IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 August 2023, accepted 1 October 2023, date of publication 11 October 2023, date of current version 18 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323035

RESEARCH ARTICLE

Self-Organized Wireless Sensor Network (SOWSN) for Dense Jungle Applications

GALANG P. N. HAKIM[1,2], MOHAMED HADI HABAEBI[2], (Senior Member, IEEE), MD. RAFIQUL ISLAM[2], (Senior Member, IEEE), ABDULLAH ALGHAIHAB[3], SITI HAJAR BINTI YUSOFF[2], (Member, IEEE), AND ERRY YULIAN T. ADESTA[4]

[1]Department of Electrical Engineering, Universitas Mercu Buana, Jakarta 11650, Indonesia
[2]Department of Electrical and Computer Engineering, International Islamic University Malaysia (IIUM), Kuala Lumpur, Selangor 53100, Malaysia
[3]Department of Electrical Engineering, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[4]Department of Industrial Engineering Safety and Health, Universitas Indo Global Mandiri (UIGM), Palembang 30129, Indonesia

Corresponding author: Mohamed Hadi Habaebi (habaebi@iium.edu.my)

The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project no (IFKSUOR3-104-1).

ABSTRACT To facilitate wireless sensor networks deployment in dense jungle environments, the challenges of unreliable wireless communication links used for routing data between nodes and the gateway, and the limited battery energy available from the nodes must be overcome. In this paper, we introduce the Self-Organized Wireless Sensor Network (SOWSN) to overcome these challenges. To develop the traits needed for such SOWSN nodes, three types of computational intelligence mechanisms have been featured in the design. The first feature is the introduction of Multi Criteria Decision Making (MCDM) algorithm with simple Additive Weight (SAW) function for clustering the SOWSN nodes. The second feature is the introduction of the fuzzy logic ANFIS-optimized Near Ground Propagation Model to predict the wireless transmission link quality and power transfer between transmitters. The third feature is the introduction of the (Levenberg Marquardt artificial neural network (LM-ANN) for Adaptive and Dynamic Power Control to further optimize the transmitter power levels, radio modulation, Spreading Factor configurations, and settings of the employed SOWSN LoRaWAN nodes based on predicted wireless transmission link quality parameters. The introduced features were extensively evaluated and analyzed using simulation and empirical measurements. Using clustering, near-ground propagation, and adaptive transmission power control features, a robust wireless data transmission system was built while simultaneously providing power conservation in SOWSN operation. The payload loss can be improved using SAW clustering from 1275-bytes to 5100-bytes. The result of power conservation can be seen from the reduction of transmission power in SOWSN nodes with the increase of transmission time (TOA) as its side effect. With the original power transmission at 20-dBm, original TOA time at 96.832-milliseconds for all nodes, and SNR 3 as input, transmission power was reduced to 12.76-dBm and the TOA increased to 346.78-milliseconds for all nodes.

INDEX TERMS Self organized WSN, cluster routing, ANFIS, transmit power control, SAW, routing.

I. INTRODUCTION

The Wireless Sensor Network (WSN) has been classified as the next generation network in 2010 by ITU [1]. This network contains interconnected sensor nodes that exchange sensed data using wireless technologies such as LoRa [2], ZigBee [3], and many others [4]. Using its characteristics solution which is node to node communication (adhoc network) it has low cost for deploying it [5], and it can be used everywhere from mountain [6], forest [7], cities [8], even in underwater [9] application. Even though WSN has many advantages, this network also has many disadvantages

The associate editor coordinating the review of this manuscript and approving it for publication was Kashif Saleem.

such as limited energy [10] and data transmission problem such as interference [11], delay [12], packet loss [13], and others [14].

To guarantee the quality of service in wireless data transmission in Wireless Sensor Network, a lot of researchers propose to employ a routing algorithm [15], which controls the behavior of the network. The algorithm decides how every node in Wireless Sensor Network should communicate with each other. This algorithm has a single purpose which is to deliver observation data using point to point node communication, from the farthest node to the base station closer node with minimum error. Routing in Wireless Sensor Network has few methods, from direct routing to cluster routing. Direct (flat) routing is defined when every node in the network will send its observe data directly from the nodes to the sink node (gateway) [16]. Cluster routing is defined when every node will perform a cluster of nodes, select its cluster head node, send its data to the cluster head node, and the cluster head node will then send its children data node to the sink node or another nearest cluster head node. Although direct routing offers simplicity for its routing, however clustered routing is preferred because it can save more energy, better network communication, efficient topology management, and many others [16], [17]. One of the many methods for clustering is to use Simple Additive Weight (SAW) algorithm. SAW is a Multi Criteria Decision Making (MCDM) algorithm that was used as a foundation in many clustering algorithms such as EMTARP [18], Enhanced GRP [19], EACO [20], and many more.

For adaptive power transmission model in ultrahigh-voltage (UHV) power substation, Sun uses linear quadratic Gaussian (LQG) method to develop a path loss model based on state-space model of wireless networks and use SNIR as its measurable parameter [21]. Vidhya in his paper explains the strategy to construct a classification-based Adaptive Transmission Power control. The method uses RSSI, link quality indicator, neighbor node distance, receiver power, and non-linear models to maintain reliable communication with minimal energy use [22]. Philip in his paper explains the strategy for power reduction by varying the transmission power of each node using distance between transmitter and receiver from GPS Coordinates [23].

In this paper we propose to use algorithm such as SAW to select Cluster Head (CH) routing in Self Organized Network Routing. After SAW MCDM for CH selection, we propose to model WSN environment in the jungle. Based on those models we can build near ground propagation model also adaptive and dynamic transmission power control model to enhance power conservation. This SOWSN Cluster routing in total can provide us an adaptive and dynamic power control transmission based on wireless physical channel selection such as MATPoC [24], and also provide a robust wireless transmission data transfer throughout the WSN network. For easy understanding we have provided a simplified diagram of this work as shown in figure 1.

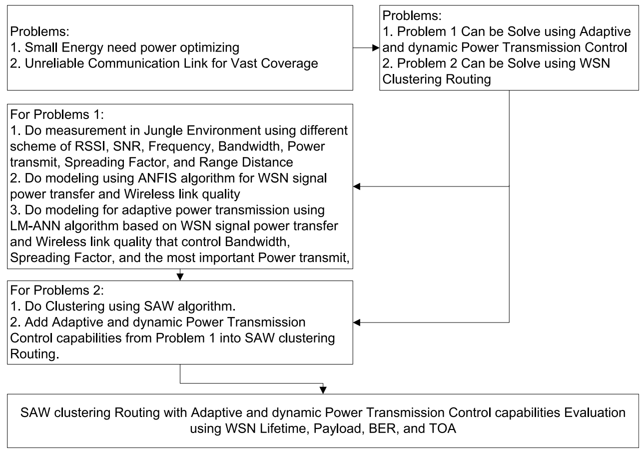


FIGURE 1. Simplified work diagram.

II. METHODS AND MATERIAL

A. FIELD MEASUREMENT EQUIPMENT

To do measurement for WSN environment modelling, we propose to use LoRaWAN radio transceiver in pair (transmitter and receiver). The microcontroller was included to control data transmission and pre-process data for the transmitter and receiver. The microcontroller on the transmitter side was programmed to allow it to instruct the LoRa transceiver to broadcast a packet of data every 100 microseconds. On the receiver side, the LoRaWAN transceiver was instructed to read incoming packet data, extract RSSI and SNR values, and transfer them directly to our laptop. Every 5 meters of measurement, from 5 meters to 100 meters, the walk test method was employed (please see figure 2). A pair of LoRaWAN transceivers were placed on top of the ground. For a LoRaWAN transceiver pair, the maximum antenna height above the ground was less than 30 cm. The configuration details for measurement equipment can be seen in table 1 and figure 2.

TABLE 1. LoRaWAN radio setting.

Parameter	Value
Frequency	433, 868, 920 MHz
Bandwidth	125, 250, 500 KHz
Spreading Factor	7-12
Antenna Gain	0 dBi
Tx- power	20 dBm
Measurement Parameter	RSSI, SNR

B. FIELD MEASUREMENT JUNGLE ENVIRONMENT

We chose a jungle site for our real-time measurement because the WSN was usually used in those environments, one of example WSN used in jungle such as illegal logging monitoring [25], Forest Fire monitoring [26], and others. Overgrown jungle vegetation covers a substantial portion of the landscape. Between the trees and larger plants in this jungle are

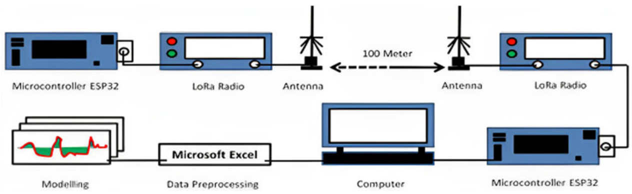


FIGURE 2. Field measurement equipment.

several smaller plants. Human movement through or around a jungle can occasionally be made difficult or impossible due to the extremely dense vegetation on the ground. This location will prevent a radio signal from a LoRaWAN transmitter from traveling directly to a LoRaWAN receiver, this small, thick, and also massive density of tall trees will obstruct the radio signal. The field measurement site at jungle can be seen in figure 3 below.

