**PAPER** • **OPEN ACCESS**

# The Indoor Positioning System Using Fingerprint Method Based Deep Neural Network

To cite this article: R F Malik *et al* 2019 *IOP Conf. Ser.: Earth Environ. Sci.* **248** 012077

# The Indoor Positioning System Using Fingerprint Method Based Deep Neural Network

**R F Malik[1*], R Gustifa[1], A Farissi[1], D Stiawan[1], H Ubaya[1], M R Ahmad[2] and A S Khirbeet[2]**

[1]Communication Network and Information Security Research Lab, Faculty of Computer Science, Universitas Sriwijaya, Jalan Srijaya Negara Bukit Besar, Bukit Lama, Ilir Bar. I, Palembang, Sumatera Selatan 30128 - Indonesia
[2]Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka, Jalan Hang Tuah Jaya, 76100 Durian Tunggal, Melaka - Malaysia

*rezafm@unsri.ac.id

**Abstract.** Highly dynamic indoor environments being one of the challenge in the Indoor Positioning System (IPS). Collecting the Received Signal Strength (RSS) value from every Wi-Fi access point known fingerprint method is presented by previous researchers. They proposed with different techniques in fingerprint methods to compete similar existing technology such as GPS in term of accuracy. The drawback using fingerprint is the IPS cannot maintain the high performance constantly. In this research, we propose the Deep Neural Network (DNN) algorithm for improving the fingerprint method in the IPS. Basically, the fingerprint method consists of two phases, Online and Offline phases. In the off-line, RSS values will be collected from several coordinates as known reference points and stored in the database. The online phase has different step which the current position will be compared to RSS values stored in the database. The DNN method was used to calculate the closest position estimation probability. The IPS using DNN was successfully applied using 5 layers consisting of a 1 input layer, 3 hidden layers and 1 output layer. The input and hidden layer have 28 nodes for each layers and output layer has 2 nodes. The simulation results from RSS data set has achieved 2 meters accuracy. It concluded that DNN performance depends on the number of hidden layers and the number of nodes in each hidden layer.
Keywords: Indoor Positioning System, Fingerprint method, Deep Neural Network

## 1. Introduction
The Indoor Positioning System (IPS) is a system that makes the location as an entity and estimate the object location. Location Based Service (LBS) is one of application use the IPS and integrating with existing wireless technology for indoor and outdoor environments [1]. Z.Liu, X. Luo, T. He in their paper improve indoor position estimation system using Weight K-Nearest Neighbour (W-KNN) algorithm, the result obtained the positioning error value was better 0.1% than original algorithm with accuracy 2.438 m [2]. Technology of indoor position estimation system becomes popular research, both the level of accuracy and the method used. because based statistics 80% - 90% of people spend time at indoors. Accuracy in the position estimation system was expected to be higher than previous research [3] [4]. Accuracy of position estimation system at outdoor has an average value 10 meters which implemented by Global Positioning System (GPS) technology [5]. It is a reason why an indoor position estimation system has a large enough market potential to be developed, if the accuracy can be improved

more less than outdoor. In this paper, the IPS is use fingerprint method and applied it into Deep Neural Network (DNN) algorithm. The DNN algorithm will apply to solve a lack of accuracy in building with architectural complexity including the faraday cage effect. It is why this paper will use the DNN which is part of the Deep Learning algorithm to reduce the weakness.

## 2. The Indoor Positioning System (IPS)

At first IPS used for location estimation for cell-based phones, whose the accuracy was denote by the size of the cell [3][6]. But gradually, this accuracy is no longer suitable for use, and there are new approaches such as network-based (TOA, E-OTD), handset-based (GPS) or hybrid approach (A-GPS) that performs more accurately when compared to cell- based [6]. At previous research, IPS integrating with the location based service (LBS) divided into locations-tracking service and position-aware service [7]. The location-tracking service provides user location information and position aware service and estimate user location. Thus, GPS is a location-tracking service and use a satellite technology to estimate a location information. This approach has markable improve an accuracy especially in outdoor [6].

In this research, the system will be integrated into Wi-Fi technology, positioning technology and location information management will provide a service based on geographic location [8]. For example, In searching for a place such as the location of restaurants, hotels, stations and indoor or other outdoor locations. It is a job for IPS to find the location. As location indicator, IPS basically using wireless devices to collect signals as initial information and calculate the object position. The IPS does not require extra hardware while user will prefer to use the GPS [9] [10]. However, it works optimally at outdoor environment but has poor location estimation at indoor environment. In an indoor environment, GPS has lack to denote specific location. and fingerprint method can be used as a method for object position estimation system in indoor environment [3].

