IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 3, 2022, accepted June 10, 2022, date of publication June 14, 2022, date of current version June 23, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3183113*

# A Fast Indoor Positioning Using a Knowledge-Distilled Convolutional Neural Network (KD-CNN)

**AQILAH BINTI MAZLAN[1], YIN HOE NG[1], AND CHEE KEONG TAN[2], (Member, IEEE)**
[1]Faculty of Engineering, Multimedia University, Cyberjaya 63100, Malaysia
[2]School of Information Technology, Monash University Malaysia, Subang Jaya 47500, Malaysia

Corresponding author: Yin Hoe Ng (yhng@mmu.edu.my)

This work was supported by the Fundamental Research Grant Scheme (FRGS), Ministry of Higher Education Malaysia under Grant FRGS/1/2019/ICT02/MMU/03/13.

**ABSTRACT** Fingerprint-based indoor positioning systems (F-IPS) may provide inexpensive solutions to GPS-denied environments. Most F-IPSs adopt traditional machine learning for position prediction, resulting in low accuracy. Deep neural networks (DNN) were recently employed for F-IPSs to minimize positioning errors. Nevertheless, a DNN-IPS fails to guarantee high accuracy in dynamic environments as it is sensitive to changes in the input data. A convolutional neural network (CNN) is recommended to replace DNN due to its capability to learn the overall topology of fingerprinting images and capture highly abstract features. Due to the convolution process and image representation, CNN-IPS incurs prohibitive storage and computational requirement for implementation on resource-limited devices. This paper incorporates knowledge distillation (KD) into CNN-IPS to distil knowledge from large deep CNNs into small CNNs. The pre-trained teacher network uses the soft probability output where the score vector from the trained network is converted into a probability distribution, which is softened by the temperature hyperparameter, leading to a more simplified model. Based on the numerical results, KD-CNN-IPS manifests better localization performance where 79.84% of the positioning errors are within 2 meters while its testing time is only 79.68% of that of the teacher model. Compared to the CNN-IPS, KD-CNN-IPS with precisely the same architecture and size could achieve a performance improvement of 13.65% in terms of the average positioning error.

**INDEX TERMS** Indoor positioning, fingerprint, received signal strength indicators, knowledge distillation, convolutional neural networks.

## I. INTRODUCTION

The proliferation of smartphones and intelligent wearable devices in the last decade has led to the emergence of location-aware computing (LAC) and location-based services (LBS), which have immensely transformed human lifestyles. LAC and LBS are value-added platforms or systems that acquire the users' geographical locations using an indoor positioning algorithm and provide the related services based on their location information. Due to its social and commercial values [1], [2], indoor positioning technology has recently attracted much attention from industry and academia, with the market size estimated to skyrocket from USD 7 billion in 2021 to about USD 20 billion in 2026 [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Le Hoang Son.

Unlike outdoor localization, the indoor environment is often more complicated and complex, usually characterized by the existence of many obstacles, non-line-of-sight (NLoS) of reference objects, severe signal variation and fluctuation, presence of ambient noise, frequent environmental and layout changes, etc. Despite such a complex indoor environment, high positioning accuracy is still anticipated to ensure satisfactory LAC and LBS.

### A. RELATED WORKS

Traditional indoor localization system (IPS) employs trilateration [4] and triangulation [5] approaches, which strictly necessitate line-of-sight (LoS) wireless signals for accurate position estimation [6]. These positioning schemes do not perform effectively and precisely indoors, with many obstacles or objects preventing LoS transmission.

To solve this NLoS issue, fingerprint-based IPS [6]–[10] is a promising alternative that can be used to provide satisfactory localization accuracy in a complex indoor environment. In general, fingerprinting is a process of accumulating wireless signals at different reference points (RPs) from multiple access points (APs) and associating the signals with indoor locations. A target's position is specifically characterized by the measured signal patterns such as a vector of received signal strength indicators (RSSIs) from multiple APs located at different places. Compared to trilateration and triangulation, fingerprint-based IPS guarantees high feasibility for indoor deployment as it requires neither the exact AP locations nor angle and distance measurements. Among various wireless signals, WiFi [11] and Bluetooth signals [12] are the most prevalent fingerprinting choice as WiFi APs are ubiquitous and Bluetooth transceiver is available cheaply on every smartphone.

Additionally, the advancement of WiFi and Bluetooth technologies, leading to the development of WiFi 6 [13] and Bluetooth 5 [14], has ensured more stable signals and broader coverage which can indirectly improve the performance of indoor positioning. Compared to WiFi access points, BLE beacons are inexpensive and easy to deploy. In this work, we will focus on Bluetooth fingerprinting, where the Bluetooth signals are generated by Bluetooth Low Energy (BLE) beacons, which will be collected as the vectors of BLE RSSI.

Bluetooth fingerprinting is conducted in two phases: 1) an offline phase (site survey); 2) an online phase (instant query) [15]. Firstly, a site survey is performed by gathering the vectors of RSSIs of all the detectable BLE signals at many distinct RPs of known locations where each RP is denoted by its own fingerprint. The collected RSSI vectors are then stored at a central database for online query later. Secondly, in the online phase, a target sample (the measured RSSI vector) at a location is fed back to the database for RSSI similarity inspection to estimate the target's position based on the most similar vectors where the set of RPs whose fingerprints closely correlate with the target's RSSI. Many conventional indoor positioning systems adopt traditional machine learning methods such as $K$-nearest neighbor (KNN) [16], support vector machine (SVM) [17], and random forest [18] to estimate the target's position based on the measured RSSI in the online phase. Although these approaches are simple for implementation, they lack the ability to entirely gain from the training data to retain the complex features [19], leading to low positioning accuracy.

Recently, a deep neural network (DNN) [20] has been proposed for indoor positioning to minimize the positioning error. A DNN model is capable of analyzing the RSSI values of the target area collected in the offline stage along with its corresponding location with one label. Subsequently, a user's actual location is learned based on this trained model. During the training phase of the DNN-based indoor positioning, the weight and bias value of each node of the hidden layers and output layer are adjusted using a backpropagation algorithm to characterize the input-output relationship. Nevertheless, recent studies have shown that DNN has encountered hurdles in guaranteeing high positioning accuracy in some dynamic environments as DNN is sensitive to changes in the input data [21] and requires sufficient input training data [22] for satisfactory positioning performance.

Convolutional neural network (CNN) based classifiers have been recommended for indoor positioning technology because CNN is more robust to the fluctuation of received signals caused by multipath effects [21]. The work in [21]–[23] has adopted a CNN model to improve the performance of the WiFi fingerprinting for indoor localization by addressing the drawbacks encountered by the DNN technique. The empirical results presented in [21] have manifested that the proposed CNN-based indoor positioning system is capable of attaining an improvement of 2.52% in terms of positioning accuracy compared to the DNN model. The superior performance of the CNN model is mainly due to the capability of CNN to learn the overall topology of an image of a radio map of fingerprints and highly abstract features [21] of the image compared to DNN that merely considers the RSSI vectors. Moreover, a CNN has deep feed-forward architecture and has an incredible ability to generalize better than networks with fully connected layers [24].

### B. MOTIVATIONS AND CONTRIBUTIONS

Although CNN is more robust than DNN for indoor positioning, there are still some challenges to deploying such a system for real-time application, especially on wearable devices with minimal resources. Furthermore, the convolution process in CNN might require higher computational power, leading to higher training and testing time. The prohibitive computational requirement has deterred the motivation to adopt CNN for indoor positioning. To address this issue, some forms of model compression or acceleration techniques can be incorporated into the CNN-based IPS to reinforce the efficiency of the deep learning model. Among those model compression techniques, knowledge distillation (KD) [25] is a viable approach to distill knowledge learned from a more extensive deep neural network into a small neural network so that the distilled network is less computationally demanding.

Most of the existing works done on KD focus on compressing DNNs. A survey [26] on KD has been extensively carried out, showing that the distilled student network can be deployed in many applications such as visual recognition, speech recognition, and natural language processing (NLP). Additionally, KD is also applicable to other tasks, including adversarial attacks, data augmentation, and data privacy and security. Another review on various KD architectures can be found in [27], which summarizes the performance of KD incorporated onto different CNN variances in terms of their complexity and performance loss. Recently, KD has been adopted to compress deep CNN to enable its application on embedded devices with limited computation, power, and memory resources [28]. In [29], a KD with a higher-dimensional hint layer is introduced to

learn from a complex deep CNN model for multi-class object detection. To the best of our knowledge, there is no prior work on KD for IPS and CNN-based IPS.

