

A DEEP LEARNING MODEL IMPLEMENTATION BASED ON RSSI FINGERPRINTING FOR LORA-BASED INDOOR LOCALIZATION

Irsan Taufik Ali

*Department of Electrical Engineering¹
irsan.taufik@ui.ac.id*

Abdul Muis

*Department of Electrical Engineering¹
muis@ui.ac.id*

Riri Fitri Sari

*Department of Electrical Engineering¹
riri@ui.ac.id*

*¹Universitas Indonesia
Depok, Indonesia, 16424*

Abstract

LoRa technology has received a lot of attention in the last few years. Numerous success stories about using LoRa technology for the Internet of Things in various implementations. Several studies have found that the use of LoRa technology has the opportunity to be implemented in indoor-based applications. LoRa technology is found more stable and is more resilient to environmental changes. Environmental change of the indoor is a major problem to maintain accuracy in position prediction, especially in the use of Received Signal Strength (RSS) fingerprints as a reference database. The variety of approaches to solving accuracy problems continues to improve as the need for indoor localization applications increases. Deep learning approaches as a solution for the use of fingerprints in indoor localization have been carried out in several studies with various novelties offered. Let's introduce a combination of the use of LoRa technology's excellence with a deep learning method that uses all variations of measurement results of RSS values at each position as a natural feature of the indoor condition as a fingerprint. All of these features are used for training in-deep learning methods. It is DeepFi-LoRaIn which illustrates a new technique for using the fingerprint data of the LoRa device's RSS device on indoor localization using deep learning methods. This method is used to find out how accurate the model produced by the training process is to predict the position in a dynamic environment. The scenario used to evaluate the model is by giving interference to the RSS value received at each anchor node. The model produced through training was found to have good accuracy in predicting the position even in conditions of interference with several anchor nodes. Based on the test results, DeepFi-LoRaIn Technique can be a solution to cope with changing environmental conditions in indoor localization.

Keywords: Internet of Things, LoRa, Indoor Localization, RSSI Fingerprinting, Deep Learning.

DOI: 10.21303/2461-4262.2021.001620

1. Introduction

In recent years several studies have been carried out to solve the problem of Indoor localization with varying degrees of accuracy. The proposed indoor positioning system uses a variety of technologies such as Radio Frequency Identification (RFID), WiFi, Bluetooth Low Energy (BLE), Zigbee, Ultra-Wideband (UWB), Visible Light Communication (VLC) as well as vision-based technology. A few years ago the use of Low Power Wide Area (LPWA) LoRa technology was widely used to localize applications. LoRa is generally projected for outdoor applications. LoRa is generally projected for outdoor applications. Many publications have reported successful implementation of LoRa in a variety of outdoor applications including [1–4]. The LoRa property can also be used for indoor scenarios as in [5, 6] including for localization applications [7–9].

In general, regional characteristics and obstacles greatly affect the propagation of telecommunications signal waves regardless of technology. The density, height of buildings, environmental conditions, and contours of the area greatly affect the use of Internet of Things Technology such as LoRa for indoor localization applications. Sadowski et al [10] found that LoRa has a poor performance for indoor localization however on the other side diverse the development of IoT LoRa technology was found to have some advantages as an alternative wireless-based technology.

In other research Islam et al. [7]. LoRa technology was found to be more stable than WiFi and BLE and is more resilient to environmental changes. LoRa operates in the sub-GHz band, which makes it obtain more penetration capability so it is more resistant to noise and multipath so that LoRa becomes the best choice for indoor localization in large rooms and high rise buildings. In a recent 2020 study, Anjum et al. [11] chose LoRa as a device in the positioning system and it was found that LoRa could be a viable solution for the Indoor localization system. Diverse results were obtained from various studies in the use of LoRa in indoor locations. Here it is possible to see that there are many opportunities to improve LoRa performance in indoor locations. With the right approach, this property makes LoRa a promising choice for indoor localization systems.

One popular approach to indoor localization systems is the use of radiofrequency fingerprinting such as WiFi. Fingerprinting is a localization technique used for positioning based on the measurement of Received Signal Strength Intensity (RSSI) on several wireless access points. Positioning is based on a fingerprinting database as a reference containing RSS values that represent each position in the room. Fingerprinting based localization generally consists of two basic phases [12]. First, is the off-line phase, which is also called the training phase, and second, is the on-line phase, which is also called the test phase. In the off-line training stage, machine learning methods can be used to train and store fingerprints that contain all RSS data. Such machine learning methods not only to reduce computational complexity but also to gain core features in RSS for better localization performance. K-nearest-neighbor (KNN) algorithm, artificial neural network, and supporting vector machines, as a popular machine learning method that has been applied for fingerprinting-based indoor localization fingerprinting [13, 14].

Indoor localization systems use RSS data as fingerprint information [15] because of the simplicity of RSS operation and low hardware requirements. However, there are two disadvantages to the RSS method. First, the RSS data collected will be different for the same position from time to time due to the influence of various transmission conditions in the indoor environment. Changes in environmental conditions in a room such as changes in the location and addition of furniture, the addition of other equipment as well as changes in room partitions are a problem in the application of indoor localization. Changes in these conditions will greatly affect variations in RSS data obtained so that it can cause significant errors in positioning [16], especially in the use of fingerprint techniques for indoor localization. In general, there are three main problem formulations captured in the description of the above mention problem:

1. How to reduce the influence of changes in environmental conditions that are dynamic to minimize a very significant difference between the RSS value obtained at the off-line training stage with the RSS value obtained at the testing stage.
2. What is the right solution to address variations in RSS values that will occur in the use of fingerprinting techniques for indoor localization?
3. How to minimize positioning errors caused by variations in RSS value differences in each position due to dynamic environmental conditions.

The following studies have been conducted to overcome the dynamic environmental conditions in indoor localization.

In paper [17] Seong et al. proposed a new database creation method based on the Log-distance Path loss Model to delete RSSI data from unnecessary Wi-Fi, and produce an Access Point (AP) database that can be updated depending on changes in the indoor environment. The proposed algorithm has a higher position resolution than the existing fingerprinting and can improve positioning accuracy due to low dependence on the reference point. To develop the database, the AP signal is not regularly filtered by using Hausdorff distance. There is an opinion that this is not necessary, by using the method of deepening the data of abnormal measurement results can enrich the features of a position that will be the main source of learning at the training stage.

Luo et al. [18] using autonomous robots that patrol the path that has been determined to detect changes in the environment and continue to collect RSS measurement data and map it as a solution. RSS and map measurements are used to build and update databases. Position detection accuracy is maintained by always updating the fingerprinting database if environmental changes are detected. This research was conducted in a simulation with controlled change scenarios and in a limited environment.

Let's offer a different technique to cope with changes in dynamic environments without continuously updating when conditions change. The idea is quite simple, namely by recording all

variations of RSS measurement results in each position. Let's make variations of the measurement results as patterns of disturbance in the room that will be included in off-line training. Disturbances can be caused by many factors such as interference, attenuation of multipath, reflection, deflection, diffraction, and channel fading. Our study does not discuss in detail the types of disturbances that cause variations in RSS measurement results in one position. Let's consider these variations to be natural features of important environmental conditions as a source of learning. This paper contributes to overcoming the problems outlined above. In this paper, let's use the term DeepFi-LoRaIn to describe the use of LoRa device fingerprinting RSS data in indoor localization using the deep learning method. Some of these contributions are as follows:

1. Using LoRa technology as an option for indoor localization systems with good penetration capability so that it is more resistant to noise and multipath, resulting in a more stable RSS measurement value.
2. Using variations in the measurement results of RSS values in a position as a natural feature of the condition of the room without normalizing the data as a fingerprinting database. All of these features are used for training on deep learning methods.
3. Using a deep learning approach to solve the problem of changing environmental conditions to see the accuracy of position predictions.
4. Conduct testing of models that have been trained to be able to predict position in natural conditions using a variety Scenarios of RSS values and changes in the value of RSS in some Anchor Node (AN) that does not exist at the training stage.

