

Received March 18, 2020, accepted April 16, 2020, date of publication April 27, 2020, date of current version May 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990450

# Indoor Positioning System Using Artificial Neural Network With Swarm Intelligence

CHIA-HSIN CHENG<sup>1</sup>, (Member, IEEE), TAO-PING WANG<sup>1</sup>,  
AND YUNG-FA HUANG<sup>2</sup>, (Member, IEEE)

<sup>1</sup>Department of Electrical Engineering, National Formosa University, Yunlin 63201, Taiwan

<sup>2</sup>Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan

Corresponding author: Yung-Fa Huang (yfahuang@cyut.edu.tw)

**ABSTRACT** The global positioning system is the most popular outdoor positioning system; however, it is inapplicable to indoor environments. In this study, we created a hybrid algorithm, which applies an artificial neural network to resolve the problems of indoor positioning and a swarm-intelligence algorithm to perform the time-consuming task of adjusting parameters. Experiment results demonstrate that the positioning accuracy of the proposed hybrid algorithm is equivalent to the exhaustive method with search times that are far shorter.

**INDEX TERMS** Swarm intelligence, artificial neural network, indoor positioning.

## I. INTRODUCTION

Internet-of-Things (IoT) has been a technology that is extensively being used in various fields such as safety, transportation, industrial, healthcare and building [1]–[3]. In the world of IoT, location tracking becomes the next frontier. Location tracking for humans is already becoming an integral part of our lives, especially for navigation and safety [2], [4]. The global positioning system is the most popular outdoor positioning system; however, the direct-view transmission requirement precludes its application in indoor environments [5]–[7]. In this study, we used an artificial neural network (ANN) to resolve the problem of indoor positioning. ANNs are used in a wide range of applications, such as those encountered in image and signal processing [8]; however, optimizing the parameters often can be time-consuming. In recent years, swarm intelligence, an integral part in the field of artificial intelligence, is gradually gaining prominence in various researches [9], [10]. Swarm-intelligence algorithms with a combination of ANN model have been considered and developed in many areas with high reliability [11]. We therefore employed a swarm-intelligence algorithm to optimize the parameters in the ANN. A similar approach was adopted in [12], where the Bat algorithm (BA) was used to optimize the number of neurons in the hidden layer of a neural network. It was also used to adjust the

The associate editor coordinating the review of this manuscript and approving it for publication was Il-sun You<sup>1</sup>.

learning rate and inertia in predicting the output power of grid-connected photovoltaic systems. In [13], particle group optimization was used to optimize the learning rate and the number of neurons in the hidden layer of a neural network.

This paper applied two types of community intelligence algorithm to neural networks, and then conducted experiments applying the proposed hybrid algorithm to indoor regional positioning and indoor coordinate positioning. Finally, we compared the experiment results with those obtained using the conventional exhaustive method.

The remainder of this paper is organized as follows. Section II outlines signal propagation channel models and swarm algorithms. Section III presents the proposed swarm optimization algorithms for localization. Section IV presents our experiment results and discusses the performance of various positioning systems. Conclusions are drawn in Section V.

## II. BACKGROUND

### A. LOGNORMAL SHADOWING MODEL

The lognormal shadowing model can be used in outdoor and indoor environments [14], [15], wherein path loss is calculated as follows:

$$PL(d) [dB] = PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (1)$$

where  $PL(d)$  refers to path loss for the distance between the transmitter and the receiver,  $d_0$  is the close in distance, and “close-in distance” is a reference point for the measurement of radio field strength (generally close to the transmitter),

$n$  is the path loss exponent (generally determined by the channel environment) [14], and  $X_\sigma$  is a random variable with a normal distribution. The random variable has an average value of 0 with standard deviation of  $\sigma$  [14].

The power measured at distance  $d$  can be expressed as

$$P_r(d) [dBm] = P_t [dBm] - PL(d) [dB] \quad (2)$$

where  $P_r(d)$  represents the power measured at distance  $d$  and  $P_t$  is the transmitting power.

Combining Eqs. (1) and (2), gives us the following:

$$P_r(d) [dBm] = P_t - PL(d_0) - 10n \log\left(\frac{d}{d_0}\right) - X_\sigma \quad (3)$$

We adopted the scheme outlined in to determine power A and combined it with Eq. (3) to obtain the following:

$$P_r(d) [dBm] = A - 10n \log\left(\frac{d}{d_0}\right) - X_\sigma \quad (4)$$

Equation (4) can be used to simulate the received signal strength over various distances in a variety of environments.

## B. FINGERPRINTING TECHNIQUE

Fingerprinting involves mapping the relationship between RSSI and a real-world location within the positioning environment [16], [17]. Fingerprinting can generally be divided into an offline phase and an online phase [7], [16], [17]:

### 1) OFFLINE PHASE

The offline phase involves collecting the features and locations of various points in the positioning environment in order to establish a radio map.

### 2) ONLINE PHASE

The online phase involves inputting unknown data in order for the algorithm to estimate the position of the target.

## C. PARTICLE SWARM OPTIMIZATION

The particle swarm optimization (PSO) algorithm is a swarm-intelligence algorithm proposed by Yang *et al.* [18], Kennedy and Eberhart [19], del Valle *et al.* [20], and Tsujimoto *et al.* [21] inspired by the foraging behavior of birds. PSO is based on the sharing of experiences and information between individuals in order for the group to gradually move toward an ill-defined but beneficial goal [21]. The updating formulas for velocity and position used in PSO are respectively presented as follows:

$$V_i^t = w \cdot V_i^{t-1} + c_1 \cdot rand \cdot (pbest_i - X_i^{t-1}) + c_2 \cdot rand \cdot (gbest - X_i^{t-1}) \quad (5)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (6)$$

where  $t$  is the iteration index and  $i$  is the individual index. The particle position for the  $i$ -th individual of the  $t$ -th iteration is represented as  $X_i^t$ , whereas the current speed of the  $i$ -th individual of the  $t$ -th iteration is recorded as  $V_i^t$ . Note that

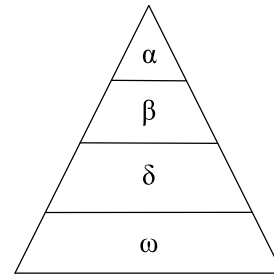


FIGURE 1. Social hierarchy of grey wolf [22].

