electronics

Article

Indoor 3D Localization Scheme Based on BLE Signal Fingerprinting and 1D Convolutional Neural Network

Shangyi Yang, Chao Sun and Youngok Kim *

Electronic Engineering Department, Kwangwoon University, Seoul 01897, Korea; shangyigoo@gmail.com (S.Y.); sunchao2601@gmail.com (C.S.)
* Correspondence: kimyoungok@kw.ac.kr

Abstract: Indoor localization schemes have significant potential for use in location-based services in areas such as smart factories, mixed reality, and indoor navigation. In particular, received signal strength (RSS)-based fingerprinting is used widely, given its simplicity and low hardware requirements. However, most studies tend to focus on estimating the 2D position of the target. Moreover, it is known that the fingerprinting scheme is computationally costly, and its positioning accuracy is readily affected by random fluctuations in the RSS values caused by fading and the multipath effect. We propose an indoor 3D localization scheme based on both fingerprinting and a 1D convolutional neural network (CNN). Instead of using the conventional fingerprint matching method, we transform the 3D positioning problem into a classification problem and use the 1D CNN model with the RSS time-series data from Bluetooth low-energy beacons for classification. By using the 1D CNN with the time-series data from multiple beacons, the inherent drawback of RSS-based fingerprinting, namely, its susceptibility to noise and randomness, is overcome, resulting in enhanced positioning accuracy. To evaluate the proposed scheme, we developed a 3D positioning system and performed comprehensive tests, whose results confirmed that the scheme significantly outperforms the conventional common spatial pattern classification algorithm.

Keywords: 1D conventional neural network; 3D localization; indoor positioning; BLE signal

check for updates

Citation: Yang, S.; Sun, C.; Kim, Y. Indoor 3D Localization Scheme Based on BLE Signal Fingerprinting and 1D Convolutional Neural Network. *Electronics* **2021**, *10*, 1758. https://doi.org/10.3390/electronics10151758

Academic Editor: Martin Reisslein

Received: 29 June 2021
Accepted: 20 July 2021
Published: 22 July 2021

1. Introduction

Indoor localization schemes, which are also termed as positioning schemes, have received significant attention recently because of their potential for use in areas such as smart factories, mixed reality, indoor navigation, security and advertising services [1]. In general, indoor localization can provide the following benefits: better user experience for navigating indoor spaces, where GPS is not practical; enabling smart building operations and enhancements; improving the efficiency of robots or unmanned aerial vehicles (UAVs) in smart factories, and allowing users to find equipment with ease [2,3]. However, most studies so far have focused on estimating the two-dimensional (2D) position of the target. However, it is also essential to determine the three-dimensional (3D) position, that is, the height as well, of the target (for instance, the height of a robot's arm or UAV in a smart factory, height of equipment in a building, and height of security features).

Most current indoor localization technologies are based on time, angle, or electromagnetic wave data; these are also known as the time of arrival, angle of arrival and received signal strength (RSS) schemes, respectively [4,5]. Among the various indoor localization schemes available, the RSS-based scheme is used widely owing to its simplicity and low hardware requirements. RSS-based methods can be divided into two categories: those involving trilateration based on the range estimated from the RSS values and the fingerprinting method, which is based on RSS fingerprint matching. Considering the variations in the RSS values in indoor spaces, however, it is difficult to accurately determine the distance information from the RSS data.

Previous studies on indoor localization suggest that several methods use the fingerprint matching scheme as the basic scheme for the localization of the target. The main idea is to first build a fingerprint database that collects the surrounding signatures at every predefined location in the areas of interest. Subsequently, the position of the target is estimated by matching the measured fingerprint with the database. Many researchers have striven to exploit the RSS value signatures for RSS-based fingerprinting techniques owing to the simplicity and low hardware requirements of the process. The first fingerprinting method based on a Wi-Fi device was introduced in [6]. The authors determined the fingerprints of the RSS value and then used a deterministic method, namely, the k-nearest neighbors technique, for position estimation. Subsequently, RSS measurements from other transmitter devices, such as RFIDs [7], Zigbee [8] and Bluetooth devices [9], have also been used for localization based on the fingerprinting technique. Moreover, classical machine learning methods such as the support vector machine model have been employed with the RSS fingerprinting technique [10]. In [11], a probabilistic Bayesian method was introduced to determine the difference between the test and saved RSS values. The main challenge in the case of RSS-based fingerprinting localization methods is that their positioning accuracy is readily affected by the random fluctuations in the RSS values caused by fading and the multipath effect. In addition, the complexity of the matching algorithm as well as that of classical machine learning algorithms increases significantly as the number of positions to be estimated increases. Thus, one requires more storage space and additional computing resources.

With the emergence of graphical processing units (GPUs), the convolutional neural network (CNN) became a focus of research interest again in 2012 [12]. Significant advances were made in image processing through the development of CNNs such as AlexNet, Zfnet, and GoogLeNet [13,14]. Since CNNs exhibit improved performance by extracting more features from raw data during image classification, many researchers have tried to use them with one-dimensional (1D) signals such as temporal signals. Moreover, a 1D CNN was developed recently to reduce the computational complexity for 1D signals [15,16].