The rest of the paper is outlined as follows. Related works are presented in Section 2. In Section 3 let's present the proposed system Deep-Fi LoRaIn. The result and validation are presented in Section 4. Discussion of experimental results is presented in Section 5. Finally, Section 6 shows the conclusion.

2. Related Work

Several studies have been conducted to improve the accuracy of positioning in the indoor localization system. Changes in the dynamic indoor environment are a challenge for researchers to maintain accuracy in positioning, especially in the use of fingerprinting RSS as a reference database. Some of the following research has tried to overcome the problems that have been stated previously.

Long-time before other studies were carried out in 2005 the paper [19] initiated a basic theory and developed an algorithm to build a localization and tracking system in a room with Zero-configuration. The technique introduced can produce a reliable system to support service and location-based network management. The localization algorithm takes input in the form of an on-line measurement of the Received Signal Strength (RSS) between the client and the adjacent AP, to estimate the client's location. Online RSS measurements between AP 802.11 are used to capture in real-time the effects of multi-path RF on temperature and humidity variations, opening and closing doors, furniture relocation, and human mobility during RSS measurements. This technique is also quite responsive to environmental dynamics because the impact of changes in physical characteristics has been explicitly found in the mapping between RSS value and actual geographical distance. This rationale is used as a reference in our research. Characteristics of environmental changes that affect variations in RSS values are the main features that represent natural conditions at a point. Variation of features is needed as a source of training in the proposed deep learning method.

A deep learning approach as a solution to the use of fingerprinting on indoor localization has been conducted in several studies with a variety of novelty offered. Some studies use Channel State Information (CSI) as Wi-Fi signal fingerprinting data and other studies use RSSI as fingerprinting. Wang et al. [13], produced a novelty using the deep learning method in the Indoor localization system. Indoor fingerprinting uses CSI which is trained using a deep learning approach called Deep learning fingerprinting (DeepFi). The DeepFi system architecture includes an off-line training phase and an online localization phase. In the off-line training stage, deep learning is used to train all deep network weights as fingerprints. Besides, the greedy learning algorithm is used to train each layer to reduce complexity. In the on-line localization phase, the probabilistic method is used based on the radial basis function to obtain an approximate location. The experimental results are presented to confirm that DeepFi can effectively reduce location errors compared to the previous

three methods namely FIFS [20], Horus [21], and Maximum Likelihood [22]. The results obtained that DeepFi is more accurate in predicting the location of a device compared to the FIFS, Horus, and Maximum Likelihood methods. Let's also use the term DeepFi in the development of research-based on RSSI fingerprinting data on LoRa devices – DeepFi-LoRaIn.

The use of CSI was also followed by researchers [23], but by using an integrated approach. The approach consists of three stages. First, using the Local Outlier Factor (LOF), the anomaly detection algorithm is used to correct abnormal data. Second, in the training phase, 3 DNN models were trained to classify fingerprint regions using CSI data that was processed from 3 antennas. Third, in the testing phase, a model fusion method called the Group Method of Data Handling (GMDH) was adopted to integrate 3 prediction results of several models and provide final position results. Test-bed experiments are carried out in empty corridors, and the final position accuracy reaches at least 97 %.

The CellinDeep in research [24] used a deep learning approach to Android-based cellular communication for indoor localization. The system produces this accuracy better than other cellular-based indoor systems at least 350 %. CellinDeep produces savings of at least 93.45 % power compared to WiFi-based techniques. Paper [25] used deep learning methods that focus even more on design problems including the influence of different hyperparameters, avoiding overfitting, and training algorithms. Some things done in the paper are used as references, such as the determination of several hyperparameters that are proven to be able to improve the optimization and accuracy of a model in the training and prediction process. Other guidelines [26] derived from several experiments that were carried out widely are also used as a reference for solving problems. The use of Python in Tensorflow along with Keras modules as a deep learning framework makes it easy for us to solve problems in deep learning in indoor localization as in previous studies [25, 27].

3. Proposed System Deep-Fi LoRaIn

The method used in this study is to combine the use of devices and simulations. The device used is the LoRa device to obtain RSS values at each position and NS3 simulators to simulate the LoRa network and Tensorflow Keras is used to simulate the process of deep learning and position prediction test. Following Fig. 1 shows the overall research flow.

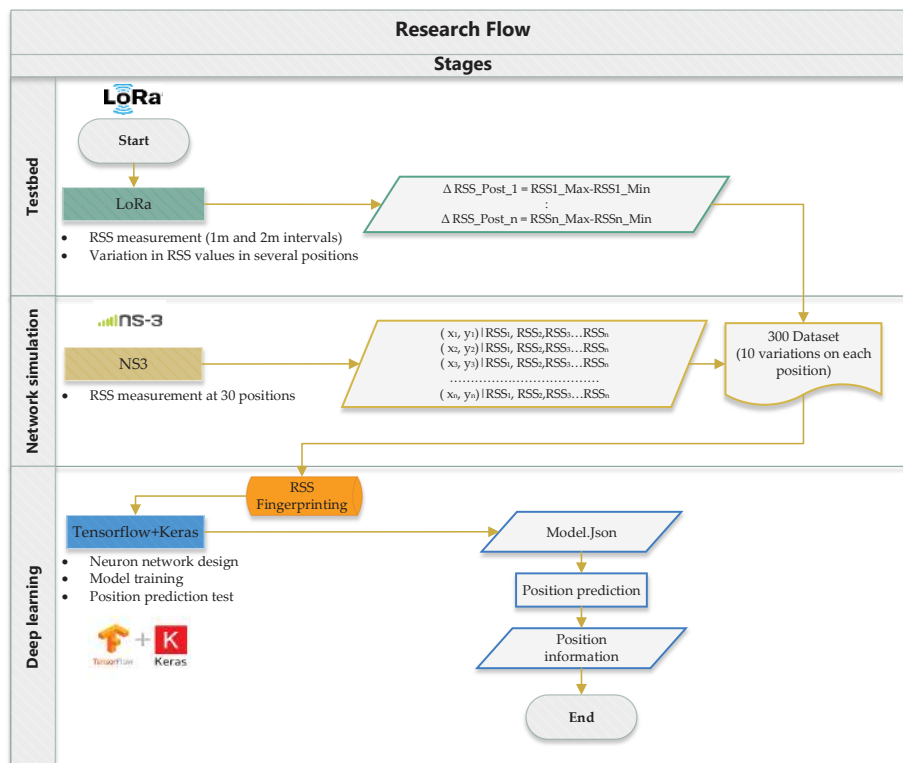


Fig. 1. Research flow chart

The research flow is divided based on the three stages, The first step measuring RSS using a testbed, the second stage measuring RSSI at 30 positions using network simulation and the third stage is designing a neural network architecture. Detail of the flow of the research is described in the following stages as shown in **Fig. 1**.

3. 1. RSS measurement using testbed

Testbed was conducted to determine the RSSI characteristics of LoRa devices in a room and find variations in RSS values in several positions. This variation of RSS values will then be used to form a dataset that will be used in training. To obtain the existing pattern of disturbances in the room such as interference, attenuation of multipath, reflection, deflection, diffraction, and channel fading using data obtained based on direct measurements in the room using a testbed. During testing, let's used the ESP32 LoRa SX1278 device as an anchor node and as a tracked node. LoRa device specification is shown in **Table 1**.

Table 1

LoRa device specification

Parameter	LoRa Value
Radio	SX1278
Frequency band	433 MHz
Anchor height	2 m
Tracked node height	1.5 m
Radiation pattern	Omnidirectional
Sensitivity	-148 dBm, +20 dBm

The room size is 11 meters long and 7 meters wide. The building wall material is made of 70 % Concrete board and 30 % glass with a thickness of 0.75 cm. The ceiling is made of plaster-board with a thickness of 0.5 cm. In the room, there are 28 cubical tables with particle board material with a thickness of 2 cm and a thin steel storage cabinet with a thickness of 1.5 mm.

RSS measurement is conducted using 1 device as the anchor node and 1 device as tracked nodes. The measurement scenario is done by shifting the tracked node away from the anchor node at the intervals of 1 meter and 2 meters in the line-of-sight. Measurements are made in several points and then recorded in the database using a computer as a data logger. Measurements are made at each point every 2 seconds with a duration of 2 minutes for each position as shown in **Fig. 2**.