$c_1$  and  $c_2$  are acceleration coefficients,  $w$  is the inertia weight coefficient,  $rand$  is a random number between [0,1],  $pbest_i$  refers to the best position experienced by the  $i$ -th particle,  $gbest$  refers to the best position experienced by all particles, and  $V_{max}$  is the maximum speed of individuals [20], [21].  $V_{max}$  determines the resolution (or precision) of the area between the current position and the best position. If  $V_{max}$  is too high, the particles may fly through the solution, and if  $V_{max}$  is too small, the particles cannot be explored enough, which leads to a local figure of merit. This restriction has three purposes: to prevent computational overflow; to implement artificial learning and attitude changes; to determine the granularity of the search for problem space. The inertia weight  $w$  keeps the particles inertia, which makes them have a tendency to expand the search space and has the ability to explore new areas. The acceleration constants  $c_1$  and  $c_2$  represent the weights of the statistical acceleration terms that push each particle toward the  $pbest_i$  and  $gbest$  positions.

## D. GRAY WOLF OPTIMIZER

The grey wolf optimization (GWO) algorithm is a swarm-intelligence algorithm developed by Mirjalili *et al.* [22] and Das *et al.* [23], in which individuals are assigned different social status levels from top to bottom as follows: leader ( $\alpha$ ), deputy ( $\beta$ ), and subordinates ( $\delta$  and  $\omega$ ). In assessing search outcomes,  $\alpha$ ,  $\beta$ , and  $\delta$  represent the first, second, and third best solutions, whereas all of the other inferior solutions are designated  $\omega$ . During the search process, each individual is guided by  $\alpha$ ,  $\beta$ , and  $\delta$  in the process of updating their position, as shown in Fig. 1.

During the hunt, the prey is surrounded by the wolves, the process of which can be modeled mathematically as follows:

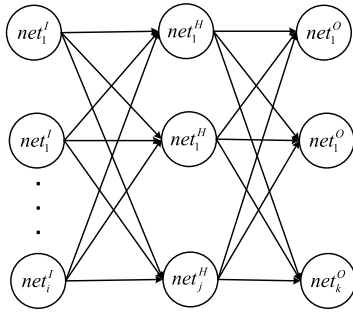
$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right| \quad (7)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (8)$$

where  $\vec{D}$  is the distance between the individual and the prey,  $t$  is the iteration index,  $\vec{X}_p$  is the position of the prey,  $\vec{X}$  is the position of the individual, and  $\vec{A}$  and  $\vec{C}$  are coefficients respectively defined as follows [24]:

$$\vec{A} = 2 \vec{a} \cdot \vec{r}_1 - \vec{a} \quad (9)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (10)$$



**FIGURE 2.** Schematic diagram showing architecture of typical back-propagation neural network.

where  $\vec{r}_1$  and  $\vec{r}_2$  are random number between [0, 1] and  $\vec{a}$  decreases linearly from 2 to 0 through each iteration. The position of an individual is updated based on the positions of  $\alpha$ ,  $\beta$ , and  $\delta$  as follows:

$$\begin{cases} \vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right| \\ \vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right| \\ \vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \end{cases} \quad (11)$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \\ \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \end{cases} \quad (12)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (13)$$

After calculation of (11),  $\vec{D}_\alpha$ ,  $\vec{D}_\beta$  and  $\vec{D}_\delta$  can be obtained, and then substituted into (12), then  $\vec{X}_1$ ,  $\vec{X}_2$  and  $\vec{X}_3$  can be obtained, and then  $\vec{X}_1$ ,  $\vec{X}_2$  and  $\vec{X}_3$  are substituted into (14),  $\vec{X}(t+1)$  can be obtained.  $\vec{X}(t+1)$  is the position to which the individual will move during the  $(t+1)$ th iteration.

### E. ARTIFICIAL NEURAL NETWORK

Artificial neural networks (ANNs) are one approach to artificial intelligence (AI) [25], [26] which simulate the behavior of biological brains based on a system of interconnected neurons used to compute values from inputs. They represent a form of machine learning, which have proven particularly effective in pattern recognition. Back-propagation neural networks employ the fine-tuning of weights within a neural net based on the error rate obtained in the previous epoch [27] (i.e., iteration). The weights are tuned with the aim of reducing error rates in order to make the model more reliable. The ultimate objective is to identify the weights that would allow the neural network to map arbitrary inputs to outputs. The first step involves determining the number neurons in the input layer, hidden layer, and output layer. Assume that the hidden layer is a single layer with  $j$  neurons, the neurons in the output layer indicate the training position, and the number of neurons in the output layer is indicated by  $k$ . Figure 2 presents the architecture of a typical back-propagation neural network.

The Back Propagation Neural Network first calculates the output of each hidden neuron, as follows:

$$net_j^H = f \left( \sum_{i=1}^{I_{max}} w_{ij} net_i^I \right) \quad (14)$$

where the function of  $f$  is the hyperbolic activation function defined as

$$f = \frac{1}{1 + e^{-net}} \quad (15)$$

where  $net_i^I$  is the  $i$ -th neuron of the input layer and the data of the neuron in the input layer is the actual input of the network as well as the output of the neuron;  $net_j^H$  refers to the  $j$ -th neuron of the hidden layer;  $w_{ij}$  refers to the weighting between the  $i$ -th neuron of the input layer and the  $j$ -th neuron of the hidden layer;  $I_{max}$  is the total number of neurons in the input layer; and  $H_{max}$  is the total number of neurons in the hidden layer. The output of each neuron in the output layer is then calculated as follows:

$$net_k^O = f \left( \sum_{i=1}^{H_{max}} w_{jk} net_i^H \right) \quad (16)$$

where  $net_k^O$  is the  $k$ -th neuron of the output layer, and  $w_{jk}$  is the weighting between  $j$ -th neuron of the hidden layer and  $k$ -th neuron of the output layer. This output is the actual output of the network. The BPNN adjusts the weighting of the network according to the error between the expected output and the actual output of the network.