In this paper, we propose an indoor 3D localization scheme based on both fingerprinting and a 1D CNN. Instead of using the conventional fingerprint matching method, in the proposed scheme, the 3D positioning problem is transformed into a classification problem, and a 1D CNN model that uses the RSS time-series data from Bluetooth low-energy (BLE) beacons is used for classification. The contributions of this study can be summarized as follows:

1. We propose an indoor 3D localization scheme based on both fingerprinting and a 1D CNN. While most of the studies so far have focused on estimating the 2D positional information of the target, we propose a 3D localization scheme based on the fingerprinting technique. We convert the 3D positioning problem into a classification problem by dividing the 3D space into a set of unit cubic grids and process the RSS time-series BLE signal as a 1D signal in order to solve the localization problem using a 1D CNN.
2. We develop a 3D positioning system, which consists of BLE beacons and a Raspberry Pi receiver, for evaluating the performance of the proposed scheme. Using the developed system, time-series RSS data are collected from the beacons at each location, and the collected data are divided into training and testing datasets for the 1D CNN model.
3. We evaluate the performance of the proposed scheme through comprehensive tests. First, the convergence and accuracy of the 1D CNN scheme are evaluated, and the effects of data preprocessing and the kernel size on the proposed scheme are investigated. Next, we compare the classification accuracy of the proposed scheme with that of the conventional common spatial pattern (CSP) algorithm.

The remainder of this paper is organized as follows. In Section 2, we briefly review some relevant literatures and compare their strength and weakness. In Section 3, we introduce the characteristics of the BLE signal and describe the developed 3D localization system. In Section 4, the indoor 3D localization scheme based on both fingerprinting and a 1D CNN is proposed. The performance of the proposed scheme is evaluated through tests, whose results are described in Section 5. Finally, the conclusions of this study are summarized in Section 6. Note that abbreviation and description in this manuscript are summarized in Appendix A (see Table A1).

2. Related Works

In indoor fingerprint positioning, it is more common to use WI-FI signal as the measurements. Baoqi Huang et al. introduced an eight-layers DNN model for indoor Wi-Fi signal fingerprinting localization in [17]. In signal processing, they utilized stacked auto encoders to extract representative features from the collected data and defined a special loss function to train DNN. However, positioning errors were beyond 2 m in indoor environment. In [18], Yifan Wang et al. developed a Wi-Fi fingerprint location recognition DNN method based on geometric distribution of fingerprint points to estimate the user's position, and then exploited the constrained Kalman filter algorithm and the hidden Markov model to optimize the final results. However, this fuse method led to complexity of algorithm and has a large error in the specific location. Although Wi-Fi signals have the physical characteristics like high transmit power, more bandwidth, and large coverage, the cheaper and more low-power BLE sensor is used in related fields. In [19], Charu Jain et al. compared a variety of machine learning methods based on BLE signal in solving floor classification problem. However, they cannot cover the more developed neural network. Since University of Toronto Geoffrey E. Hinton's research group won the championship through the constructed CNN network, AlexNet, in the 2012 ImageNet Large-Scale Visual Recognition Challenge [13], CNN has not only attracted the attention of many researchers in the image field, but also it keeps going to conquer the battlefield in other fields. In [20], Danshi Sun et al. utilized CNN to achieve positioning in multi-floor large-area indoor environment. They converted the BLE RSS value into "fingerprint image" to train the 2D CNN and to predict floor categories, and then combining with magnetic field data to locate the transmitter's position. In [21], they carried out a PSO-aided 2D CNN architecture for the indoor positioning system, in which PSO is used for optimal parameter selection. However, both of them restructured the time-series BLE signal vector into image-liked matrix. This will lead to high time complexity and unwarranted execution time of the localization method. In [22], Kodai Tasaki et al. developed a 3D CNN based on BLE RSS value to against statistically fluctuated signal due to random wireless channels. By taking the spatiotemporal structure of RSSI data set into consideration, the results showed a good result in fingerprinting than 2D CNN. However, it needs a spatial correlation for all the obtained RSS values, and special and complex operations in constructing the input dataset.

For indoor localization, the information of height is also important factor in industrial or commercial scenarios, the above-mentioned articles work well in floor and 2D positioning, but they did not focus on much 3D position information. In addition, due to the temporal feature of collecting BLE signal data, it is necessary to develop a 1D CNN to fully take advantage of the temporal signal. In this paper, we use the 1D CNN algorithm to accomplish the 3D spatial position classification problem, which will achieve more accurate and efficient positioning. In addition, this method has excellent positioning accuracy on small spatial environment.

3. System Description

3.1. Received Signal Strength (RSS) of BLE Signal

The received power of a BLE signal in indoor environments decreases as the distance of propagation increases. Therefore, the RSS is reflective of the distance between the transmitter and receiver. In conventional RSS-based ranging schemes, the log-normal path loss model is generally used, and the RSS value is expressed as follows [23]:

$$\mathrm{RSS[dBm]} = 10 \times log\frac{P_r}{P_{ref}}, \tag{1}$$

where P_{ref} is the reference power, P_r is the received power (calculated as $P_r = P_t \times G_t \times G_r \times (\lambda/4\pi d)^2$), and P_t is the transmitted power. G_t and G_r denote the antenna gains of the transmitter and receiver, respectively. λ is the wavelength of the radio wave and d is the distance between the transmitter and receiver.