FIGURE 3. Jungle field measurement environment.

C. SIMPLE ADDITIVE WEIGHT ALGORITHM

This algorithm is also known as weighted addition method [27]. This algorithm concept is to see for all weighted sum on each alternative on all attributes. This algorithm was developed by Fishburn to show arranged product sets using priority orderings and assignments [28]. In general, the multi criteria decision problem (MCDM) is defined on set of alternatives with set of decision criteria. If we assume that all criteria are the benefit criteria, then the higher value is the better alternative. Hence, Preference for each alternative was given by:

$$A = \sum_{j=1}^{n} w_j r_{ij} \tag{1}$$

where:

w_j = weighted value for alternatives j = 1,2,…,n

r_{ij} = performance value of alternative A, i=1,2,…,m, and j = 1,2,…,n

D. ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) ALGORITHM

Zadeh developed the foundation of fuzzy in 1975 based on Linguistic Variable and its Application to Approximate Reasoning [29]. The fuzzy rule was subsequently created to model the qualitative elements of human knowledge (reasoning based on experience) [30] and address the issue using those as its foundations. Even though fuzzy is typically employed for control [31], including robot movement [32], speed control [33], and many other [34], [35], however a fuzzy system can be applied to anything from detection [36] until forecasting [37]. In 1985, Takagi and Sugeno develop fuzzy model using implications and reasoning for industrial control [38]. The Fuzzy Sugeno equation can be seen from equation 2 through equation 7.

Each output for step one, is indicated by O_i^1 which to increase the degree of membership.

$$O_i^1 = \mu A_i(x) \ and \ O\frac{1}{i} = \mu B_i(x), \quad i = 1, 2 \tag{2}$$

where:

i = each node of ANFIS architecture.

A, B = is the linguistic label.

x = is the input to node i. (such as small, large, etc.).

Every membership function type is usable in this stage. However, to offer maximum equal to 1 and minimum equal to 0, generalized bell types were employed. hence:

$$\mu Ai(x) = \frac{1}{1 + \left(\frac{x - c_i}{a_i}\right)^{2*b_i}} \tag{3}$$

where:

a,b,c = is the parameter set.

By multiplying the two input signals, the second step contributes to the firing strength of fuzzy inference. The is represented by each node.

$$O_i^2 = \mu A_i(x).\mu B_i(x), \quad i = 1, 2 \tag{4}$$

The following phase involved applying normalization for each fuzzy inference firing.

$$O_i^3 = \overline{W}_i = \frac{W_i}{W1 + W2}, \quad i = 1, 2 \tag{5}$$

where:

W = is the firing strength of the node.

$\overline{W}$ = is the normalized firing strength of the node.

The following phase involved a calculation based on the parameters of the rule consequent in the following phase:

$$O_i^4 = \overline{W}_i.F_i = \overline{W}_i.(P_i x + Q_i x + R_i x), \quad i = 1, 2 \tag{6}$$

where:

P,Q,R = is the parameter set

The final phase computes the overall output by adding up all of the input signals:

$$Output = O_i^5 = \sum_{k=0}^{n} \overline{W}_i.F_i = \frac{\sum_{k=1}^{n} W_i.F_i}{\sum_{k=1}^{n} W_i} \tag{7}$$

Jang Jyh Shing Roger created Adaptive Neuro Fuzzy Inference System (ANFIS) in 1993 [39] based on fuzzy Takagi Sugeno's if-then rules. The fuzzy inference system can adapt organically using the ANFIS approach based on its training data. The Takagi-Sugeno fuzzy inference system serves as the foundation for the artificial neural network technique that makes up ANFIS [40]. This method can combine the benefits of fuzzy logic and neural networks into a single framework. A collection of fuzzy IF-THEN rules that can be learned to estimate nonlinear functions govern how this inference system operates. Consequently, ANFIS is regarded as a Universal Estimator (universal assessor). Equations 2 through 7 make up the core algorithm used by ANFIS, which uses a fuzzy Sugeno algorithm. Jang then used gradient descent and chain rule to optimize its parameter. However, to do this, we must be aware of the error rate for data training for each node output, since ANFIS learns through chain rule and gradient descent. The ANFIS architecture can be seen from figure 4 and ANFIS algorithm can be seen from equation 8 through equation 13:

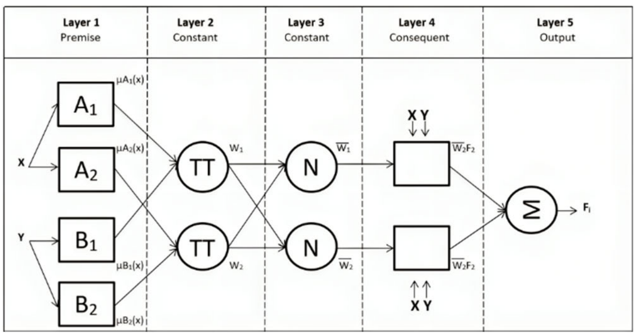


FIGURE 4. **ANFIS architecture based on Jang [40].**

If the data training sets have P numbers of inputs and the i-th position node outputs define O_i, then the error function may be calculated as follows:

$$E_p = \sum_{m=1}^{\#L} \left(T_{mp} - O_{mp}^{L}\right)^2 \tag{8}$$

where:

E_p = is the error measure which the sum of squared errors.

T_{mp} = is the m component from the P output target vector.

O_{mp}^{L} = is the m component from the actual output vector that has been delivered by the P input vector.

Hence, the error rate can be calculated as:

$$\frac{\partial E_p}{\partial O_{ip}^{k}} = \sum_{m=1}^{\#k+1} \frac{\partial E_p}{\partial O_{mp}^{k+1}} \frac{\partial O_{mp}^{k+1}}{\partial O_{ip}^{k}} \tag{9}$$

where $1 \leq k \leq L-1$ is the error rate of an internal node. it is expressed as the linear combination error rate of nodes in the next step. Therefore, for all $1 \leq k \leq L$ and $1 \leq i \leq \#(k)$, we can find $\frac{\partial Ep}{\partial O_{ip}^{k}}$, using mathematical equations (8) and (9).

Thus, we have α as a parameter of the adaptive network:

$$\frac{\partial E}{\partial \alpha} = \sum_{o*\in S} \frac{\partial Ep}{\partial O*} \frac{\partial O*}{\partial \alpha} \tag{10}$$

where:

S = shows the set of nodes whose output depends on α.

Derivative for overall error measurement E concerning α is:

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^{p} \frac{\partial Ep}{\partial \alpha} \tag{11}$$

Therefore, we can write the updated mathematical equations for generic parameter α as follows:

$$\Delta\alpha = -\eta \frac{\partial E}{\partial \alpha} \tag{12}$$

where:

η = is a learning rate.

The learning rate can be written as:

$$\eta = \frac{k}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha}\right)^2}} \tag{13}$$

where:

k = is the step size or length of each gradient transition in the parametric space.

E. LEVENBERG MARQUARDT ARTIFICIAL NEURAL NETWORK (LM-ANN) ALGORITHM

A computational intelligence called ANN (Artificial Neural Network) imitates the organic neural network behavior of living things. Numerous issues, including control [41], detection [42], forecasting [43], [44], and more, have been resolved using ANN approaches. The technique operates by analyzing and computing previously approved data training [45]. Artificial neurons (ANN) are a group of nodes that mimic the neurons found in the biological brain. Like in a biological brain, each neuron has synapses that allow it to communicate with other neurons. The signal is received by neuron receiver, which then processes it before using a synapse attached to it to signal the next neurons. Neurons serve as processing components in ANN. Bias and weights govern these neurons that are coupled to other neurons. Its signal calculation is then used to generate an output after being fed via an activation function [46]. A simple equation can be used to represent one layer of an ANN neuron:

$$a_i = f_i(IW_i p + b_i) \tag{14}$$

where:

IW_i = Scalar weight.

p = Scalar input.

b_i = Scalar bias.

f_i = Transfer Function

The neural network can be constructed with numerous layers to enable it to approximate any function within a limited number of discontinuities, serving as a general function approximator. Feedforward networks are the name given to these several layers. An introductory weight and bias are used in this network training. In order to consider the optimum network behavior, it also requires a set of examples (such as network inputs p and target outputs). This set example is a loss function, according to Erickson, which is a function that assesses the discrepancy between the actual outcome and the planned output [47]. To reduce the network, mean square error, the weights and biases of the network are repeatedly changed during training (performance function). Equations 15 through 24 show the Levenberg-Marquardt algorithm.