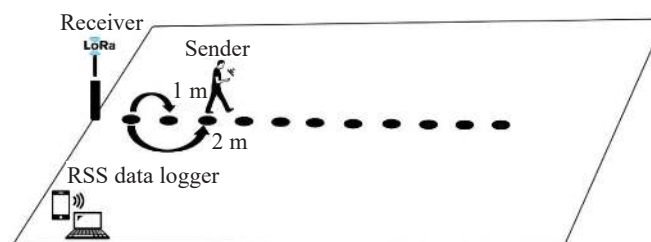


Fig. 2. RSS measurement position testbed in the interval of 1 m and 2 m

RSS characteristics obtained from the results of simulation and testbed measurements are used to find variations in RSS values at one point caused by disturbances that occur in the room. RSS measurement results are processed using spreadsheets to determine the amount of data variation as well as the range of data for the highest and lowest RSS values which are delta changes in RSS values found at each position at the time of measurement.

3. 2. Measure RSSI at 30 positions using network simulation

Network simulations using NS3 are performed to measure RSS at 30 positions based on values received at 4 anchor nodes. RSS values generated through simulation are fixed values that

represent each position. The results of this measurement are further extended using a variation pattern of values obtained in the measurement using the testbed so that each position has 10 variations of the measurement results. The total dataset generated is as much as 300 RSS fingerprinting data. The dataset in the form of RSS fingerprinting was obtained from simulations using NS-3 with a scenario of 4 anchor nodes positions in each corner of the room measuring 11X7 m. The collection of RSS values uses a node that functions as a tracked node that is positioned at 30 different points scattered in the room as shown in Fig. 3.

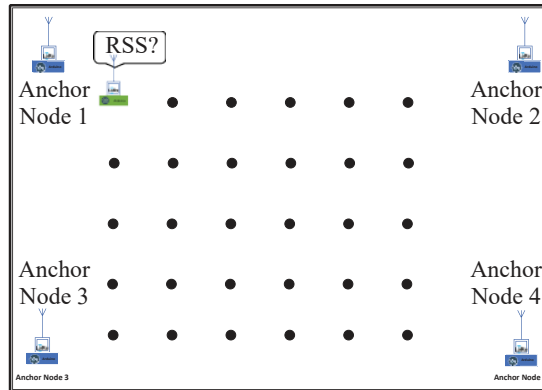


Fig. 3. RSS Fingerprint position for 30 different locations

Furthermore, the amount of RSS value sent from the tracked node will be read at each anchor node as a fingerprinting database. For communication between LoRa nodes using 433 MHz frequency with a 14 dBm sending signal strength with spreading factor 7. Tensorflow Keras simulations are used for several processes. First, to design a neuron network architecture along with hyperparameter. Second, conduct training on the fingerprinting dataset to produce the most optimal model. Third, it is used for testing in predicting position using the optimal model.

3. 3. System architecture

This section will explain system architecture in general As shown in Fig. 4. The system uses LoRa devices as sender and receiver, 4 devices as an anchor node, and 1 device as a tracked node. The system is divided into 2 stages: the off-line phase, which is called the training phase, and second, the on-line phase, which is called the test phase. The training phase is carried out to construct a database based on survey data related to RSS feeds in several positions. At this stage, a comprehensive location survey is performed to record fingerprints of the RSS magnitude of each anchor node at the target tracked node location. During the location survey, the collector must stand in each training position and do RSS scans for several at different positions. RSS value is collected at each position as a reference point in the database.

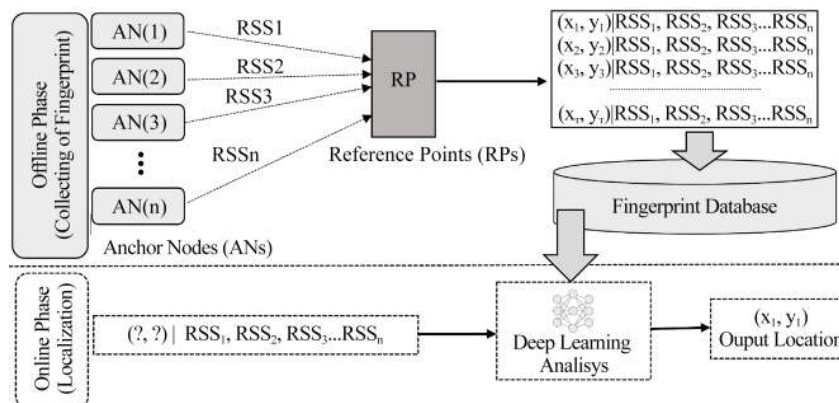


Fig. 4. System architecture

The LoRa end-device sends an uplink packet to the server. The server receives the packet and measures the RSSI value. These values are stored in a database and fingerprint maps are generated from RSSI data for the service area. In the online phase, the LoRa end-device sends an uplink packet containing RSSI information to the Server. RSSI data is sent to the position server from the position of the end device for identification.

A. Neuron Architecture

Hyperparameter is a variable that greatly affects the model output. Hyperparameter learning consists of the number of layers, the number of neurons in each layer, the choice of nonlinearity parameters, the learning rate, batch size, etc. The choice of hyperparameter is an important problem in learning. Until now, there is currently almost no integrated theory to choose the parameters correctly. However, some research experiments can be used as a reference as in the paper that has been reviewed previously, although not all hyperparameters are discussed in this paper. In this section, let's discuss how to improve the performance of deep learning networks and how to set deep learning hyperparameters. Let's focus on the parameters that affect the results of the training, to find the most accurate model.

a. Activation function

In the process of building a neural network, one of the choices is what activation function to use in the hidden layer as well as at the output layer of the network. The activation function decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron. The neural network has neurons that work in correspondence of weight, bias, and their respective activation function. In a neural network, it would update the weights and biases of the neurons based on the error at the output. This process is known as back-propagation. Activation functions make the back-propagation possible since the gradients are supplied along with the error to update the weights and biases. As the name implies, the activation function functions to determine whether the neuron must be «active» or not based on the weighted sum of the input. In general, there are 2 types of activation functions, Linear and Non-Linear Activation functions. Let's use non-linear functions namely sigmoid and Relu. There are many other activation functions however, the functions mentioned above are functions that are used in this case.

The sigmoid activation or is also called a logistic function has been the default choice for activation functions for a long time. The sigmoid function is the continuous activation function most commonly used to classify 2 classes or groups of data. The Sigmoid function has a range from 0 to 1.

In hidden units, the nonlinear activation function $h(\cdot)$ Used is the logistic sigmoid function, as follows [28]:

$$z_J = h(a_J) = \tanh(a_J), \quad (1)$$

the quantities a_J known as activations, where

$$\tanh(a) = \frac{(e^a - e^{-a})}{(e^a + e^{-a})} \quad (2)$$

a sum-of-squares error, in which the output units have linear activation functions, so that At output units, for regression problems, the usual activation function used is the identity function [28]:

$$y_k = a_k. \quad (3)$$

For classification problems, the usual activation function used is a logistical sigmoid function, i. e.:

$$y_k = \sigma(a_k), \quad (4)$$

where

$$\sigma(a) = \frac{1}{1 + e^{-a}}. \quad (5)$$

The ReLU function is the most popular function currently used with several success stories in terms of image recognition and voice recognition. Basically, ReLU does a «threshold» from 0 to infinity. ReLU can also cover up weaknesses held by Sigmoid and Tanh. It is known that the Rectified Linear Unit (ReLU), is formulated as a function [29]

$$f(x) = \max(0, x). \quad (6)$$

b. Learning rate

The use of learning rate parameters has an important influence on the time needed to achieve the desired target. Will slowly optimize the value of weight changes and produce smaller errors. The learning rate variable states a constant that is between 0.1–0.9. This value shows the speed of learning from the network. If the learning rate used is too small then too many epochs are needed to achieve the desired target value, causing the training process to take a long time. The greater the value of learning rate is used, the network training process will be faster, but if it is too large it will cause the network to become unstable and cause repeated error values back and forth between certain values, thus preventing errors from reaching the expected target. Therefore the selection of the value of the variable learning rate must be as optimal as possible to obtain a fast training process [30].