When the RSS value at 1 m is used as the reference power value, we can calculate the distance, d, from the current RSS value as follows [24]:

$$\mathrm{d[m]} = 10^{(P_0 - RSS)/10n} \tag{2}$$

where P_0 is the RSS value at 1 m, and n is the path loss parameter, which is different in various spaces: 1.4–1.9 for corridors and 2 for large open rooms [25]. Figure 1 shows the results of our real experiments to investigate the changes in the RSS value with the travelled distance, which will be the basic characteristic element that we will use to distinguish the different positions. As can be seen from the figure, the RSS values of the two beacons are not the same, even though both beacons are of the same model and from the same manufacturer.

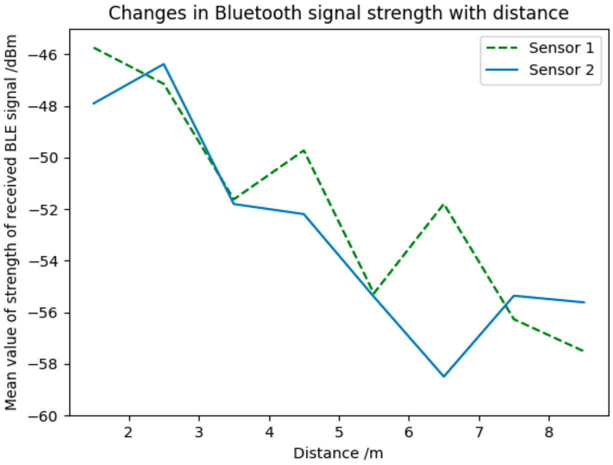

Figure 1. Changes in BLE RSS value with distance during experiments.

3.2. Developed 3D Localization System

Owing to the presence of shadow areas and the multipath effect in indoor environments, the deployment of BLE beacons affects the signal reception, which, in turn, affects the positioning accuracy. According to the layout principle proposed in [26], a spacing of 4–6 m is optimal for Bluetooth placement plans when no obstacle exists between the transmitter and receiver. Therefore, we arranged 8 BLE beacons at the top corners of a 3D space with dimensions of 4.0 m × 2.0 m × 3.0 m (length × width × height) in a general office building in our university. Figure 2 shows a schematic of the 3D localization system, including the coordinates and deployment locations of the of BLE beacons.

Figure 3 shows an overview of the data flow for the developed 3D positioning system. In this system, we used a third-generation Raspberry Pi (Model B) as the BLE signal receiver and data uploader. The entire process was divided into three stages. First, the BLE signal receiver scans the beacon signals in the surroundings and extracts the RSS value of the received packet from the beacon frame and beacon ID. Once the RSS value has been determined, the Raspberry Pi functions as the gateway device to upload the extracted data to the local database. A timestamp corresponding to the moment of reception is also added to update its own real-time clock from the network time protocol server over the Wi-Fi router device. Finally, the data are used to train the neural network and predict the 3D position of the receiver.

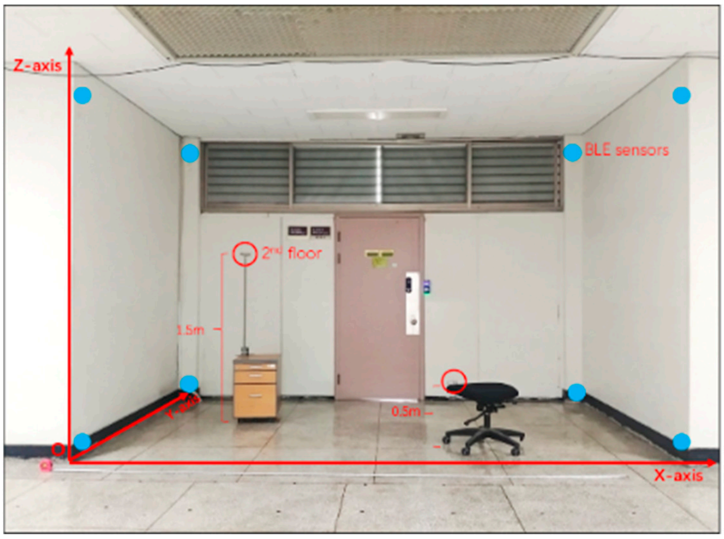

Figure 2. Actual experimental environment and BLE sensor deployment locations.

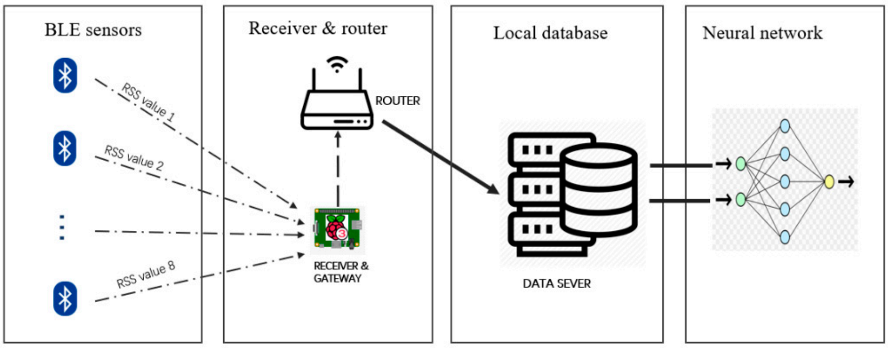

Figure 3. Overview of data flow for developed 3D positioning system.

