

Research Article

Evaluation of a High-Accuracy Indoor-Positioning System with Wi-Fi Time of Flight (ToF) and Deep Learning

Doan Perdana^(b),¹ I Made Arya Indra Tanaya,² Abdul Aziz Marwan^(b),³ and Fityanul Akhyar^(b)⁴

¹Advanced Creative Networks Research Center in Telkom University, Kabupaten Bandung, Indonesia ²PT. Bale Teknologi Bali, Denpasar, Indonesia ³Det setuent of Electrical Engineering, Telkow University, Kalupaten Bandung, Indonesia

³Department of Electrical Engineering, Telkom University, Kabupaten Bandung, Indonesia ⁴Intelligence System Laboratory, Telkom University, Kabupaten Bandung, Indonesia

Correspondence should be addressed to Doan Perdana; doanperdana@telkomuniversity.ac.id

Received 16 August 2022; Revised 27 October 2022; Accepted 13 December 2022; Published 4 April 2023

Academic Editor: Roberto Nardone

Copyright © 2023 Doan Perdana et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article proposes the use of Wi-Fi ToF and a deep learning approach to build a cheap, practical, and highly-accurate IPS. To complement that, rather than using the classic geometrical approach (such as multilateration), it uses a more data-driven approach, i.e., the location fingerprinting technique. The fingerprint of a location, in this case, is a set of Wi-Fi ToFs between the target device and an access point (AP). Therefore, the number of APs in the area dictates the set size. The location fingerprinting technique requires a collection of fingerprints of various locations in the area to build a reference database or map. This database or map contains the information used to carry out the main task of the location fingerprinting technique, namely, estimating the position of a device based on its location fingerprint. For that task, we propose using a fully connected deep neural network (FCDNN) model to act as a positioning engine. The model is given a location fingerprint as its input to produce the estimated location coordinates as its output. We conduct an experiment to analyze the impact of the available AP pair in the dataset, from 1 unique AP pair, 2 AP pairs, and more, using WKNN and FCDNN to compare their performance. Our experimental results show that our IPS, DeepIndoor, can achieve an average positioning error or mean square error of 0.1749 m, and root mean square error of 0.5740 m in scenario 3, where 1–10 AP pairs or the raw dataset is used.

1. Introduction

The goal of 5G and IoT applications is to improve people's daily lives by transforming a variety of things from conventional to intelligent [1, 2]. Through efficient packet radio access and adjustable bandwidth, it will offer better data speeds and reduced latency [3]. Among them, the deep learning theory is considered one of the most promising techniques to tackle tremendous highdimensional data [4]. Many of those applications require location-related information to deliver their services. The majority of location-based services (LBSs) for outdoor environments are possible due to GNSSs and Global Navigation Satellite System [5–7] and Global Positioning System (GPS) [8]. The state of the art of outdoor positioning technologies nowadays can be considered mature [9] and sufficient in terms of fulfilling the related service requirements such as QoS [10] and user mobility [11] that depends on the network architecture used [12]. Unfortunately, that is not the case for indoor scenarios. Localizing an object or route indoors using GPS is usually not feasible due to the loss of the signal emitted by its satellites [13]. The complexity of indoor environments with walls and various objects contributes to this phenomenon. Regarding the numerous potential applications that can be enabled, e.g., indoor wayfinding, asset tracking, and crowd monitoring; it is unfortunate that there is no standardized solution for indoor positioning systems (IPSs) yet [5].

However, the topic of indoor positioning solutions has gained substantial interest among industries and academia [14–17]. The current landscape of IPS underlying connectivity technologies mainly consists of Bluetooth, Wi-Fi, Zigbee, RFID (radio frequency identification), and UWB (ultra-wideband). Each of these technologies comes with its characteristics in enabling an IPS. The characteristics of each mentioned technology in enabling an IPS are listed in Table 1.

In the case of where the users are people, Bluetooth and Wi-Fi are typically the preferred option over the other technologies as both are available in most smartphones nowadays. Regarding the deployment cost, Wi-Fi can be preferred over Bluetooth as the deployment of Wi-Fi access points (APs) in indoor facilities are more common than Bluetooth beacons. It makes deploying a Wi-Fi-based IPS cheaper than a Bluetooth-based IPS since there is no need to implement new infrastructure in the area. On the UWB side, it leads in terms of accuracy (see Table 1).

However, the availability of UWB in smartphones is not very common yet besides its high deployment cost [22]. This makes a UWB-based IPS not as practical as either a Wi-Fibased IPS or a Bluetooth-based IPS. Although the building blocks of making an IPS have been available, realizing a cheap, practical, and highly-accurate IPS remains a challenge [23–26]. It proposes DeepIndoor, a Wi-Fi-based IPS utilizing the time of flight of Wi-Fi signals and a deep learning approach. DeepIndoor leverages the advantages of a Wi-Fi-based IPS in terms of its practicality and the low deployment cost and combines them with a deep learning approach to improve its accuracy. It uses a data-driven approach for the location inference technique to work with deep learning, i.e., location fingerprinting.