In this work, a CNN-based IPS is developed in which the offline phase utilizes the BLE-generated fingerprints, which are transformed into fingerprint images for CNN training and testing. The developed CNN model is used as a teacher model, while the simplified CNN model with a smaller filter size in the convolution layers is considered a student model. A novel KD framework is proposed in this work to distill the knowledge gained from the teacher model to the student model. To show the importance and benefit of using KD, the positioning errors of the student models with and without the proposed KD are compared. It is noticed that the proposed knowledge-distilled CNN-based IPS (KD-CNN-IPS) can achieve much better positioning accuracy compared with the one without KD while achieving marginal loss in terms of positioning accuracy as compared the teacher model (CNN-IPS). In summary, the contributions of this paper are summarized as follows:

1. A CNN-based IPS (CNN-IPS) is developed where the RSSI vectors are converted to fingerprint images before inputting them into the CNN model. This model is considered as the teacher model without any model compression. The developed CNN model is simplified by reducing the number of epochs and the filter size in the convolution layers to minimize the testing time. This simplified CNN model is treated as a student model. In this work, we will consider two simplified student models with different numbers of epochs and filter sizes.
2. A novel KD framework is proposed to distill knowledge from the teacher model to the student model. In the proposed KD-CNN-IPS framework, the soft probability output is used by the pre-trained teacher network, where a vector of scores is produced by the trained neural network, which is then converted into a probability distribution. The temperature hyperparameter is introduced in this framework to soften the probability distribution, which might not be informative for the student network.
3. A comprehensive investigation is conducted to analyze the performance of the proposed KD-CNN-IPS (student model with KD) as compared to the CNN-IPS (teacher model) in terms of their 2D and 3D positioning errors, testing loss, and testing time. Besides, the proposed KD-CNN-IPS is also compared with the student model without KD to highlight the importance and benefit of using KD.

### C. PAPER ORGANIZATION

The rest of this paper is organized as follows. In Section II, the CNN architecture for IPS is described where the teacher and student models are established. A novel KD framework is proposed in Section III, and the mechanism for distilling knowledge from the teacher model to the student models is also demonstrated. Section IV describes the dataset preparation and presents a comprehensive investigation of the proposed KD-CNN-IPS from various aspects. The comparative results to highlight the benefit of using KD on IPS are also shown in Section IV. The paper ends with some insightful concluding remarks in Section V.

## II. CONVOLUTIONAL NEURAL NETWORK-BASED INDOOR POSITIONING SYSTEM

This section provides a brief review of the CNN-based indoor positioning system (CNN-IPS) considered in this study. Figure 1 depicts the architecture of CNN-IPS, which consists of an input layer, convolutional layers, leaky ReLu activation layer, max-pooling layers, a flatten layer, and a dense layer. The CNN-IPS works as follows. Let $\{\boldsymbol{r}^n | n = 1, 2, \ldots, N\}$ and $\{\boldsymbol{y}^n | n = 1, 2, \ldots, N\}$ denote the input RSSI and the corresponding ground-truth label datasets of $N$ samples, respectively. More explicitly, $N = \sum_{m=1}^{M} g_m$, where $M$ represents the total number of distinct location classes and $g_m$ is the total number of samples in the $m$th location. The $n$th input sample of the CNN-IPS is a one-dimensional (1D) RSSI vector which can be expressed as

$$\boldsymbol{r}^n = \begin{bmatrix} r_1^n & r_2^n & \cdots & r_K^n \end{bmatrix} \tag{1}$$

where $r_k^n$, $k = 1, 2, \ldots K$, and $K$ denotes the RSSI of the $k$th access point for the $n$th sample and the total number of access points used for positioning, respectively. Next $\boldsymbol{r}^n$ will be converted to a two-dimensional (2D) fingerprint image $\boldsymbol{X}^n$ by reshaping $\boldsymbol{r}^n$ into a square matrix of size $Q_1 \times Q_1$. As such, each of the RSSI values in $\boldsymbol{r}^n$ corresponds to a pixel in $\boldsymbol{X}^n$. If $K \neq c^2$ where $c$ is an integer, $\boldsymbol{r}^n$ will be padded with zeroes before it is transformed to 2D fingerprint image. The 2D fingerprint image serves as an input to the convolutional layer, followed by an activation layer and a pooling layer. The output of the pooling layer will then be processed by the second convolutional layer. Subsequently, the images produced by the second convolutional layer will be flattened and sent to the dense network. Finally, in the last layer of the dense network, a predicted output vector $\hat{\boldsymbol{y}}^n$ of size $1 \times M$, which can be written as

$$\hat{\boldsymbol{y}}^n = \begin{bmatrix} \hat{y}_1^n & \hat{y}_2^n & \cdots & \hat{y}_M^n \end{bmatrix} \tag{2}$$

will be generated. In the following, the working principle of each of the layers will be explained in detail.

### A. CONVOLUTIONAL LAYERS

The role of the convolutional layers is to perform feature extraction. To accomplish this, a set of learnable filters, also known as kernels, are applied to convolve the input fingerprint images in parallel. This process results in a set of outputs known as feature maps. More explicitly, the kernels function as a sliding window that learns the local features of the fingerprint image by convolving with the entire region of the input image with an overlapping distance. The key hyperparameters of this layer are the convolutional kernel

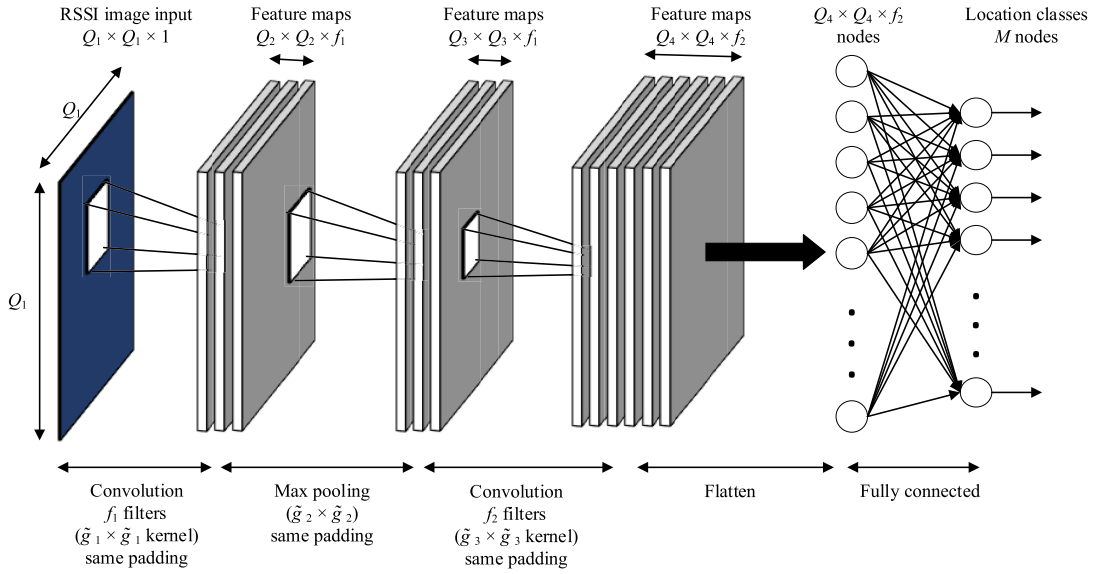


**FIGURE 1.** The general structure of CNN-IPS.

size, the number of filters, stride, and fillings which would determine the size of the output. The size of the kernels has to be smaller than that of the input image [30].

On the other hand, the stride is the number of element shifts that the kernel moves over an input along a particular axis, and it could be deployed to control the density of convolving. To capture the maximum number of features, a stride of 1 could be used as it provides the maximum overlapping between the kernel and the input. As the stride increases, the receptive field overlap reduces because the kernel would shift over the input with larger step size, and this results in a smaller feature map. Also, the choice of stride value has a significant impact on the computational complexity of the network during both the training and testing phases. To avoid the convolutional operation from inducing information loss on the border of the image, the same padding could be employed by inserting zero values to the outer frame of the input image.

Mathematically, the $(i, j)$ element of the feature map $\boldsymbol{S}$ can be expressed as [30]

$$\begin{aligned} \boldsymbol{S}(i,j) &= (\boldsymbol{H} * \boldsymbol{V})(i,j) \\ &= \sum_a \sum_b \boldsymbol{V}(i+a, j+b)\boldsymbol{H}(a,b) \end{aligned} \tag{3}$$

where $*$ denotes the 2D convolution operator, $\boldsymbol{V}(a, b)$ and $\boldsymbol{H}(a, b)$ represent the $(a, b)$ elements of the input image $\boldsymbol{V}$ and convolution kernel $\boldsymbol{H}$, respectively.

### B. ACTIVATION LAYER

To enable the CNN-IPS to capture the complex nonlinear relationship between 2D fingerprint input and the output location, various types of nonlinear activation functions could be harnessed. Conventional activation functions such as logistic sigmoid and hyperbolic tangent functions suffer from gradient vanishing, which makes the network parameters of deep CNN difficult to be trained effectively as the slopes of the functions tend to be zero when the input of the activation function $x$ is too small or too large. To mitigate this issue, rectified linear unit (ReLU) activation function could be employed. Mathematically, ReLU is defined as

$$f_{\text{ReLU}}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $x$ is the input of the activation function. The function returns zero value when the input is negative, and it produces exactly the same value as the input when the input is positive. Compared to the logistic sigmoid and hyperbolic tangent functions, ReLU could achieve a much faster computing rate. This is due to the fact that division is necessitated to compute the derivatives for the logistic sigmoid and hyperbolic tangent functions as the functions involve exponential operation while the gradient of ReLU is a constant. However, ReLU suffers from a dead neuron, i.e., the neuron will be deactivated when the input is negative, causing some neurons to be untrained during the entire training phase. To address this problem, a leaky ReLU (LReLU) activation function, which is one of the variants of ReLU, is utilized in this study. The LReLU function could be written as

$$f_{\text{LReLU}}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{otherwise} \end{cases} \tag{5}$$

Unlike the ReLU function, the LReLU function assigns a small positive gradient $\alpha$ for negative input.