$$E(n) = \frac{1}{2}\sum\nolimits_{k=1}^{m} e_k^2(n) \,;\, e_k(n) = d_k(n) - y_k(n) \quad (15)$$

where:
n= is a training epoch.
m= number of outputs.
d = desired (target) output.
y_k = actual ANN output.
E = mean-squared error surface.
The weight update can be written as:

$$W(n+1) = W(n) + \Delta W(n) \quad (16)$$

Thus,

$$\Delta W(n) = -\eta \frac{\partial E(n)}{\partial W(n)} = -\eta \nabla E(n) \quad (17)$$

where:
W = is weight update.
η = is a training rate.
Due to the inherent inaccuracy of serial computation, including layer-by-layer base calculation, every technique that uses the straightforward steepest descent minimization based on first-order minimization is extremely slow [48]. Thus, the weight update must be carried out utilizing combinatory approaches, such as adaptive learning rate $\eta*$ and momentum term α, to prevent oscillation around a local minimum and to speed up the training.

$$\Delta W(n) = -\eta * \nabla E(n) + \alpha \Delta W(n-1) \quad (18)$$

By averaging the gradient locally, the momentum approach will attempt to obtain some information regarding the error surface's curvature. The second approach then employs optimization methods that use a second-order derivative of the performance index or cost function J(w) to minimize the error E(n).

$$\Delta J(W_{n+1}) = J(W) + \Delta W \nabla J(w) + \frac{1}{2} W \nabla^2 J(w) \quad (19)$$

where:

$$\frac{\partial E(n)}{\partial (W(n))} = \nabla J(w) = g \quad (20)$$

Which is the index of the gradient of performance, and

$$\frac{\partial^2 E(n)}{\partial^2 (W(n))} = \nabla^2 J(w) = H \quad (21)$$

Without having to compute the Hessian matrix, this algorithm was created to approach the second-order training speed [49]. The hessian matrix can be expressed as follows where the performance function is the sum of squares, as in training feed-forward networks:

$$H(n) = J^T(n) J(n) \quad (22)$$

and the gradient can be written as:

$$g(n) = J^T(n) E(n) \quad (23)$$

where:
J = is the Jacobian matrix that contains the first derivatives of the network errors concerning the weights and biases.
In comparison to computing the Hessian matrix, computing the Jacobian matrix can be done using a normal backpropagation method. This approximate representation of the Hessian matrix [48] is then used by the LM-ANN method in the Newton-like update formula that follows.

$$W(n+1) = W(n) - \left[J(n)^T J(n) + \mu I\right]^{-1} J^T(n) E(n) \quad (24)$$

where:
μ = is a nonnegative scalar, that controls both magnitude and direction.
I = is an identity matrix.
This is exactly like Newton's approach when the scalar is zero. This transforms into a gradient descent with a tiny step size, though, when is large. Newton's approach is renowned for achieving minimum mistakes with more accuracy and speed. As a result, the goal of this method is to transition as rapidly as feasible to Newton's method. As a result, it drops following each improvement in performance function (imagine a step that is successful) and only increases when a tentative step would improve performance function. By doing it this way, the algorithm's performance function will always be decreased. This made the algorithm one of the quickest techniques for training feed-forward neural networks of a moderate size (up to several hundred weights) [49], [50]. The block diagram for Levenberg–Marquardt algorithm training can be seen in figure 5 below.

III. MODEL BASED ON FIELD MEASUREMENT

In this section we would like to present field measurement result, continue with clustering SAW Clustering Model that use field measurement result data, then we present the ANFIS Near Ground Propagation Model for power transfer and wireless link quality, and Neural Network model for adaptive and dynamic transmission power control.

A. FIELD MEASUREMENT RESULT

54 distinct LoRaWAN radio configurations were used in the measurement at this jungle location. The GPS coordinates for the site were 6°21′28.2″ South 106°54′16.4″ East (East suburban Jakarta). The measurement results such as RSSI and SNR parameter can be seen in figure 6 and figure 7 based on different LoRaWAN configurations, such as frequency,range, bandwidth, and spreading factors.

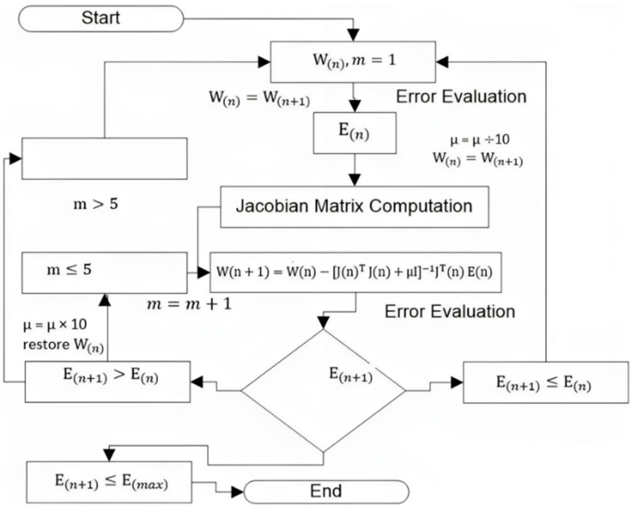


FIGURE 5. Block diagram for training using Levenberg–Marquardt algorithm according to Yu and Wilamowski [51].

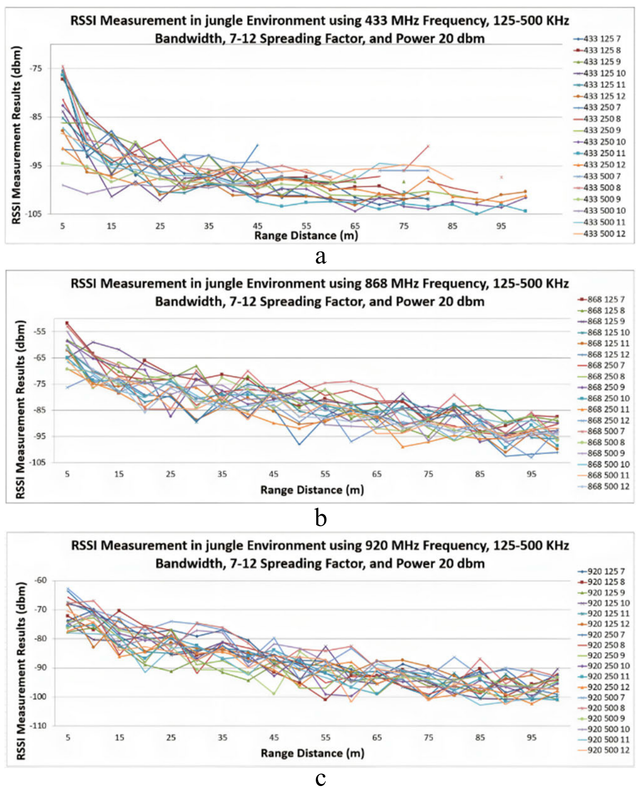


FIGURE 6. RSSI measurement in jungle location with frequency (top) 433 MHz, (middle) 868 MHz, (below) 920 MHz.

B. SAW CLUSTER ROUTING MODEL

The simulation was done in matlab to simulate in area 200 × 200 meter, with 20 WSN nodes that placed randomly. The routing protocols first step would be selection of cluster head using SAW MCDM algorithm and to find how many CH children node follower. Because we are using simulation, the ANFIS near ground propagation model was used to determine

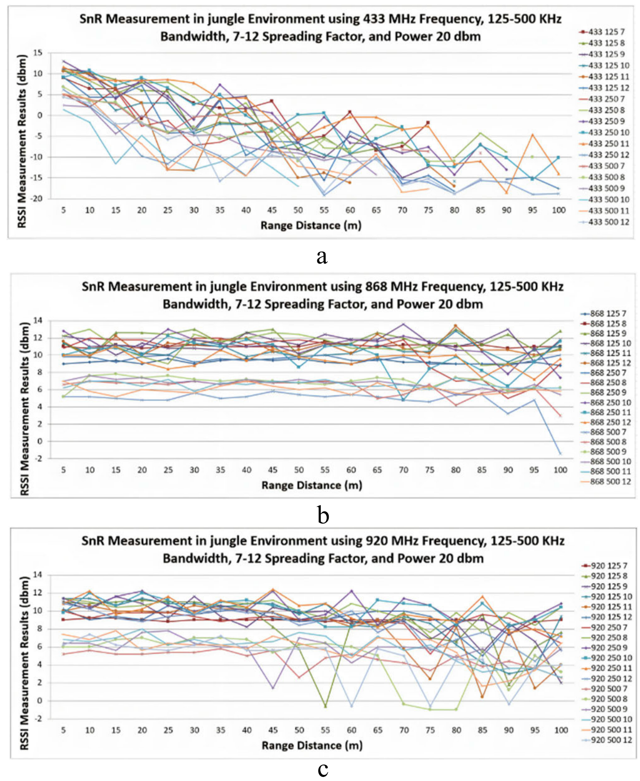


FIGURE 7. SNR measurement in jungle location with frequency (top) 433 MHz, (middle) 868 MHz, (below) 920 MHz.