The location fingerprinting technique can be considered more robust than the classic geometrical approaches (such as multilateration). It does not rely on line-of-sight (LOS) communication to make a good estimation. In location fingerprinting, a location is estimated based on its fingerprint (or set of features), which in this case is a set of Wi-Fi time of flights (ToFs). For that task, it proposes using a fully connected deep neural network (FCDNN) model to act as a positioning engine. The model is given a location fingerprint as its input to produce the estimated location coordinates as its output. The successful applications of deep learning in various domains [27] and as computing resources become cheaper and more available, it encourage us to apply it in this domain. By doing this research, our major contributions can be seen as follows:

- (i) We design a cheap, practical, and highly-accurate IPS using Wi-Fi ToF and a deep learning approach.
- (ii) We conduct extensive experiments to evaluate the condition of available AP pair scenarios and optimize the performance of the WKNN algorithm and our positioning engine or DeepIndoor on a publicly available dataset which can be accessed in [28].
- (iii) We detail the structure and configuration of our positioning engine to encourage its applications in other testbeds or perhaps to work with other features than Wi-Fi ToF for future developments.

The rest of this article is divided into several sections: Section 2 presents some previous related works, Section 3 details our system model, Section 4 covers our experiment settings, Section 5 shows our results and findings, and Section 6 provides the conclusions of this research.

2. Related Works

Indoor positioning system or IPS consists of radio frequency-based system and nonradio-frequency-based system. In frequency-based system, namely, Wi-Fi, there are several localization parameters that consist of distance based and direction based [29]. In distance based, there are signal based and time based. In signal based, there are RSSI and CSI. While in time based, there are ToF and RTT. The localization parameter, ToF is a time difference between time of departure in APs and time of arrival in users. The disadvantages are time synchronization, needed for both APs and the user, and higher cost. The strengths include great resistance to multipath effects and high localization accuracy. The other localization parameter, RSSI is a received signal strength indicator that computes distance by power loss and the signal strength deficiency between APs and the user. The weakness is prone to the noise, multipath effects, and NLoS, and the strength is easy to implement and no synchronization of time and additional hardware is needed. Hence, if the priority is the high accuracy, then the ToF localization parameter can be considered over RSSI. However, the RSSI localization option might be preferred over ToF if the priority is low price. To estimate the user position, the positioning algorithm is needed to calculate the localization parameter.

There are the range-based method like trilateration and the range-free method like fingerprint to utilize the localization parameter. The user position estimation for both the methods in a 2D space requires measurements from at least three APs [30], and at least four APs needed for 3D space.

Ma et al. [31] proposed a novel positioning algorithm to improve positioning result of Wi-Fi RTT ranging. They also explained a characteristic of Wi-Fi fine time measurement (FTM). From the results, the proposed approach achieved a localization error of 1.20 m for static and 1.31 m for dynamic positioning.

Zhou et al. [32] proposed a novel indoor-positioning system algorithm with matrix completion and anchor selection. From the results, the proposed approach achieved a localization error of 1.52 m.

In fingerprinting algorithm, there are deterministic approach such as the Kalman filter, NN, KNN, WKNN, SVM, DT, PCA, and neural networks and the probabilistic approach such as Gaussian distribution, particle filter, Kernel method, hidden Markov model, and Naive Bayes method. For example, using the fingerprinting algorithm and Kalman filter, Giovanelli et al. [33] proposed a novel indoor-positioning system with ToF and RSSI data fusion. The mean RMS error of data fusion is about 50% lower than

Parameters	Technologies				
	Bluetooth	Wi-Fi	Zigbee	UWB	RFID
Typical accuracy	2-5 m [16]	<5 m [16]	RSSI	<50 cm [16]	<2 m [16]
Transmission range	~30 m [17]	~50 m [17]	10-100 m [18]	10 m [18]	$< 100 \text{ m}^{1}$
Power consumption	Very low [19]	Moderate [19]	Low [19]	Ultralow [18]	High [20]
Embedded in smart phones	Yes	Yes	No	Uncommon	Uncommon
Deployment cost	Moderate	Low [17]	Moderate	High [19]	Depends ²

TABLE 1: The characteristics of IPS underlying technologies.

¹The transmission range of RFID is below 100 m in free space [20]. ²The deployment cost of an RFID-based IPS depends on the utilized positionning algorithm [21].

when just RSSI data are used, 5.69 m and 2.78 m, respectively. The proposed system utilizes both ToF and RSSI as the location-dependent characteristic. The ToF measurement might have large fluctuations because of the jitter, the limited resolution of time intervals, or a combination of both. Hence, it can be reduced with averaging [34–36]. Although, the impact of ToF on range measurements may decrease with distance, though, the uncertainty of RSSI measurements may grow with distance. Thus, the RSSI and ToF data may complement each other.