c. Adam optimization

Adam (Adaptive Moment Estimation) is a first-order stochastic-based optimization algorithm of stochastic objective functions, based on adaptive estimates of low-order moments that can be used instead of the classical stochastic gradient descent procedure to update the network weight iteratively based on training data. Adam is currently a popular algorithm in the field of deep learning because it achieves good results quickly. Empirical results show that Adam works well in practice and is better than other stochastic optimization methods. Adam is relatively easy to configure where the default configuration parameters work well in most problems. In several papers [31, 32], Adam is recommended as an optimization algorithm used for deep learning applications. Adam keeps an exponentially decaying average of past gradients m_t . m_t and v_t are values of the first moment which is the Mean and the second moment which is the uncentered variance of the gradients respectively [33]. The formulas for the first Moment (mean) and the second moment (the variance) of the Gradients. Where β_1 and β_2 is Exponential decay rates:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad (7)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}. \quad (8)$$

Then the final formula for the parameter update is:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t, \quad (9)$$

where θ_t is updated parameters.

d. Number of layers

Bengio et al. [26], suggests the use of a higher number of hidden neuron layers to provide significant expressive power. A large number of hidden layers provide more capacity for more complex function models with weaknesses that are difficult in the training process. For regression and classification tasks, and not learning representations, it is also recommended to use the same layer size rather than decreasing or increasing the number of layers. The fully-connected layer is the layer in which all the activation neurons from the previous layer are all connected with neurons in the next layer as well as ordinary artificial neural networks. Every activation from the previous layer needs to be converted into one-dimensional data before it can be connected to all neurons in the Fully-connected layer. The difference between the fully-connected layer and the ordinary convolution layer

is that the neurons in the convolution layer are connected only to certain regions of the input, while the fully-connected layer has a whole connected neuron. However, the two layers still operate the dot product, so the function is not so different. The neural network configuration is shown in Fig. 5.

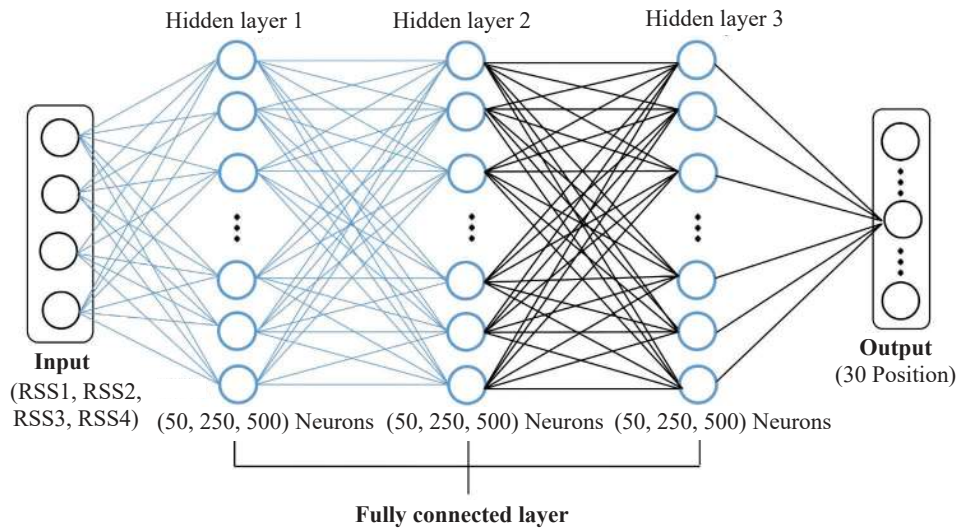


Fig. 5. Neural network configuration

Recommendations from some of the above experimental results are used as a reference to determine hyperparameter in this paper. Let's propose to use three hidden layer neural networks that are fully connected by changing the number of neurons in each hidden layer during training. All hidden layers are equipped with the ReLU non-linearity activation function and for output using the Sigmoid activation function. For the output layer, let's use a dropout layer with a 50 percent reduction rate. Its weight is initialized by using the random procedure suggested by [34]. Neural networks are trained using Adam's algorithm with a learning rate of 0.001, batch size 8. Training data uses 70 % of the 300 data available on the datasets while the test data uses 30 % of the 300 data available on the datasets. Input is the RSSI value obtained from 4 different anchor nodes. The output is an approximate location with 30 labels in a multi-categorical format that represents each position. The following **Table 2** shows the parameters used.

Table 2
Parameter Set Up

Parameter	Parameter Value
Model type	Sequential
Hidden layer (Activation)	ReLu
Output layer (Activation)	Sigmoid
Hidden layer number	3
Neuron number	50, 250, 500 fully connected layer
Optimizer	Adam
Data set size	300
Data training	70 %
Data test	30 %
Learning rate	0.001
Batch size	8
Epochs	100, 125, 150, 200, 250

The main objective of the training stage is to find the best model with the highest accuracy, which will then be used as a reference to determine the position at the testing stage. In this scenario,

the training is repeated to find the best model. At each parameter switch, the number of neurons and epochs is set with different values. For the parameter number of neurons using 3 different values, namely 50, 250, and 500 neurons with all layers connected. Epochs parameter uses 5 different values namely, 100, 125, 150, 200, and 250 number of epochs. The results of the training model will be discussed in result and validation in the next session.

B. Testing Scenarios

The dataset used is in the form of a fingerprint database which is the result of RSS measurements using the previous NS-3 simulation stage. The determination of variations in data features is done randomly uses on RSS values of 30 positions that have been obtained on measurements using NS-3. The range of variations in RSS value generated refers to the delta values obtained in the measurement using the testbed. Each position produces 10 variations of RSS values, resulting in 300 datasets of RSS values from 30 positions on the 4 anchors.

The test stage for predicting the position consists of 4 scenarios. The first scenario is called variable position testing which aims to measure the accuracy of the prediction results of each feature variation that each position has. In this test, the RSS input values are taken randomly from various features representing the 30 available positions from the RSS fingerprinting dataset.

The second scenario is called an adaptive test which aims to measure the accuracy of prediction of positions if there is a change in the indoor environment. Adaptive testing consists of 3 scenarios namely the scenario of changes in the value of RSS on 1 anchor node, 2 anchor nodes, and 3 anchor nodes at each position randomly. Changing the RSS value is done to simulate interference with the environment that can occur when predicting position. Disturbances can change the measurement results of RSS values at certain anchor nodes.

4. Result and Validation

4. 1. Training model result

Based on the results of several pieces of training using a combination of different numbers of neurons and epochs, let's obtain varying model accuracy values as shown in **Table 3** below.

Training using 50 neurons; the number of epochs 100 produces an accuracy value of the model 84.44 %, the number of epochs 125 has an accuracy of the model 86.67 %, the number of epochs 150 obtains an accuracy value of the model 84.44 %, the number of epochs 200 gets an accuracy value of the model 87.78 % and the accuracy of the model the highest is 90 % on the number of epochs 250.

Furthermore, training uses 250 neurons; the number of epochs 100 obtained the accuracy value of the model 81.11 %, the number of epochs 125 resulted in the accuracy of the model 83.33 %, the number of epochs 150 obtained the accuracy value of the model 85.56 %, the number of epochs 200 obtained the accuracy value of the model 87.78 % and the accuracy value of the model the highest was obtained on epochs 250 with a model accuracy value of 88.89 %.

The training uses 500 neurons; the number of epochs 100 produces an accuracy value of the model 82.22 %, the number of epochs 125 obtained the accuracy of the model 85.56 %, the number of epochs 150 the accuracy value of the model is 84.44 %, the number of epochs 200 gets the accuracy value of the model 87.78 % and the accuracy value of the model the number of epochs 250 is 92.22 %. From the overall results of the training, the highest accuracy model obtained on the use of the number of 500 neurons on epochs 250 was 92.22 %.