the RSSI and the SNR value that we have based on each nodes position. However, in the real world where we don't know the exact location of each node's position, we can get the RSSI and the SNR value just by receiving the data packet of each nodes wireless data transmission. However, in this simulation we are using the maximum value for each parameter for cluster head selection. The initial parameter would be 100 meter maximum range, 500 KHz bandwidth, 20 dBm for power transmission, and 7 Spreading factor. This first step then continues with calculation Cluster Head selection with the highest score. The criteria for this would be the quantity of node follower with weight 0.1, the connection with other cluster head or gateway with weight 1, and minimum RSSI level at −100 dBm would be weight 1. Hence the SAW Cluster Routing Model can be seen on equation 25:

$$A = \sum\nolimits_{j=1}^{n} \left(1\ r_{ij} + 1 r_{ij} + 0.1\ r_{ij}\right) \tag{25}$$

where:

r_{ij} = performance value of alternative A, i=1,2,…,m, and j =1,2,…,n.

A_i = Selection Node output value.

C. ANFIS FOR ENVIRONMENTAL MODELING POWER TRANSFER AND WIRELESS LINK QUALITY NEAR GROUND PROPAGATION MODEL

Levenberg Marquardt Artificial Neural Network only created an adaptive and dynamic transmission power control model.

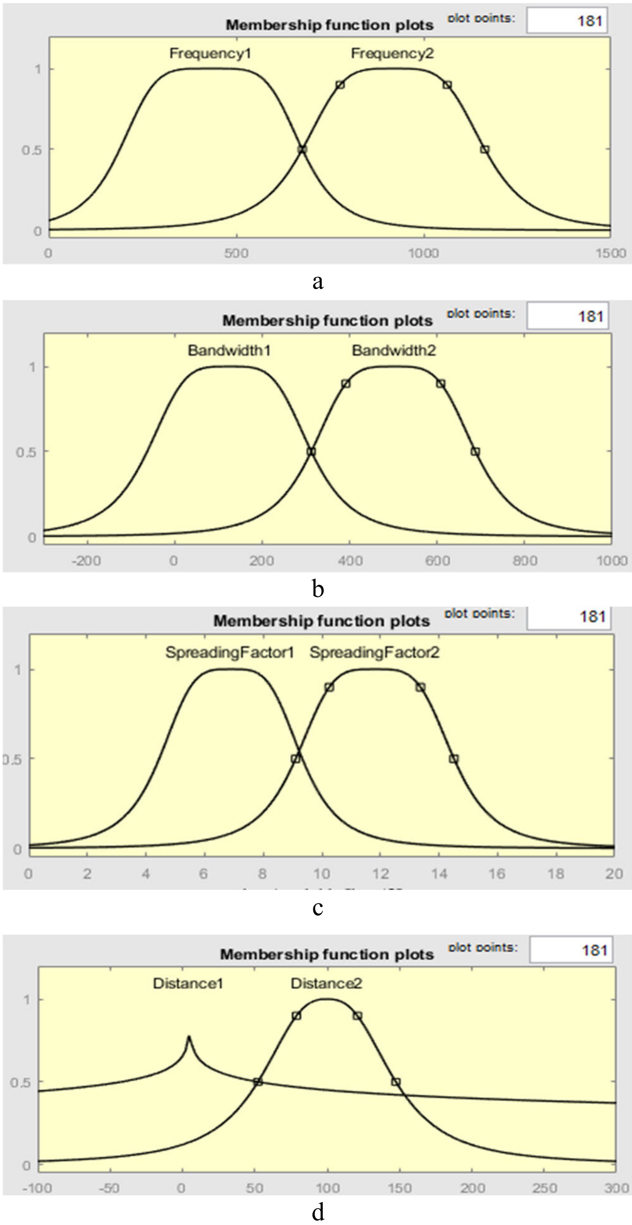


FIGURE 8. ANFIS for Power Transfer (RSSI) Near Ground Propagation Model in jungle Environment, function of (a) Frequency 433 MHz, 868 MHz, and 920 MHz (b) bandwidth 125 KHz, 250 KHz, and 500 KHz, (c) Spreading factor from 7 to 12, (d) Distance from 5 meter to 100 meter.

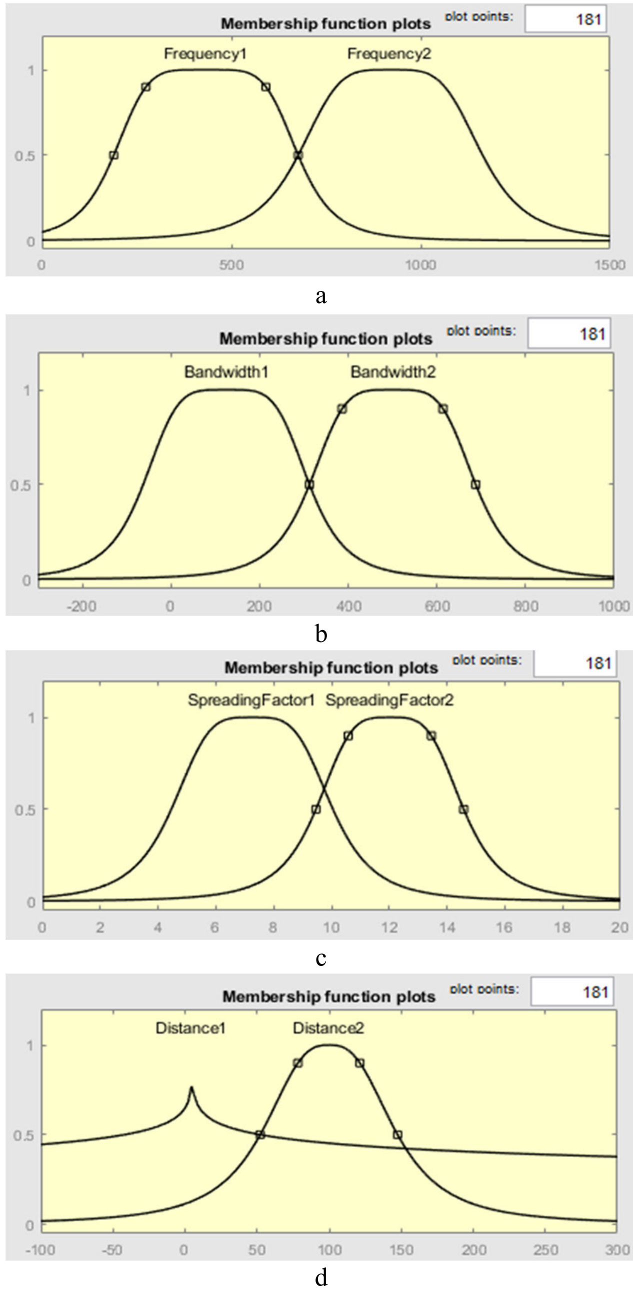


FIGURE 9. ANFIS for Wireless Link Quality (SNR) Near Ground Propagation Model in Jungle Environment SNR Model(a) Block Diagram (b) function of bandwidth 125 KHz, 250 KHz, and 500 KHz, (c) function of Spreading factor from 7 to 12, (d) function of range from 5 meter to 100 meter.

To provide the same environmental simulation, we need another method. This environmental simulation will create RSSI and SNR value with specific configurations for spreading factor, bandwidth, power transmission, and distance. This environmental simulation can be modelled using ANFIS. Once assisted by ANFIS algorithm, then we can make Power Transfer and Wireless Link Quality Near Ground Propagation Model in jungle environment. For Power Transfer Model, The RSSI in field measurement data in figure 6 will act as data training for ANFIS. Thus, using generalized bell function in equation 3 we present the fuzzification of RSSI field measurement on figure 8 and table 2.

For Wireless link quality model, The SNR in field measurement data in figure 7 will act as data training for ANFIS. Thus, using generalized bell function in equation 3 we present the fuzzification of SNR field measurement on figure 9 and table 2.

Because ANFIS model was based on fuzzy rule reasoning. Therefore, the fuzzy ANFIS rule was in general form, and it can be used for ANFIS near ground propagation for power transfer and wireless link quality model. Therefore, for ANFIS Rule can be seen on table 3 below:

TABLE 2. ANFIS near ground propagation model, fuzzification input for power transfer (RSSI) and wireless link quality (SNR) model.