### C. POOLING LAYER

Pooling is a pivotal step in CNN-IPS that enhances the efficiency of the training process. In the convolutional layers, a large number of feature maps would be produced. To reduce

the computational cost in training the network, a pooling function could be adopted to downsample the feature maps of the previous convolutional layer by preserving useful details and discarding irrelevant information. By doing so, the spatial dimension of the feature maps could be effectively reduced. Besides that, pooling could also reduce the risks of the network from being overfitted and enhance invariance to small shifts and distortions on the inputs. The most widely adopted pooling operation is max pooling, which extracts patches from the input feature maps and selects the maximum value in each patch. Mathematically, the max pooling operation [31] could be expressed as

$$\sigma_{uij} = \max_{(p,q)\in R_{ij}} x_{upq} \tag{6}$$

where $\sigma_{uij}$ denotes the output of pooling operator for the $u$th feature map and $x_{upq}$ refers to the $(p, q)$ element that is within the pooling window $R_{ij}$, which represents a local neighborhood around the position $(i, j)$.

### D. FULLY CONNECTED LAYER

The layers mentioned above are responsible for extracting valuable features for positioning. To classify the location, the feature maps generated by the final convolution layer flattened and connected to the fully connected layers (a.k.a. dense layers). In the fully connected layers, each of the neurons is linked to all the neurons in adjacent layers. Besides that, each neuron in the fully connected layer is followed by an activation function. For the last fully connected layer, it has the same number of neurons as the number of location classes, and the softmax activation function is utilized to compute the probability for each of the classes. The softmax activation could be computed as follows:

$$f_{\text{softmax}}\left(x_j\right) = \frac{e^{x_j}}{\sum_{l=1}^{L} e^{x_l}} \tag{7}$$

where $x_j$ indicates logit of the $j$th neuron, $l = 1, 2, \ldots, L$, and $L$ is the total number of neurons for the fully connected layer considered. Specifically, the softmax activation function maps the features learned by the last fully-connected layer to a set of probability values that range from 0 to 1, and the sum of all the outputs in this layer is 1. The predicted location can then be obtained by selecting the output class with the highest probability.

To train the CNN-IPS, an adaptive moment estimation (Adam) optimizer is employed to minimize the cross-entropy loss between the predicted output and the ground truth label, which can be computed as

$$\begin{aligned} L_{\text{CE}}\left(\boldsymbol{z}^n, \boldsymbol{y}^n\right) &= H\left(f_{\text{softmax}}(\boldsymbol{z}^n), \boldsymbol{y}^n\right) \\ &= -\sum_{k=1}^{M} f_{\text{softmax}}(z_k^n) \log\left(y_k^n\right) \end{aligned} \tag{8}$$

where $H(\psi, \xi) = -\sum_{k=1}^{M} \psi_k \log(\xi_k)$ refers to the cross-entropy loss function and $\boldsymbol{z}^n = \begin{bmatrix} z_1^n & z_2^n & \cdots & z_M^n \end{bmatrix}$ is a vector of logits produced by the last fully connected layer for the $n$th input sample. One salient feature of Adam is that it utilizes adaptive learning rates for each network weight. The iterative update rules for Adam [32] are given as follows:

$$w_j^{(i+1)} = w_j^{(i)} - \lambda \times \frac{\frac{\theta_j^{(i)}}{1-\gamma_1}}{\sqrt{\frac{\omega_j^{(i)}}{1-\gamma_2}} + \varepsilon} \tag{9}$$

where $w_j^{(i)}$ denotes the $j$th weight at iteration $i$, $\lambda$ signifies the learning rate, $\gamma_1$ and $\gamma_2$ represent the hyperparameters for the optimizer, $\epsilon$ is an extremely small constant value used to avoid division by zero, $\theta_j^{(i)}$ and $\omega_j^{(i)}$ are the first and second gradient moments of the past gradients, respectively. Specifically, $\theta_j^{(i)}$ and $\omega_j^{(i)}$ can be formulated as

$$\theta_j^{(i)} = \gamma_1 \times \theta_j^{(i-1)} + (1 - \gamma_1) \times \delta_j^{(i)} \tag{10}$$

and

$$\omega_j^{(i)} = \gamma_2 \times \omega_j^{(i-1)} + (1 - \gamma_2) \times \left(\delta_j^{(i)}\right)^2, \tag{11}$$

respectively, where $\delta_j^{(i)}$ is the gradient of the cost function with respect to the weight.

## III. KNOWLEDGE DISTILLATION BASED INDOOR POSITIONING SYSTEM

In this section, the proposed methodology is introduced. Although CNN-IPS possesses a powerful learning capability, it is too complex to be deployed on edge-computing systems. To overcome this challenge, in this paper, we aim to leverage response-based knowledge distillation (KD) scheme developed in [33] to effectively transfer useful knowledge learned by a pre-trained complex CNN-IPS which is also known as teacher model and abbreviated as CNN-IPS (Teacher), to a lightweight CNN-IPS structure termed as student model and abbreviated as CNN-IPS (Student) so that better control over the compression-performance tradeoff could be achieved.

Figure 2 depicts the block diagram of the proposed knowledge distillation framework for indoor positioning. As illustrated in the figure, firstly, the 1D input RSSI vectors are transformed into 2D fingerprint images. Next, the pre-optimized teacher model maps the 2D input fingerprint images to $\boldsymbol{z} = [z_1 z_2 \cdots z_M]$ on the last fully connected layer. Then, the temperature-scaled softmax activation function is applied on the logits to generate soft labels $\rho_i$, which can be formulated as

$$\rho_i = f_{\text{TS-Softmax}}\left(z_i\right) = \frac{e^{\frac{z_i}{T}}}{\sum_{j=1}^{M} e^{\frac{z_j}{T}}} \tag{12}$$

where $T \geq 1$ represents the temperature parameter. Note that $f_{\text{TS-Softmax}}(z_i) = f_{\text{softmax}}(z_i)$ when $T = 1$. In this case, the

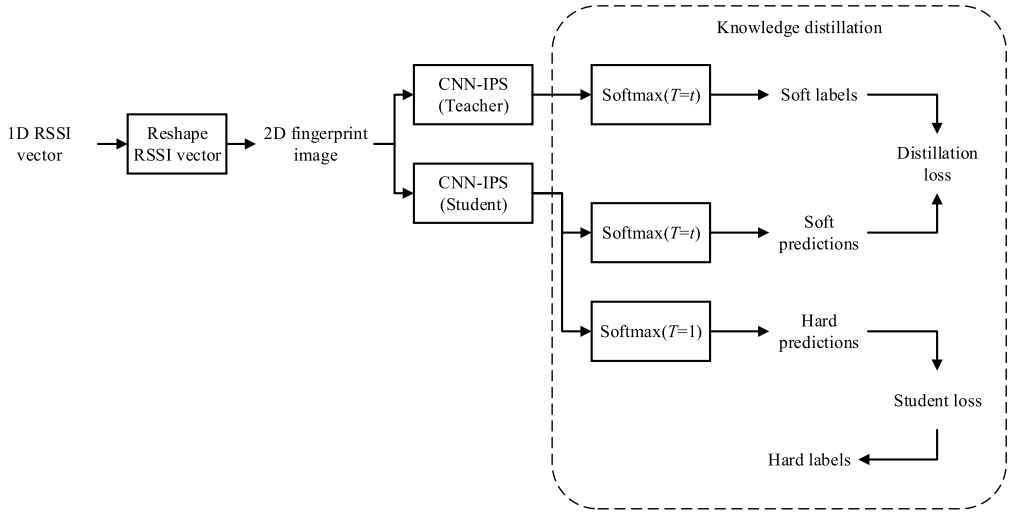


**FIGURE 2.** Block diagram of the proposed knowledge distillation framework for indoor positioning.

output of the softmax function that corresponds to the correct class will have a very high probability, while the probabilities for all other classes will be small. As a result, the information gained will be limited, and it might not be useful to train the student model. To address this issue, the logits of the teacher model could be scaled by a temperature parameter to enhance the relativeness of the outputs of different neurons in the last fully connected layer. Generally, higher temperatures will result in a softer probability distribution which makes the inter-class relationships become more detectable.