### F. STANDARDIZATION

Before training the neural network, the input data undergoes preprocessing to obtain an average value of roughly 0 and reduce the standard deviation in order to reduce the likelihood of weight divergence during training [28]. Preprocessing also reduces the likelihood of producing infinity or negative infinity cases. The preprocessing method used in this paper is standardization, the mathematical formula of which is as follows:

$$x' = \frac{(x - \mu)}{\sigma} \quad (17)$$

where  $\mu$  is the average of the original data set,  $\sigma$  is the standard deviation of the original data set,  $x$  is the original data, and  $x'$  is the result after standardization [28], [29].

## III. METHODOLOGY

### A. CHANNEL MODEL

In this study, the lognormal shadowing model was used to simulate the channel model in the experimental environment. The signal strength measured by the receiver at distance  $d_0$  was obtained from real-world measurements. The transmitter was a D-Link DIR-600M Wireless N150, and the RSSI receiver was the D1 mini Wi-Fi module. The parameter settings are detailed in Table 1.

Our positioning system was implemented in two experimental environments: and an open environment (Scenario A)

TABLE 1. Parameters of channel model.

Parameter	Value
Path loss exponent $n$	1.8
Close-in distance $d_0$	0.01 m
Signal strength $A$ at $d_0$	-20 dBm
Standard deviation $\sigma$	3, 9 dB

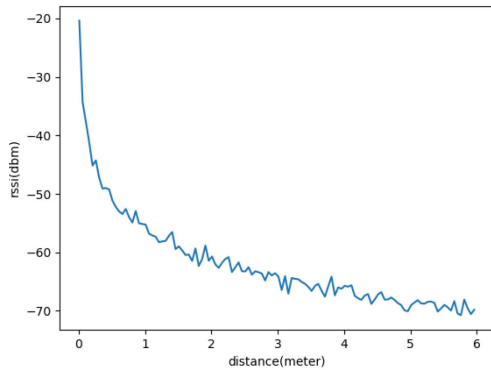


FIGURE 3. Channel model for scenario A ( $X_\sigma = 3$  dB).

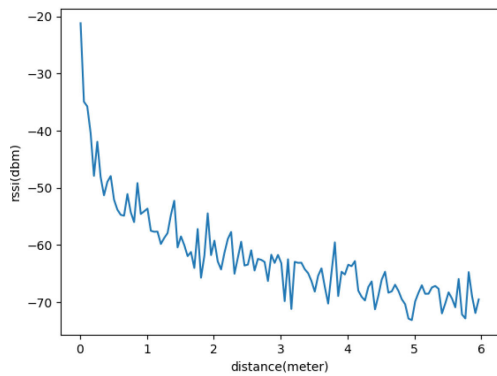


FIGURE 4. Channel model for scenario B ( $X_\sigma = 9$  dB).

a complex environment (Scenario B). The open environment is on the top floor of the building, so it is a light of sight (LOS) environment. Scenario A can be attributed to less interference, since the obstacles are less. The complex environment is inside the classroom. The main interference is come from the more obstacles in the classroom. We defined various indoor environments by adjusting the standard deviation of the parameters. A smaller standard deviation (representative of an indoor environment without obstacles) produces a stable model, whereas a larger standard deviation (representative of a complex indoor environment) produces an unstable model.

In the experiment, the environments were simulated as follows: Scenario A (Gaussian random variables with standard deviation of 3dB) and Scenario B (Gaussian random variables with standard deviation of 9dB) These are shown in Table 2 and Figs. 3-4.

From the Figs 3-4, the curve in the channel model indicates that an increase in distance reduces the slope of the

TABLE 2. Experiment scenarios.

Scenario	Standard deviation
A	3 dB
B	9 dB

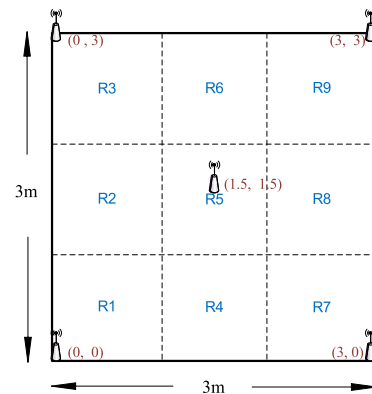


FIGURE 5. Experiment map.

curve, particularly at distances exceeding 3 m. This leads to a drop in resolution, which reduces positioning accuracy. Thus, we sought to keep the range of the indoor channel model to within 3 m.

### B. EXPERIMENT ENVIRONMENT

According to the channel model in the previous sub-section III.A, the anchor nodes in the experimental map were placed in each of the four corners as well as in the center to reduce uncertainty in regional positioning [30]. The map used in the experiments is shown in Fig. 5.

The experiment was conducted indoors in a square area (length = 3m and width = 3m). In the experiment on regional positioning, the overall map was divided into nine regions (length = 1m and width = 1m).

All training points were generated randomly. Both in the regional and coordinate positioning experiments, we used 1800 data points for training and 900 data points for testing. In regional positioning, the output was the area in which the target was located; in coordinate positioning, the outputs were the x coordinate and y coordinate pertaining to the target.

The sharing of Wi-Fi radio channels by different systems or devices using the same frequency band (e.g., Bluetooth devices and microwave ovens) often results in co-channel interference when multiple devices are operating in the same area. RSSI can be profoundly affected by this interference as well as fading and shadowing. We sought to reduce environmental shadowing effects by performing arithmetic mean filtering during the collection of RSSI values. This involved sampling RSSI values 100 times in the same location and then averaging the results to reduce the impact of signal variation on the positioning algorithm.

### C. EXHAUSTIVE LOCALIZATION METHOD

We adopted the exhaustive method to determine the best learning rate and optimal number of neurons in the hidden layer, with the aim of maximizing accuracy in regional positioning and minimizing the average distance error, which are respectively defined using Eqs. (18) and (19). Neurons in the hidden layer had a search range of 1 to 100 with a search interval of 1. The learning rate was given a search range of 0.01 to 1 with a search interval of 0.01.