Rizk et al. [37] presents an indoor-positioning system with Wi-Fi RTT and RSSI. To solve the problem of signal fluctuations, interference from fingerprinting, multipath propagation errors, and NLOS transmissions, the proposed system achieved a localization error of 0.51 m and 0.59 m, respectively, for office and lab environments.

Singh et al. [38] presents an overview of machine learning-based indoor-positioning system with Wi-Fi RSSI fingerprints. The survey provided an ML-based Wi-Fi RSSI fingerprinting for indoor localization and a comparison of their performance. The performance of ML prediction models such as DT, SVM, KNN, ANN, MLP, CNN, RNN, and DQN has been compared based on classification accuracy, positioning error, robustness, scalability, complexity, localization space, and database used. Also, the author evaluated that CNN [39] has high robustness, high scalability, and low complexity. Then, from the summarized view of indoor localization schemes table, it can be concluded that KNN could increase the robustness and decrease the positioning error, while PCA could decrease the complexity to reduce the computational time. They also summarized the lists of available open-source datasets.

Chin et al. [40] proposed a MIMO-based indoor positioning with CSI data using the artificial neural network. They compare the performance of GCNN, CNN, and FCNN. The error distance that is below 0.2 m is more than 90% for the GCNN, error distance that is below 0.2 m is 75% for the proposed CNN, and error distance is all above 0.4 m for the FCNN.

3. System Model

3.1. Location Fingerprinting. Location fingerprinting is a location inference technique that utilizes locationdependent characteristics to infer where the estimated location is [41]. A fingerprint, in this context, is a set of characteristics or features that characterize a location. As it utilizes Wi-Fi ToF for this research, a fingerprint is a set of Wi-Fi ToFs. The location fingerprinting technique consists of two stages: (i) the offline stage and (ii) the online stage. The features of various locations in the testbed are collected to build a reference database or map in the offline stage. This database contains various fingerprints with their respective location coordinates. In the online stage, the features of an unknown location are collected to create its fingerprint. The fingerprint of the unknown location is then compared with the fingerprints stored in the reference database to estimate where the unknown location is. A high-level view of the location fingerprinting technique is depicted in Figure 1.

3.2. Wi-FiToF. Measuring the ToF of a Wi-Fi signal has been made possible from the time the fine time measurement (FTM) protocol was introduced in the IEEE 802.11-2016 standard. The communication between a client device and an AP under the FTM protocol is shown in Figure 2.

The ToF between two devices is perceived as half of the round-trip time (RTT). Thus, the ToF between a client device and an AP (as illustrated in Figure 2) can be calculated as follows:

$$T_{\rm O}F = \frac{(t_4 - t_1) - (t_3 - t_2)}{2},\tag{1}$$

where t_1 , t_2 , t_3 , and t_4 are timestamps recorded on the local device denoting the time of arrival (ToA) or time of departure (ToD) of the corresponding message.

This research uses a publicly available dataset (that can be accessed at [28]) for our experiments. The dataset consists of records of Wi-Fi signals traveling from a transmitting (T_x) device to a receiving (R_x) device. The ToD and ToA of the corresponding Wi-Fi signal are available in each record. The ToF of each record can be formulated as follows:

$$\psi = ToF_{T_x,R_x} - e$$

$$= ToF_{R_x} - ToD_{T_x},$$
(2)

Feature based on ToF for fingerprint, ToF from T_x to R_x , ToA of the signal assessment R_x , ToD of the signal assessment T_x , and the assessment error path indicates ψ , ToF_{T_x,R_x} , ToF_{R_x} , ToD_{T_x} , and e, respectively. Note that the value of e is not provided in the dataset. However, it considers the value of e as part of the characteristics that are willing to be captured since it represents the area condition.



FIGURE 1: A high-level view of the location fingerprinting technique.

3.3. FCDNN as a Positioning Engine. The role of an FCDNN as the positioning engine, in this case, is to estimate the position of the client device based on its location fingerprint. In the input layer of the FCDNN, the number of neurons is equal to the number of features in the dataset. Additionally, the number of neurons in the output layer depends on the 2-D or 3-D space coordinate. In our case, there are 6 available features in the dataset and 10 APs in the area, and they use the 3-D Cartesian coordinate system to represent the client device location. However, we just used 5 features and omitted the column that displays the AP index in order to evaluate the simulated situations. The structure of the FCDNN for such a case is shown in Figure 3. On the other hand, the number of hidden layers and their neurons is not specified initially (presented in another section).

3.4. Using the FCDNN. To use the FCDNN, it needs to feed it with an input of a location fingerprint. In Figure 3, it is shown in the input layer that each element of the input is connected to a neuron. Therefore, for the input of $X = (x1, ..., xN) \in \mathbb{R}^N$, the activation value of each neuron in the input layer is as follows:

$$a_j^{[1]} = x_j,$$
 (3)

where x_j denotes the j^{th} element of the input and $a_j^{[1]}$ denotes the activation value of the j^{th} neuron in the input layer.