Linguistic Variable	Fuzzification input for RSSI			Fuzzification input for SNR		
	a	b	c	a	b	c
Frequency 1	244	2.385	433	244	2.555	433
Frequency 2	244	2.059	920	244	2.078	920
Bandwidth 1	188	2.01	125	188	2.26	125
Bandwidth 2	188	2.006	500	188	2.197	500
Spreading Factor 1	2.44	1.934	6.891	2.56	1.966	7.255
Spreading Factor 2	2.7	1.995	11.81	2.3	1.906	12.02
Distance 1	47.5	0.142	4.996	47.5	0.137	5.004
Distance 2	47.7	1.341	99.85	47.7	1.38	99.85

And for fuzzy constant output for ANFIS near ground propagation power transfer and wireless link quality model can be seen in table 4 below:

Thus, the mathematical model for ANFIS near ground propagation for power transfer (RSSI) and wireless link quality (SNR) model as can be seen in equation 26 below:

$$B_i = \frac{\sum_{k=0}^{n} \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^{2b_i}\right]} + \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^{2b_i}\right]} F_A^P}{\sum_{k=0}^{n} \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^{2b_i}\right]} + \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^{2b_i}\right]}} \tag{26}$$

where:

x = Input variables such as frequency, bandwidth, spreading factor, and range.

$a_i b_i c_i$ = parameter of generalize bell function such as a_i Describe the broadness, of the membership function input of frequency, bandwidth, spreading factor, and range, while b_i describe the form of the curve on either side of the middle of frequency, bandwidth, spreading factor, and range, and while c_i describe the center of the membership function of frequency, bandwidth, spreading factor, and range.

F_A^P = Defines the output variables constant Level generated automatically by ANFIS based on number of fuzzy rules/inferences. in this case the output variable would be RSSI and SNR.

n = Total summation of ANFIS Inference/Rule firing Strength index.

B_i = output such as RSSI or SNR.

D. LM-ANN FOR ADAPTIVE AND DYNAMIC TRANSMISSION POWER CONTROL

Assisted with Levenberg Marquardt Artificial Neural Network algorithm then we can make adaptive and dynamic transmission power control model in jungle environment. The RSSI and the SNR in field measurement data in figure 6 and figure 7 will act as data training for it. The proposed LM-ANN model is shown in equation 27 below.

TABLE 3. ANFIS near ground propagation model, inference/rule for firing strength power transfer and wireless link quality model.

Rule Index	Rule Firing	Rule Output
1	If input1 is frequency1 and input2 is bandwidht1 and input3 is Spreading factor1 and input4 is distance1	Then output1 is out1mf1
2	If input1 is frequency1 and input2 is bandwidht1 and input3 is Spreading factor1 and input4 is distance2	Then output1 is out1mf2
3	If input1 is frequency1 and input2 is bandwidht1 and input3 is Spreading factor2 and input4 is distance1	Then output1 is out1mf3
4	If input1 is frequency1 and input2 is bandwidht1 and input3 is Spreading factor2 and input4 is distance2	Then output1 is out1mf4
5	If input1 is frequency1 and input2 is bandwidht2 and input3 is Spreading factor1 and input4 is distance1	Then output1 is out1mf5
6	If input1 is frequency1 and input2 is bandwidht2 and input3 is Spreading factor1 and input4 is distance2	Then output1 is out1mf6
7	If input1 is frequency1 and input2 is bandwidht2 and input3 is Spreading factor2 and input4 is distance1	Then output1 is out1mf7
8	If input1 is frequency1 and input2 is bandwidht2 and input3 is Spreading factor2 and input4 is distance2	Then output1 is out1mf8
9	If input1 is frequency2 and input2 is bandwidht1 and input3 is Spreading factor1 and input4 is distance1	Then output1 is out1mf9
10	If input1 is frequency2 and input2 is bandwidht1 and input3 is Spreading factor1 and input4 is distance2	Then output1 is out1mf10
11	If input1 is frequency2 and input2 is bandwidht1 and input3 is Spreading factor2 and input4 is distance1	Then output1 is out1mf11
12	If input1 is frequency2 and input2 is bandwidht1 and input3 is Spreading factor2 and input4 is distance2	Then output1 is out1mf12
13	If input1 is frequency2 and input2 is bandwidht2 and input3 is Spreading factor1 and input4 is distance1	Then output1 is out1mf13
14	If input1 is frequency2 and input2 is bandwidht2 and input3 is Spreading factor1 and input4 is distance2	Then output1 is out1mf14
15	If input1 is frequency2 and input2 is bandwidht2 and input3 is Spreading factor2 and input4 is distance1	Then output1 is out1mf15

$$C_i = purelin(IW_{i,n} \sum_{i=1}^{ni} (tansig(\sum_{i=1}^{ni} (IW_{i,n} p_i + b_i))) + b_i) \tag{27}$$

where:

IW_i = Scalar weight.

p = Scalar input such as frequency, RSSI, and SNR.

b_i = Scalar bias.

C_i = Scalar output such as Power transmits, bandwidth, and spreading factor.

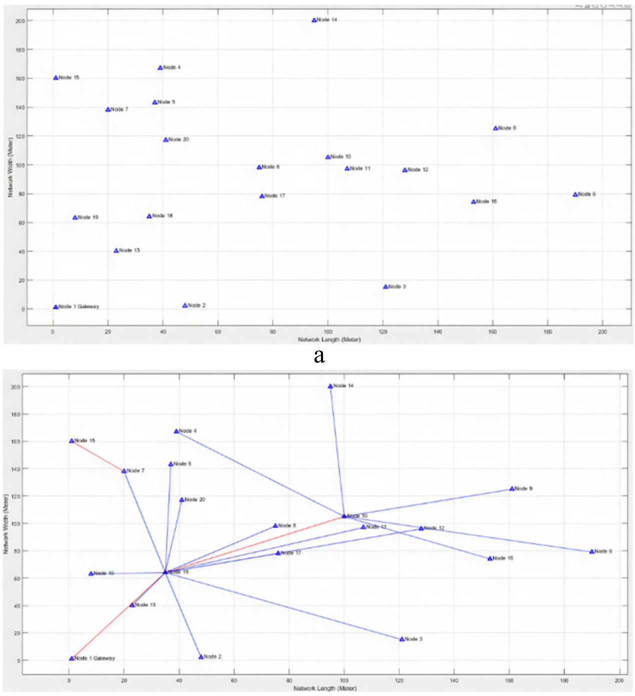
In this model we have 3 input and 3 output. The frequency, RSSI, and SNR are proposed as input elements, while Power transmits, bandwidth, and spreading factor are proposed for output element.

IV. SOWSN ROUTING MODEL EVALUATION

In this section, the evaluation of each SOWSN Feature model will be presented. The first model would be SAW clustering model, the second would be ANFIS near ground model together with LM-ANN adaptive and dynamic transmission power control model.

A. EVALUATION USING PAYLOAD PARAMETER

In this evaluation we have used 20 WSN nodes placed randomly in 200-meter square. Using equation 25 then we can see the evaluation result from clustered routing in figure 10 below:



a

b

FIGURE 10. 20 WSN node cluster Simulation. (a) Random Node placement. (b) Clustering using SAW Model.

As we see from figure 10 above, the SAW Cluster Routing Model has managed to cluster against all nodes. It shows us that this model has created 2 main clusters with node 18 and 10 acting as cluster head, and node 7 acting as subcluster head. The blue line indicates normal transmission link, and red line indicates backbone transmission link.

Theoretically, according to the LoRaWAN datasheet the maximum payload was 255 bytes for single transmission [52]. Thus, we can calculate how many payloads will be dropped or will be succeed using this routing model. WSN Transmission Payload success was a cumulative from total transmission from all nodes in this network. hence, we can see the payload success in wireless transmission can be seen on figure 11 below:

As we see from figure 11 above, before clustering with SAW model, because of their range that exceeds almost 200 meter (example from node 14 that send data to node 1

TABLE 4. ANFIS near ground propagation model, constant output power transfer and wireless link quality model.

Fuzzy Constant Output F_A^P	Jungle Environment	
	SNR (dB)	RSSI (dBm)
1	14.25	-82.95
2	-19.52	-114.2
3	13.2	-90.66
4	-30.35	-111.9
5	5.06	-87.94
6	-17.21	-104
7	2.233	-97.4
8	-32.28	-97.25
9	11.77	-60.58
10	9.915	-107.6
11	12.61	-67.59

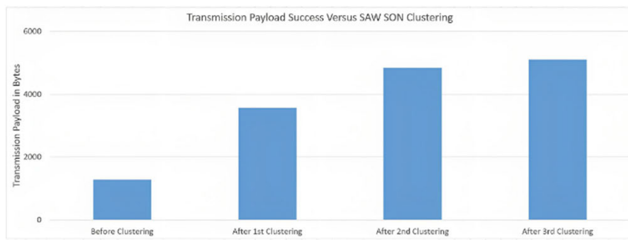


FIGURE 11. Wireless sensor network Transmission Payload success in effect of SAW SOWSN clustering routing.

gateway) it will cause payload to be lost or missing. By clustering with just using SAW model, we manage to improve this condition from just receiving 1275 bytes and in the end, we have received 5100 bytes from all nodes.

