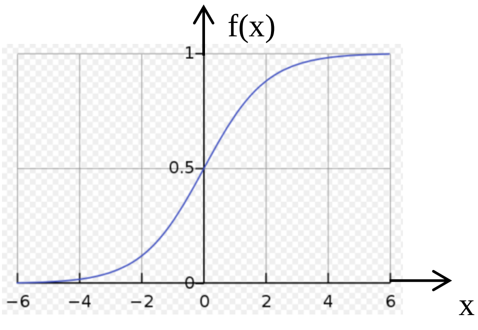

Figure 5: **Sigmoid activation function**

Training model during learning process will have 100 sets of 8 RSSI data, 2 outputs of grid (x, y) coordinates and 1 hidden layer of 8 neurons. Fig. 3 displays the architecture of SFFNN. Once the neural network training is completed, 8 RSSI values received from unknown object location are applied at the inputs of the neural network and the output coordinates will be obtained as the target object coordinates.

100 sets of 8 RSSI values are applied at the 8 inputs and 100 target coordinates are applied at the output of SFFNN. The network is trained with these values. In localisation stage, 8 unknown RSSI values are applied to the input of the trained network. Unknown object coordinates of (x, y) is generated at the network output. A sample of results is tabulated in Tab. 2.

4.3. MFFNN localization

Multiple Feed-Forward Artificial Neural Network model is similar to SFFNN with the exception that it has multiple hidden layers between input and output layers, [27]. Input signal moves to output through several hidden layers. In this study 8 inputs, 5 hidden layers of 8 neurons each and 2 outputs of (x, y) coordinates are deployed for MFFNN localizations. Similar training process is completed where 100 sets of 8 RSSI values are applied as inputs and (x, y) coordinates of each grid point are given as 2 outputs. Input RSSI values passed through 5 hidden layers and ended at 2 outputs corresponding to grid coordinates. Same sigmoid activation function is used during the learning process to translate input signals to output signals. 8 RSSI values arriving from unknown object locations are applied to trained network inputs and target output coordinates are generated at the network outputs. A sample of MFFNN localization results are given in Tab. 2. It was observed that there is not much error distance difference between SFFNN and MFFNN.

4.4. MBPNN localization

Multiple Back-Propagated Neural Network is a combination of several neural network models, [28]. It has multiple hidden layers between input and output layers. Feed-forward networks are straight forward network that relates the inputs and outputs. In order to make the neural networks suitable in specific tasks, the connections between neurons must be chosen carefully and the weights on the connections must be selected properly. The signals, reaching to output layer, are back propagated to input layer. This forward and backward propagation between input and output layers several times helps to correct the weights between neurons. This way the error between the desired output and the real output will be reduced. The network must determine that how the error changes as the weight changes. Hence back propagation algorithm is the most widely used algorithm to calculate the error derivative of weights. Fig. 2 illustrates the back–propagation model neural network. Learning process is carried out similarly by training the network with input RSSI values and the output grid coordinates. RSSI values arriving from unknown object location are applied as input and the output target coordinates are derived. A sample of MBPNN localization results are given in Tab 3.

Table 3: Samples of MBPNN, GRNN and DNN localization results

Unknown Object coordinates (m)		**MBPNN** estimated object coordinates (m)			**GRNN** estimated object coordinates (m)			**DNN** estimated object coordinates (m)		
X_0	Y_0	X_E	Y_E	Error	X_E	Y_E	Error	X_E	Y_E	Error
2	2	2.4	2.5	0.64	2.4	2.6	0.72	2.4	2.4	0.57
2	8	2.5	8.6	0.78	2.4	8.4	0.56	2.5	8.4	0.64
3	4	3.6	4.5	0.78	3.5	4.5	0.70	3.5	4.4	0.64
5	7	5.4	7.6	0.72	5.4	7.5	0.64	5.4	7.4	0.57
6	3	6.4	3.5	0.64	6.4	3.5	0.64	6.3	3.5	0.58
8	5	8,5	5.4	0.64	8.4	5.5	0.64	8.3	5.5	0.58
12	3	12.5	3,6	0.78	12.4	3.4	0.57	12.2	3.5	0.54
14	8	14.5	8.4	0.64	14.4	8.5	0.64	14.3	8.5	0.58
16	5	16.4	5.7	0.81	16.5	5.5	0.70	16.3	5.4	0.5
18	7	18.5	7.4	0.64	18.4	7.6	0.72	18.3	7.5	0.58
		Avg. error (m)		0.71	Avg. error (m)		0.65	Avg. error (m)		0.58