For the remaining layers, each neuron is connected to all the neurons in the previous layer (see Figure 3). Determination of the neuron's activation value in the hidden and output layers is as follows:

$$\mathbf{a}_{j}^{[1]} = \max\left(0, \sum_{k,l=2,\cdots,L} w_{jk}^{[l]} a_{k}^{[l-1]} + b_{j}^{[l]}\right).$$
(4)

The activation value of j^{th} neuron in i^{th} layer, link's weight from $a_j^{[1]}$ to $a_k^{[l-1]}$, activation bias of j^{th} neuron in i^{th} layer, and number of layers represents $a_j^{[1]}$, $w_{jk}^{[l]}$, $b_j^{[l]}$, and L, respectively. Notice that in equation (4), it implements the rectified linear units (ReLU) function in calculating the activation value of the neurons.

The output of the FCDNN is generated based on the value of the neurons in the output layer. As shown in Figure 3, each neuron in the output layer is connected to an



FIGURE 2: FTM protocol illustration.



FIGURE 3: The structure of the FCDNN as a positioning engine in an area with 6 APs.

input element. Therefore, the output of the FCDNN is calculated as follows:

$$Y = (\hat{y}_i, \dots, \hat{y}_n)$$
$$= (a_1^{[L]}, \dots, a_n^{[L]}) \in \mathbb{R}^{n^{[L]}},$$
(5)

where \hat{Y} denotes the output, which is the estimated location coordinates (or label), and $n^{[L]}$ denotes the number of neurons in the output layer.

3.5. Optimizing the FCDNN. By changing the model's parameters $\theta \in \mathbb{R}^d$, such as weights and biases, the FCDNN is optimized. This process aims to make the model better at

doing its task. For that purpose, it uses gradient descent to minimize the cost function $J(\theta)$ by updating each element of θ in the opposite direction of the cost function gradient $\nabla_{\theta} j(\theta)$ w.r.t. the elements of θ [41].

The model's cost using the root mean square error (RMSE), since it considers the issue as a regression problem as follows:

$$J(\theta) = J\left(Y^{(i)}, \widehat{Y}^{(i)}\right)$$
$$= \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(Y^{(i)} - \widehat{Y}^{(i)}\right)^{2}}; \theta \in \mathbb{R}^{d}.$$
(6)

Sample count, weights and biases, true label of i^{th} sample, and estimated label of i^{th} sample indicated by $m, d, Y^{(i)}$, and $\hat{Y}^{(i)}$, respectively. Note that $J(\theta)$ and $(Y^{(i)}, \hat{Y}^{(i)})$ are treated equally as both depend on the model's parameters.

To update the parameters using gradient descent, it adopts the stochastic gradient descent (SGD) algorithm [41]. Additionally, minibatch SGD improves computing efficiency. We use the estimate of moments of gradients to hasten convergence and slow down the quick decay of learning rates [42, 43]. First, calculate $g_{t,i}$ that is the gradient of the cost function w.r.t. to the parameter $\theta_{(i)}$ at the timestep *t* as follows:

$$g_{t,i} = \nabla_{\theta_t} J(\theta_{t,i}). \tag{7}$$

After obtaining $g_{t,i}$, compute the value of m_t and v_t , which are the exponential moving averages of the gradient and the squared gradient, respectively. The computations are as follows:

$$V_t = \beta V_{t-1} + (1 - \beta)S_t,$$
 (8)

where β is the hyperparameter that controls the exponential decay rates of the corresponding moving averages and w_t and b_t are the first moment (the mean) and the second raw moment (the uncentered variance) of the gradient estimates, respectively.

Since the value of w_t and b_t is initialized as 0, weightcorrection and bias-correction to both m_t and v_t is performed to counteract the moment estimates that are biased towards zero at initial timesteps. The value of biascorrected w_t and b_t is calculated as follows:

$$w_{t} = w_{t-1} - \eta \frac{\partial L}{\partial w_{t-1}},$$

$$b_{t} = b_{t-1} - \eta \frac{\partial L}{\partial b_{t-1}},$$
(9)

where w_t and b_t denotes the weight-corrected and biascorrected, respectively. Finally, it can update the model's parameters as follows:

$$\mathbf{w} \coloneqq \mathbf{w} - \eta \nabla \mathbf{Q}_{\mathbf{i}}(\mathbf{w}),\tag{10}$$

where the parameter w that minimizes Q(w) is to be estimated. Each summand function Q_i is typically associated with the i^{th} observation in the dataset (used for training). The hyperparameter η , or the learning rate, controls the step size for each iteration.