4. Proposed Scheme

4.1. System Framework

Without a loss of generality, a 3D space can be divided into a set of unit cubic grids. The considered space is divided into M unit cubic grids, which can be considered as M distinct spatial locations. When N BLE beacons are used, a time-series of the RSS values from the N beacons can be collected at each location, and the collected data can be labeled as $J = \{1,\ 2,\ \ldots,\ m,\ldots,\ M\}$. Note that each label indicates the premeasured 3D coordinates of the corresponding cubic grid. The RSS measurement result at location m, X^m, is combined with its label, m, to form a training sample, (X^m, m), for the proposed scheme. The time-series of the RSS values at location m from the n-th beacon can be expressed as $X_n^m = (x_1, x_2, ., x_l, .., x_L)$, where x_l is the l-th RSS value and L is the length of the time-series of the RSS values.

In this study, we used N (=8) BLE beacons and divided the space into M (=16) grids with a unit size of 1 m × 1 m × 1 m, as shown in Figure 4. Therefore, the RSS value vector from the 8 beacons at a certain position, m, can be expressed as $X^m = [X_1^m, X_2^m, \ldots, X_8^m]^T$, where T represents the transpose operation, and the time-series of the RSS values for all the positions is expressed as $X = [X^1, \ldots, X^m, \ldots X^{16}]$. Note that the input for the training phase is denoted as (X_{Tr}, J), while X_{pr} represents the input data for the prediction to evaluate the performance of the proposed scheme. In the prediction phase, the location of the target is estimated based on its RSS value, X_{pr}. The layout of the proposed scheme is shown in Figure 5. As can be seen from the figure, the scheme consists of two phases: the training phase and the positioning phase. The 1D CNN model is trained using the training dataset in the training phase. Next, the trained model is used to predict the location of the target from the input data, X_{pr}.

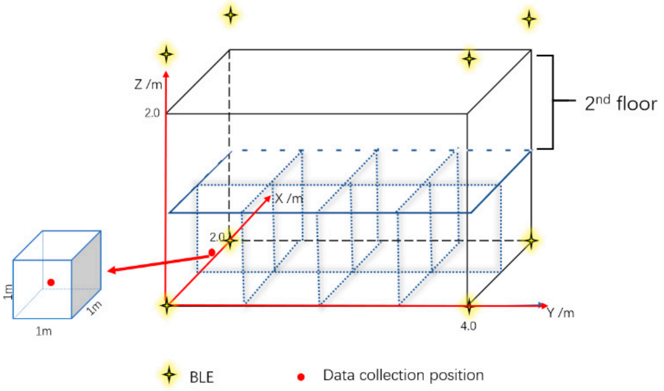

Figure 4. Division of positional spatial grids.

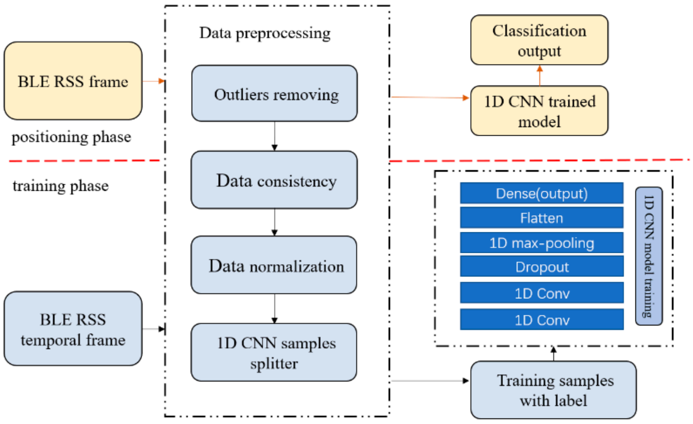

Figure 5. Layout of proposed scheme.

4.2. Data Preprocessing

Generally, it is essential to perform data preprocessing for efficient model convergence when using a neural network. The following data preprocessing methods are employed in the proposed scheme.

4.2.1. Homogenization of RSS Values

Theoretically, a signal scanning process should be enough to obtain the RSS values of all the available BLE sensors in the surrounding environment. In actual implementations, however, no signal scanning process can obtain all the signals because of differences in the signal strength through the different propagation channels and the resulting packet loss. In addition, the receiver may not be able to obtain the same number of temporally consecutive data values from all the beacons owing to the differences in the sampling time. If the length of the samples for each label is not the same, a bias can occur in the training phase. Therefore, it is essential to construct a homogeneous dataset from the heterogeneous dataset. Hence, we constructed RSS value vectors of the same length from consecutive samples to ensure that the input data requirement for the 1D CNN was met. In the proposed scheme, the minimum principle is adopted to prepare the valid signal frame for training the 1D CNN. This means that we chose the sample with the minimum length as the benchmark for all the samples at all 16 sampling positions.