The same 2D input fingerprint images are also fed to the student network in parallel. Then, the student network will produce both the hard and soft outputs using (7) and (12), respectively. It is also important to highlight that the same set of temperature parameters will be used for both the teacher and student models to generate the soft labels. To train the student network, the following loss function [34] is adopted:

$$L = \alpha L_{\mathrm{CE}}\left(z_s^n, \boldsymbol{y}^n\right) + \beta L_{\mathrm{KD}}\left(z_s^n, z_t^n\right) \tag{13}$$

where $L_{\mathrm{CE}}\left(z_s^n, \boldsymbol{y}^n\right)$ represents the student loss (crossentropy loss between the predicted output and the ground truth label), $L_{\mathrm{KD}}\left(z_s^n, z_t^n\right)$ is the distillation loss, $\alpha$ and $\beta$ signify the user-defined weight factors that are utilized to control the importance of $L_{\mathrm{CE}}\left(z_s^n, \boldsymbol{y}^n\right)$ in relation to $L_{\mathrm{KD}}\left(z_s^n, z_t^n\right)$. Mathematically, $L_{\mathrm{KD}}\left(z_s^n, z_t^n\right)$ can be formulated as

$$L_{\mathrm{KD}}\left(z_s^n, z_t^n\right) = T^2 D_{\mathrm{KL}}\left(f_{\mathrm{TS-Softmax}}\left(z_s^n\right), f_{\mathrm{TS-Softmax}}\left(z_t^n\right)\right) \tag{14}$$

where $D_{\mathrm{KL}}(\psi, \xi) = \sum_{k=1}^{M} \psi_k \log\left(\psi_k / \xi_k\right)$ is the Kullback-Leibler (KL) divergence, $z_s^n$ and $z_t^n$ are the output logit vectors of the teacher and student models for the $n$th input sample, respectively. In this work, the weighted average between $L_{\mathrm{CE}}\left(z_s^n, \boldsymbol{y}^n\right)$ and $L_{\mathrm{KD}}\left(z_s^n, z_t^n\right)$ is used, i.e., $\alpha$ is configured as $1 - \beta$ and $\alpha \in [0, 1]$, and these weights could e set by the distillation designers based on their system's equirements. Since the objective function used to train the udent model is composed of a weighted average of $L_{\mathrm{CE}}\left(z_s^n, \boldsymbol{y}^n\right)$ and $L_{\mathrm{KD}}\left(z_s^n, z_t^n\right)$, this enables the student model to mimic the teacher's softened outputs while benefitting from the ground truth labels to achieve better accuracy.

## IV. EXPERIMENTAL ENVIRONMENTS AND DATASETS

To evaluate the effectiveness and the applicability of the proposed scheme in multi-floor environments, an indoor positioning testbed was set up, and an extensive fingerprint measurement campaign was carried out at the second and third floors of Wing B of the Faculty of Engineering (FOE), Multimedia University, on its campus in Cyberjaya, Malaysia. In what follows, this dataset will be abbreviated as the MMU-FOE dataset. Figures 3 and 4 show the floor layouts and the example pictures of the evaluation site, respectively. The dimension for each of the floors is 53.67 m $\times$ 10.36 m, and thus the total area of the evaluation site is approximately around 1112 $\mathrm{m}^2$. The spatial distribution of the pre-determined reference points is also illustrated in Figure 3. In total, 253 reference points were chosen for BLE fingerprint data collection, and the grid spacing between two adjacent reference points was 1 m. More specifically, the total number of reference points on the second and third floors are 125 and 128, respectively. At each reference spot, 50 fingerprint measurements were collected to capture the full variation of detectable RSSIs.

In this work, all the BLE beacons used were Sensoro SmartBeacon-4AA Pro. As shown in Figure 3, a total of 16 BLE beacons were employed, with 8 BLE beacons were placed on each floor. All the BLE beacons were attached to the wall with a placement height of 2 m from the floor in order to minimize the impact of the multipath. The transmission power and the advertising interval of the BLE beacons were configured as 4 dBm and 100 ms, respectively.

Measurement of the fingerprint was performed using a smart mobile device equipped with the Sensoro application. During the fingerprint measurement, the smartphone was

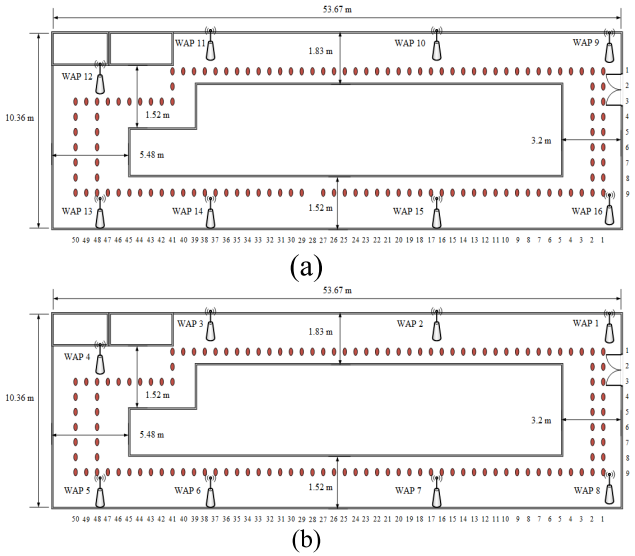


**FIGURE 3.** Layouts of the experimental environments (a) 2nd floor of MMU FOE building and (b) 3rd floor of MMU FOE building. The red dots indicate the reference points for the location fingerprints.

**FIGURE 4.** Example pictures of the experimental environments.

always held at the same orientation and maintained at the height of 1.5 m from the floor to follow the typical height and orientation when an indoor positioning user holds the smartphone in front of his/her chest. Besides that, for the sake of consistency, in order to reduce the people presence effect, measurements were conducted during holiday time.

The fingerprint database has 20 attributes, which consists of the measured RSSIs collected from 16 BLE beacons, the corresponding floor number and coordinates $(\tilde{x}_n, \tilde{y}_n, \tilde{z}_n)$. To be more specific, $\tilde{z}_n$ is the relative height of the reference point from the second floor. Since the floor-to-floor height from the second floor to the third floor is 4.01 m, the values of $\tilde{z}_n$ for all the reference points in the second and third floors are set as 0 and 4.01, respectively.

Figure 5 illustrates the example of input fingerprint images and the images produced by a different layer of CNN-IPS. The pixel's brightness depends on the recorded RSSI values, and a higher RSSI value will result in a brighter pixel.

## V. RESULTS AND ANALYSIS

In this section, the performance of the proposed scheme is extensively evaluated and benchmarked with those of the teacher and student counterparts by utilizing the MMU-FOE dataset as described in Section IV.

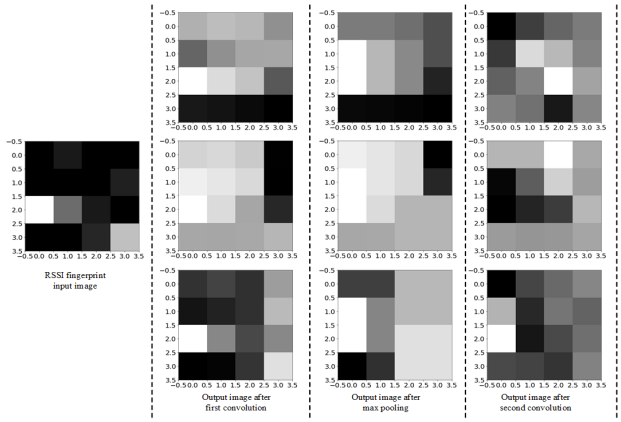


**FIGURE 5.** Example images of the fingerprint data.

### A. SIMULATION SETTING

The simulations are performed using Python 3.7.12. To establish the deep learning models, the powerful Keras 2.7.0 is adopted. The measured RSSIs of the MMU-FOE dataset range from −100 dBm to −57 dBm, and these RSSI values are normalized before they are processed by the machine learning-based positioning techniques. Throughout our simulations, the dataset is randomly split into 70% for training, and the remaining data is used for testing. A complex model of CNN-IPS is employed as a teacher model, and it is termed CNN-IPS (TM). Besides that, we also consider another two compact models of CNN-IPS of different sizes that are abbreviated as CNN-IPS (M1) and CNN-IPS (M2). To evaluate the performance of the proposed KD-CNN-IPS, two knowledge distilled CNN models that are trained under the supervision of CNN-IPS (TM) are considered, namely the KD-CNN-IPS (M1) and KD-CNN-IPS (M2). In Table 1, the detailed configuration of the hyperparameters for various techniques considered are summarized. Note that the architecture and configuration of KD-CNN-IPS (M1) and KD-CNN-IPS (M2) are precisely the same as those of the CNN-IPS (M1) and CNN-IPS (M2), respectively.

To provide a comprehensive analysis, the following evaluation metrics are used to quantify the performance of various machine learning techniques considered:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{15}$$

$$\text{precision} = \frac{TP}{TP + FP} \tag{16}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{17}$$

where $TP$, $TN$, $FP$, and $FN$ denote the true positive, true negative, false positive, and false negative, respectively. Since the proposed IPS is a multi-class problem, $TP$, $TN$, $FP$, and $FN$ are calculated using the average over all possible classes.