Table 3

Training model accuracy

Epochs	Model Accuracy (%)		
	50 Neurons	250 Neurons	500 Neurons
100	84.44	81.11	82.22
125	86.67	83.33	85.56
150	84.44	85.56	85.56
200	87.78	87.78	90.00
250	90.00	88.89	92.22

In **Fig. 6** it is possible to see a graph of the highest model training results with an accuracy of 92.22 % using 500 neurons and 250 epochs. In the graph, it is possible to see the results of the training model that shows that there is no overfitting where the accuracy value obtained during training does not exceed the accuracy value obtained during training.

Fig. 7 shows a graph of the loss model where the amount of the loss value obtained in testing is below the loss value during training. This model is the best model obtained from the overall results of the training model used as a reference to predict the position at a later stage.

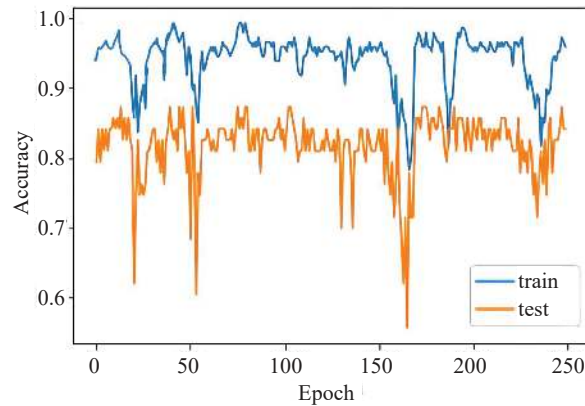


Fig. 6. Result of validation of the training model

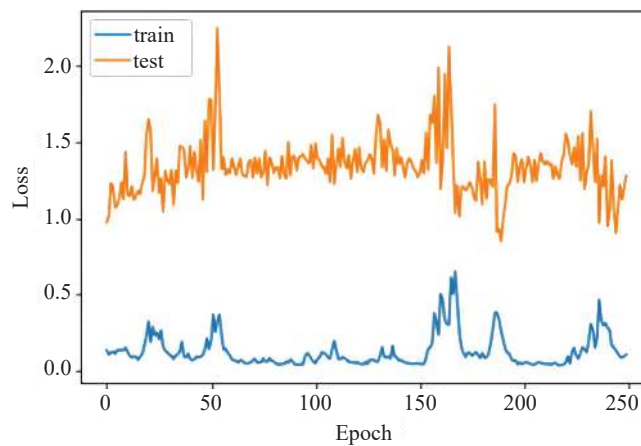


Fig. 7. Result of validation of the loss model

4. 2. Varied positions testing results

In the varied position testing stages, the input RSS values at 4 AN that have been randomly selected from each position. The test produces predictive results that correspond to 30 actual positions as shown in **Table 4**.

Table 4
Prediction result of 30 positions for Varied position test and

Input (RSS1,RSS2,RSS3,RSS4)	Real Position	Predict Position
1	2	3
[-39.0871,-54.1477,-62.7648,-65.0123]	1	1
[-42.8357,-54.3426,-62.9856,-64.6272]	2	2
[-46.9621,-53.7561,-64.4561,-62.8952]	3	3
[-51.8922,-46.9673,-61.3546,-63.5343]	4	4
[-54.5921,-42.5249,-63.6538,-63.2173]	5	5

Continuation of Table 4

1	2	3
[-56.2007,-37.1931,-64.6808,-62.7425]	6	6
[-48.1532,-58.1024,-58.1733,-63.1034]	7	7
[-48.7533,-54.9862,-60.3752,-61.7533]	8	8
[-51.8532,-53.2734,-60.8533,-61.4732]	9	9
[-52.8637,-52.8543,-62.8632,-61.8656]	10	10
[-56.3754,-50.6482,-62.7563,-62.1064]	11	11
[-56.5783,-48.3858,-63.2738,-58.2567]	12	12
[-54.6392,-58.8634,-55.5383,-59.6374]	13	13
[-54.1073,-58.8536,-57.5993,-56.1274]	14	14
[-55.6499,-56.8695,-57.4759,-58.4178]	15	15
[-55.1962,-54.3759,-59.9853,-58.1652]	16	16
[-58.7353,-54.9637,-58.8263,-57.1823]	17	17
[-60.1938,-52.5638,-61.2933,-57.2034]	18	18
[-59.0723,-60.9722,-52.1733,-57.5743]	19	19
[-58.8653,-60.1062,-52.9724,-56.0363]	20	20
[-59.9627,-58.9366,-53.6428,-55.7453]	21	21
[-59.8714,-59.1291,-55.2549,-53.6468]	22	22
[-60.9632,-58.2134,-56.1633,-52.9833]	23	23
[-62.3647,-59.4629,-59.3562,-52.4728]	24	24
[-61.3849,-63.7005,-43.1623,-56.7086]	25	25
[-61.9673,-63.9763,-47.1734,-53.8364]	26	26
[-61.1635,-62.9634,-49.9534,-52.9534]	27	27
[-61.9726,-62.4838,-51.5368,-48.8253]	28	28
[-64.1834,-62.1723,-55.4738,-46.8356]	29	29
[-63.2739,-61.8364,-56.1836,-43.9673]	30	30

The 30 input RSS values that are used represent 30 positions at the test stage resulting in the exact position prediction results corresponding to the actual position. As represented in **Fig. 8**.

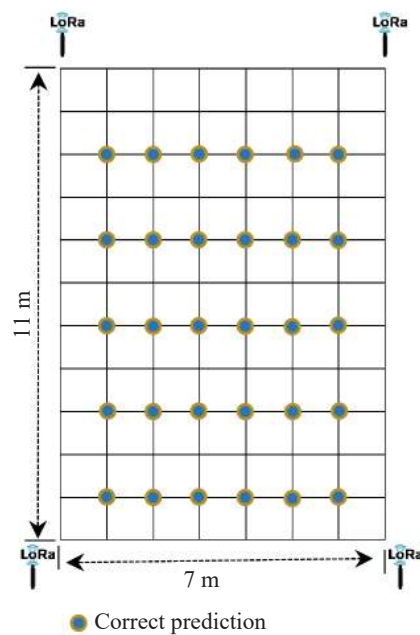


Fig. 8. Prediction result of 30 positions for varied position testing

Prediction results for positions 1 to 30 are very much in line with the actual position. These results indicate that the model obtained from the training results functions optimally as a reference at the prediction stage.

4.3. Adaptive 1 node test result

In the adaptive 1 node test, the changes in RSS values in several AN positions do not significantly influence the predicted results. A prediction error occurred at 4 points out of 30 test points. The test result data can be seen in the following **Table 5**.

Table 5

Prediction result of 30 positions for adaptive test

Input (RSS1,RSS2,RSS3,RSS4)	Real Position	Predict Position
[-39.0871, -54.1477, -60.5478, -65.0123]	1	1
[-40.8647, -54.3426, -62.9856, -64.6272]	2	1
[-46.9621, -50.7685, -64.4561, -62.8952]	3	3
[-51.8922, -46.9673, -61.3546, -55.8673]	4	11
[-54.5921, -42.5249, -63.6538, -63.2173]	5	5
[-56.2007, -37.1931, -62.7332, -62.7425]	6	6
[-48.1532, -55.8757, -58.1733, -63.1034]	7	7
[-48.7533, -54.9862, -60.3752, -59.1857]	8	8
[-52.3569, -53.2734, -60.8533, -61.4732]	9	9
[-52.8637, -52.8543, -62.8632, -62.7356]	10	10
[-57.9563, -50.6482, -62.7563, -62.1064]	11	11
[-56.5783, -48.3858, -60.8686, -58.2567]	12	12
[-54.6392, -55.875, -55.5383, -59.6374]	13	12
[-54.1073, -58.8536, -55.9636, -56.1274]	14	14
[-56.1356, -56.8695, -57.4759, -58.4178]	15	15
[-56.9866, -54.3759, -59.9853, -58.1652]	16	16
[-58.7353, -56.1345, -58.8263, -57.1823]	17	16
[-60.1938, -52.5638, -60.4736, -57.2034]	18	18
[-59.0723, -60.9722, -52.1733, -58.1355]	19	19
[-58.8653, -61.8653, -52.9724, -56.0363]	20	20
[-59.9627, -58.9366, -53.6428, -54.8674]	21	21
[-58.6537, -59.1291, -55.2549, -53.6468]	22	22
[-60.9632, -58.2134, -56.1387, -52.9833]	23	23
[-63.1698, -59.4629, -59.3562, -52.4728]	24	24
[-61.3849, -63.7005, -42.8756, -56.7086]	25	25
[-61.9673, -63.1899, -47.1734, -53.8364]	26	26
[-62.8754, -62.9634, -49.9534, -52.9534]	27	27
[-61.9726, -62.4838, -50.9532, -48.8253]	28	28
[-64.1834, -63.3861, -55.4738, -46.8356]	29	29
[-63.2739, -61.8364, -55.4637, -43.9673]	30	30

Fig. 9 exhibits 4 position prediction errors, 2 of them just shifted to the closest position, which was position 2 and position 17 prediction.