4.2.2. Elimination of Outlier Values

Outliers are generated when the sensor is switched on or off or when there is significant interference, such as that from human activity. To reduce the effect of outliers, the interquartile range (IQR) method has been introduced [27]. The idea of this method is to first rank the data and then choose the interquartile points, denoted as Q1, Q2 and Q3, in ascending order. Then, using the first quartile point, Q1, and the third quartile point, Q3, the reliable interquartile range can be obtained as follows:

$$\mathrm{IQR} = \mathrm{Q3} - \mathrm{Q1}, \tag{3}$$

After obtaining the IQR, the first and second inner limitation (IL) values can be calculated as follows:

$$\text{1st IL: Q1} - 1.5 \times \mathrm{IQR}, \tag{4}$$

$$\text{2nd IL: Q3} + 1.5 \times \mathrm{IQR}, \tag{5}$$

If the RSS value is bigger than the second IL value or smaller than the first IL value, it is regarded as an outlier, while the data values that lay within the confidence interval, that is, between the first and second IL values, can be trusted, as shown in Figure 6. Therefore, we can construct a training dataset of the form (X_{Tr}, J) from the raw observations (X_R, J). This process ensures that the trained model is not polluted by unstable BLE RSS outlier values.

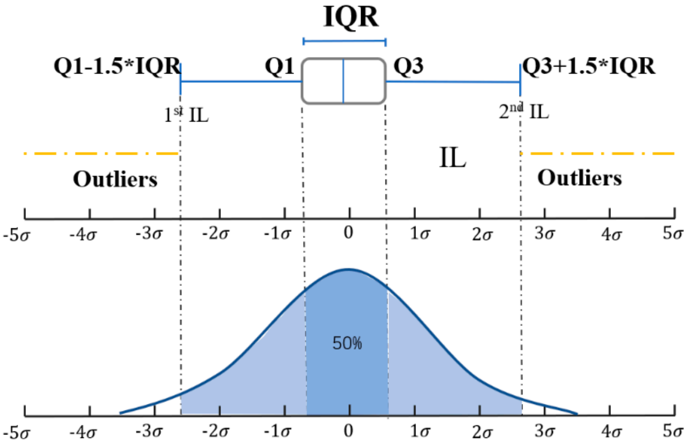
IQR
Q1-1.5*IQR
Q1
Q3
Q3+1.5*IQR
1st IL
2nd IL
IL
Outliers
Outliers
-5σ -4σ -3σ -2σ -1σ 0 1σ 2σ 3σ 4σ 5σ
50%
-5σ -4σ -3σ -2σ -1σ 0 1σ 2σ 3σ 4σ 5σ

Figure 6. Removal of outliers using IQR method.

4.2.3. Data Normalization

The BLE RSS values typically ranged from −70 dB (lowest) to −30 dB (highest). However, we normalized the scale of the RSS values because the input values should be limited to the range (0, 1) for ensuring that the CNN training efficiency and coverage speed are high [28]. The min–max normalization method was adopted for this [29]:

$$x_i' = \frac{(x_i - x_{min})}{(x_{max} - x_{min})} \tag{6}$$

where $\mathrm{x_{min}}$ is the minimum RSS value of the data collected from a beacon. Note that the measured values of all the beacons were normalized independently for every location, instead of normalizing the measured values together.

4.3. 1D CNN Model

In the case of conventional schemes, the theoretical relationship between the RSS value and the distance is used to estimate the location of the target. It is known that theoretically the time-series of the RSS values for a specific location does not change significantly over time. Based on this characteristic, many researchers have introduced

RSS-based fingerprinting schemes based on statistical features such as the entropy, mean, and variance of the time-series of the RSS values to estimate the location. However, this requires designing and extracting features related to the temporal characteristics of RSS values based on the specific situation and thus is not a universal approach [30–32].

Since CNNs show excellent performance with respect to the extraction of additional features from raw data during image classification, many researchers have attempted to use them with 1D signals such as temporal signals. Moreover, a 1D CNN was recently developed to reduce the computational complexity for 1D signals. Since the 3D positioning problem was transformed into a classification problem and the RSS time-series of the BLE signals was considered a 1D signal in the proposed scheme, a 1D CNN was adopted for solving the problem. Figure 7 shows the general process of 1D convolution. Randomly initialized filters, which also termed as kernels, perform convolution extraction, and then scan the entire input data along a certain stride. The extracted outputs make up the feature map.

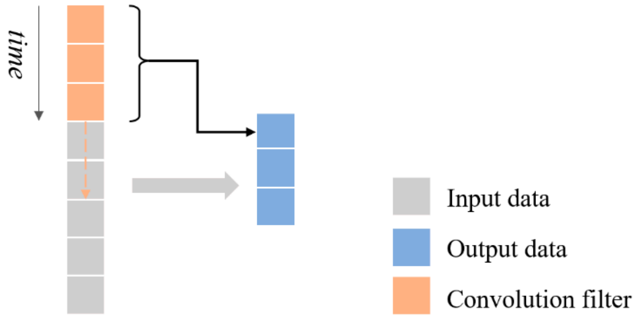

Figure 7. General process of 1D convolution for temporal data.