4. Experiment Settings

4.1. The Testbed Map. The dataset used in this research is obtained from [28]. The dataset consists of ToF measurements at 4410 locations in the given area. Before creating the fingerprints of each location, we check if all of the APs are accessible at each of those 4410 locations. The locations where a pair of 1, 2, 3, and more APs in the area can be heard. Then, we make 3 scenarios for those APs pair locations. Where, 1 AP pair, 2-10 AP pairs, 1-10 AP pairs, or raw dataset is simulated as shown in Table 2. The ToF measurement samples at 4410 different locations in the given area. In our experiment, our goal is to estimate the client device location based on the Wi-Fi ToFs between the client device and the available APs in the area. Table 3 provides an illustration of measurement data at a specific place, where for every client location (X, Y, Z), there are ψ or the ToFbased feature between the Tx device or the user device and Rx device or APs. Then, these features are available from distance (m) in the dataset. Some client location (X, Y, Z)may have 1, 2, 3, 4, 5, or more available ToF-based feature from nearby APs. For example, in Table 4, in the dataset there are 4 available ToF-based features from 4 nearby APs for client location (-0.74769843 m, 7.46585460 m, and $1.4\,m)$ and $(-0.72188836\,m,\,8.71135620\,m,\,and\,\,1.4\,m).$

4.2. The Training, Validation, and Testing Dataset. From the ToF measurement samples, it can create 4410 fingerprints, and each of the fingerprints belongs to a location in the testbed. Those fingerprints are split into different sets for

TABLE 2: Simulation scenario.

AP distributions	Algorithms		
1 AP pair	WKNN	FCDNN	
2-10 AP pairs	WKNN	FCDNN	
1-10 AP pairs	WKNN	FCDNN	

TABLE 3: An example of measurement data at a certain location.

Client locations			Distance (m)	٨D	
X axis (m)	Y axis (m)	Z axis (m)	Distance (III)	AP	
-0.74769843	7.46585460	1.4	19.9000	1	
-0.74769843	7.46585460	1.4	7.0900	7	
-0.74769843	7.46585460	1.4	30.1600	3	
-0.74769843	7.46585460	1.4	5.7600	8	
-0.72188836	8.71135620	1.4	5.5000	10	
-0.72188836	8.71135620	1.4	37.9100	5	
-0.72188836	8.71135620	1.4	52.6600	9	
-0.72188836	8.71135620	1.4	53.0700	2	

TABLE 4: The fingerprints of point (-0.74769843, 7.46585460, and 1.4) and (-0.72188836, 8.71135620, and 1.4) based on the measurement data in Table 3.

F(-0.74769843, 7.46585460, 1.4)			
F _{AP1}	F _{AP7}	F _{AP3}	F _{AP8}
19.9000	7.0900	30.1600	5.7600
	F(-0.72188836,	8.71135620, 1.4)	
F _{AP10}	F _{AP5}	F _{AP9}	F _{AP2}
5.5000	37.9100	52.6600	53.0700

training and testing purpose. 80% of them are used for training, and the rest of 20% are used for testing. Thus, the number of fingerprints in the training and testing dataset are 3528 and 882, respectively. In Table 4, for the scenario 1, where 1 AP pair was filtered from the dataset with 1006 fingerprints. Thus, the number of fingerprints in the training and testing dataset are 805 and 201, respectively. In addition, the dataset containing 2689 fingerprints was filtered for scenario 2, which included 2 AP pairs and above, up to 10 AP pairs. Thus, the number of fingerprints in the training and testing dataset are 2151 and 538, respectively. Additionally, for scenario 3, where there were at least one AP pair and up to ten AP pairs were filtered out of the dataset with 4410 fingerprints. Thus, the number of fingerprints in the training and testing dataset are 3528 and 882, respectively. Notice that rounding is applied. Then, for these 3 scenarios, 2 algorithms which are WKNN and the proposed FCDNN were simulated to compare the performance to predicted user positions.

4.3. The Model's Hyperparameters. The value of each hyperparameter of our model is detailed in Table 5. There are options for batch size, epochs, hidden layer, neuron in the hidden layer, and η .



TABLE 5: The hyperparameter settings of our model.

FIGURE 4: WKNN predictions (a) 1 AP pair, (b) 2–10 AP pairs, and (c) 1–10 AP pairs.



FIGURE 5: WKNN loss (a) 1 AP pair, (b) 2-10 AP pairs, and (c) 1-10 AP pairs.

5. Results

5.1. Exploration of Model Structures. The experiments started by training WKNN and the proposed FCDNN models where each model has a different combination of an AP pair that filtered from the dataset. To obtain their accuracy, 3 scenarios were simulated to predict user locations (X, Y, Z). Thus, in terms of number of AP pairs, these models are tested for their accuracy. Note that there is a trade-off between accuracy and over-fitness; therefore, the right balance between the two is aimed. The positioning error is calculated as the *L*-2 norm between the estimated and the ground-truth position in Equation (11), where $l(Y^{(i)}, \hat{Y}^{(i)})$ denotes the positioning error of the *i*th example.

$$l\left(Y^{(i)}, \hat{Y}^{(i)}\right) = \left\|\hat{Y}^{(i)} - Y^{(i)}\right\|_{2}.$$
 (11)