Incorrect prediction results occur when using input data in position 2, position 4, position 13, and position 17. Prediction for position 2 produces output for position 1, position 4 produces prediction output position 11, position 13 produces predictive output position 12, and position 17 produces predictive output position 16.

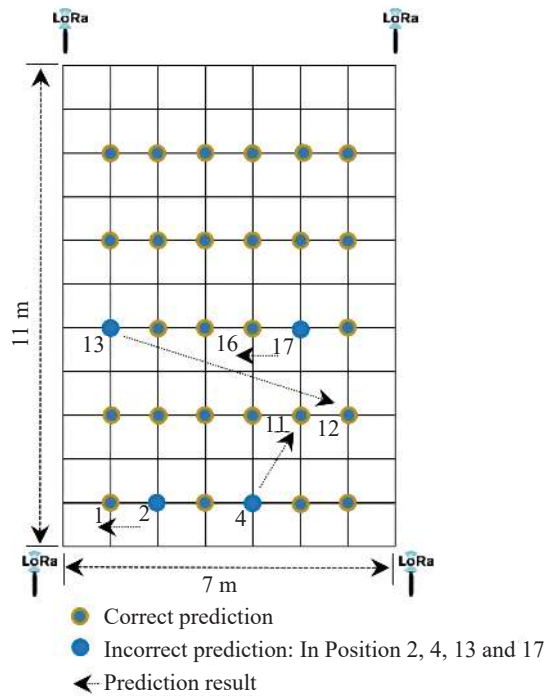


Fig. 9. Prediction result of 30 positions for adaptive 1 node test

4. 4. Adaptive 2 node test result

For adaptive 2 node testing results, prediction error increased from the previous test to 5 positions as shown in Table 6 and Fig. 10 below.

Table 6
Prediction result of 30 positions for adaptive 2 node test

Input (RSS1,RSS2,RSS3,RSS4)	Real Position	Predict Position
1	2	3
[-40.0426, -54.1477, -60.5478, -65.0123]	1	1
[-40.8647, -52.7537, -62.9856, -64.6272]	2	2
[-46.9621, -50.7685, -64.4561, -61.2937]	3	3
[-50.8663, -46.9673, -61.3546, -55.8673]	4	11
[-54.5921, -42.5249, -62.7384, -63.2173]	5	5
[-56.2007, -36.9371, -62.7332, -62.7425]	6	6
[-46.5362, -55.8757, -58.1733, -63.1034]	7	7
[-48.7533, -54.9862, -61.6365, -59.1857]	8	8
[-52.3569, -53.2734, -60.8533, -62.6382]	9	9
[-52.8637, -53.0635, -62.8632, -62.7356]	10	10
[-57.9563, -50.6482, -64.0737, -62.1064]	11	11
[-56.5783, -48.3858, -60.8686, -57.5352]	12	12
[-54.6392, -55.875, -54.2738, -59.6374]	13	14
[-56.8364, -58.8536, -55.9636, -56.1274]	14	14
[-56.1356, -58.0635, -57.4759, -58.4178]	15	15
[-56.9866, -54.3759, -59.9853, -60.1257]	16	16
[-58.7353, -56.1345, -61.8536, -57.1823]	17	16
[-60.1938, -52.5638, -60.4736, -58.0436]	18	18

Continuation of Table 6

1	2	3
$[-59.0723, -60.9722, -52.1733, -58.1355]$	19	19
$[-58.8653, -61.8653, -52.9724, -58.0364]$	20	20
$[-58.8653, -61.8653, -52.9724, -58.0364]$	21	22
$[-58.6537, -61.1873, -55.2549, -53.6468]$	22	22
$[-62.1584, -58.2134, -56.1387, -52.9833]$	23	23
$[-63.1698, -60.0354, -59.3562, -52.4728]$	24	24
$[-61.3849, -65.6353, -42.8756, -56.7086]$	25	25
$[-61.9673, -63.1899, -47.1734, -56.0627]$	26	26
$[-62.8754, -62.9634, -49.9534, -54.0263]$	27	27
$[-61.9726, -63.0527, -50.9532, -48.8253]$	28	28
$[-64.1834, -63.3861, -55.4738, -47.9267]$	29	29
$[-65.0737, -61.8364, -55.4637, -43.9673]$	30	29

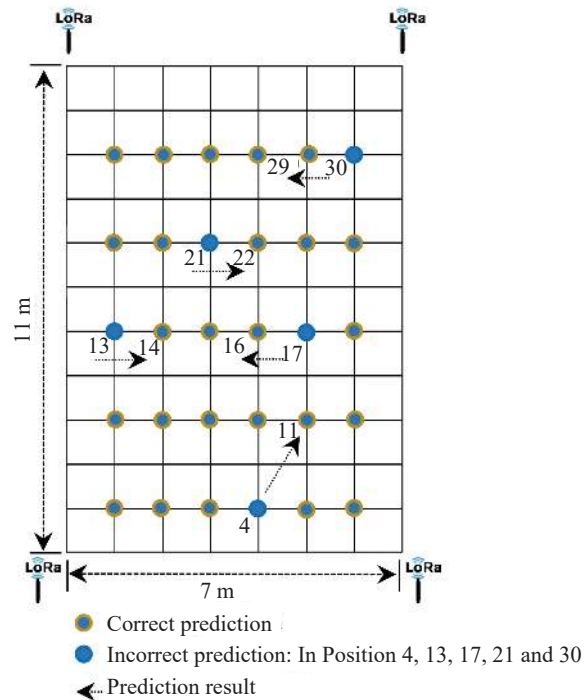


Fig. 10. Prediction result of 30 positions for adaptive 2 node test

Prediction errors occur at 5 positions namely, position 5, position 13, position 17, position 21, and position 30. Position 4 is predicted to be position 11, position 13 is predicted at the closest position is 14. At position 17 the prediction becomes position 16, Position 21 becomes position 22, and prediction result position 30 becomes position 29. From 5 prediction error results, 4 prediction positions only shift to the closest position.

4. 5. Adaptive 3 node test result

Adaptive test results for changes in 3 nodes increased from adaptive testing on 2 nodes to 9 position prediction errors. These errors can be seen in **Table 7** and **Fig. 11** below.

Prediction errors occur at 9 positions, namely, at position 4, position 8, position 13, position 17, and position 20. Furthermore, the prediction error occurs at position 21, position 22, position 28,

and position 30. For all prediction errors, 5 of them are error position predictions that also occurred in previous tests. **Fig. 11** is clearly seen from 9 prediction errors, 7 prediction errors are only shifted at the closest position, position 8 prediction becomes position 9, position 13 prediction shifts to position 14, and position 17 prediction becomes position 16. Furthermore, prediction of position 20 becomes position 19, prediction position 21 shifts to position 22, prediction of position 28 becomes position 27, and the result of the prediction of position 30 shifts to position 29. For the results of predictions that have a shift in a position somewhat far away is at position 4 and position 22. Prediction of position 4 becomes position 11 and prediction 24 shifts to position 14.