In addition, both the execution time and the average positioning error will also be examined. Execution time is an essential consideration when designing data-driven indoor

**TABLE 1.** Configuration of various models considered.

| Technique | CNN-IPS (TM) | CNN-IPS (M1) and KD-CNN-IPS (M1) | CNN-IPS (M2) and KD-CNN-IPS (M2) |
|---|---|---|---|
| No of epoch | 100 | 30 | 30 |
| Input size | 4 × 4 | 4 × 4 | 4 × 4 |
| Convolution layer | Filters : 32<br>Kernel size : (2, 2)<br>Strides : (1, 1)<br>Padding : same | Filters : 16<br>Kernel size : (2, 2)<br>Strides : (1, 1)<br>Padding : same | Filters : 4<br>Kernel size : (2, 2)<br>Strides : (1, 1)<br>Padding : same |
| Activation layer | Type : LReLU<br>$\alpha$ : 0.2 | Type : LReLU<br>$\alpha$ : 0.2 | Type : LReLU<br>$\alpha$ : 0.2 |
| Pooling layer | Type : MaxPooling2D<br>Pool size : (2, 2)<br>Strides : (1, 1)<br>Padding : same | Type : MaxPooling2D<br>Pool size : (2, 2)<br>Strides : (1, 1)<br>Padding : same | Type : MaxPooling2D<br>Pool size : (2, 2)<br>Strides : (1, 1)<br>Padding : same |
| Convolution layer | Filters : 64<br>Kernel size : (1, 1)<br>Strides : (1, 1)<br>Padding : same | Filters : 32<br>Kernel size : (1, 1)<br>Strides : (1, 1)<br>Padding : same | Filters : 8<br>Kernel size : (1, 1)<br>Strides : (1, 1)<br>Padding : same |

**TABLE 2.** Localization performance comparison of CNN-IPS and the conventional indoor positioning methods during the testing phase.

| Technique | Accuracy | 2D average positioning error | 3D average positioning error |
|---|---|---|---|
| Gaussian Naïve Bayes | 0.4360 | 2.0102 | 2.0102 |
| Decision tree | 0.5291 | 1.9439 | 2.0092 |
| K-nearest neighbor | 0.5619 | 1.5152 | 1.5178 |
| CNN-IPS | 0.5888 | 1.2403 | 1.2496 |

positioning approaches as it will affect user experience, and fast response time is desired to provide real-time indoor localization. The average positioning errors are measured using the Euclidean distance between the estimated location and the ground truth. More explicitly, two types of average positioning errors are considered, namely the 3D and the 2D average positioning errors, defined by (18) and (19), respectively.

$$e_{3D} = \frac{1}{N_1}\sum_{\eta=1}^{N_1}\sqrt{\left(\tilde{x}_\eta - \hat{x}_\eta\right)^2 + \left(\tilde{y}_\eta - \hat{y}_\eta\right)^2 + \left(\tilde{z}_\eta - \hat{z}_h\right)^2} \tag{18}$$

$$e_{2D} = \frac{1}{D}\sum_{d=1}^{D}\sqrt{\left(\tilde{x}_d - \hat{x}_d\right)^2 + \left(\tilde{y}_d - \hat{y}_d\right)^2} \tag{19}$$

where $N_1$ is the total number of test samples, $D$ represents the total number of test samples in which both the building and floor are correctly predicted, $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ and $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ are the $i$th actual and predicted coordinates, respectively.

### B. SIMULATION RESULTS AND DISCUSSION

Table 2 shows the localization performance of CNN-IPS and the conventional indoor positioning schemes during the testing phase. Three conventional indoor positioning techniques are used for performance benchmarking, namely the Gaussian Naive Bayes, decision tree, and K-nearest neighbor (KNN). As anticipated, CNN-IPS exhibits higher accuracy and lower positioning errors compared to the conventional approaches due to its capability to exploit the overall topology of fingerprinting images and disentangle highly abstract features. It is also evident in [21]–[23] that CNN-IPS also attains higher positioning accuracy compared to DNN-based IPS.

In terms of storage, the KNN technique imposes high storage requirements for the testing phase as it needs to store the entire training dataset in order to predict the user's location. Thus, it is impractical to implement memory-hungry KNN on resource-limited mobile and embedded devices. On the other hand, for all the non-lazy learners considered (Gaussian Naive Bayes, decision tree, and CNN), the execution storage space required is much smaller than that of the training as only the learnable parameters need to be stored once the machine learning models are trained. More specifically, the strict minimum memory requirement for the Gaussian Naive Bayes is to store the prior probability and probability distributions of features for each class, while the decision tree needs to keep the threshold information. As for the convolutional neural network, the weights and the biases of the network will be stored. This comparison shows that CNN-IPS is excellent in terms of positioning accuracy, but its complex architecture may be the main hurdle for practical implementation on resource-constrained portable devices.

**TABLE 3. Classification performance comparison of CNN-IPS (TM), CNN-IPS (M1) and CNN-IPS (M2).**

| Performance metric | CNN-IPS (TM) | CNN-IPS (M1) | CNN-IPS (M2) |
|---|---|---|---|
| Training accuracy | 0.6976 | 0.5426 | 0.3613 |
| Training loss | 0.8689 | 1.4021 | 2.1582 |
| Training time (s) | 219.77 | 58.72 | 25.28 |
| Testing accuracy | 0.5888 | 0.4862 | 0.3400 |
| Testing loss | 1.5074 | 1.7056 | 2.4916 |
| Testing time (s) | 0.5399 | 0.4542 | 0.4074 |
| Testing precision | 0.6454 | 0.5334 | 0.3520 |
| Testing recall | 0.5969 | 0.4859 | 0.3461 |
| Testing F1 score | 0.5709 | 0.4507 | 0.2852 |

Table 3 provides an insight into the performance of the CNN-IPS (TM), CNN-IPS (M1), and CNN-IPS (M2) in terms of accuracy, loss, precision, recall, and execution time. As anticipated, the size and structure of CNN significantly influence the localization performance. More precisely, in terms of accuracy, precision, recall, CNN-IPS (TM) is the best performer while CNN-IPS (M2) attains the worst performance. For instance, in comparison with the CNN-IPS (TM), the accuracy for CNN-IPS (M1) and CNN-IPS (M2) degrades by 22.22% and 48.21%, respectively. These performance trends could be explained as follows. Since the three techniques considered here are based on CNN, as long as the models are not overfitted, the models which are equipped with a more complex network structure not only tend to possess a better ability to disentangle the underlying complex nonlinear relationship between the input RSSIs of the BLE signals and the user positions, but also they would generally have a better capability to progressively capture a higher-level representation of the input RSSIs. As a result, CNN-IPS (TM) attains the lowest training and testing losses. However, the excellent performance of a more complex CNN model is achieved at the expense of higher computational complexity and longer execution time. As evident in Table 3, CNN-IPS (TM) has the longest training and testing time, while CNN-IPS (M2) attains the shortest execution time for both the training and testing phases.

Having studied the impacts of the network size of the CNN on the localization performance, in the following, we investigate the effects of the hyperparameters on the proposed KD-CNN-IPS scheme. Figures 6 and 7 present the 3D and 2D positioning errors for various techniques under investigation during the testing phase as a function of temperature, respectively. More explicitly, the temperature of the distilled student networks is varied from 2 to 20 and $\alpha$ is set as 0.1. From the figures, it is observed that the techniques under consideration exhibit the following increasing order of 3D and 2D average positioning errors: CNN-IPS (TM), KD-CNN-IPS (M1), CNN-IPS (M1), KD-CNN-IPS (M2), and CNN-IPS (M2). Besides that, as expected, the average positioning errors of CNN-IPS (TM), CNN-IPS (M1), and CNN-IPS (M2) are independent of the temperature, and

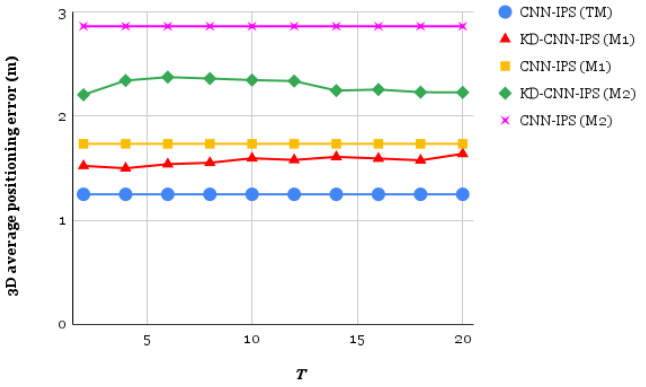


**FIGURE 6. 3D average positioning error of CNN-IPS (TM), KD-CNN-IPS (M1), CNN-IPS (M1), KD-CNN-IPS (M2), and CNN-IPS (M2) during the testing phase with varying temperature and $\alpha = 0.1$ used on the distilled student network.**

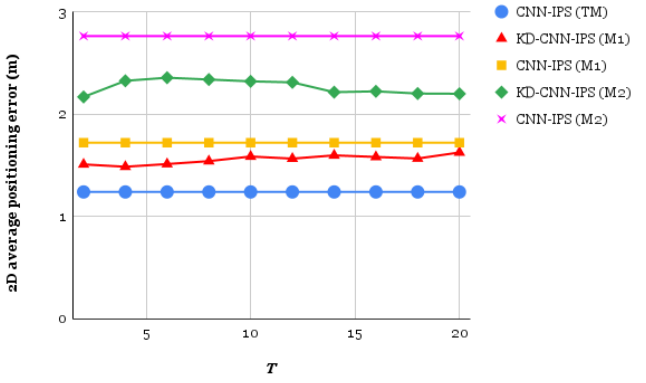


**FIGURE 7. 2D average positioning error of CNN-IPS (TM), KD-CNN-IPS (M1), CNN-IPS (M1), KD-CNN-IPS (M2), and CNN-IPS (M2) during the testing phase with varying temperature and $\alpha = 0.1$ used on the distilled student network.**

the distilled student networks outperform their counterparts without KD. Quantitatively, KD-CNN-IPS (M1) achieves a performance improvement of 13.54% and 13.65% over the CNN-IPS (M1) counterpart in terms of the 3D and 2D average positioning errors, respectively. On the other hand, the reductions in terms of the 3D and 2D positioning errors for the KD-CNN-IPS (M2) over that of the CNN-IPS (M2) counterpart are 22.62% and 20.86%, respectively. The performance advantage of the proposed KD-CNN-IPS over the CNN-IPS with identical architecture and size in terms of average positioning error is due to the reason that KD-CNN-IPS is trained by both the ground truths and the teacher model, while the CNN-IPS counterpart is only trained by the ground truth.