In the regional positioning experiment, positioning accuracy was used to evaluate the model. In the accurate positioning experiment, the average distance error between the output coordinates and the target coordinates was used to evaluate the model, as defined in the following two equations:

$$Accuracy = \frac{Correct\_num}{Total\_num} \cdot 100\% \quad (18)$$

where *Accuracy* refers to positioning accuracy, *Correct\_num* indicates the number of correct positions in the test sample, and *Total\_num* is the total number of test samples in the test.

$$d = \frac{1}{n} \sum_{i=1}^n \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2} \quad (19)$$

where *d* is the average distance error, *n* is the total number of test samples, *i* is the sample index, *x* is the output of the *x* coordinate,  $\hat{x}$  is the target output for the *x* coordinate, *y* is the output of the *y* coordinate, and  $\hat{y}$  is the target output for the *y* coordinate.

### D. PROPOSED HYBRID ALGORITHM

In this paper, we formulated a hybrid algorithm to facilitate the search for optimal parameters for the neural network (number of neurons in hidden layer and learning rate). The evaluation of parameters requires a clearly-defined fitness function. For regional positioning, we used the error rate as a fitness function, the mathematical definition of which is presented in Eq. (20). For accurate positioning, we used the average distance error between the calculated and target coordinates as a fitness value, the mathematical definition of which is presented in Eq. (19).

$$Errorrate = \frac{Error\_num}{Total\_num} \cdot 100\% \quad (20)$$

where *Errorrate* refers to the positioning error rate, *Error\_num* indicates the number of times the localization region is wrong during a given test, and *Total\_num* is the total number of test samples in the overall test process.

#### Hybrid algorithm:

1. Particle Swarm Optimization in conjunction with ANN (PSO-ANN) [12], [13]

In the proposed scheme, the PSO algorithm is used mainly to determine the best learning rate and optimal number of neurons in the hidden layer of the neural network.

The flow of the proposed PSO-ANN algorithm can roughly be divided as follows:

1. Initialize all individual locations.
2. Train neural network.

3. Enter test data and calculate individual fitness values.
  4. Determine the best solution for each individual and overall.
  5. Determine whether the maximum number of iterations has been reached. If yes, then return the best model and the corresponding test results and end the algorithm; otherwise, go to Step 6.
  6. Update individual locations.
  7. Train neural network again.
  8. Enter test data and calculate individual fitness values.
  9. Update the best solution for each individual and overall.
  10. Add an iterative index.
  11. Go back to Step 5.
2. Grey Wolf Optimization in conjunction with ANN (GWO-ANN) [12], [13]

The GWO algorithm is used mainly to identify the best combination of neural network learning rate and number of neurons in the hidden layer.

The flow of the proposed GWO-ANN algorithm can roughly be divided as follows:

1. Initialize individual locations.
2. Train neural network.
3. Enter test data and calculate individual fitness values.
4. Find  $\alpha$ ,  $\beta$ , and  $\delta$ .
5. Determine whether the maximum number of iterations has been reached. If yes, then return the best model and corresponding test results and end the algorithm; otherwise, go to Step 6.
6. Update individual locations based on formula.
7. Train neural network.
8. Enter test data and calculate individual fitness values.
9. Update  $\alpha$ ,  $\beta$ , and  $\delta$ .
10. Add an iterative index.
11. Go back to Step 5.

## IV. EXPERIMENT RESULTS

The process of the main experiment is divided into the following steps:

1. As described in Section III, we first obtained the RSSI values within a square interior space 3m × 3m. Training data (1800 data points) and test data (900 data points) were generated randomly.
2. We saved the training and test data.
3. We executed the training process using the exhaustive method, GWO-ANN, and PSO-ANN for comparison.

### A. RESULTS FOR REGIONAL POSITIONING

In regional positioning, there were 5 neurons in the input layer of the neural network; there were 9 neurons in the output layer, representing 9 regions; and RSSI values were received from 5 anchor nodes. The activation function of the hidden layer used ReLU, and the activation function of the output layer used Softmax. In regional positioning, we implemented the exhaustive method, PSO-ANN, and GWO-ANN. Table 3 and Table 4 respectively shows the details of the

**TABLE 3.** Parameters of PSO-ANN for regional positioning.

Parameter name	Value
The number of individual	5, 10
The number of iteration	50
The maximum of inertia weight	0.9
The minimum of inertia weight	0.4
Individual experience	2
Global experience	2
The maximum speed of individuals ( $V_{max}$ )	10
The number of neurons in the input layer	5
The number of neurons in the output layer	9
Activation function for the hidden layer	ReLU
Activation function for the output layer	Softmax
The search ranges of the learning rate	0.01~1
Search interval for learning rate	0.01
Range of the number of hidden neurons	1~100
batch size	500
The number of iterations for ANN	500

**TABLE 4.** Parameters of GWO-ANN for regional positioning.

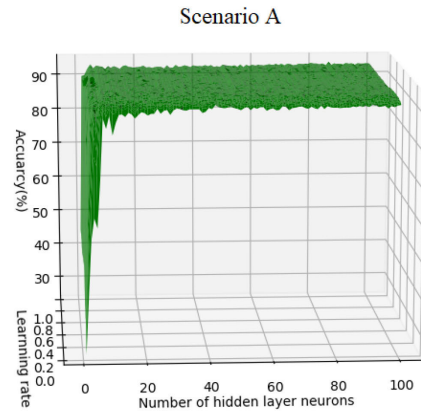
Parameter name	Value
The number of individual	5, 10
The number of iteration	50
The number of neurons in the input layer	5
The number of neurons in the output layer	9
Activation function for the hidden layer	ReLU
Activation function for the output layer	Softmax
The search ranges of the learning rate	0.01~1
Search interval for learning rate	0.01
Range of the number of hidden neurons	1~100
batch size	500
The number of iterations for ANN	500

parameter settings of PSO-ANN and GWO-ANN for regional positioning.