### *2.1 Fingerprint Method*

Localization technique being important element at IPS application [11]. Currently there are many localization and tracing system techniques that have been designed and implemented before. One of the approach is using Received Signal Strength (RSS). The RSS approach is known as the cheapest techniques and can be implemented in both indoor and outdoor environments. This RSS work to change Signal Strength from the transmitter-receiver into information entity. RSS information can use to estimate the distance between the transmitter and receiver through two methods, one of them is through the fingerprint localization method. This method analyse behavior from signal propagation and get the information about the geometry of building from several coordinates that contain the RSS values. In fingerprint method , RSS values will be formed a database whose collected from several places [12] called reference points. Through fingerprint measurement, the unknown location can be estimated by finding a match between the existing fingerprint and the previous fingerprint [13]. normally fingerprint consists of 2 phases, Online and Offline phases [14] :

#### *2.1.1 Offline*

In Offline phase a collection of RSS values from the access point in each point was collected into a database called Radio Map [8]. This radio map builds from the RSS measurements received from reference point site, Radio map can be form an average from measurements which taken also all the statistics to describing the radio fingerprint. The collection of fingerprints for all known locations is called radio map [15]

#### *2.1.2 Online*

If offline phase has purpose to build an empirical training database at each reference point. Then in online phase , data were compared with the existing RSS data and matched to be able to estimate the position of an object [14] [16]. This online phase will receive input data real time to compare. Pattern-Matching Algorithm is used to compare and matching these value [17]. Overall procedure of position estimation system is carried out by linear or non-linear mapping F : $R^N \rightarrow R^2$ [8]

IOP Conf. Series: Earth and Environmental Science **248** (2019) 012077 doi:10.1088/1755-1315/248/1/012077

### *2.2 Deep Learning*

Deep learning is a branch of Machine Learning that used to solve a complex problem. Therefore, deep learning uses more complex and deep architecture of machine learning. Thus, machine learning has limitation on the ability to process data naturally. It is why representation learning is needed to enable features to be found quickly and naturally without human intervention. However, the application is very difficult and complex to represent.

Deep learning method is a representation learning consists the stacks of simpler representations. This can be seen in the layers that are classified by tasks. For example, an image, at the first layer states that this layer consists of objects that recognize the edges. Furthermore, at second layer can be recognized a contour and angle. From the first layer and second layer can be recognized an object. The object is the third layer. At the end of the third layer forms a complete object. Point of deep learning is the feautres from existing layers are not manually input. Therefore, deep learning learn a data by self using general procedure learning. Deep learning makes a hierarchy of architectural concept from complex concept and being simpler complex concept. It is an achievement by deep learning with able to represent a problem with high flexibility [18].

#### *2.2.1 Deep Neural Network (DNN)*

DNN is an example of Deep architecture model, Deep learning is a concept of Artificial Neural Network that is in between the input and output layer. This Artificial Neural Network consists of many hidden layers. This Hidden Layer directly increases the modeling of the DNN as well as optimizing configuration.

A Neural network is said ‘deep’ if the layers are used a lot and stacked. Traditional neural network standar model is only has a few and the usually only have 2 layers. The Calculations are used in the DNN can be explained as follows [18]:

$$\mathcal{Y}_i = f(W_i^1 \mathcal{X}_1 + W_i^2 \mathcal{X}_2 + W_i^3 \mathcal{X}_3 + \cdots + W_i^m \mathcal{X}_m) = f\left(\sum_j W_i^j \mathcal{X}_j\right) \quad (1)$$

$\mathcal{X}_m$ as an output layer , $\mathcal{X}_j$ as an input layer and $W_i$ as a connector between output and input layer. Thus, the calculation is operated into activation function.