5.2. Exploration of Model Structures. Weighted k-nearest neighbor: in Figure 4, the positioning error of X, Y, and Z in the WKNN algorithm are shown. It can be seen, if the distribution of the true values is more condensed near the predicted lines and also linear along the lines, the algorithm can be considered more accurate. Additionally, because every user's z location is the same (1.4 m), the WKNN

algorithm can identify this distribution of user z positions and forecast that all user z positions will be at a single location with a value of almost 1.4 m. It may be inferred from this that the WKNN was able to discriminate between the user x, y, and z position distribution, where the user x and y position have diverse distributions. Additionally, distinguish between the user z locations and the fact that they all have the same value under a single distribution. Figures 5 and 6 show WKNN loss for K = [1, 39]. It can be seen that, 0.9424 m loss and 1.3635 m RMSE are the lowest in scenario 3 with K = 3, where the dataset used has 1 and above, up to the 10 AP pair distribution. For scenarios 1 and 2, the loss and RMSE are lowest with K = 2. Furthermore, scenario 1 with K=2 has the greatest 3.1136 m loss and 2.4771 mRMSE. Thus, in order for WKNN to achieve lower loss and RMSE, the AP pair distribution of 2 and even 1 are needed in scenario 3.

5.3. Fully Connected Deep Neural Network. In Figure 7, the positioning error of X, Y, and Z in the proposed FCDNN algorithm are shown. It can be seen, if the distribution of the true values is more condensed near the predicted lines and also linear along the lines, the algorithm can be considered more accurate. Moreover, because every user's z location is the same (1.4 m), the FCDNN method predicts



FIGURE 6: WKNN RMSE (a) 1 AP pair, (b) 2-10 AP pairs, and (c) 1-10 AP pairs.

that the user's z position has a variety of points with values ranging from 1.39 m to 1.49 m but is unable to discern the user's z position distribution. Therefore, it may be inferred that with a single distribution, the FCDNN was unable to discriminate across user z position distributions where the user x and y position have diverse distributions and were unable to distinguish between the user z places since they all had the same value under one distribution. Figures 8 and 9 show the proposed FCDNN loss for 1000 epochs for both training and testing. It can be seen that, 0.1749 m loss and 0.5740 m RMSE are the lowest in scenario 3, where the dataset used has 1 and above, up to the 10 AP pair distributions, and 0.6277 m loss and 1.1231 m RMSE are the highest in scenario 1. Thus, in order for FCDNN to achieve lower loss and RMSE, the AP pair distribution of 2 and even 1 are needed in scenario 3.

5.4. Performance Comparison of WKNN and FCDNN. In Table 6, FCDNN has lower loss and RMSE than WKNN in all 3 scenarios, and if there are ever more diverse APs accessible, the performance of both algorithms is more likely to improve. The WKNN has the lowest loss and RMSE with 0.9424 m and 1.3635 m, respectively in scenario 3 or 1–10 AP



FIGURE 7: FCDNN predictions (a) 1 AP pair, (b) 2-10 AP pairs, and (c) 1-10 AP pairs.



FIGURE 8: FCDNN loss (a) 1 AP pair, (b) 2-10 AP pairs, and (c) 1-10 AP pairs.



FIGURE 9: FCDNN RMSE (a) 1 AP pair, (b) 2-10 AP pairs, and (c) 1-10 AP pairs.

TABLE 6: The error metrics (a) WKNN and (b) FCDNN.	
--	--

Error metrics	1 AP pair (m)	2-10 AP pairs (m)	1-10 AP pairs (m)
(a) WKNN			
Loss	3.1136	1.0968	0.9424
RMSE	2.4771	1.4810	1.3635
(b) FCDNN			
Loss	0.6277	0.2142	0.1749
RMSE	1.1231	0.6532	0.5740

pairs. The FCDNN has the lowest loss and RMSE with 0.1749 m and 0.5740 m, respectively, in scenario 3 or 1–10 AP pairs or the raw dataset.

6. Conclusion

The proposed IPS, DeepIndoor, which combines Wi-Fi ToF and a deep learning approach, successfully achieves the goal of this research, namely, enabling the realization of a highaccuracy, cheap, and practical IPS. The use of a deep learning model, where it established FCDNN that was made for this purpose, allows for the high accuracy. The average positioning error of DeepIndoor is 0.1749 m and RMSE of 0.5740 m. The realization of DeepIndoor is also cheap and practical since it utilizes Wi-Fi as the underlying technology where the availability of Wi-Fi in most smartphones and the deployment of Wi-Fi networks in many indoor facilities contribute to these advantages. Therefore, accuracy will increase if there is a larger variety of the AP pair distribution available in the dataset used for training and testing.

Data Availability

The Wi-Fi ToF dataset used to support the findings of this study are available in the GitHub repository (https://github. com/intel/WiFi-Location-Core-PE-and-Measurement-Database).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors want to thank Telkom University, Bandung, for being willing to provide research funding and publication costs. FCT/MCTES partially funds this work. This work was funded by Telkom University.