B. EVALUATION USING TRANSMISSION POWER CONTROL PARAMETER

In this section the reduction on the power transmitted in each wireless sensor network node is presented. The parameter input for this method would be using frequency, RSSI, and SNR. In simulation we can get the RSSI and SNR value using ANFIS model, while adaptive and dynamic transmission power control capability will be provided with LM-ANN model. However, for frequency we are using 920 MHz only because the regulation in Indonesia [53]. While we can keep with the same RSSI values, however, we can change the SNR value which in this example would be 3 and 9. The power conservation result from each node can be seen from figure 12 below:

As we see from figure 12 above, the the LM-ANN adaptive and dynamic transmission power control model has managed to do power conservation against all nodes. It shows us The

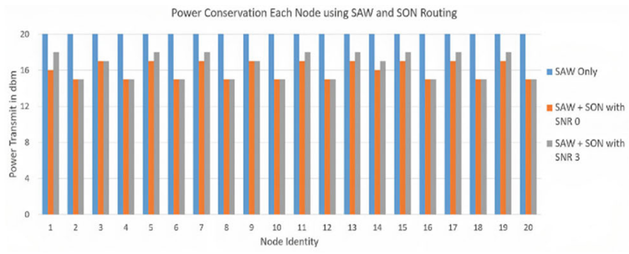


FIGURE 12. Power conservation results using ANFIS near ground propagation model and LM-ANN adaptive transmission power control model for each node.

SOWSN has managed to do power conservation for all nodes. If we are using SNR 3 then the smallest power transmit can be reduced to become 13.71 dBm and if we are using SNR 9 then the smallest power transmit can be reduced to become 12.76 dBm.

C. EVALUATION USING WSN NODE LIFETIME PARAMETER

We can calculate power conservation for WSN node Lifetime from this scenario below. The WSN application was built for soil PH monitoring in precision agriculture 4.0 application [53]. This application type doesn't need continues monitoring, it only needs just a few seconds for data collection, data transmission, and the rest of the activity will be sleep for all day long. We can use a minimize duty cycle such as 0.1 % day active (86.4 second), that can be expressed using:

$$D = \frac{T_{active}}{T_{period}} \tag{28}$$

where:
D = Duty Cycle
T_{active} = Time where WSN node active
T_{period} = Total period $T_{active} + T_{sleep}$

WSN node consumption can be modeled with battery charge consumption, and according to Orrelana the battery charge consumption component was the number of sleep and active charge consumption [54]. Then we can express:

$$CC_{cycle} = N_{sleep}CC_{sleep}T_{sleep} + N_{active}CC_{active}T_{active} \tag{29}$$

where:
$\mathrm{CC_{cycle}}$ = Total charge consumption
$\mathrm{N_{sleep}}$ = Number of deep sleeps in 1 period
$\mathrm{CC_{sleep}}$ = charge consumption on deep sleep mode
$\mathrm{N_{active}}$ = Number of active in 1 period
$\mathrm{CC_{active}}$ = charge consumption on active mode
T_{sleep} = time sleep in 1 period
T_{active} = time active in 1 period

The quantity of cycles that we can be done with a battery capacity is given by:

$$N_{cycle} = \frac{Batt_{cap}}{CC_{cycle}} \tag{30}$$

where:
N_{cycle} = Number of cycle (days) can be performed
$Batt_{cap}$ = Battery capacity

If we then substitute expressions 28 and 29 into expression 30, and in considering duty cycle only 0.1% where active mode was 86.4 second for one day period. Therefore, we can express the new charge consumption models:

$$N_{cycle} = \frac{Batt_{cap}}{N_{sleep}CC_{sleep}86313.6 + N_{active}CC_{active}86.4} \tag{31}$$

We can calculate how many days WSN network will be running based on node 18 power profile. If we take node 18 the most far coverage, and current measurement is for soil PH monitoring application [55] with battery capacity at 3000 mAH, then we can calculated the WSN network lifetime. The WSN network lifetime based on node 18 power profile can be seen in figure 13 below:

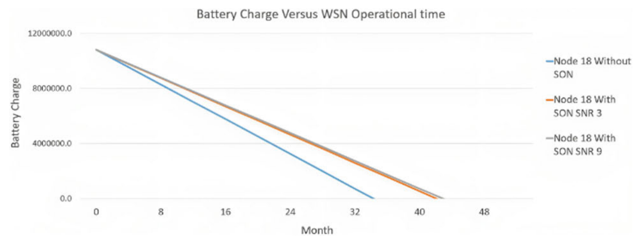


FIGURE 13. WSN lifetime is based on Node 18 power profile.

As we see from figure 13 above, the LM-ANN adaptive and dynamic transmission power control model has managed to do power conservation. In the end, it has managed to increase WSN node lifetime. This can be seen from node 18 that acts as super CH (all traffic must pass through here before going to node 1 gateway) for this network. With 0.1 duty cycle, if we just using just clustering model only, node 18 has managed to operate until 32 months. If we combine all three-feature model, node 18 has managed to operate until 42 months.

D. EVALUATION USING BIT ERROR RATE (BER) PARAMETER

For Bit Error Rate (BER) we can calculate it using Dias BER Rayleigh model. This was because we are do field measurement in jungle environment where no direct line for wireless data transmission caused by vegetation blocking, thus we get a Rayleigh environment. Dias solution for LoRaWAN wireless transmission in Rayleigh environment for BER parameter, was to model and calculate it using SNR and spreading factor parameter [56]. The Dias BER expression can be wrote:

$$P_b = \frac{2^{SF-1}}{2^{SF}-1}\left[1 - \frac{\Gamma\left(2^{SF}\right)\Gamma\left(\frac{2+(\gamma 2^{SF})}{1+(\gamma 2^{SF})}\right)}{\Gamma\left(\frac{1+2^{SF}+(\gamma 2^{SF})}{1+(\gamma 2^{SF})}\right)}\right] \tag{32}$$

where:
P_b = The average error bit probability.

γ = SNR.
Γ = Gamma function.
SF = Spreading factor.

Using expression 32 we can calculate BER for each nodes wireless data transmission to their respective cluster head. We can see the BER change caused by implementation on SOWSN model in in each node in figure 14 below:

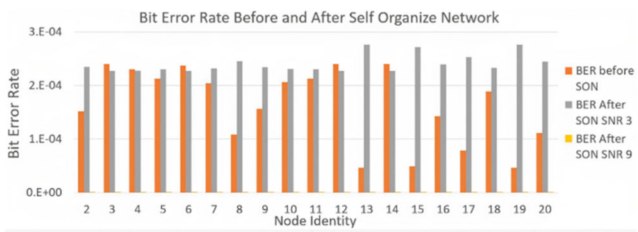


FIGURE 14. Nodes wireless data transmission BER in SOWSN implementation.

As we see from figure 14 above, before the SOWSN implementation the BER for each nodes wireless transmission has a minimum value with the smallest one at about 46×10^{-6} from node 13 and node 18. However, after SOWSN implementation, for example if we are using SNR 3 then the BER for each nodes wireless transmission has increased a lot, with the smallest BER value 226×10^{-6} from node 3 and node 12 and if we are using SNR 9 the BER for each wireless sensor network node has decrease with the smallest BER value at about 58×10^{-8} from node 2 and also node 3.

E. EVALUATION USING TRANSMISSION TIME ON AIR (TOA) PARAMETER

Time on Air means a total transmission time for one time transmission in LoRaWAN. This can be calculated using LoRaWAN TOA expression as stated in their datasheet [57], that can be seen from expression 33 below:

$$T_{Packet} = \left((n_{Preamble} + 4.25) \frac{2^{SF}}{BW} \right) + \left((8 + \max \left(ceil \left(\frac{8PL - 4SF + 28 + 16 - 20H}{4\,(SF - 2DE)} \right) \times (CR + 4)\,, 0 \right) \frac{2^{SF}}{BW} \right) \tag{33}$$

where:

T_{Packet} = Total transmission time or TOA (Time On Air).
$n_{Preamble}$ = The number of Preamble Symbols.
BW = Bandwidth.
SF = Spreading factor.
PL = The number of Payload bytes.
H = 0 when the header is enabled and 1 when no header is present.
DE = 1 when the low data rate optimization is enabled 0 for disabled.
CR = Is the coding rate from 1 to 4.