In addition, it is also noteworthy that the performance of the proposed KD-CNN-IPS scheme is insensitive to the temperature when the temperature is varied from 2 to 20 as only a minimal variation in terms of the average positioning errors is noticed. As can be seen from the figures, KD-CNN-IPS (M1) and KD-CNN-IPS (M2) attain the best results when the temperature is configured as 4 and 20, respectively, as these settings could provide the best tradeoff between the relativeness and distinctiveness of the network

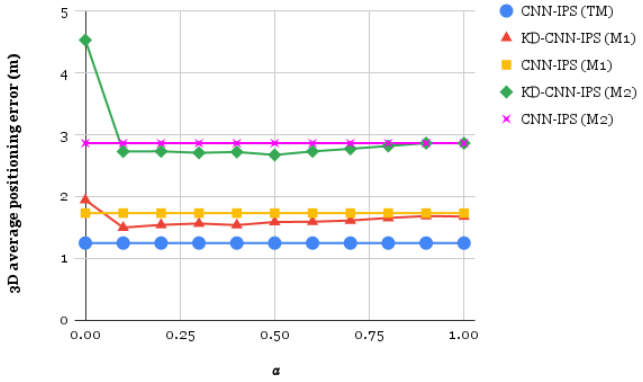


**FIGURE 8.** 3D average positioning error of CNN-IPS (TM), KD-CNN-IPS (M1), CNN-IPS (M1), KD-CNN-IPS (M2), and CNN-IPS (M2) for $T = 4$ and different $\alpha$ during the testing phase.

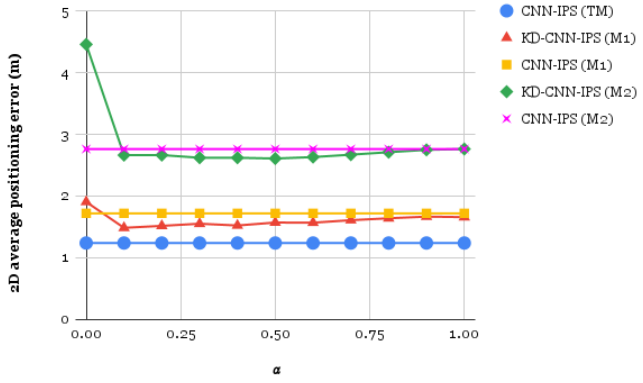


**FIGURE 9.** 2D average positioning error of CNN-IPS (TM), KD-CNN-IPS (M1), CNN-IPS (M1), KD-CNN-IPS (M2), and CNN-IPS (M2) for $T = 4$ and different $\alpha$ during the testing phase.

outputs. To be more precise, a higher value of $T$ could provide more helpful information on the inter-class relations at the expense of the distinctiveness of the outputs and vice versa.

To gain further insight into the performance of the knowledge distilled student network, Figures 8 and 9 characterize the 3D and 2D average positioning errors during the testing phase for varying values of $\alpha$, respectively. To ease the comparison to the baseline models, the performance of CNN-IPS (TM), CNN-IPS (M1), and CNN-IPS (M2) are included as well. A detailed examination on Figures 8 and 9 reveals that the schemes under consideration demonstrate similar performance trends as those observed in Figures 6 and 7, i.e., both the 3D and 2D average positioning errors increases in the following order: CNN-IPS (TM), KD-CNN-IPS (M1), CNN-IPS (M1), KD-CNN-IPS (M2) and CNN-IPS (M2). Besides that, it is also observed that there is a substantial improvement in terms of the 3D and 2D average positioning errors of KD-CNN-IPS (M1) and KD-CNN-IPS (M2) when $\alpha$ is increased from 0 to 0.1. This is because $L = L_{\text{KD}}\left(z_s^n, z_t^n\right)$ when $\alpha = 0$. In this case, both the KD-CNN-IPS (M1) and KD-CNN-IPS (M2) are only trained using distillation loss, but they fail to exploit useful information gained from student loss. On the other hand, $\alpha$ does not have a noticeable

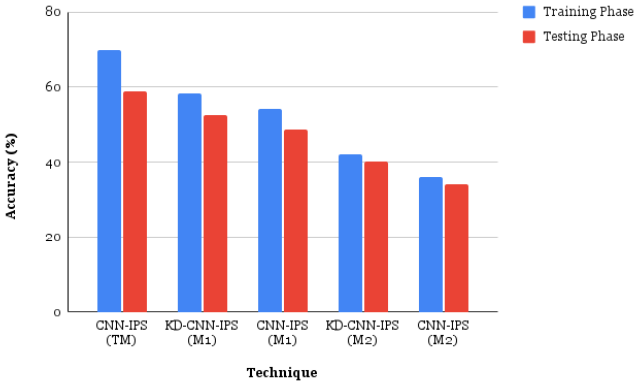


**FIGURE 10.** Accuracy of CNN-IPS (TM), KD-CNN-IPS (M1), CNN-IPS (M1), KD-CNN-IPS (M2), and CNN-IPS (M2) for both the training and testing phases.

impact on the 3D and 2D average positioning errors of the proposed KD-CNN-IPS schemes when $\alpha$ is varied from 0.1 to 1 as only a slight variation of performance is noticed. From the figures, KD-CNN-IPS (M1) and KD-CNN-IPS (M2) attain the best performance when $\alpha$ is set as 0.1 and 0.6, respectively, because these settings could effectively strike a balance between the contribution of $L_{\text{CE}}\left(z_s^n, \boldsymbol{y}^n\right)$ and $L_{\text{KD}}\left(z_s^n, z_t^n\right)$. Therefore, the aforementioned setting of $\alpha$ will be used for KD-CNN-IPS (M1) and KD-CNN-IPS (M2) in the following simulations.

Figure 10 illustrates the accuracy of various schemes considered for both the training and testing phases. From the figure, it is apparent that the highest accuracy is obtained by the CNN-IPS (TM), in which its accuracy for the training and testing phases are 69.76% and 58.88%, respectively. As expected, the accuracies of CNN-IPS (M1) and CNN-IPS (M2) are lower as compared to that of CNN-IPS (TM). When CNN-IPS (M1) and CNN-IPS (M2) are trained under the guidance of CNN-IPS (TM) via KD, an improvement in terms of accuracy is observed in comparison to the student counterparts. More precisely, for both the training and testing phases, the accuracy of KD-CNN-IPS (M1) is around 4% higher than that of the CNN-IPS (M1). On the other hand, it is observed that KD-CNN-IPS (M2) could attain around 6% of increment in terms of accuracy over CNN-IPS (M2) for both the training and testing phases. The good performance of KD-CNN-IPS (M1) and KD-CNN-IPS (M2) is due to their capability to preserve the high-quality performance of the complex teacher model by leveraging on the informative dark knowledge, i.e., softened probabilities gained from the teacher network.

As shown in Figure 11, the trends of the 3D and 2D average positioning errors for all compared techniques in both the training and testing phases are in good agreement with the observations in Figure 10. Since all the techniques considered here are based on classification predictive modeling, good accuracy in predicting the location class will lower average positioning error. Importantly, it is observed that the 3D and 2D average positioning errors of the proposed KD-CNN-IPS (M1) and KD-CNN-IPS (M2) are approximately 15.81% and

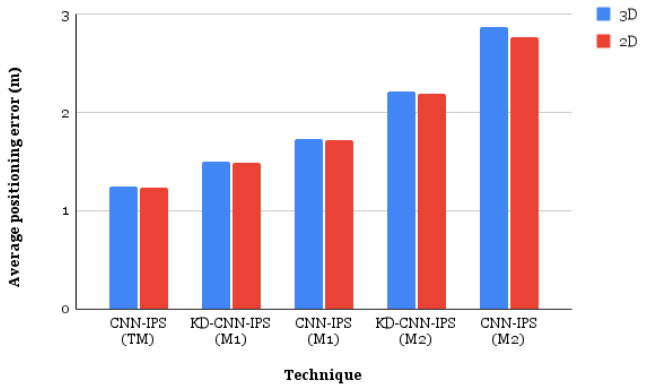


**FIGURE 11.** 3D and 2D average positioning errors for CNN-IPS (TM), KD-CNN-IPS (M1), CNN-IPS (M1), KD-CNN-IPS (M2), and CNN-IPS (M2) during the testing phase.