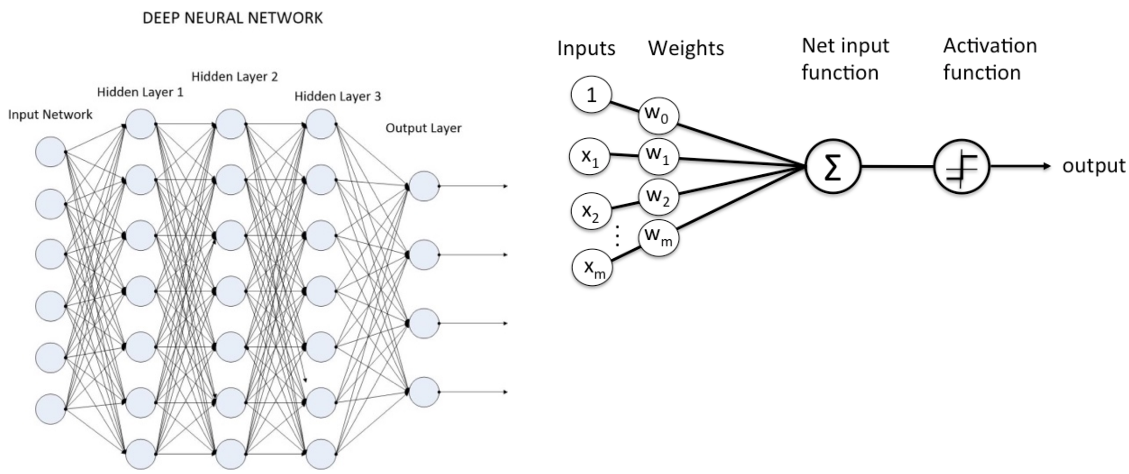


**Figure 1.** DNN Scheme [19]

## 3. Research Methodology

The research of the indoor position estimation system will be applied at building with adjoin 4 rooms inside. Each room will be determined reference points and the point will be taken as offline data. There are 16 access points will cover the building. The floor plan as depicted in Figure 2 with 4 rooms are shown.

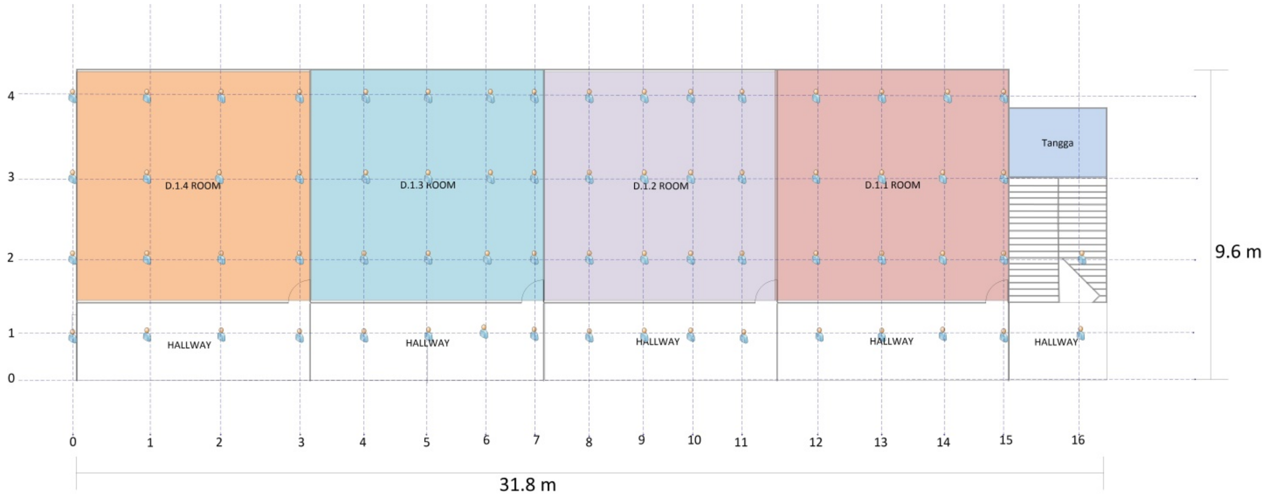


**Figure 2.** Floor plan

At the floor plan, there is 83 reference points and each point will be take the training data in the Received Signal Strength (RSS) value. This RSS was taken using NetSurveyor software. Then the RSS values entered into database with .csv format. As mention in the section 2.1, the indoor positon system generally divided into 2 phases, offline phase and online phase.

### *3.1 Offline Phase*

Offline phase is the step of taking RSS values data and storing data in radio map. The tool to take and store data using NetSurveyor software. The range of RSS value between -100 dBm (weak signal) until -40 dBm (strength signal). Thus, the next step is a data normalization with formula as follows:

$$\mathcal{X}_i = \frac{(\mathcal{X} - \mu)}{\sigma} \qquad (2)$$

where $\mathcal{X}_i$ is RSS value that have done normalization. $\mathcal{X}$ is the values and will being normalization. $\mu$ is average of dataset meanwhile $\sigma$ was a standar deviation. Collecting data is measured per each reference point. In building, it training for 83 points and will store in database.

### *3.2 Online Phase*

Online phase is step for implemented the DNN at real time positioning. There is a 1 input, 1 output and a number hidden layer with symbolized by $h^d$ in the DNN. The Output layer is symbolized by $L^t$ it means the position estimation of Mobile Terminal (MT) or object obtained from RSS as follows:

$$L^t \quad = F(v^t) \qquad (3)$$

where $L^t$ is the position of the MT (Mobile Terminal) at time t . The F function is obtained from a database fingerprint pre-built that represents nonlinear operations between RSSI values and object positions. In a neuron, as represented by $L^t$ receives $x_1 \dots\dots.. x_d \in R$ as input. This input can originated by data or previous output layer $h^0 = x$. $x$ is an early input from RSS values and represented by $h^0$. at the connection between the output $L^t$ and input $x_1 \dots\dots.. x_d \in \mathbb{R}$ is named as weights and symbolized by $w_{1j} \dots\dots\dots w_{dj} \in \mathbb{R}$ . There is a bias vector layer k and symbolized by $b^k$. The layer modelling is shown as follow:

$$h^d = \varphi(W^d h^{d-1} b^d) \tag{4}$$

and the formula is calculated to get input value before function activation calculation ( ) : $h^d$

$$\begin{aligned} \mathcal{Y}_i &= f(W_i^1 \mathcal{X}_1 b^1 + W_i^2 \mathcal{X}_2 b^2 + W_i^3 \mathcal{X}_3 b^3 + \dots + W_i^d \mathcal{X}_d b^d) \\ &= f(\sum_j W_i^j \mathcal{X}_j b^j) \end{aligned} \tag{5}$$

This operation has 2 separate operations namely linear operations and non-linear / activation operations. This activation function is applied to the output layer. The activation function was used in this study is Sigmoid Logistic: $\sigma(a) = \frac{1}{1+\exp(-a)}$ . Sigmoid logistic activation function is used to determine the next neuron described at hidden layer calculation formula as follows:

$$h^1 = 1/(1 + \exp(-w^1 v^t - b^1)) \tag{6}$$

$$h^2 = 1/(1 + \exp(-w^2 h^1 - b^2)) \tag{7}$$

$$h^3 = 1/(1 + \exp(-w^3 h^2 - b^3)) \tag{8}$$

When output of hidden layer 1 was obtained between hidden layer 1 and next hidden layer connection, the value of the weight will be updated. The purpose of this action in order to the value of input layer is have learning process by updating the weight value. The Next Output from DNN positioning is calculated on the following calculation:

$$P(L^t = l_i \mid v^t), i = 1, \dots\dots, N \tag{9}$$

This model is the probability of the position tool in $l_i$i at time t according to the time $v^t$. $l_i$ itself is adjusted to the reference $i^{th}$. At the fourth layer output the activation function can be used if there is only 1 neuron at its output or term by regression. While at this research used more than one neuron therefore will used softmax regression at output layer $o^t$ to determine the probability of the position estimation described as follows:

$$P(L^t = l_i \mid v^t) = o_i^t = \frac{\exp(-\varpi_i h^3 - b_i)}{\sum_i \exp(-\varpi_i h^3 - b_i)} \tag{10}$$

where $\varpi_i$ denotes the weight connection between third hidden layers $h^3$ and output layer o . meanwhile $b_i$ is bias from output layer.

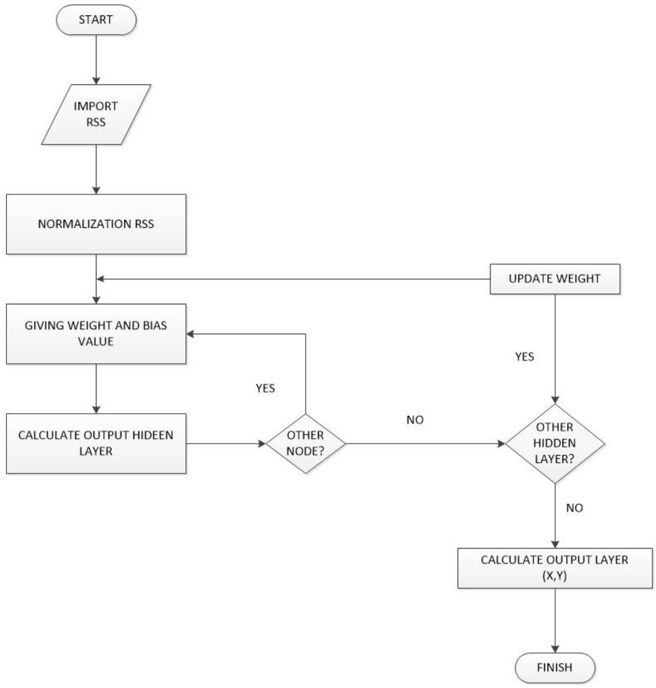


**Figure 3.** Flowchart Software Deep Learning (DNN)

## 4. Result and Discussion

The data that was collected are RSS Values fom access point. This RSS described by dBm format. Data was taken at predefined reference point with distance between per 2 meters. An example of the result can be seen in figure 4. The detail data is shown at table 3:

**Table 1.** Detail Data RSS

| Location | Amount Reference point (Number) | Amount RSS data detected (Number) | Amount RSS data used (Number) |
|---|---|---|---|
| Room D.1.1 | 16 | 690 | 448 |
| Room D.1.2 | 16 | 842 | 448 |
| Room D.1.3 | 16 | 1.117 | 448 |
| Room D.1.4 | 16 | 861 | 448 |

Before online phase, first step is offline phase. In offline phase will collect data from RSS and store the data in database. The RSS data are shown in Table 2 as follows:

**Table 2.** Input Layer

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | -77 | Ap8 | -68 | Ap15 | -110 | Ap22 | -110 |
| Ap2 | -100 | Ap9 | -68 | Ap16 | -100 | Ap23 | -72 |
| Ap3 | -77 | Ap10 | -100 | Ap17 | -100 | Ap24 | -72 |
| Ap4 | -67 | Ap11 | -100 | Ap18 | -95 | Ap25 | -72 |
| Ap5 | -67 | Ap12 | -100 | Ap19 | -110 | Ap26 | -110 |
| Ap6 | -66 | Ap13 | -83 | Ap20 | -110 | Ap27 | -110 |
| Ap7 | -68 | Ap14 | -110 | Ap21 | -110 | Ap28 | -100 |

Next step is normalizing the data. This normalization is used calculation (1) (2), then the results shown in Table 3:

**Table 3.** Normalization Data

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | 0,721 | Ap8 | 1,335 | Ap15 | 0,848 | Ap22 | 0,848 |
| Ap2 | 0,848 | Ap9 | 1,335 | Ap16 | 0,848 | Ap23 | 1,062 |
| Ap3 | 0,721 | Ap10 | 0,848 | Ap17 | 0,848 | Ap24 | 1,062 |
| Ap4 | 1,403 | Ap11 | 0,848 | Ap18 | 0,507 | Ap25 | 1,062 |
| Ap5 | 1,403 | Ap12 | 0,848 | Ap19 | 0,848 | Ap26 | 0,848 |
| Ap6 | 1,471 | Ap13 | 0,311 | Ap20 | 0,848 | Ap27 | 0,848 |
| Ap7 | 1,335 | Ap14 | 0,848 | Ap21 | 0,848 | Ap28 | 0,848 |

*4.1 Deep Neural Network (DNN) Modeling*

DNN is a neural network have many hidden layers and stacked. If neural network has input layer, output layer and just 1 or 2 hidden layers. DNN can have more than 2 hidden layers. DNN was built from many nodes which connected by weights. It is illustrate in Figure 1. Modeling of this calculation was taken from sample point 1. Amount of input layer nodes, hidden layer 1, hidden layer 2, hidden layer 3, output layer is 28,28,28,2 respectively.

*4.1.1 Input Layer to Hidden Layer 1*

The calculation will be using the DNN modeling formula. As shown in Table 4 as a sample. First Weight were given random with limit 0 till 1 and bias with values default 1.

**Table 4**. Weight Values

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | 0,009 | Ap8 | 0,667 | Ap15 | 0,322 | Ap22 | 0,233 |
| Ap2 | 0,073 | Ap9 | 0,996 | Ap16 | 0,153 | Ap23 | 0,794 |
| Ap3 | 0,747 | Ap10 | 0,655 | Ap17 | 0,936 | Ap24 | 0,743 |
| Ap4 | 0,807 | Ap11 | 0,751 | Ap18 | 0,981 | Ap25 | 0,578 |
| Ap5 | 0,462 | Ap12 | 0,151 | Ap19 | 0,895 | Ap26 | 0,887 |
| Ap6 | 0,457 | Ap13 | 0,046 | Ap20 | 0,551 | Ap27 | 0,340 |
| Ap7 | 0,179 | Ap14 | 0,141 | Ap21 | 0,758 | Ap28 | 0,202 |

Then input layer and weight also bias are calculated using formula (5) (6) . Result of this calculation shown at Table 5:

**Table 5.** Output Hidden Layer 1

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | 0,00649 | Ap8 | 0,89030 | Ap15 | 0,27302 | Ap22 | 0,19749 |
| Ap2 | 0,06196 | Ap9 | 1,32943 | Ap16 | 0,13037 | Ap23 | 0,84342 |
| Ap3 | 0,53907 | Ap10 | 0,55537 | Ap17 | 0,79398 | Ap24 | 0,78905 |
| Ap4 | 1,13220 | Ap11 | 0,63716 | Ap18 | 0,49736 | Ap25 | 0,61382 |
| Ap5 | 0,64818 | Ap12 | 0,12867 | Ap19 | 0,75863 | Ap26 | 0,75236 |
| Ap6 | 0,67233 | Ap13 | 0,01446 | Ap20 | 0,46738 | Ap27 | 0,28845 |
| Ap7 | 0,23893 | Ap14 | 0,11994 | Ap21 | 0,64267 | Ap28 | 0,17165 |
| Sum Total | | | | 1,2412 | | | |

Output value represents the value of 1 node at the point of the layer. The result of h1 output will be used as input on hidden layer 1 to hidden layer 2. Then the same calculation with same input value but different weight to get the value of next node

### *4.1.2 Hidden Layer 1 to Hidden Layer 2*

Next the same calculation was also calculate for the next reference point (node). output value from hidden layer 1 will being input value for next hidden layer. But before that the weight must updated. Updating weight was calculated because input value have learning process. The result of this weight update will be used as the weight value for the next hidden layer.