Before doing calculations, we need to do configuration for LoRaWAN modem. The LoRaWAN Modem configuration for this simulation such as, Header 1, Data Optimizer 0, Coding Rate 4/5, LoRaWAN packet configuration such as Payload Length 255 Bytes, and total Preamble Length 6.25 Symbols. Thus, we can get TOA for each WSN node wireless transmission link in figure 15 below:

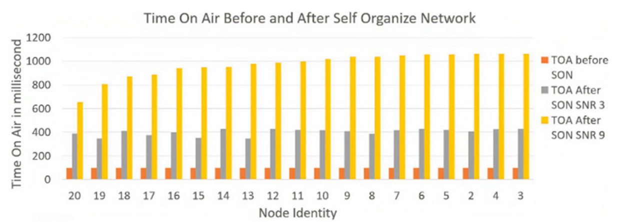


FIGURE 15. Wireless data transmission TOA in SOWSN implementation.

As we see from figure 15 above, before SOWSN implementation the TOA for each nodes wireless transmission has minimum value with the smallest one at about 96.8 millisecond for all nodes. After SOWSN implementation the TOA for each nodes wireless transmission with SNR 3 has increased with the smallest TOA value at about 346.78 millisecond for node 13 and node 19. However, with SNR 9 the TOA has increased more with the smallest value at about 654.7 millisecond for node 20.

V. CONCLUSION

In this paper, we present a SOWSN Cluster Routing to enhance wireless sensor network operation. The simulation of SOWSN Cluster Routing was made by 3 different methods such as SAW for Clustering, ANFIS near ground propagation model for simulated environment, and Neural Network for adaptive and dynamic transmission power control model. Using these 3 methods we managed to create cluster routing method while simultaneously creating power conservation in wireless sensor network operation. To validate this claim, the evaluation is conducted using Payload, WSN Lifetime, BER and TOA parameter calculations. The payload loss can be improved using SAW clustering from 1275 bytes to 5100 bytes. The power conservation used for transmission power control parameter can be seen to be reduced from 20 dbm to just 12.76 dbm. This reduction in the end will cause an increase in WSN lifetime from 32 months to 42 months. ANFIS provides a model for environment simulation which is power transfer and wireless link quality in near ground propagation. The LM ANN provides power transmission reduction effect by creating an adaptive and dynamic transmission power control model. The BER parameter can increase or decrease according to the SNR that we use, however the TOA parameter will always increase because of adaptive and dynamic transmission power control model. The original TOA time for all node was at 96.832 millisecond. However, with SNR 3 TOA time will increase starting from 346.78 millisecond and With SNR 9 TOA time will increase starting from 654.7 millisecond.

REFERENCES

[1] *Requirements for Support of Ubiquitous Sensor Network (USN) Applications and Services in the NGN Environment*, ITU-T, document ITU-T Y.2221, 2010.

[2] A. M. Baharudin and W. Yan, "Long-range wireless sensor networks for geo-location tracking: Design and evaluation," in *Proc. Int. Electron. Symp. (IES)*, Sep. 2016, pp. 76–80, doi: 10.1109/ELECSYM.2016.7860979.

[3] F. Arslan, "On the wireless sensor network for medical instruments monitoring system," *Int. J. Sci. Eng. Res.*, vol. 9, no. 8, pp. 88–96, Aug. 2018.

[4] C. Del-Valle-Soto, C. Mex-Perera, J. A. Nolazco-Flores, R. Velázquez, and A. Rossa-Sierra, "Wireless sensor network energy model and its use in the optimization of routing protocols," *Energies*, vol. 13, no. 3, pp. 1–33, 2020, doi: 10.3390/en13030728.

[5] J. Polo, G. Hornero, C. Duijneveld, A. García, and O. Casas, "Design of a low-cost wireless sensor network with UAV mobile node for agricultural applications," *Comput. Electron. Agricult.*, vol. 119, pp. 19–32, Nov. 2015, doi: 10.1016/j.compag.2015.09.024.

[6] S. A. Malek, S. D. Glaser, and R. C. Bales, "Wireless sensor networks for improved snow water equivalent and runoff estimates," *IEEE Access*, vol. 7, pp. 18420–18436, 2019, doi: 10.1109/ACCESS.2019.2895397.

[7] A. Balamurugan, M. P. Kumar, and R. M. Kumar, "Intelligent application of WSN for forest monitoring," *Int. J. Innov. Technol. Exploring Eng.*, vol. 9, no. 6, pp. 89–94, Apr. 2020, doi: 10.35940/ijitee.f3581.049620.

[8] H. Sharma, A. Haque, and F. Blaabjerg, "Machine learning in wireless sensor networks for smart cities: A survey," *Electronics*, vol. 10, no. 9, p. 1012, Apr. 2021, doi: 10.3390/electronics10091012.

[9] M. Ahmed, M. Salleh, M. I. Channa, and M. F. Rohani, "Energy efficient routing protocols for underwater wireless sensor networks: A review," *TELKOMNIKA, Telecommun. Comput. Electron. Control*, vol. 15, no. 1, p. 212, Mar. 2017, doi: 10.12928/TELKOMNIKA.v15i1.4706.

[10] G. Liu, Z. Liu, V. S. Sheng, L. Zhang, and Y. Yang, "A novel energy-efficient, static scenario-oriented routing method of wireless sensor network based on edge computing," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–25, Jan. 2022, doi: 10.1155/2022/3450361.

[11] N. Azmi, L. M. Kamarudin, M. Mahmuddin, A. Zakaria, A. Y. M. Shakaff, S. Khatun, K. Kamarudin, and M. N. Morshed, "Interference issues and mitigation method in WSN 2.4 GHz ISM band: A survey," in *Proc. 2nd Int. Conf. Electron. Design (ICED)*, Aug. 2014, pp. 403–408, doi: 10.1109/ICED.2014.7015839.

[12] Y. Yun and Y. Xia, "Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications," *IEEE Trans. Mobile Comput.*, vol. 9, no. 9, pp. 1308–1318, Sep. 2010, doi: 10.1109/TMC.2010.76.

[13] A. Akbas, H. U. Yildiz, B. Tavli, and S. Uludag, "Joint optimization of transmission power level and packet size for WSN lifetime maximization," *IEEE Sensors J.*, vol. 16, no. 12, pp. 5084–5094, Jun. 2016, doi: 10.1109/JSEN.2016.2548661.

[14] R. A. Uthra and S. V. K. Raja, "QoS routing in wireless sensor networks—A survey," *ACM Comput. Surv.*, vol. 45, no. 1, pp. 1–12, Nov. 2012, doi: 10.1145/2379776.2379785.

[15] J. Kumar, V. Rishiwal, and M. I. Ansari, "Quality of service in wireless sensor networks: Imperatives and challenges," *Int. J. Sensors, Wireless Commun. Control*, vol. 9, no. 4, pp. 419–431, Sep. 2019, doi: 10.2174/2210327909666190129154033.

[16] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, no. 8, pp. 11113–11153, Aug. 2012, doi: 10.3390/s120811113.

[17] A. Zeb, A. K. M. M. Islam, M. Zareei, I. A. Mamoon, N. Mansoor, S. Baharun, Y. Katayama, and S. Komaki, "Clustering analysis in wireless sensor networks: The ambit of performance metrics and schemes taxonomy," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 7, Jul. 2016, Art. no. 4979142, doi: 10.1177/155014774979142.

[18] B.-S. Kim, B. Shah, F. Al-Obediat, S. Ullah, K. Kim, and K.-I. Kim, "An enhanced mobility and temperature aware routing protocol through multi-criteria decision making method in wireless body area networks," *Appl. Sci.*, vol. 8, no. 11, p. 2245, Nov. 2018, doi: 10.3390/app8112245.

[19] B.-S. Kim, S. Ullah, K. H. Kim, B. Roh, J.-H. Ham, and K.-I. Kim, "An enhanced geographical routing protocol based on multi-criteria decision making method in mobile ad-hoc networks," *Ad Hoc Netw.*, vol. 103, Jun. 2020, Art. no. 102157, doi: 10.1016/j.adhoc.2020.102157.

[20] X. Xu, Y. Li, T. Huang, Y. Xue, K. Peng, L. Qi, and W. Dou, "An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks," *J. Netw. Comput. Appl.*, vol. 133, pp. 75–85, May 2019, doi: 10.1016/j.jnca.2019.02.008.

[21] W. Sun, K. Wei, Z. Liu, Q. Li, and X. Xu, "Linear quadratic Gaussian control for wireless communication reliability for a mobile monitoring robot in a UHV power substation," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4149–4159, Sep. 2022, doi: 10.1109/JSYST.2022.3143866.

[22] S. S. Vidhya, M. Senthilkumar, and V. A. Narayanan, "An empirical investigation based quality of service aware transmission power prediction in low power networks," *Sādhanā*, vol. 47, no. 4, p. 239, Nov. 2022, doi: 10.1007/s12046-022-01982-4.