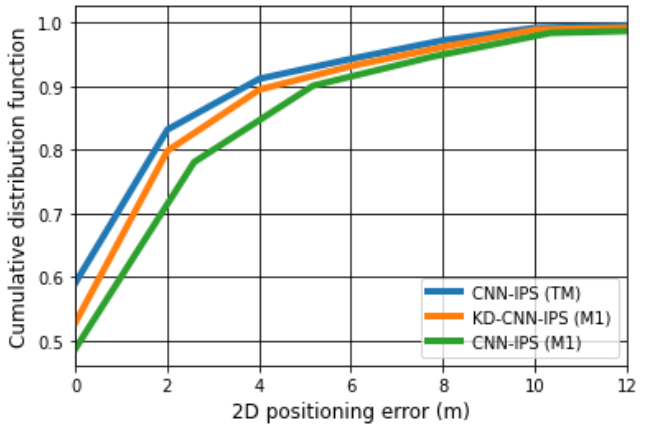


**FIGURE 12.** CDF of 2D positioning error for CNN-IPS (TM), KD-CNN-IPS (M1), and CNN-IPS (M1) during the testing phase.

26.35% lower than their student counterparts. Besides that, it is worth mentioning that the proposed KD-CNN-IPS (M1) only suffers from slight performance degradation in terms of average positioning errors as compared to the CNN-IPS (TM) as its 3D and 2D average positioning errors are only 0.25 m and 0.246 m higher than those of the CNN-IPS (TM). On the other hand, compared to the CNN-IPS (TM), the 3D and 2D average positioning of CNN-IPS (M1) is 0.4849 m and 0.481 m higher.

Figures 12-15 shed lights on distribution of the positioning errors for all compared methods using the cumulative distribution function (CDF). Generally, the results demonstrate that the proposed approach can achieve a higher probability of better localization than its counterpart without KD. Quantitatively, from Figures 12 and 13, it is noticed that the probability of positioning errors within 2 m by the proposed KD-CNN-IPS (M1) is 0.7984. In contrast, the ones for the cases of CNN-IPS (TM) and CNN-IPS (M1) are 0.8318 and 0.7709, respectively. Besides that, we can also observe from Figures 12 and 13 that nearly 89.47% of the positioning errors of KD-CNN-IPS (M1) are within 4 m. On the other hand, as can be seen from Figures 14 and 15, the probability of attaining less than 2 m of average positioning error by

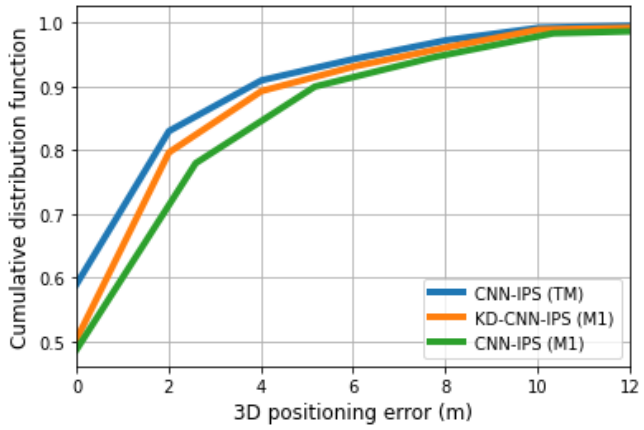


**FIGURE 13.** CDF of the 3D positioning error for CNN-IPS (TM), KD-CNN-IPS (M1), and CNN-IPS (M1) during the testing phase.

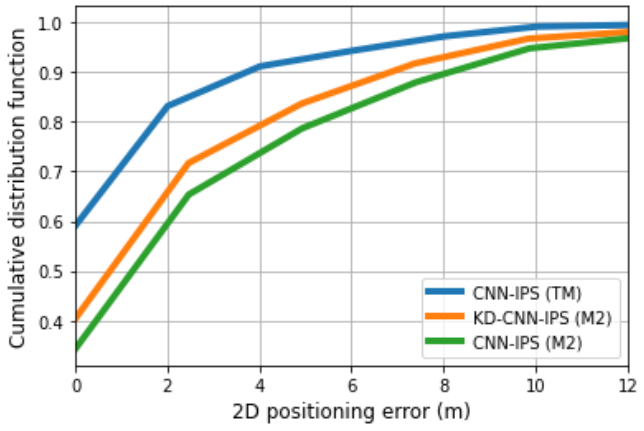


**FIGURE 14.** CDF of 2D positioning error for CNN-IPS (TM), KD-CNN-IPS (M2), and CNN-IPS (M2) during the testing phase.

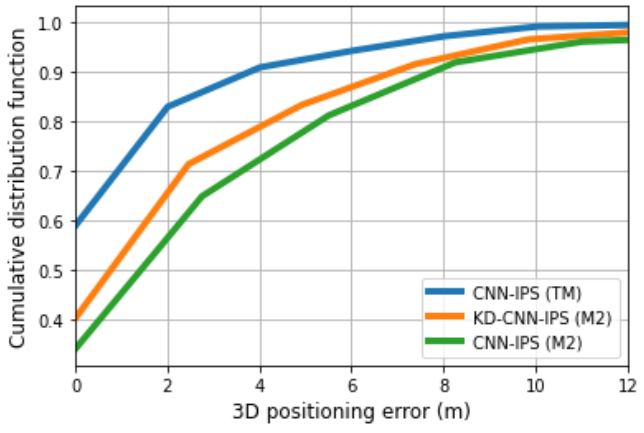


**FIGURE 15.** CDF of 3D positioning error for CNN-IPS (TM), KD-CNN-IPS (M2), and CNN-IPS (M2) during the testing phase.

KD-CNN-IPS (M2) is 70.26%, which is 6.36% higher than that of the CNN-IPS (M2).

Figures 16 shows the average testing time for various techniques considered. As evident in Figure 16, the average testing time of the proposed distilled student networks is

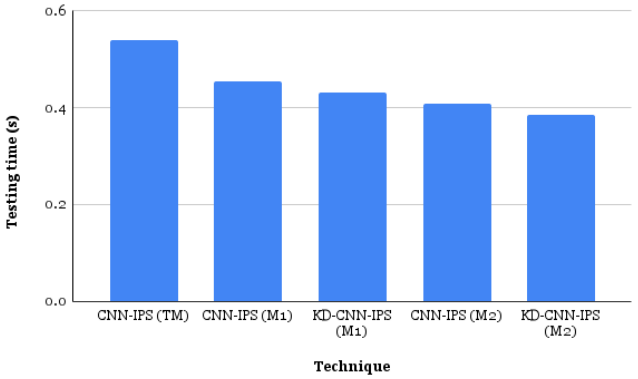


**FIGURE 16.** Testing time of CNN-IPS (TM), CNN-IPS (M1), KD-CNN-IPS (M1), CNN-IPS (M2) and KD-CNN-IPS (M2).

shorter than that of the teacher network. This is because the testing time incurred by the CNN is dictated by the structure and complexity of the model. To be more precise, a more complex CNN model will result in a longer testing time and vice versa. Quantitatively, the average testing time associated with the KD-CNN-IPS (M1) is only 79.68% of that of the CNN-IPS (TM), while for the KD-CNN-IPS (M2), the average testing time is 71.53% of that of the CNN-IPS (TM).

## VI. CONCLUSION

In this work, a novel fast indoor localization scheme based on a convolutional neural network is proposed. By leveraging on the knowledge distillation, the intricate mapping between the input RSSIs and location information, which has been disentangled and learned by the complex and powerful pre-trained teacher network, could be effectively distilled to a lightweight and low-complexity student model for positioning. To assess the effectiveness and practicality of the proposed method, an indoor localization testbed has been set up to collect a multi-floor indoor positioning dataset, and extensive simulations have been carried out. Further, an in-depth study of the effects of hyperparameters of proposed KD-CNN-IPS has also been presented. Numerical results reveal that the proposed KD-CNN-IPS is capable of attaining significant improvement in localization performance viz a viz the CNN with identical architecture and size while achieving a substantial reduction in testing time as compared to the complicated teacher network. Essentially, for KD-CNN-IPS (M1), 79.84% of the positioning errors are within 2 m, and its average testing time is only 79.68% of that of the teacher model. Compared to the student counterpart, the average positioning error of KD-CNN-IPS (M1) is approximately 15.81% lower. With these remarkable benefits, it can be concluded that the proposed KD-CNN-IPS is an appealing indoor positioning approach for real-time deployment on resource-constrained devices. The possible future direction of research may include investigating the performance of the KD-CNN-IPS in complex environments and extending KD-CNN-IPS to semi-supervised learning to fully exploit both the labeled and unlabeled location fingerprint data.