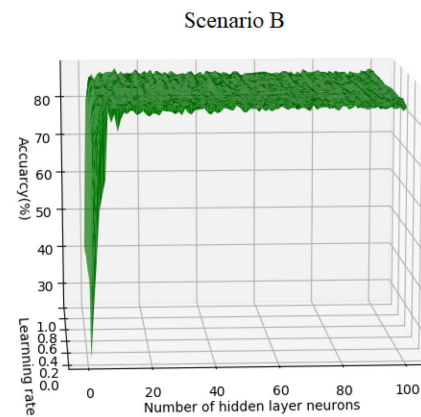
The exhaustive method was implemented using sample data collected in Scenario A and B. The objective was to identify the optimal combination of learning rate and the number of neurons in the hidden layer. The learning rate was given a search range of 0.01 to 1. The search interval for the learning rate was 0.01, whereas the search interval for the number of neurons was 1. Following the experiment, the learning rate, the number of hidden layer neurons, and their relationship with positioning accuracy were assembled as a three-dimensional graph, for use in identifying the optimal combination of parameters, as shown in Fig. 6 (Scenario A) and Fig. 7 (Scenario B).

Under Scenario A, the exhaustive method achieved maximum positioning accuracy of 94.22% within a search time of 40,764 seconds. Under Scenario B, maximum positioning accuracy dropped to 88.11% within a search time of 40,845 seconds.

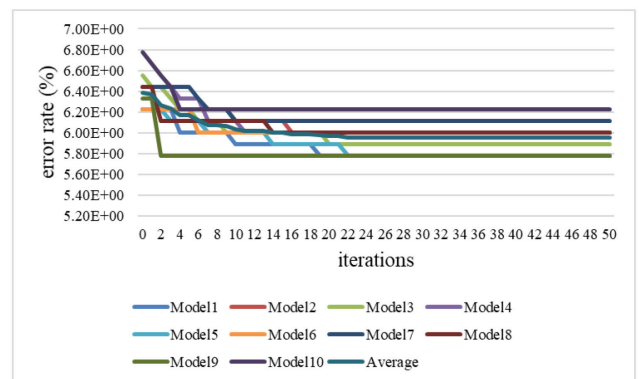
The experiments using the PSO-ANN algorithm included 5 or 10 individuals. The number of iterations of the batch-size and neural network were set to 500, and the number of iterations of the hybrid algorithm was set to 50. In setting the parameters for PSO, we referred to the suggestions in [20], where  $c_1$  and  $c_2$  were respectively set at 2, the maximum



**FIGURE 6.** 3D graph showing ANN parameters (learning rate and number of hidden neurons) and regional positioning accuracy obtained using exhaustive method under Scenario A.



**FIGURE 7.** 3D graph showing ANN parameters (learning rate and number of hidden neurons) and regional positioning accuracy obtained using the exhaustive method under Scenario B.



**FIGURE 8.** PSO-ANN iteration curve indicating accuracy in regional positioning using 5 individuals under Scenario A.

inertia weight was set at 0.9, and the minimum inertia weight was set at 0.4. Throughout this iterative procedure, the inertia weight was reduced linearly from the maximum value to the minimum.

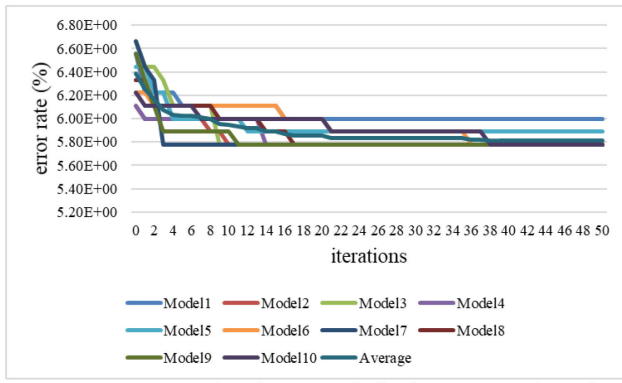


FIGURE 9. PSO-ANN iteration curve indicating accuracy in regional positioning using 10 individuals under Scenario A.

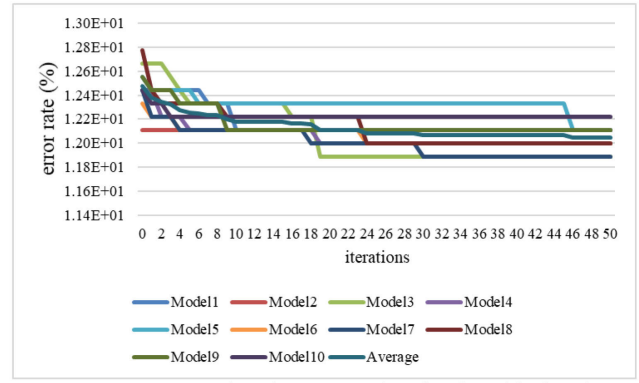


FIGURE 11. PSO-ANN iteration curve of regional positioning for 10 individuals under Scenario B.

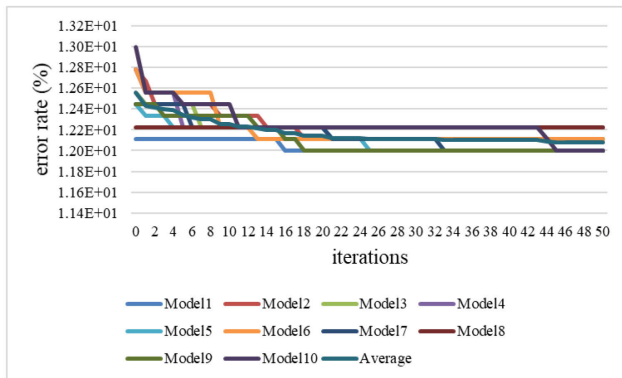


FIGURE 10. PSO-ANN iteration curve indicating accuracy in regional positioning using 5 individuals under Scenario B.

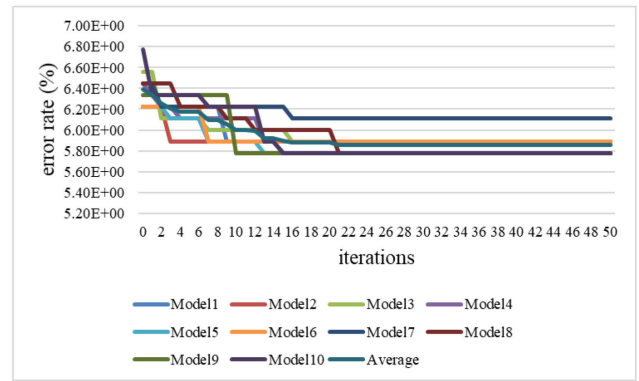


FIGURE 12. GWO-ANN iteration curve indicating accuracy in regional positioning using 5 individuals under Scenario A.