[23] M. S. Philip and P. Singh, "Adaptive transmit power control algorithm for dynamic LoRa nodes in water quality monitoring system," *Sustain. Comput., Informat. Syst.*, vol. 32, Dec. 2021, Art. no. 100613, doi: 10.1016/j.suscom.2021.100613.

[24] M.-Y.-M. Mirza, G. Ahmed, and N. M. Khan, "Model-based adaptive transmission power control (MATPoC) for wireless sensor networks in fading environment," *J. Circuits, Syst. Comput.*, vol. 26, no. 9, Sep. 2017, Art. no. 1750143, doi: 10.1142/S0218126617501432.

[25] G. A. Mutiara, N. Suryana, and O. Mohd, "WSN nodes power consumption using multihop routing protocol for illegal cutting forest," *TELKOMNIKA, Telecommun. Comput. Electron. Control)*, vol. 18, no. 3, p. 1529, Jun. 2020, doi: 10.12928/TELKOMNIKA.v18i3.14844.

[26] N. Varela, D.-M. Jorge, A. Ospino, and N. A. L. Zelaya, "Wireless sensor network for forest fire detection," *Proc. Comput. Sci.*, vol. 175, pp. 435–440, Jan. 2020, doi: 10.1016/j.procs.2020.07.061.

[27] R. Meri, "Simple additive weighting (SAW) method on the selection of new teacher candidates at integrated Islamic elementary school," *Int. J. Inf. Syst.*, vol. 4, no. 1, pp. 428–435, 2020.

[28] P. C. Fishburn, "Letter to the editor—Additive utilities with incomplete product sets: Application to priorities and assignments," *Oper. Res.*, vol. 15, no. 3, pp. 537–542, Jun. 1967, doi: 10.1287/opre.15.3.537.

[29] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, 1975.

[30] N. Yusof, N. Bahiah, M. Shahizan, and Y. Chun, "A concise fuzzy rule base to reason student performance based on rough-fuzzy approach," in *Fuzzy Inference System: Theory and Applications*. London, U.K.: IntechOpen, 2012.

[31] I. Iswanto and I. Ahmad, "Second order integral fuzzy logic control based rocket tracking control," *J. Robot. Control*, vol. 2, no. 6, pp. 594–604, 2021, doi: 10.18196/jrc.26142.

[32] A. Adriansyah, Y. Gunardi, B. Badaruddin, and E. Ihsanto, "Goal-seeking behavior-based mobile robot using particle swarm fuzzy controller," *TELKOMNIKA, Telecommun. Comput. Electron. Control*, vol. 13, no. 2, p. 528, Jun. 2015, doi: 10.12928/telkomnika.v13i2.1111.

[33] R. Kristiyono and W. Wiyono, "Autotuning fuzzy PID controller for speed control of BLDC motor," *J. Robot. Control*, vol. 2, no. 5, pp. 400–407, 2021, doi: 10.18196/jrc.25114.

[34] B. AlKhlidi, A. T. Abdulsadda, and A. Al Bakri, "Optimal robotic path planning using intlligents search algorithms," *J. Robot. Control*, vol. 2, no. 6, pp. 519–526, 2021, doi: 10.18196/26132.

[35] G. Wu, C. Cui, and G. Wu, "Dynamic modeling and torque feedforward based optimal fuzzy PD control of a high-speed parallel manipulator," *J. Robot. Control*, vol. 2, no. 6, pp. 527–538, 2021, doi: 10.18196/jrc.26133.

[36] D. Sahid and M. Alaydrus, "Multi sensor fire detection in low voltage electrical panel using modular fuzzy logic," in *Proc. 2nd Int. Conf. Broadband Commun., Wireless Sensors Powering (BCWSP)*, Sep. 2020, pp. 31–35, doi: 10.1109/BCWSP50066.2020.9249400.

[37] J. G. Carvalho and C. T. Costa, "Identification method for fuzzy forecasting models of time series," *Appl. Soft Comput.*, vol. 50, pp. 166–182, Jan. 2017, doi: 10.1016/j.asoc.2016.11.003.

[38] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985, doi: 10.1109/TSMC.1985.6313399.

[39] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, Jun. 1993, doi: 10.1109/21.256541.

[40] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River, NJ, USA: Prentice-Hall, 1997.

[41] A. S. Rizal, A. Adriansyah, S. Budiyanto, S. C. Haryanti, and U. A. Rachmawati, ''Overcurrent relay coordination using an adaptive neuro fuzzy inference systems (ANFIS),'' *Electrotehnica, Electronica, Automatica*, vol. 68, no. 3, pp. 55–62, Sep. 2020, doi: 10.46904/eea.20.68.3.1108007.
[42] I. Y. Pratama, A. Wahab, and M. Alaydrus, ''Deep learning for assessing unhealthy lettuce hydroponic using convolutional neural network based on faster R-CNN with inception v2,'' in *Proc. 5th Int. Conf. Informat. Comput. (ICIC)*, Nov. 2020, pp. 1–6, doi: 10.1109/ICIC50835.2020.9288554.
[43] D. O. Oyewola and A. F. Augustine, ''Predicting impact of COVID-19 on crude oil price image with directed acyclic graph deep convolution neural network,'' *J. Robot. Control*, vol. 2, no. 2, pp. 103–109, 2021, doi: 10.18196/jrc.2261.
[44] Z. Dzulfikri, N. Nuryanti, and Y. Erdani, ''Design and implementation of artificial neural networks to predict wind directions on controlling yaw of wind turbine prototype,'' *J. Robot. Control*, vol. 1, no. 1, pp. 20–26, 2020, doi: 10.18196/jrc.1105.
[45] S. Sinha, U. Srivastava, V. Dhiman, and S. Mishra, ''Performance assessment of deep learning procedures on malaria dataset,'' *J. Robot. Control*, vol. 2, no. 1, pp. 12–18, 2020, doi: 10.18196/jrc.2145.
[46] Z. E. Mohamed, ''Using the artificial neural networks for prediction and validating solar radiation,'' *J. Egyptian Math. Soc.*, vol. 27, no. 1, pp. 1–12, Dec. 2019, doi: 10.1186/s42787-019-0043-8.
[47] B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline, ''Machine learning for medical imaging,'' *RadioGraphics*, vol. 37, no. 2, pp. 505–515, Mar. 2017, doi: 10.1148/rg.2017160130.
[48] S. Mammadli, ''Financial time series prediction using artificial neural network based on Levenberg–Marquardt algorithm,'' *Proc. Comput. Sci.*, vol. 120, pp. 602–607, Jan. 2017, doi: 10.1016/j.procs.2017.11.285.
[49] M. T. Hagan and M. B. Menhaj, ''Training feedforward networks with the Marquardt algorithm,'' *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994, doi: 10.1109/72.329697.
[50] S. Roweis, ''Levenberg–Marquardt optimization,'' Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 1996. [Online]. Available: https://cs.nyu.edu/~roweis/notes/lm.pdf
[51] H. Yu and B. M. Wilamowski, ''Levenberg–Marquardt training,'' in *Intelligent Systems*, 1st ed. Boca Raton, FL, USA: CRC Press, 2011.
[52] *Datasheet SX1276/77/78/79—137 MHz to 1020 MHz Low Power Long Range Transceiver*, Semtech, Camarillo, CA, USA, 2020.
[53] *Peraturan Menteri Komunikasi Dan Informatika Republik Indonesia Nomor 1 Tahun 2019 Tentang Penggunaan Spektrum Frekuensi Radio Berdasarkan Izin Kelas*, Kominfo, Jakarta, Indonesia, 2019.
[54] C. J. García-Orellana, M. Macías-Macías, H. M. González-Velasco, A. García-Manso, and R. Gallardo-Caballero, ''Low-power and low-cost environmental IoT electronic nose using initial action period measurements,'' *Sensors*, vol. 19, no. 14, p. 3183, Jul. 2019, doi: 10.3390/s19143183.
[55] E. Ramadhan, G. P. N. Hakim, M. H. I. Hajar, and A. Firdausi, ''Soil energy harvesting and its review on WSN node operation,'' in *Proc. Int. Symp. Electron. Smart Devices (ISESD)*, Jun. 2021, pp. 1–6, doi: 10.1109/ISESD53023.2021.9501765.
[56] C. F. Dias, E. R. de Lima, and G. Fraidenraich, ''Bit error rate closed-form expressions for LoRa systems under Nakagami and Rice fading channels,'' *Sensors*, vol. 19, no. 20, p. 4412, Oct. 2019, doi: 10.3390/s19204412.
[57] *SX1272/3/6/7/8: LoRa Modem Designer's Guide AN1200.13*, Semtech Corporation, Camarillo, CA, USA, 2013.