Table 7

Prediction result of 30 positions for adaptive 3 node test

Input (RSS1,RSS2,RSS3,RSS4)	Real Position	Predict Position
[-40.0426,-54.1477,-60.5478,-66.0527]	1	1
[-40.8647,-52.7537,-64.9657,-64.6272]	2	2
[-49.8546,-50.7685,-64.4561,-61.2937]	3	3
[-50.8663,-46.9673,-64.9665,-55.8673]	4	11
[-54.5921,-42.5249,-62.7384,-64.7854]	5	5
[-54.8754,-36.9371,-62.7332,-62.7425]	6	6
[-46.5362,-55.8757,-58.1733,-63.8094]	7	7
[-50.6342,-54.9862,-61.6365,-59.1857]	8	9
[-52.3569,-54.8375,-60.8533,-62.6382]	9	9
[-53.9446,-53.0635,-62.8632,-62.7356]	10	10
[-57.9563,-50.6482,-64.0737,-63.0463]	11	11
[-55.1856,-48.3858,-60.8686,-57.5352]	12	12
[-54.6392,-55.875,-54.2738,-58.9547]	13	14
[-56.8364,-58.8536,-55.9636,-56.6484]	14	14
[-56.1356,-58.0635,-57.4759,-58.8536]	15	15
[-56.9866,-54.3759,-60.1757,-60.1257]	16	16
[-56.0251,-56.1345,-61.8536,-57.1823]	17	16
[-60.5424,-52.5638,-60.4736,-58.0436]	18	18
[-59.0723,-60.4746,-52.1733,-58.1355]	19	19
[-59.0824,-59.8635,-53.1072,-55.8254]	20	19
[-60.8678,-58.9366,-55.0374,-54.8674]	21	22
[-58.6537,-61.1873,-55.2549,-55.1072]	22	14
[-62.1584,-58.2134,-56.1387,-53.9673]	23	23
[-63.1698,-60.0354,-59.3562,-53.0828]	24	24
[-63.2438,-65.6353,-42.8756,-56.7086]	25	25
[-62.0854,-63.1899,-47.1734,-56.0627]	26	26
[-62.8754,-64.8432,-49.9534,-54.0263]	27	27
[-61.9726,-63.0527,-50.9532,-51.1756]	28	27
[-65.0547,-63.3861,-55.4738,-47.9267]	29	29
[-65.0737,-61.8364,-55.4637,-45.0565]	30	29

The results of varied position test and the adaptive test showed that the model used as a predict reference to make predictions is quite adaptive to changes. Disturbance simulations do not significantly affect the predicted results. **Table 8** summarizes the final results for a varied position testing and adaptive testing.

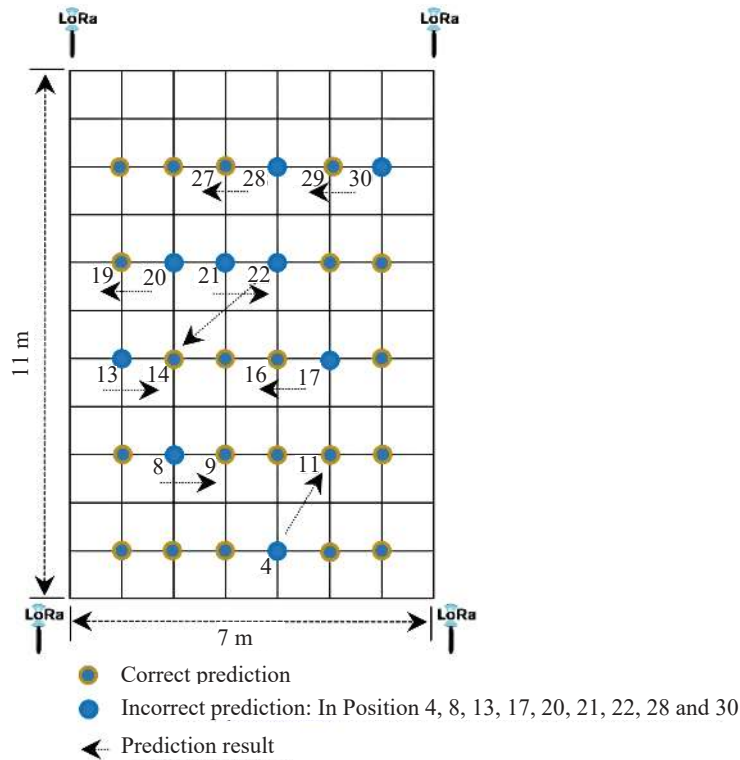


Fig. 11. Prediction result of 30 positions for adaptive 3 node test

Table 8
Accuracy of predictions

Predict Test	Incorrect	Correct	Accuracy (%)
Varied positions	0	30	100
Adaptive 1 node	4	26	86.66
Adaptive 2 node	5	25	83.33
Adaptive 3 node	9	21	70

The results of the predictions in varied position testing reach 100 % accuracy and the adaptive 1 node testing results obtained the accuracy level of 86,66 %, The 2 node adaptive test results produce an accuracy of 83.33 % and for the 3 nodes adaptive test the prediction accuracy results are obtained at 70 %. From the overall prediction results using 4 test scenarios, the average accuracy obtained is far above 50 %, which is 85 %.

5. Discussion of Experimental Results

Three main problems to be solved in this research are reducing the environmental influence on the results of RSS measurements, addressing the variability of RSS values, and positioning accuracy. Therefore, this study aims to develop techniques that can improve the ability and accuracy of object position detection for indoor localization applications in a dynamic environment. Development and optimization of the use of fingerprint techniques for indoor localization applications using a deep learning approach. The following are the results obtained from the study.

Environmental influences that cause differences in the RSS value at the time of fingerprinting data collection with the RSS value at the time of testing can be minimized by taking advantage of the LoRa device. Here the researchers see that there are still opportunities to improve LoRa performance in indoor locations. With the right approach, these properties make LoRa a promising choice for indoor localization systems.

Variation in the change in the RSS measurement value at each position due to changing environmental conditions which can serve as a feature that helps improve the training process. The right hyperparameter setting based on best practice experience produces the best model with the highest accuracy of 92.22 %, as shown in **Fig. 2**. The best model with the highest accuracy is used as a reference to obtain position prediction results with high accuracy.

The results of the position variation test and adaptive test in **Table 8** show that the deep learning approach can be a solution to address changes in environmental conditions that occur in cases of indoor localization. Referring to the results of the simulation of environmental changes carried out in the adaptive test scenario, it shows that the use of a deep learning approach is proven to be able to overcome changes in environmental conditions that occur in indoor environments with accurate predictions.

With the concept and use of simple infrastructure, this method is possible to be implemented in an Indoor Localization system. Compared to the concept applied in previous studies, the use of the DeepFi-LoRaIn concept does not require additional equipment that is always operating indoors to monitor and record changes that occur in the environment. The resulting training model is able to maintain a good accuracy of position prediction despite changing conditions in the environment.

This research was conducted with several limitations, including the use of limited space and static object detection. The future challenge is the implementation in a wider indoor environment using moving object prediction. The advantages of LoRa's reach can be used to implement indoor localization between buildings. Improved position prediction accuracy can still be improved by increasing the number of RSS fingerprint features that represent each changing environment as training data predicts the existence of immovable objects using.

6. Conclusions

The results of variation and adaptive tests showed the average accuracy obtained is 85 %, even the result of predictions in varied position testing reaches 100 % accuracy. That result showed the deep learning approach can be a solution to overcome changes in environmental conditions that occur in the case of indoor localization. Variations in changes in the measurement value of RSS in each position are caused by changes in environmental conditions that can represent the real conditions in the environment. Variations in RSS values in each position serve as features that help to improve the training process. The use of the deep learning approach is proven to be able to cope with changes in environmental conditions that occur in the indoor environment with accurate prediction accuracy. The characteristic of RSS LoRa fingerprint in indoor location is promising to be implemented due to it is relatively more stable to environmental changes with the result that is no significant shift in RSS values that are between in training stages and the prediction stages.

Acknowledgement

We thank Ministry of Research and Higher Education of Republic of Indonesia for financial support for this research under the PDD Grant number NKB-423/UN2.RST/HKP.05.00/2020.