**Table 6.** Input Hidden Layer 1

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | 0,77577 | Ap8 | 0,42391 | Ap15 | 0,33668 | Ap22 | 0,90659 |
| Ap2 | 0,59344 | Ap9 | 0,04791 | Ap16 | 0,55658 | Ap23 | 0,61196 |
| Ap3 | 0,38113 | Ap10 | 0,94374 | Ap17 | 0,06567 | Ap24 | 0,06392 |
| Ap4 | 0,88646 | Ap11 | 0,18571 | Ap18 | 0,04646 | Ap25 | 0,74935 |
| Ap5 | 0,09998 | Ap12 | 0,97975 | Ap19 | 0,28051 | Ap26 | 0,67479 |
| Ap6 | 0,35394 | Ap13 | 0,51614 | Ap20 | 0,62466 | Ap27 | 0,16596 |
| Ap7 | 0,53679 | Ap14 | 0,57224 | Ap21 | 0,85314 | Ap28 | 0,55890 |

**Table 7.** Weight and bias Hidden Layer 1

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | 0,282 | Ap8 | 0,281 | Ap15 | 0,620 | Ap22 | 0,070 |
| Ap2 | 0,674 | Ap9 | 0,608 | Ap16 | 0,019 | Ap23 | 0,374 |
| Ap3 | 0,120 | Ap10 | 0,270 | Ap17 | 0,953 | Ap24 | 0,955 |
| Ap4 | 0,761 | Ap11 | 0,055 | Ap18 | 0,365 | Ap25 | 0,259 |
| Ap5 | 0,756 | Ap12 | 0,998 | Ap19 | 0,387 | Ap26 | 0,549 |
| Ap6 | 0,931 | Ap13 | 0,215 | Ap20 | 0,454 | Ap27 | 0,253 |
| Ap7 | 0,213 | Ap14 | 0,999 | Ap21 | 0,468 | Ap28 | 0,597 |

1

Then input layer and weight also bias are calculated using formula (5) (7). Result of this calculation shown at Table 8:

**Table 8.** Output Hidden Layer 1

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | 0,21877 | Ap8 | 0,11912 | Ap15 | 0,20891 | Ap22 | 0,06346 |
| Ap2 | 0,40010 | Ap9 | 0,02912 | Ap16 | 0,01057 | Ap23 | 0,22881 |
| Ap3 | 0,04596 | Ap10 | 0,25518 | Ap17 | 0,06257 | Ap24 | 0,06104 |
| Ap4 | 0,67460 | Ap11 | 0,01020 | Ap18 | 0,01695 | Ap25 | 0,19445 |
| Ap5 | 0,07558 | Ap12 | 0,97798 | Ap19 | 0,10856 | Ap26 | 0,37080 |
| Ap6 | 0,32951 | Ap13 | 0,11112 | Ap20 | 0,28397 | Ap27 | 0,04203 |
| Ap7 | 0,11434 | Ap14 | 0,57173 | Ap21 | 0,39995 | Ap28 | 0,33414 |
| Sum total value | | 6,3196 | | | | | |
| Output h2 point 1 | | 0,99820258 | | | | | |

Output value represents the value of 1 node at the point of the layer. The result of h2 output will be used as input on hidden layer 2 to hidden layer 3. Then the same calculation with same input value but different weight to get the value of next node

*4.1.3 Hidden Layer 2 to Hidden Layer 3*
Same as previous calculation, it will use output hidden layer 2 as input hidden layer

**Table 9.** Input Hidden Layer 3

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | 0,998203 | Ap8 | 0,9986 | Ap15 | 0,999344 | Ap22 | 0,998023 |
| Ap2 | 0,998432 | Ap9 | 0,99883 | Ap16 | 0,998658 | Ap23 | 0,9995425 |
| Ap3 | 0,999690 | Ap10 | 0,999622 | Ap17 | 0,99907 | Ap24 | 0,99827 |
| Ap4 | 0,99917 | Ap11 | 0,999609 | Ap18 | 0,998123 | Ap25 | 0,9984 |
| Ap5 | 0,09998 | Ap12 | 0,99883 | Ap19 | 0,997912 | Ap26 | 0,998199 |
| Ap6 | 0,99946 | Ap13 | 0,999619 | Ap20 | 0,99936 | Ap27 | 0,9994 |
| Ap7 | 0,99866 | Ap14 | 0,99738 | Ap21 | 0,998544 | Ap28 | 0,996278 |

Output Hidden layer 3 is calculated using formula (5) (7)