References

- M. R. Palattella, M. Dohler, A. Grieco et al., "Internet of things in the 5G era: enablers, architecture, and business models," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 510–527, 2016.
- [2] M. A. Ezechina, K. K. Okwara, and C. A. U. Ugboaja, "The Internet of Things (Iot): a scalable approach to connecting everything," *International Journal of Engineering Sciences*, vol. 4, no. 1, pp. 9–12, 2015.
- [3] K. Yasaka and O. Abe, "Deep learning and artificial intelligence in radiology: current applications and future directions," *PLoS Medicine*, vol. 15, no. 11, Article ID e1002707, 2018.
- [4] A. A. Cardoso and F. H. T. Vieira, "Adaptive fuzzy flow rate control considering multifractal traffic modeling and 5G communications," *PLoS One*, vol. 14, no. 11, Article ID e0224883, 2019.
- [5] A. Basiri, E. S. Lohan, T. Moore et al., "Indoor location based services challenges, requirements and usability of current solutions," *Computer Science Review*, vol. 24, pp. 1–12, 2017.

- [6] P. Pascacio, S. Casteleyn, J. Torres-Sospedra, E. S. Lohan, and J. Nurmi, "Collaborative indoor positioning systems: a systematic review," *Sensors*, vol. 21, no. 3, p. 1002, 2021.
- [7] W. Jiang, Y. Li, C. Rizos, B. Cai, and W. Shangguan, "Seamless indoor-outdoor navigation based on GNSS, INS and terrestrial ranging techniques," *Journal of Navigation*, vol. 70, no. 6, pp. 1183–1204, 2017.
- [8] J. Kunhoth, A. Karkar, S. Al-Maadeed, and A. Al-Ali, "Indoor positioning and wayfinding systems: a survey," Humancentric Comput," *Information Science*, vol. 10, no. 1, pp. 1– 41, 2020.
- [9] G. Xu, "Method of enhancing the accuracy of indoor positioning (RSSI to UL-AOA)," *Journal of Physics: Conference Series*, vol. 1871, no. 1, Article ID 012076, 2021.
- [10] D. Wicaksono, D. Perdana, and R. Mayasari, "Design and analysis automatic temperature control in the broiler poultry farm based on wireless sensor network," in *Proceedings of the* 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITI-SEE), pp. 450–455, Yogyakarta, Indonesia, November 2017.
- [11] R. C. Manurung, D. Perdana, and R. Munadi, "Performance evaluation Gauss-Markov mobility model in vehicular ad-hoc network with spearman correlation coefficient," in *Proceedings of the 2016 International Seminar on Intelligent Technology and its Applications (ISITIA)*, pp. 351–356, Lombok, Indonesia, July 2016.
- [12] B. Pamukti and D. Perdana, "Non-linear effects of high rate soliton transmission on DWDM optical fiber communication system," in *Proceedings of the 2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 26–30, Yogyakarta, Indonesia, August 2016.
- [13] D. Deliyska, N. Yanev, and M. Trifonova, "Methods for developing an indoor navigation system," E3S Web of Conferences, vol. 280, Article ID 04001, 2021.
- [14] C. C. Yeh, K. J. Jhang, and C. C. Chang, "An intelligent indoor positioning system based on pedestrian directional signage object detection: a case study of Taipei Main Station," *Mathematical Biosciences and Engineering*, vol. 17, no. 1, pp. 266–285, 2020.
- [15] R. Simedroni, "Indoor positioning using decawave MDEK1001," in *Proceedings of the 2020 International Workshop on Antenna Technology (iWAT)*, pp. 1–4, Bucharest, Romania, February 2020.
- [16] R. Carotenuto, M. Merenda, D. Iero, and F. G Della Corte, "Simulating signal aberration and ranging error for ultrasonic indoor positioning," *Sensors*, vol. 20, no. 12, p. 3548, 2020.
- [17] N. R. Mrindoko and L. M. Minga, "A comparison review of indoor positioning techniques," *International Journal of Computer*, vol. 21, no. 1, pp. 42–49, 2016.
- [18] G. M. Mendoza-Silva, J. Torres-Sospedra, and J. Huerta, "A meta-review of indoor positioning systems," *Sensors*, vol. 19, no. 20, p. 4507, 2019.
- [19] S. Subedi and J. Y. Pyun, "Deep learning and artificial intelligence in radiology: current applications and future directions," *Journal of Sensors*, vol. 2017, Article ID 9742170, 16 pages, 2017.
- [20] F. Mazhar, M. G. Khan, and B. Sallberg, "Precise indoor positioning " using UWB: a review of methods, algorithms and implementations," *Wireless Personal Communications*, vol. 97, no. 3, pp. 4467–4491, 2017.
- [21] K. A. Nguyen, Z. Luo, G. Li, and C. Watkins, "A review of smartphones based indoor positioning: challenges and applications," 2020, https://arxiv.org/abs/2006.02251.