References

- [1] Harris, N., Curry, J. (2018). Development and Range Testing of a LoRaWAN System in an Urban Environment. *World Academy of Science, Engineering and Technology. International Journal of Electronics and Communication Engineering*, 12 (1), 47–55. 5.
- [2] Nor, R. F. A. M., Zaman, F. H., Mubdi, S. (2017). Smart traffic light for congestion monitoring using LoRaWAN. 2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC), 132–137. doi: <http://doi.org/10.1109/ICSGRC.2017.8070582>
- [3] Mdhaffar, A., Chaari, T., Larbi, K., Jmaiel, M., Freisleben, B. (2017). IoT-based health monitoring via LoRaWAN. *IEEE EUROCON 2017 – 17th International Conference on Smart Technologies*, 519–524. doi: <http://doi.org/10.1109/EUROCON.2017.8011165>
- [4] Li, L., Ren, J., Zhu, Q. (2017). On the application of LoRa LPWAN technology in Sailing Monitoring System. 2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS), 77–80. doi: <http://doi.org/10.1109/WONS.2017.7888762>

- [5] Ke, K.-H., Liang, Q.-W., Zeng, G.-J., Lin, J.-H., Lee, H.-C. (2017). A LoRa wireless mesh networking module for campus-scale monitoring. Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks. doi: <https://doi.org/10.1145/3055031.3055034>
- [6] Wixted, A. J., Kinnaird, P., Larijani, H., Tait, A., Ahmadinia, A., Strachan, N. (2016). Evaluation of LoRa and LoRaWAN for wireless sensor networks. 2016 IEEE SENSORS. doi: <https://doi.org/10.1109/icsens.2016.7808712>
- [7] Islam, B., Islam, M. T., Kaur, J., Nirjon, S. (2019). LoRaIn: Making a Case for LoRa in Indoor Localization. 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). doi: <https://doi.org/10.1109/percomw.2019.8730767>
- [8] Lam, K.-H., Cheung, C.-C., Lee, W.-C. (2017). LoRa-based localization systems for noisy outdoor environment. 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). doi: <https://doi.org/10.1109/wimob.2017.8115843>
- [9] Fargas, B. C., Petersen, M. N. (2017). GPS-free geolocation using LoRa in low-power WANs. 2017 Global Internet of Things Summit (GIoTS). doi: <https://doi.org/10.1109/giots.2017.8016251>
- [10] Sadowski, S., Spachos, P. (2018). RSSI-Based Indoor Localization With the Internet of Things. IEEE Access, 6, 30149–30161. doi: <https://doi.org/10.1109/access.2018.2843325>
- [11] Anjum, M., Khan, M. A., Hassan, S. A., Mahmood, A., Qureshi, H. K., Gidlund, M. (2020). RSSI Fingerprinting-Based Localization Using Machine Learning in LoRa Networks. IEEE Internet of Things Magazine, 3 (4), 53–59. doi: <https://doi.org/10.1109/iotm.0001.2000019>
- [12] Bahl, P., Padmanabhan, V. N. (2000). 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). doi: <https://doi.org/10.1109/infcom.2000.832252>
- [13] Wang, X., Gao, L., Mao, S., Pandey, S. (2017). CSI-based Fingerprinting for Indoor Localization: A Deep Learning Approach. IEEE Transactions on Vehicular Technology, 66 (1), 763–776. doi: <https://doi.org/10.1109/tvt.2016.2545523>
- [14] Li, J., Li, Y., Ji, X. (2016). A novel method of Wi-Fi indoor localization based on channel state information. 2016 8th International Conference on Wireless Communications & Signal Processing (WCSP). doi: <https://doi.org/10.1109/wcsp.2016.7752710>
- [15] Wang, B., Zhou, S., Liu, W., Mo, Y. (2015). Indoor Localization Based on Curve Fitting and Location Search Using Received Signal Strength. IEEE Transactions on Industrial Electronics, 62 (1), 572–582. doi: <https://doi.org/10.1109/tie.2014.2327595>
- [16] Xiao, Y., Zhang, S., Cao, J., Wang, H., Wang, J. (2017). Exploiting distribution of channel state information for accurate wireless indoor localization. Computer Communications, 114, 73–83. doi: <https://doi.org/10.1016/j.comcom.2017.10.013>
- [17] Seong, J.-H., Seo, D.-H. (2017). Environment Adaptive Localization Method Using Wi-Fi and Bluetooth Low Energy. Wireless Personal Communications, 99 (2), 765–778. doi: <https://doi.org/10.1007/s11277-017-5151-x>
- [18] Luo, R. C., Hsiao, T. J. (2019). Dynamic Wireless Indoor Localization Incorporating With an Autonomous Mobile Robot Based on an Adaptive Signal Model Fingerprinting Approach. IEEE Transactions on Industrial Electronics, 66 (3), 1940–1951. doi: <https://doi.org/10.1109/tie.2018.2833021>
- [19] Lim, H., Kung, L.-C., Hou, J. C., Luo, H. (2006). Zero-Configuration, Robust Indoor Localization: Theory and Experimentation. Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications. doi: <https://doi.org/10.1109/infocom.2006.223>
- [20] Xiao, J., Wu, K., Yi, Y., Ni, L. M. (2012). FIFS: Fine-Grained Indoor Fingerprinting System. 2012 21st International Conference on Computer Communications and Networks (ICCCN). doi: <https://doi.org/10.1109/icccn.2012.6289200>
- [21] Youssef, M., Agrawala, A. (2005). The Horus WLAN location determination system. Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services – MobiSys '05. doi: <https://doi.org/10.1145/1067170.1067193>
- [22] Brunato, M., Battiti, R. (2005). Statistical learning theory for location fingerprinting in wireless LANs. Computer Networks, 47 (6), 825–845. doi: <https://doi.org/10.1016/j.comnet.2004.09.004>
- [23] Yin, Y., Song, C., Li, M., Niu, Q. (2019). A CSI-Based Indoor Fingerprinting Localization with Model Integration Approach. Sensors, 19 (13), 2998. doi: <https://doi.org/10.3390/s19132998>
- [24] Rizk, H., Torki, M., Youssef, M. (2019). CellinDeep: Robust and Accurate Cellular-Based Indoor Localization via Deep Learning. IEEE Sensors Journal, 19 (6), 2305–2312. doi: <https://doi.org/10.1109/jsen.2018.2885958>
- [25] Xiao, L., Behboodi, A., Mathar, R. (2017). A deep learning approach to fingerprinting indoor localization solutions. 2017 27th International Telecommunication Networks and Applications Conference (ITNAC). doi: <https://doi.org/10.1109/atnac.2017.8215428>
- [26] Bengio, Y. (2012). Practical Recommendations for Gradient-Based Training of Deep Architectures. Neural Networks: Tricks of the Trade, 437–478. doi: https://doi.org/10.1007/978-3-642-35289-8_26

- [27] Riyaz, S., Sankhe, K., Ioannidis, S., Chowdhury, K. (2018). Deep Learning Convolutional Neural Networks for Radio Identification. *IEEE Communications Magazine*, 56 (9), 146–152. doi: <https://doi.org/10.1109/mcom.2018.1800153>
- [28] Bishop, C. (2006). *Pattern recognition and machine learning*. Springer, 738.
- [29] Nair, V., Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *Proceedings of the Proceedings of the 27th international conference on machine learning (ICML-10)*.
- [30] Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep learning*. Adaptive computation and machine learning. MIT Press.
- [31] Ruder, S. (2016). An overview of gradient descent optimization algorithms. Available at: <https://arxiv.org/pdf/1609.04747.pdf>
- [32] Hijazi, S., Kumar, R., Rowen, C. (2015). Using convolutional neural networks for image recognition. Cadence Design Systems Inc. Available at: https://ip.cadence.com/uploads/901/cnn_wp-pdf
- [33] Kingma, D. P., Ba, J. (2014). Adam: A Method for Stochastic Optimization. Available at: <https://arxiv.org/pdf/1412.6980.pdf>
- [34] He, K., Zhang, X., Ren, S., Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE International Conference on Computer Vision (ICCV)*. doi: <https://doi.org/10.1109/iccv.2015.123>

Received date 22.10.2020

Accepted date 28.01.2021

Published date 29.01.2021

© The Author(s) 2021

*This is an open access article under the CC BY license
(<http://creativecommons.org/licenses/by/4.0>).*