In the next section, the 1D CNN model of the proposed scheme is described in detail. We used five different layers, which are the convolutional layer, pooling layer, dropout layer, fully connected (FC) layer, and output layer.

4.3.1. Convolutional Layer and Pooling Layer

The function of the convolutional layer is to extract the feature map. In the convolutional layer, filters that are randomly generated using different initialization values traverse every sample, X^m, of the input training dataset, (X_{Tr}, J), along a specific stride and extract features from it. In this manner, the feature map is obtained as the output of the convolution layer, as shown in Figure 7. The number of filters used affects the resolution of the feature output. Generally, the higher the number of filters used, the higher the number of features extracted from the original signal and thus the higher the resolution. The hyperparameters of the convolution layer include the convolution filter size and the stride size, and these determine the size of the feature map. For example, a convolution layer without padding produces an output volume of [16 × 1 × 12] if it uses 12 filters, whose window size is 3, and the stride step is 2 and input volume is [32 × 1].

The output of the convolutional layer exhibits information redundancy and thus a high computing cost. The function of the pooling layer is to down-sample and resample the input data to extract additional features and compress the data to improve the computational efficiency. The pooling function abstracts the input data within the window interval and regards the output as a representative value for the pooled RSS features, as shown in Figure 8. The main parameter of this layer is the stride size, which determines the width of the information extracted.

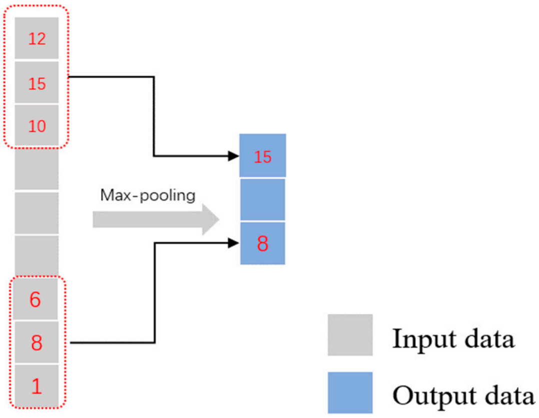

Figure 8. Max pooling operation.

Both max pooling and average pooling are commonly used pooling functions. The max pooling function calculates the maximum value of the RSS value vector within the window, while the average pooling function calculates the mean value of the window. According to the relevant theory, during feature extraction, errors arise primarily because of two factors: (1) the increase in the variance of the estimates caused by the restricted neighborhood size and (2) the offset of the estimated mean value caused by the parameter error of the convolution layer. In general, the average pooling function can reduce the first error, while the max pooling function can reduce the second error. Hence, the max pooling function is used in the proposed scheme. For the entire network, this merely meant the down-sampling of the results obtained from the upper layer and reducing the number of training parameters to avoid overfitting.

4.3.2. Dropout Layer, FC Layer, and Output Layer

The dropout layer is used to solve the problem of overfitting in deep learning [33]. The underlying idea of the dropout layer is to randomly disconnect nodes at a given rate. In the proposed scheme, this layer is placed after the convolutional layer to improve the network diversity. After cross-validation, the best results were obtained when the implicit node dropout rate was set to 0.5.

The output of the max pooling layer is stored in a long vector after passing through the flatten layer, and the data become one-dimensional and is used as the input of the FC layer. The FC layer is usually at the tail of the CNN, and it is similar to the most common artificial neural network named the Dense layer. In the FC layer, all the neurons are fully connected by weight, and each neuron has a class score. The cumulative sum of the neurons is the input to next output layer.

The SoftMax function, which is used widely for multiple classifications, is employed as the output activation function in the output layer. In the case of the considered system, the class with the highest probability amongst the 16 categories is taken as the estimated label for the corresponding input. Since the sum of the probabilities for all the classes is equal to 1, we can estimate the location of the target in terms of the spatial coordinates by performing the regression, as follows:

$$\textstyle\sum_{i=1}^{16} P(i) \times C, \tag{7}$$

where C is the set of all the predefined coordinates for all the 16 reference locations, and $P(i)$ is the estimated probability of each class. ReLU was selected as the activation function for all the applicable layers except the output layer, where the SoftMax function was used. Adam was used as the optimization algorithm instead of the classical stochastic gradient descent method to update the network weights iteratively based on the training data [34]. The categorical cross-entropy loss function was adopted because it is well suited for tasks involving multiple classifications. The parameters for the 1D CNN model are listed in Table 1.

Table 1. Parameters for 1D CNN model.

Parameter	Value
Hidden layers	4
Hidden activation	ReLU
Output activation	SoftMax
Optimizer	Adam
Loss function	Cross-entropy

4.3.3. Summary of 1D CNN Model of Proposed Scheme

In a typical CNN, the data type is generally a single-channel grayscale image or a three-channel color image. Analogously, the data used in this study can be considered multichannel monoscale (gray) images. However, in contrast to the case for actual images, the total number of beacons was taken to be the number of channels for training the 1D CNN model. In other words, the number of channels was set to 8 because 8 BLE beacons were used. The RSS value sequences from all the BLEs were divided into individual samples, with each consisting of 32 RSS values, in a sequential and nonoverlapping manner based on the chronological order of reception.