4.5. GRNN localization

This localization depends on the joint probability density function of x and y position coordinates given the RSSI training dataset [29]. Estimated value is the most probable value of y represented by Eq. (8). Basic statistics of the input variables and the target variables which are used in Eq. (10) are given in Tab. 4.

Generalized regression neural networks toolbox of Matlab is used to generate the GRNN model and carry out the training on input dataset. A sample of GRNN localization results are presented in Tab. 3.

Table 4: **Overall statistical values of input dataset for each Transmitter across the fingerprint map.**

Tx/RSSI	Min dB	Max dB	Mean dB	Std σ
A	-92	-59	-60	4.64
B	-90	-57	-61	4.59
C	-94	-62	-65	4.37
D	-91	-62	-65	4.32
E	-96	-45	-52	7.17
F	-94	-45	-49	6.90
G	-90	-58	-62	7.21
H	-88	-58	-62	7.11

4.6. DNN localization

In this study, deep neural network is considered as a feed-forward neural network. It has large number of hidden layers, 20 of them, with 8 neurons each. 1 input layer of 8 neurons and 1 output layers of 2 neurons are also deployed. Neural designer software has employed to determine the unknown object coordinates. A sample of DNN localization results is presented in Tab. 3.

5. Results

Fingerprint database is built across a test area and it is used as training data for various neural network models. This paper has presented a measurement and neural Network simulation-based localization study. A summary of localization accuracies is presented in Tab. 5.

Table 5: Summary of average localization accuracies

Average Localization Accuracies					
k-NN	**SFFNN**	**MFFNN**	**MBPNN**	**GRNN**	**DNN**
1m	0.83m	0.81m	0.71m	0.65m	0.58m

Initially classical weighted fingerprint localization technique is implemented. Positioning accuracy of a grid distance is calculated between the actual and estimated object positions.

There are 200 training points across the test area with 1 metre distances between them. Unknown object locations are estimated using the learned network with these training points. There are 8 anchor points where the RF transmitters are located. As a result, there are 8 RSSI values recorded at each training grid point. In this study, large numbers of unknown object locations are tested to estimate their positions.

Overview of several neural network models such as SFFNN, MFFNN, MBPNN, GRNN and DNN are presented for the application of fingerprint localisation. Each model has generated its own localization accuracies. It was observed that as the number of hidden layers is increased the localization accuracies are also increased. Graphical representation of target object coordinates and the estimated target object coordinates with different neural network techniques are given in Fig. 6.

SFFNN and MFFNN models did not show large accuracy difference between them. On the other hand, MBPNN and GRNN models have displayed considerable localization accuracy, around 30% more, compared to classical k-NN technique. The best neural network model was found to be DNN model.

Large number of hidden layers is employed in this model. The localization accuracies with DNN model increased %50 relative to k-NN model.

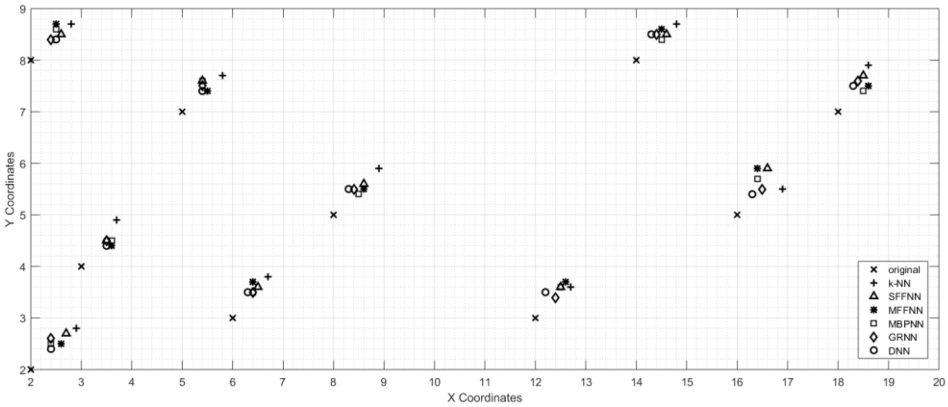

Figure 6: Graphical view of sample target coordinates and estimated target coordinates with different neural network techniques.