**Table 10.** Weight and bias Hidden Layer 3

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | 0,605 | Ap8 | 0,819 | Ap15 | 0,786 | Ap22 | 0,777 |
| Ap2 | 0,205 | Ap9 | 0,089 | Ap16 | 0,170 | Ap23 | 0,717 |
| Ap3 | 0,376 | Ap10 | 0,442 | Ap17 | 0,890 | Ap24 | 0,666 |
| Ap4 | 0,768 | Ap11 | 0,801 | Ap18 | 0,288 | Ap25 | 0,602 |
| Ap5 | 0,432 | Ap12 | 0,272 | Ap19 | 0,190 | Ap26 | 0,232 |
| Ap6 | 0,553 | Ap13 | 0,912 | Ap20 | 0,261 | Ap27 | 0,907 |
| Ap7 | 0,281 | Ap14 | 0,518 | Ap21 | 0,219 | Ap28 | 0,728 |

1

**Table 11.** Output Hidden Layer 3

| Access point | RSS value | Access Point | RSS Value | Access Point | RSS Value | Access Point | RSS Value |
|---|---|---|---|---|---|---|---|
| Ap1 | 0,603913 | Ap8 | 0,81804 | Ap15 | 0,785344 | Ap22 | 0,77664 |
| Ap2 | 0,205477 | Ap9 | 0,088966 | Ap16 | 0,16974 | Ap23 | 0,71636 |
| Ap3 | 0,3760835 | Ap10 | 0,442727 | Ap17 | 0,888728 | Ap24 | 0,66493 |
| Ap4 | 0,767362 | Ap11 | 0,80036 | Ap18 | 0,287498 | Ap25 | 0,601215 |
| Ap5 | 0,43177 | Ap12 | 0,271996 | Ap19 | 0,18988 | Ap26 | 0,23266 |
| Ap6 | 0,55226 | Ap13 | 0,910109 | Ap20 | 0,26092 | Ap27 | 0,903624 |
| Ap7 | 0,28061 | Ap14 | 0,51826 | Ap21 | 0,218767 | Ap28 | 0,72208 |
| Sum total value | | 14,486 | | | | | |
| Output h3 point 1 | | 0,99999949 | | | | | |

Output value represents the value of 1 node at the point of the layer. The result of h2 output will be used as input on Hidden layer 3 to Output Layer. Thus, the same calculation with same input value but different weight to get the value of next node.

*4.1.4 Hidden layer 3 to Output layer*
At the end of the layer output value in this modeling will applied softmax regression it aimed to determaining probability estimation because the output value expected are more than 1 neuron or node This is shown in the calculation formula (10). The results of this calculation as follows:

**Table 12.** Result Output Layer (X,Y)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Output | X | 0,2210 | 0,6816 | 0,8383 | 0,1575 | ΣX | $exp^x$ |
| | | 0,6677 | 0,1155 | 0,4535 | 0,2192 | 13,2406 | 1,7769E-06 |
| | | 0,8885 | 0,1233 | 0,4010 | 0,5890 | | |
| | | 0,1821 | 0,3513 | 0,3530 | 0,4914 | | |
| | | 0,7656 | 0,4185 | 0,9731 | 0,8108 | | |
| | | 0,3880 | 0,3989 | 0,8206 | 0,7373 | | |
| | | 0,4140 | 0,2193 | 0,4872 | 0,0725 | | |
| Output layer | Y | 0,7630 | 0,0277 | 0,5660 | 0,6592 | ΣY | $exp^y$ |
| | | 0,3285 | 0,6100 | 0,6728 | 0,2305 | 15,4898 | 1,8743E-07 |
| | | 0,6511 | 0,9308 | 0,7180 | 0,7668 | | |
| | | 0,8522 | 0,5746 | 0,1535 | 0,0691 | | |
| | | 0,1583 | 0,2368 | 0,1865 | 0,7068 | | |
| | | 0,4734 | 0,5067 | 0,9059 | 0,7606 | | |
| | | 0,7542 | 0,8309 | 0,5033 | 0,8916 | | |

Then the real result of X and Y :

**Table 13.** Output of Output layer (X,Y)

| Coordinate | DNN Value |
|---|---|
| X | 0.904583038 |
| Y | 0.095416962 |

From the result, it is obtained DNN as coordinate value (X,Y). To get the accuracy value, X and Y have to convert into meter then compare it with real object position and calculate the error. Based on the result, the error is 2.0 meter. It is shown in Figure 4.