## REFERENCES

[1] S. Tomažič, "Indoor positioning and navigation," *Sensors*, vol. 21, no. 14, pp. 1–4, Jul. 2021, doi: 10.3390/s21144793.
[2] P. Mahida, S. Shahrestani, and H. Cheung, "Deep learning-based positioning of visually impaired people in indoor environments," *Sensors*, vol. 20, no. 21, pp. 2–17, Oct. 2020, doi: 10.3390/s20216238.
[3] *The Global Indoor Location Market Size to Grow From USD 7.0 Billion in 2021 to USD 19.7 Billion by 2026, at a Compound Annual Growth Rate (CAGR) of 22.9%*. Globe Newswire. Accessed: Nov. 7, 2021. [Online]. Available: https://www.globenewswire.com/news-release/2021/10/13/2313337/0/en/The-global-Indoor-location-market-size-to-grow-from-USD-7-0-billion-in-2021-to-USD-19-7-billion-by-2026-at-a-Compound-Annual-Growth-Rate-CAGR-of-22-9.html
[4] Y. Liu, Z. Yang, X. Wang, and L. Jian, "Location, localization, and localizability," *J. Comput. Sci. Technol.*, vol. 25, no. 2, pp. 274–297, 2010, doi: 10.1007/s11390-010-9324-2.
[5] C. Gentile, N. Alsindi, R. Raulefs, and C. Teolis, *Geolocation Techniques: Principles and Applications*, New York, NY, USA: Springer, 2013.
[6] A. Poulose, J. Kim, and D. S. Han, "A sensor fusion framework for indoor localization using smartphone sensors and Wi-Fi RSSI measurements," *Appl. Sci.*, vol. 9, no. 20, pp. 4–17, Oct. 2019, doi: 10.3390/app9204379.
[7] A. Poulose and D. S. Han, "Performance analysis of fingerprint matching algorithms for indoor localization," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2020, pp. 661–665, doi: 10.1109/ICAIIC48513.2020.9065220.
[8] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of WiFi based localization for smartphones," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw. (Mobicom)*, 2012, pp. 305–316, doi: 10.1145/2348543.2348581.
[9] W. Sun, J. Liu, C. Wu, Z. Yang, X. Zhang, and Y. Liu, "MoLoc: On distinguishing fingerprint twins," in *Proc. IEEE 33rd Int. Conf. Distrib. Comput. Syst.*, Jul. 2013, pp. 226–235, doi: 10.1109/ICDCS.2013.41.
[10] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Non-line-of-sight identification and mitigation using received signal strength," *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, pp. 1689–1702, Mar. 2015, doi: 10.1109/TWC.2014.2372341.
[11] L. M. Ni, Y. Liu, Y. Cho Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun. (PerCom).*, Mar. 2003, pp. 407–415, doi: 10.1109/PERCOM.2003.1192765.
[12] N. Swangmuang and P. Krishnamurthy, "Location fingerprint analyses toward efficient indoor positioning," in *Proc. 6th Annu. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2008, pp. 100–109, doi: 10.1109/PERCOM.2008.33.
[13] *IEEE 802.11ax: The Sixth Generation of WiFi White Paper*. CISCO. Accessed: Oct. 17, 2021. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/wireless/white-paper-c11-740788.html
[14] E. Au, "Bluetooth 5.0 and beyond [standards]," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 119–120, Jun. 2019, doi: 10.1109/MVT.2019.2905520.
[15] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: Mobile phone localization via ambience fingerprinting," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2009, pp. 261–272, doi: 10.1145/1614320.1614350.
[16] P. Torteeka and X. Chundi, "Indoor positioning based on Wi-Fi fingerprint technique using fuzzy K-nearest neighbor," in *Proc. 11th Int. Bhurban Conf. Appl. Sci. Technol. (IBCAST) Islamabad, Pakistan, 14th 18th January*, Jan. 2014, pp. 461–465, doi: 10.1109/IBCAST.2014.6778188.
[17] C. Figuera, J. L. Rojo-Álvarez, M. Wilby, I. Mora-Jiménez, and A. J. Caamaño, "Advanced support vector machines for 802.11 indoor location," *Signal Process.*, vol. 92, no. 9, pp. 2126–2136, Sep. 2012, doi: 10.1016/j.sigpro.2012.01.026.
[18] X. Guo, N. Ansari, L. Li, and H. Li, "Indoor localization by fusing a group of fingerprints based on random forests," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4686–4698, Dec. 2018, doi: 10.1109/JIOT.2018.2810601.
[19] M. Ibrahim, M. Torki, and M. ElNainay, "CNN based indoor localization using RSS time-series," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 1044–1049, doi: 10.1109/ISCC.2018.8538530.
[20] H. Lu, G. Xingli, L. Shuang, Z. Heng, L. Yaning, and Z. Ruihui, "Indoor positioning technology based on deep neural networks," in *Proc. Ubiquitous Positioning, Indoor Navigat. Location-Based Services (UPINLBS)*, Mar. 2018, pp. 1–6, doi: 10.1109/UPINLBS.2018.8559721.

[21] J.-W. Jang and S.-N. Hong, "Indoor localization with WiFi fingerprinting using convolutional neural network," in *Proc. 10th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2018, pp. 753–758, doi: 10.1109/ICUFN.2018.8436598.
[22] X. Song, X. Fan, X. He, C. Xiang, Q. Ye, X. Huang, G. Fang, L. L. Chen, J. Qin, and Z. Wang, "CNNLoc: Deep-learning based indoor localization with WiFi fingerprinting," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2019, pp. 589–595, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00139.
[23] A. Mittal, S. Tiku, and S. Pasricha, "Adapting convolutional neural networks for indoor localization with smart mobile devices," in *Proc. Great Lakes Symp. (VLSI)*, May 2018, pp. 117–122, doi: 10.1145/3194554.3194594.
[24] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network—A deep learning approach," *Proc. Comput. Sci.*, vol. 132, pp. 679–688, Jan. 2018, doi: 10.1016/j.procs.2018.05.069.
[25] A. Alkhulaifi, F. Alsahli, and I. Ahmad, "Knowledge distillation in deep learning and its applications," *Peer J. Comput. Sci.*, vol. 7, no. 474, pp. 1–24, Apr. 2021, doi: 10.7717/peerj-cs.474.
[26] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distil-lation: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1789–1817, Mar. 2021, doi: 10.1007/s11263-021-01453-z.
[27] P. Chen, S. Liu, H. Zhao, and J. Jia, "Distilling knowledge via knowledge review," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5006–5015, doi: 10.1109/CVPR46437.2021.00497.
[28] W. Ahmed, A. Zunino, P. Morerio, and V. Murino, "Compact CNN structure learning by knowledge distillation," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6554–6561, doi: 10.1109/ICPR48806.2021.9413006.
[29] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. 31st Int. Conf. Neural Inf. Process (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 742–751.
[30] D. Sun, E. Wei, L. Yang, and S. Xu, "Improving fingerprint indoor localization using convolutional neural networks," *IEEE Access*, vol. 8, pp. 193396–193411, 2020, doi: 10.1109/ACCESS.2020.3033312.
[31] D. Yu, H. Wang, P. Chen, and Z. Wei, "Mixed pooling for convolutional neural networks," in *Proc. Int. Conf. Rough Sets Knowl. Technol.*, Oct. 2014, pp. 364–375, doi: 10.1007/978-3-319-11740-9_34.
[32] 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, pp. 1–20, Jul. 2019, doi: 10.3390/s19143127.
[33] G. Hinton, O. Vinyals, and J. Deans, "Distilling the knowledge in a neural network," in *Proc. Deep. Learn. Represent. Learn. Workshop (NIPS)*, 2015, pp. 1–9.
[34] A. Ozerov and N. Q. K. Duong, "Inplace knowledge distillation with teacher assistant for improved training of flexible deep neural networks," in *Proc. 29th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2021, pp. 1356–1360, doi: 10.23919/EUSIPCO54536.2021.9616244.

**AQILAH BINTI MAZLAN** was born in Perak, Malaysia, in 1999. She is currently pursuing the degree in electronics engineering with the Faculty of Engineering, Multimedia University, Malaysia. Her research interests include machine learning and indoor positioning.

**YIN HOE NG** received the B.Eng. degree (Hons.) in electronics engineering, the M.Eng.Sc. and Ph.D. degrees from Multimedia University in 2004, 2008, and 2013, respectively. He is currently a Senior Lecturer with the Faculty of Engineering, Multimedia University. He is also a Chartered Engineer (C.Eng.) and a Professional Technologist (P.Tech.) registered with the Engineering Council United Kingdom and the Malaysia Board of Technologists, respectively. His current research interests include advanced signal processing techniques for digital communication systems, machine learning, and indoor positioning.

**CHEE KEONG TAN** (Member, IEEE) received the B.Eng. degree in electronics (telecommunication), the M.Eng.Sc. degree in information, communication and technology and the Ph.D. degree in information, communication and technology from Multimedia University, Malaysia. He is currently a Senior Lecturer with the School of Information Technology, Monash University Malaysia. He has carried out projects for telecommunication companies and cellular service providers, which led to the development of a few patents on wireless algorithm and protocol design. He is the main contributing authors to more than 20 international journal articles. His current research interests include radio resource management, 5G networks, indoor positioning scheme, game theory, machine learning, and artificial intelligence.