- [22] T. Kim Geok, "Review of indoor positioning: radio wave technology," *Journal of Navigation*, vol. 70, no. 6, pp. 1183– 1204, 2017.
- [23] Y. B. Bai, S. Wu, H. R. Wu, and K. Zhang, "Overview of RFIDbased indoor positioning technology," GSR, vol. 2012, 2012.
- [24] H. Xu, M. Wu, P. Li, F. Zhu, and R. Wang, "An RFID indoor positioning algorithm based on support vector regression," *Sensors*, vol. 18, no. 5, p. 1504, 2018.
- [25] C. Lamoureux and R. Chelouah, "Fusion particle and fingerprint recognition for indoor positioning system on mobile," *Engineering Applications of Artificial Intelligence*, vol. 98, Article ID 104082, 2021.
- [26] Y. S. Kuo, P. Pannuto, K. J. Hsiao, and P. Dutta, "Luxapose: indoor positioning with mobile phones and visible light," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, pp. 447–458, Maui, HA, USA, September 2014.
- [27] S. Fakhoury, V. Arnaoudova, C. Noiseux, F. Khomh, and G. Antoniol, "Keep it simple: is deep learning good for linguistic smell detection?" Comput," in *Proceedings of the 2018 IEEE 25th International Conference on Software Analysis*, *Evolution and Reengineering (SANER)*, pp. 602–611, Campobasso, Italy, March 2018.
- [28] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur, "Reference-PE-and-Measurements-DB-for-WiFi-Time-based-Scalable-Location," 2018, https://git-hub.com/intel/ Reference-PE-and-Measurements-DB-for-Wi-Fi-Timebased-Scalable-Location.
- [29] T. Kim Geok, K. Zar Aung, M. Sandar Aung et al., "Review of indoor positioning: radio wave technology," *Applied Sciences*, vol. 11, no. 1, p. 279, 2020.
- [30] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 689–705, 2007.
- [31] C. Ma, B. Wu, S. Poslad, and D. R. Selviah, "Wi-Fi RTT Ranging Performance Characterization and Positioning System Design," *IEEE Transactions on Mobile Computing*, vol. 21, no. 2, pp. 740–756, 2020.
- [32] M. Zhou, Y. Li, Y. Wang, Q. Pu, X. Yang, and W. Nie, "Device-to-device cooperative positioning via matrix completion and anchor selection," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5461–5473, 2022.
- [33] D. Giovanelli, E. Farella, D. Fontanelli, and D. Macii, "Bluetooth-based indoor positioning through ToF and RSSI data fusion," in *Proceedings of the 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, Nantes, France, September 2018.
- [34] D. Macii, A. Colombo, P. Pivato, and D. Fontanelli, "A data fusion technique for wireless ranging performance improvement," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 1, pp. 27–37, 2013.
- [35] T. Gadeke, J. Schmid, M. Kr " uger, J. Jany, W. Stork, and K. D. M. uller-Glaser, "A bi-modal ad-hoc localization scheme for wireless networks based on RSS and ToF fusion," in *Proceedings of the 10th Workshop on Positioning, Navigation and Communication (WPNC)*, pp. 1–6, Dresden, Germany, March 2013.
- [36] D. Giovanelli, B. Milosevic, and E. Farella, "Bluetooth low energy for data streaming: application-level analysis and recommendation," in *Proceedings of the 2015 6th International Workshop on Advances in Sensors and Interfaces* (*IWASI*), pp. 216–221, Gallipoli, Italy, June 2015.

- [37] H. Rizk, A. Elmogy, and H. Yamaguchi, "A robust and accurate indoor localization using learning-based fusion of wi-fi RTT and RSSI," *Sensors*, vol. 22, no. 7, p. 2700, 2022.
- [38] N. Singh, S. Choe, and R. Punmiya, "Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview," *IEEE Access*, vol. 9, pp. 127150–127174, 2021.
- [39] W. Njima, I. Ahriz, R. Zayani, M. Terre, and R. Bouallegue, "Deep CNN for indoor localization in IoT-sensor systems," *Sensors*, vol. 19, no. 14, p. 3127, 2019.
- [40] W. L. Chin, C. C. Hsieh, D. Shiung, and T. Jiang, "Intelligent indoor positioning based on artificial neural networks," *IEEE Network*, vol. 34, no. 6, pp. 164–170, 2020.
- [41] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, https://arxiv.org/abs/1609.04747.
- [42] X. Liu, Z. Pan, H. Yang, X. Zhou, W. Bai, and X. Niu, "An Adaptive Moment estimation method for online AUC maximization," *PLoS One*, vol. 14, no. 4, Article ID e0215426, 2019.
- [43] Y. Peng, Y. Shi, and C. Wang, "An enhanced reliable access scheme for massive IoT applications in ubiquitous IoT systems," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 7241362, 15 pages, 2022.