RSSI based neural network simulations have shown that localization accuracies are drastically improved compared to classical k-NN technique. A fair size database of 200 points is sufficient enough to achieve good localization accuracies. Future work will consist of advance DNN studies with different number of hidden layers and introduction of back propagation between these layers using different applications..

REFERENCES

[1] Bahl, P., Padmanabhan , V. N., RADAR: an in-building RF-based user location and tracking system, *Proceedings* IEEE INFOCOM 2000 Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), Tel Aviv, Israel, 2000, Vol *2*., pp. 775-784, doi: 10.1109/INFCOM.2000.832252

[2] Ansari , J., Riihijarvi , J. , Mahonen, P., Combining Particle Filtering with Cricket System for Indoor Localization and Tracking Services, 2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, Athens, Greece, 2007, pp. 1-5, doi: 10.1109/PIMRC.2007.4394578

[3] Le Dortz, N., Gain , F. , Zetterberg, P., WiFi fingerprint indoor positioning system using probability distribution comparison, 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 2012, pp. 2301-2304. doi:10.1109/ICASSP.2012.6288374

[4] Wang, X., Gao, L., Mao ,S., PhaseFi: Phase Fingerprinting for Indoor Localization with a Deep Learning Approach, 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015, pp. 1-6 ,doi: 10.1109/GLOCOM.2015.7417517

[5] Ilias , B., Shukor, S. A. A., Adom, A. H., Rahim, N. A., Ibrahim , M. F., Yaacob, S., Indoor mobile robot localization using KNN, 2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Batu Ferringhi, Malaysia, 2016, pp. 211-216, doi: 10.1109/ICCSCE.2016.7893573

[6] Gansemer, S., Großmann, U., Hakobyan, S., RSSI-based Euclidean Distance algorithm for indoor positioning adapted for the use in dynamically changing WLAN environments and multi-level buildings, 2010 International Conference on Indoor Positioning and Indoor Navigation, Zurich, 2010, pp. 1 – 6, 10.1109/IPIN.2010.5648247

[7] Ibrahim , A., Rahim ,S. K. A. , Mohamad, H., Performance evaluation of RSS-based WSN indoor localization scheme using artificial neural network schemes, 2015 IEEE 12th Malaysia International Conference on Communications (MICC), Kuching,Malaysia, 2015, pp. 300-305, doi: 10.1109/MICC.2015.7725451

[8] Hayashi ,Y., Sakata, M., Gallant, S.I., Multi-Layer Versus Single-Layer Neural Networks and an Application to Reading Hand-Stamped Characters, International Neural Network Conference, 1990, vol. 2, pp. 781-784

[9] Rahman , M.S., Park ,Y., Kim , K. D., RSS-Based Indoor Localization Algorithm for Wireless Sensor Network Using Generalized Regression Neural Network, *Arabian Journal for Science and Engineering*, 37(2012), 4, pp 1043–1053

[10] Zbeda , R. , Nathan, P., Multilayer neural network with back propagation: hardware solution to learning XOR, *Journal of Computing Sciences in Colleges*, 20(2005), 5, pp. 144-146