In order to ensure a wider feature extraction range, which is expected to yield more features, in the case of the input data, we did not go directly to the pooling layer after performing one convolution. Instead, we performed the convolutional extraction twice, which means after the first convolution layer, we added second convolution layer to execute feature extraction again. Thus, in this manner, we not only limited the number of parameters but also improved feature extraction.

To begin with, all the input samples are in the form of $[P \times 32 \times 8]$, where P is the number of samples, 32 is the RSS value length of one sample, and 8 is the number of BLEs. It can be processed by two times of conventional operation with 32 filters, of which window size is 3 and stride is 1 in default and no padding used, and then output map size is $[P \times 28 \times 32]$. After that, the dropout layer is added to mitigate the effects of overfitting. By being made to pass through the max pooling layer with size 2, stride 2 and no padding used, the local features in the form of $[P \times 28 \times 32]$ are down-sampled to local features in the form of $[P \times 14 \times 32]$. Next, the data passes through the flatten layer and are converted into 1D vector data of the form $[P \times 448]$ and then fed to the FC layer. After the data has passed through the FC layer, the output layer with the SoftMax function is used to determine the label corresponding to the predicted result. As a summary, conventional filter size and max pooling size are 3 and 2, separately, the stride size are 1 and 2 successively, both of them were no padding used and the number of datapoints in the training batch was set at 32, the number of parameters for the entire network was 12,698 from model own statistics.

5. Performance Evaluation

The performance of the proposed scheme was evaluated through comprehensive tests. First, the convergence and accuracy of the 1D CNN scheme were evaluated, and the effects of data preprocessing and the kernel size on the proposed scheme were investigated. Next, the location estimation performance of the proposed scheme was evaluated. We compared the classification accuracy of the proposed scheme with that of the conventional CSP algorithm. All the tests were performed using Python 3.8 on a desktop equipped with an Nvidia GeForce GTX 1650 GPU and an AMD FX(tm)-6300 3.50 GHz six-core central processing unit. FeasyBeacon 5Mart FSC-BP104, which is a Bluetooth 5.0 BLE smart beacon with a TI CC2640R2F chipset and works at an ISM frequency of 2.4 GHz, was used [35]. We set the data transmission interval to 100 ms and the transmission power to +5 dBm. In addition, we set the broadcast mode as the transmission format, and the broadcast packets followed the specifications designed by the company.

5.1. Loss and Accuracy Performance of Proposed Scheme

The effectiveness of the proposed scheme was evaluated based on the loss function and accuracy of the training process. The validation process was performed for up 20 epochs. During the tests, approximately 200 samples were collected at each predefined location, and the length of the RSS value for each sample was set to 32. The complete dataset of 3200 samples was randomly divided as follows: 70% training data, 10% validation data, and 20% test data. The order of the samples in the training data set was randomly shuffled.

Figure 9 shows the loss and accuracy performance of the proposed scheme. As can be seen from the figure, both curves changed significantly and converged at approximately Epoch 3. Since the convergence rate was high, it can be concluded that the dataset was suitable for the proposed scheme.

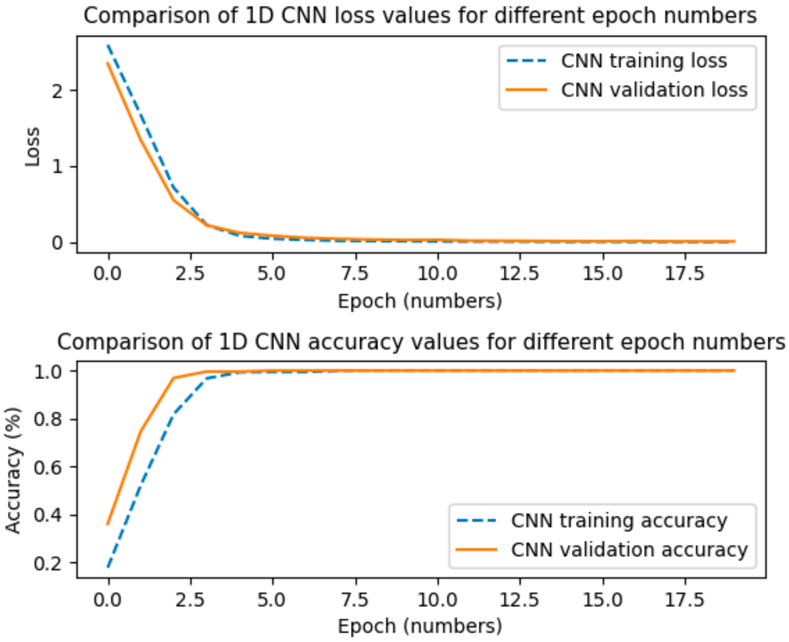

Figure 9. Loss and accuracy performance of 1D CNN model.

5.2. Effect of Data Preprocessing on Proposed Scheme

A major benefit of neural networks is that prior knowledge of the noise distribution is not required. Noisy RSS value measurements can be used directly to train the network, and the neural network is capable of characterizing the noise and compensating for it to determine the target position with accuracy. To estimate the effect of outlier preprocessing on the training of the 1D CNN model, comparative tests were performed; the results are shown in Figure 10. As can be seen from the figure, the dataset subjected to outlier preprocessing resulted in better loss and accuracy performance than that not subjected to it. In addition, it can be seen from the loss function curve that the convergence point, for the dataset not subjected to outlier preprocessing is at approximately Epoch 10. Thus, convergence in this case took three times longer than that for the dataset subjected to outlier preprocessing (approximately, Epoch 3). This means that noisy interference or outlier values can add to the complexity of network learning and that data preprocessing is necessary for efficient model training.