TABLE 5. Experiment results under Scenario A.

Method	Accuracy (%)	Time (sec)
Exhaustive	94.22	40764.928
PSO-ANN(5)	94.04	1064.425
PSO-ANN(10)	94.19	2062.167
GWO-ANN(5)	94.14	1032.209
GWO-ANN(10)	94.18	2095.551

The PSO-ANN algorithm was run ten times (Models 1-10) and the average error rate values were recorded. Figures 8 and 9 respectively present the results for 5 and 10 individuals in Scenario A. Figures 10 and 11 respectively present the results for 5 and 10 individuals in Scenario B. We observed a decrease in the error rate with an increase in the number of iterations, thereby demonstrating the effectiveness of hybrid algorithm. The superior performance achieved under Scenario A can be attributed to less interference.

Similarly, the experiments using the GWO-ANN algorithm were also conducted using 5 or 10 individuals. Figures 12 and 13 respectively present the results for 5 and 10 individuals in Scenario A. Figures 14 and 15 respectively present the results for 5 and 10 individuals in Scenario B.

From the results summary in Table 5, we can see that under Scenario A, positioning accuracy of the hybrid algorithms

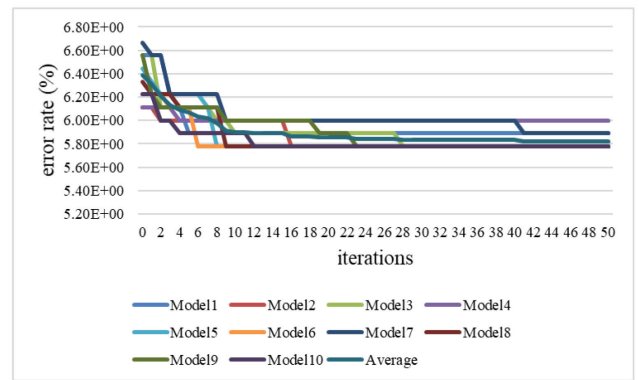


FIGURE 13. GWO-ANN iteration curve indicating accuracy in regional positioning using 10 individuals under Scenario A.

and the baseline differed by less than 1%. The average search duration using the PSO-ANN was 97.39% less than the exhaustive method when using 5 individuals and 94.94% less when using 10 individuals. The average search duration using the GWO-ANN was 97.47% less than the exhaustive method when using 5 individuals and 94.86% less when using 10 individuals.

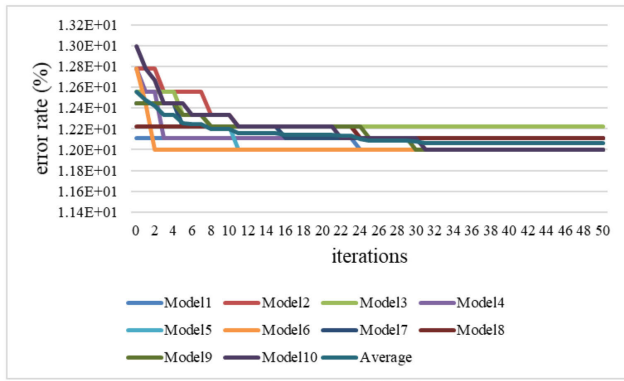


FIGURE 14. GWO-ANN iteration curve indicating accuracy in regional positioning using 5 individuals under Scenario B.

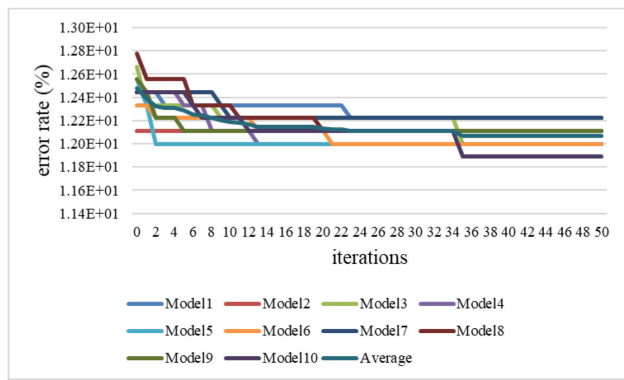


FIGURE 15. GWO-ANN iteration curve indicating accuracy in regional positioning using 10 individuals under Scenario B.

TABLE 6. Experiment results under Scenario B.

Method	Accuracy (%)	Time (sec)
Exhaustive	88.11	40845.242
PSO-ANN(5)	87.92	1029.232
PSO-ANN(10)	87.96	1995.961
GWO-ANN(5)	87.93	1008.902
GWO-ANN(10)	87.93	2054.526

As shown in Table 6, under Scenario B the positioning accuracy of the hybrid algorithms and the baseline differed by less than 1%. The average search duration using the PSO-ANN was 97.48% less than the exhaustive method when using 5 individuals and 95.11% less when using 10 individuals. The average search duration using the GWO-ANN was 97.53 less than the exhaustive method when using 5 individuals and 94.97% less when using 10 individuals.

**B. RESULTS OF POSITIONING ACCURACY**

In estimating coordinate positioning accuracy, we used 5 neurons in the input layer of the neural network with the RSSI values of 5 anchor nodes as an input, with 2 neurons in the output layer. The x coordinate was used as an output, and the y coordinate used a hidden layer. The Sigmoid function was the activation function in the hidden layer, whereas the

TABLE 7. Parameters of PSO-ANN for coordinate positioning.