[11] Xiao ¸ L., Behboodi , A. , Mathar , R., A deep learning approach to fingerprinting indoor localization solutions, 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, Australia, 2017, pp. 1-7, doi: 10.1109/ATNAC.2017.8215428
[12] Yang, K., Artificial Neural Networks (ANNs): A New Paradigm for Thermal Science and Engineering, ASME. J. Heat Transfer, 130(2008), 9, pp. 093001-1 - 093001-19, doi:10.1115/1.2944238.
[13] Esfe ,M. H., Saedodin , S., Sina , N., Afrand , Rostami, M., S., Designing an artificial neural network to predict thermal conductivity and dynamic viscosity of ferromagnetic nanofluid, International Communications in Heat and Mass Transfer, 68(2015),pp. 50-57
[14] Halsted , T. , Schwager, M., Distributed multi-robot localization from acoustic pulses using Euclidean distance geometry, 2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), Los Angeles, CA, USA, 2017, pp. 104-111, doi: 10.1109/MRS.2017.8250938
[15] Peng , Y., Fan , W., Dong , X. , Zhang , X., An Iterative Weighted KNN (IW-KNN) Based Indoor Localization Method in Bluetooth Low Energy (BLE) Environment, 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, 2016, pp. 794-800
[16] Koyuncu , H., Yang , S. H., Comparison of Indoor localization techniques by using reference nodes and weighted k-NN algorithms, *Recent Advances in Information Science*, 8(2012), pp. 46-5, ISBN: 978-1-61804-140-1
[17] El Assaf ¸ A., Zaidi , S., Affes , S. , Kandil, N., Robust ANNs-Based WSN Localization in the Presence of Anisotropic Signal Attenuation, *in IEEE Wireless Communications Letters*, 5(2016), 5, pp. 504-507, doi: 10.1109/LWC.2016.2595576
[18] Borenovic , M. , Neskovic, A., ANN based models for positioning in indoor WLAN environments, 2011 19thTelecommunications Forum (TELFOR) *Proceedings of Papers*, Belgrade, Serbia, 2011, pp. 305-312 ,doi: 10.1109/TELFOR.2011.6143551
[19] Zouari , R., Zayani , R. ,Bouallegue, R., Indoor localization based on feed-forward Neural Networks and CIR fingerprinting techniques, 2014 IEEE Radio and Wireless Symposium (RWS), Newport Beach, CA, 2014, pp. 271-273, doi: 10.1109/RWS.2014.6830093
[20] Murugadoss, R. , Ramakrishnan, M., Universal approximation of nonlinear system predictions in sigmoid activation functions using artificial neural networks, 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 2014, pp. 1-6, doi: 10.1109/ICCIC.2014.7238539
[21] Cruz-López , J. A., Boyer , V. , El-Baz, D., Training Many Neural Networks in Parallel via Back-Propagation, 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Lake Buena Vista, FL, 2017, pp. 501-509, doi: 10.1109/IPDPSW.2017.72
[22] Nayak , J., Naik ,B. , Behera, H. S., A hybrid PSO-GA based Pi sigma neural network (PSNN) with standard back propagation gradient descent learning for classification, 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, India, 2014, pp. 878-885, doi: 10.1109/ICCICCT.2014.6993082
[23] Specht , D. F., A general regression neural network, in IEEE Transactions on Neural Networks, 2(1991), 6, pp. 568-576, doi: 10.1109/72.97934
[24] Duin , R. P. W., On the Choice of Smoothing Parameters for Parzen Estimators of Probability Density Functions, in IEEE Transactions on Computers, C-25(1976),11, pp. 1175-1179, doi: 10.1109/TC.1976.1674577
[25] Lee , N. , Han, D., Magnetic indoor positioning system using deep neural network, 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN*)*, Sapporo, Japan, 2017, pp. 1-8. doi: 10.1109/IPIN.2017.8115887
[26] Wang , R., Kwong, S., Jiang ,Q. , Wong, K. C., Active Learning Based on Single-Hidden Layer Feed-Forward Neural Network, 2015 IEEE International Conference on Systems, Man, and Cybernetics, Kowloon, 2015, pp. 2158-2163, doi: 10.1109/SMC.2015.377
[27] Youssefi , B., Mirhassani , M. , Wu, J., Efficient mixed-signal synapse multipliers for multi-layer feed-forward neural networks, 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), Abu Dhabi, UAE, 2016, pp. 1-4, doi: 10.1109/MWSCAS.2016.7870144
[28] Keller , J. M., Liu , D., Fogel , D. B., Multilayer Neural Networks and Backpropagation, in Fundamentals of Computational Intelligence:Neural Networks, Fuzzy Systems, and Evolutionary Computation , 1(2016), pp.400-405
[29] Cui , H. , Tu , N., Generalized Regression Neural Networks Based HVDC Transmission Line Fault Localization, 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 2015, pp. 25-29, doi: 10.1109/IHMSC.2015.103