**Figure 4.** Mapping Accuracy

## 5. Conclusion

The modeling of the position estimation system using DNN was successfully applied using 5 layers consisting of 1 input layer with the number of nodes of 28 nodes, 3 hidden layers with each node number of 28 nodes and 1 output layer with the number of nodes 2 (X, Y). From the results shown the DNN value is affected by the number of hidden layers and the number of nodes in each hidden layer. The value of DNN is the value of X and Y. These values are used as a coordinate point in determining the position of the object and the accuracy is 2 meters

## 6. Reference

[1] H. Yang and S.-L. Lin, "User Adoption of Location-Based Service," pp. 51–56, 2018.
[2] Z. Liu, X. Luo, and T. He, "Indoor Positioning System Based on the Improved W-KNN Algorithm," *2017 IEEE 2nd Adv. Inf. Technol. Electron. Autom. Control Conf.*, pp. 1355–1359, 2017.
[3] G. Felix, M. Siller, and E. N. Alvarez, "A fingerprinting indoor localization algorithm based deep learning," *2016 Eighth Int. Conf. Ubiquitous Futur. Networks*, pp. 1006–1011, 2016.
[4] X. Gan, B. Yu, L. Huang, and Y. Li, "Deep learning for weights training and indoor positioning using multi-sensor fingerprint," *2017 Int. Conf. Indoor Position. Indoor Navig. IPIN 2017*, vol. 2017–Janua, no. September, pp. 1–7, 2017.
[5] G. Wu and P. Tseng, "A Deep Neural Network-Based Indoor Positioning Method using Channel State Information," pp. 290–294, 2018.
[6] I. A. Junglas and R. T. Watson, "LOCATION-BASED SERVICES " Federal law enforcement attempts to use," *Commun. ACM March Commun. ACM*, vol. 51, no. 3, pp. 65–69, 2008.
[7] L. Barkuus and A. Dey, "Location-Based Services for Mobile Telephony : a Study of Users' Privacy Concerns," *Proc. INTERACT 2003, 9TH IFIP TC13 Int. Conf. Human-Computer Interact.*, pp. 1–5, 2003.
[8] T. Chuenurajit, S. Phimmasean, and P. Cherntanomwong, "Robustness of 3D indoor localization based on fingerprint technique in wireless sensor networks," *2013 10th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol.*, vol. 20, no. 3, pp. 1–6, 2013.
[9] P. C. Paul Castro, "A Probabilistic Room Location Service for Wireless Networked Environments," *UbiComp '01 3rd Int. Conf. Ubiquitous Comput.*, pp. 18–34, 2001.
[10] B. a. Forouzan, *Data Communications and Networking - Global Edition*. 2012.
[11] Y. Gu, Y. Chen, J. Liu, and X. Jiang, "Online deep intelligence for Wi-Fi indoor localization," *UbiComp ISWC 2015 - Proc. 2015 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. Proc. 2015 ACM Int. Symp. Wearable Comput.*, pp. 29–32, 2015.
[12] T. Alhmiedat and G. Samara, "An Indoor Fingerprinting Localization Approach for ZigBee Wireless Sensor Networks," vol. 105, no. 2, pp. 190–202, 2013.
[13] C. Laoudias, P. Kemppi, and C. G. Panayiotou, "Localization using Radial Basis Function Networks and Signal Strength Fingerprints in WLAN," 2009.
[14] L. Jiang, "A Wlan Fingerprinting Based Indoor Localization Technique," 2012.
[15] S. Sorour, Y. Lostanlen, and S. Valaee, "RSS based indoor localization with limited deployment load," *GLOBECOM - IEEE Glob. Telecommun. Conf.*, pp. 303–308, 2012.
[16] E. S. Pasinggi, S. Sulistyo, and B. S. Hantono, "Sistem Penentuan Posisi Di Dalam Ruangan Dengan Metode Fingerprint ( Knn )," pp. 6–8, 2015.
[17] G. Felix, M. Siller, and E. N. Alvarez, "A fingerprinting indoor localization algorithm based deep learning," *2016 Eighth Int. Conf. Ubiquitous Futur. Networks*, pp. 1006–1011, 2016.
[18] Y. LeCun, Y. Bengio, G. Hinton, L. Y., B. Y., and H. G., "Deep learning," *Nature*, 2015.
[19] L. Deng and D. Yu, "Deep Learning," pp. 3–4.

## Acknowledgements

This article's publication is supported by the United States Agency for International Development (USAID) through the Sustainable Higher Education Research Alliance (SHERA) Program for

Universitas Indonesia's Scientific Modelling, Application, Research and Training for City-centered Innovation and Technology (SMART CITY) Project, grant #AID-497-A-160004, Sub Grant #IIE00000078-UI-1.

This article is presented at the International Conference on Smart City Innovation 2018 that supported by the United States Agency for International Development (USAID) through the Sustainable Higher Education Research Alliance (SHERA) Program for Universitas Indonesia's Scientific Modeling, Application, Research and Training for City-centered Innovation and Technology (SMART CITY) Project, Grant #AID-497-A-1600004, Sub Grant #IIE-00000078-UI-1