Parameter name	Value
The number of individual	5
The number of iteration	50
The maximum of inertia weight	0.9
The minimum of inertia weight	0.4
Individual experience	2
Global experience	2
The maximum speed of individuals ( $V_{max}$ )	10
The number of neurons in the input layer	5
The number of neurons in the output layer	2
Activation function for the hidden layer	Sigmoid
Activation function for the output layer	Identity
The search ranges of the learning rate	0.01~1
Search interval for learning rate	0.01
Range of the number of hidden neurons	1~100
batch size	500
The number of iterations for ANN	500

TABLE 8. Parameters of PSO-ANN for coordinate positioning.

Parameter name	Value
The number of individual	5
The number of iteration	50
The number of neurons in the input layer	5
The number of neurons in the output layer	2
Activation function for the hidden layer	Sigmoid
Activation function for the output layer	Identity
The search ranges of the learning rate	0.01~1
Search interval for learning rate	0.01
Range of the number of hidden neurons	1~100
batch size	500
The number of iterations for ANN	500

TABLE 9. Experiment results under Scenario A.

Method	Average error distance (m)	Time (sec)
Exhaustive	0.1027	33024.031
PSO-ANN(5)	0.1103	818.889
PSO-ANN(10)	0.104	1623.268
GWO-ANN(5)	0.103	809.248
GWO-ANN(10)	0.1027	1623.24

identity function was used in the output layer. Analysis was performed on the exhaustive method, PSO-ANN, and GWO-ANN. The number of iterations of the batch-size and neural network were set to 500, and the number of iterations of the hybrid algorithm was set to 50. In setting the parameters for PSO, we referred to the suggestions in [20], where  $c_1$  and  $c_2$  were respectively set at 2, the maximum inertia weight was set at 0.9, and the minimum inertia weight was set at 0.4. Throughout this iterative procedure, the inertia weight was reduced linearly from the maximum value to the minimum. Table 7 and Table 8 respectively shows the details of the parameter settings of PSO-ANN and GWO-ANN for coordinate positioning.

The summarized results in Table 9 reveal that under Scenario A, the average positioning error of the hybrid algorithms and the baseline differs by less than 0.01 meters. The



TABLE 10. Experiment results under Scenario B.

Method	Average error distance (m)	Time (sec)
Exhaustive	0.2101	33088.663
PSO-ANN(5)	0.2133	812.36
PSO-ANN(10)	0.212	1631.412
GWO-ANN(5)	0.2113	812.942
GWO-ANN(10)	0.2104	1623.305

average search duration using the PSO-ANN was 97.52% less than the exhaustive method when using 5 individuals and 95.08% less when using 10 individuals. The average search duration using the GWO-ANN was 97.55% less than the exhaustive method when using 5 individuals and 95.08% less when using 10 individuals.

As shown in Table 10, under Scenario B the average positioning error of the hybrid algorithms and the baseline differs by less than 0.01 meters. The average search duration using the PSO-ANN was 97.54% less than the exhaustive method when using 5 individuals and 95.07% less when using 10 individuals. The average search duration using the GWO-ANN was 97.54% less than the exhaustive method when using 5 individuals and 95.09% less when using 10 individuals.

The results in Tables 5, 6, 9, 10 show that the exhaustive method achieved the highest accuracy; however, the time cost was far too high. Furthermore, the accuracy of the exhaustive method was less than 1% higher than the average accuracy obtained using the hybrid algorithm. The performance of the hybrid algorithm with 10 individuals was higher than with 5 individuals; however, the average search time was nearly double, and the performance improvement was less than 1%. In regional positioning, the best overall trade-off between accuracy and time cost would be the hybrid GWO-ANN algorithm with 5 individuals, which provides higher average accuracy than PSO-ANN with lower time cost.

## V. CONCLUSION

In this paper, we augmented a neural network algorithm with two swarm-intelligence algorithms (particle swarm optimization and the gray wolf algorithm) to identify the optimal combination of learning rate and number of hidden neurons. We then assessed their performance in localization accuracy using the exhaustive method as a baseline. Overall, the magnitude of the errors decreased with an increase in the number of iterations. Under identical conditions, the hybrid algorithms achieved the same results as the exhaustive method (difference of less than 1%/0.01 meters); however, the proposed scheme reduced the average search duration by roughly 95%. In future research, we will seek to determine whether the performance of the hybrid algorithm could be improved through the adoption of additional parameters.

## REFERENCES

- [1] C. Gritti, M. Önen, R. Molva, W. Susilo, and T. Plantard, "Device identification and personal data attestation in networks," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 9, no. 4, pp. 1–25, Dec. 2018.
- [2] I. V. Kottenko, I. Saenko, and A. Branitskiy, "Applying big data processing and machine learning methods for mobile Internet of Things security monitoring," *J. Internet Services Inf. Secur.*, vol. 8, no. 3, pp. 54–63, Aug. 2018.
- [3] M. Park, J. Seo, J. Han, H. Oh, and K. Lee, "Situational awareness framework for threat intelligence measurement of Android malware," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl. (JoWUA)*, vol. 9, no. 3, pp. 25–38, 2018.
- [4] U. Noriki, S. Goshi, and S. Yoshitaka, "Device-to-device communication based DTN for disaster information system by using emergent user policy and locational information," *J. Internet Services Inf. Secur.*, vol. 9, no. 3, pp. 41–51, Aug. 2019.
- [5] Y. Zhuang, L. Hua, L. Qi, J. Yang, P. Cao, Y. Cao, Y. Wu, J. Thompson, and H. Haas, "A survey of positioning systems using visible LED lights," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1963–1988, 3rd Quart., 2018.
- [6] S. Sadowski and P. Spachos, "RSSI-based indoor localization with the Internet of Things," *IEEE Access*, vol. 6, pp. 30149–30161, 2018.
- [7] B. Molina, E. Olivares, C. E. Palau, and M. Esteve, "A multimodal fingerprint-based indoor positioning system for airports," *IEEE Access*, vol. 6, pp. 10092–10106, 2018.
- [8] N. Zeng, Z. Wang, B. Zineddin, Y. Li, M. Du, L. Xiao, X. Liu, and T. Young, "Image-based quantitative analysis of gold immunochromatographic strip via cellular neural network approach," *IEEE Trans. Med. Imag.*, vol. 33, no. 5, pp. 1129–1136, May 2014.
- [9] N. Zeng, Z. Wang, H. Zhang, K. Kim, Y. Li, and X. Liu, "An improved particle filter with a novel hybrid proposal distribution for quantitative analysis of gold immunochromatographic strips," *IEEE Trans. Nanotechnol.*, vol. 18, pp. 819–829, Aug. 2019.
- [10] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, and Y. Li, "A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease," *Neurocomputing*, vol. 320, pp. 195–202, Dec. 2018.
- [11] P. G. Asteris and M. Nikoo, "Artificial bee colony-based neural network for the prediction of the fundamental period of infilled frame structures," *Neural Comput. Appl.*, vol. 31, no. 9, pp. 4837–4847, Jan. 2019.
- [12] M. K. Ramawan, Z. Othman, S. I. Sulaiman, I. Musirin, and N. Othman, "A hybrid bat algorithm artificial neural network for grid-connected photovoltaic system output prediction," in *Proc. IEEE 8th Int. Power Eng. Optim. Conf. (PEOCO)*, Langkawi, Malaysia, Mar. 2014, pp. 619–623.
- [13] S. K. Gharghan, R. Nordin, M. Ismail, and J. A. Ali, "Accurate wireless sensor localization technique based on hybrid PSO-ANN algorithm for indoor and outdoor track cycling," *IEEE Sensors J.*, vol. 16, no. 2, pp. 529–541, Jan. 2016.
- [14] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [15] J. J. Wang, J. G. Hwang, and J. G. Park, "A novel indoor ranging method using weighted autofrequent RSSI measurements," in *Proc. 11th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Surfers Paradise, QLD, Australia, Dec. 2017, pp. 1–5.
- [16] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.
- [17] A. Khalajmehrabadi, N. Gatsis, and D. Akopian, "Modern WLAN fingerprinting indoor positioning methods and deployment challenges," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1974–2002, 3rd Quart., 2017.
- [18] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, M. Karamanoglu, *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications*. Amsterdam, The Netherlands: Elsevier, 2013.
- [19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, vol. 4, 1995, pp. 1942–1948.
- [20] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.
- [21] T. Tsujimoto, T. Shindo, and K. Jinano, "The neighborhood of canonical deterministic PSO," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, Jun. 2011, pp. 1811–1817.
- [22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

- [23] K. R. Das, D. Das, and J. Das, "Optimal tuning of PID controller using GWO algorithm for speed control in DC motor," in *Proc. Int. Conf. Soft Comput. Techn. Implementations (ICSCTI)*, Faridabad, India, Oct. 2015, pp. 108–112.
- [24] M. F. Hassanin, A. M. Shoeb, and A. E. Hassanien, "Grey wolf optimizer-based back-propagation neural network algorithm," in *Proc. 12th Int. Comput. Eng. Conf. (ICENCO)*, Cairo, Egypt, Dec. 2016, pp. 213–218.
- [25] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.
- [26] F. A. Ruslan, A. M. Samad, Z. M. Zain, and R. Adnan, "Flood water level modeling and prediction using NARX neural network: Case study at Kelang river," in *Proc. IEEE 10th Int. Colloq. Signal Process. Appl.*, Mar. 2014, pp. 204–207.
- [27] F. Du and W. Du, "Novel smith predictor and BPNN for wireless networked control systems," in *Proc. Asia-Pacific Conf. Inf. Process.*, Shenzhen, China, Jul. 2009, pp. 63–66.
- [28] S. I. Koval, "Data preparation for neural network data analysis," in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (EIConRus)*, Moscow, Russia, Jan. 2018, pp. 898–901.
- [29] K. Jaafar, N. Ismail, M. Tajjudin, R. Adnan, and M. H. F. Rahiman, "Z-score and feedforward neural network (FFNN) for flood modelling at Kuala Krai station," in *Proc. 7th IEEE Control Syst. Graduate Res. Colloq. (ICSGRC)*, Shah Alam, Malaysia, Aug. 2016, pp. 92–97.
- [30] C.-H. Cheng and Y. Yan, "Indoor positioning system for wireless sensor networks based on two-stage fuzzy inference," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 5, May 2018, Art. no. 155014771878064.



**TAO-PING WANG** received the B.S. and M.S. degrees in electrical engineering from National Formosa University, Huwei, Taiwan, in 2017 and 2019, respectively, where he is currently pursuing the M.S. degree in electrical engineering. His main research interests include wireless sensor networks and embedded systems.



**YUNG-FA HUANG** (Member, IEEE) received the Diploma degree in electrical engineering from the National Taipei University of Technology, Taipei, in 1982, the M.Eng. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1987, and the Ph.D. degree in electrical engineering from National Chung Cheng University, Chiayi, Taiwan, in 2002. From 1982 to 1984, he was with the Air Forces for the Military Service, Taiwan. From 1987 to 2002, he was an Instructor

with the Chung Chou Institute of Technology, Yuanlin, Taiwan, where he was an Associate Professor with the Department of Electrical Engineering, from February 2002 to July 2004. From August 2004 to July 2007, he was an Associate Professor with the Graduate Institute of Networking and Communication Engineering, Chaoyang University of Technology, Taichung, Taiwan. From August 2007 to July 2008, he was the Head of the Department of Computer and Communication Engineering and the Chair of the Graduate Institute of Networking and Communication Engineering, Chaoyang University of Technology. From August 2008 to July 2010, he was the Head of the Department of Information and Communication Engineering, Chaoyang University of Technology, where he has been a Professor, since September 2011. His current research interests include multiuser detection in OFDM-CDMA cellular mobile communication systems, communication signal processing, fuzzy systems, and wireless sensor networks. He has served as the Co-Chair for the IEEE SMC Society Technical Committee on Intelligent Internet Systems.



**CHIA-HSIN CHENG** (Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from National Chung Cheng University, Chiayi, Taiwan, in 1997 and 2007, respectively. From August 2011 to January 2019, he was an Associate Professor with the Department of Electrical Engineering, National Formosa University, Huwei, Taiwan, where he is currently a Professor and the Chair. His research interests are in the areas of the Internet of Things (IoT), communication

networks, interference cancellation, spread-spectrum systems, wireless sensor networks, and embedded systems.

• • •