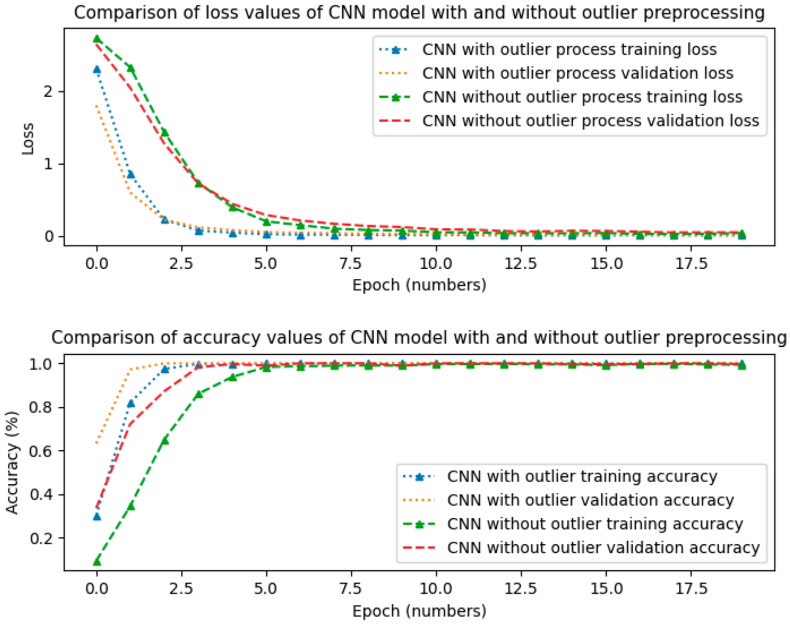

Figure 10. Comparison of performance of 1D CNN model using datasets with and without outlier values.

5.3. Effect of Kernel Size on Proposed Scheme

The effect of the kernel size used for convolution was also evaluated to optimize the performance of the 1D CNN. We used kernel sizes of 3, 6 and 12 to test the 1D CNN model in terms of loss and accuracy. The results are shown in Figure 11.

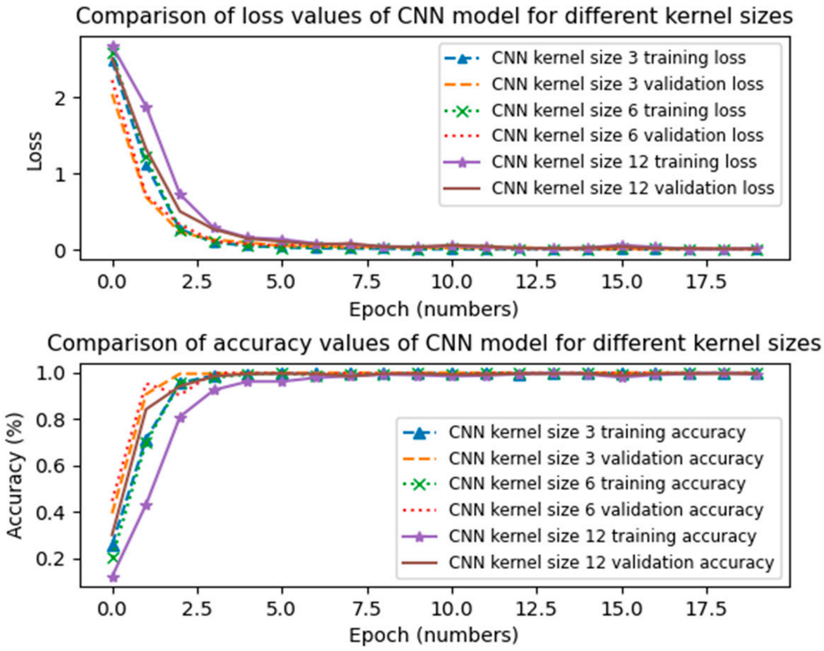

Figure 11. Comparison of performance of 1D CNN model for kernel sizes of 3, 6, and 12.

As can be seen from the figure, the performance deteriorated as the kernel size was increased. Specially, both the loss and the accuracy were the worst for the kernel size of 12. On the other hand, for the kernel sizes of 3 and 6, the performances were similar. This means that a large convolution window is not preferable for extracting more reliable features from a large set of widely fluctuating RSS values. In addition, a large window also increases the computational burden. Thus, the size of the convolution kernel was set to 3.

5.4. Position Estimation Performance

After the completion of the training and validation processes, we evaluated the performance of the proposed scheme in 3D position estimation using the test dataset, X_{pr}. Since the 1D CNN model provides the probability of each possible category of the target's location, the coordinates of the target can be calculated using Equation (7). To allow for a visual comparison of the estimated and actual positions, we tested 340 samples from all 16 categories. The results are plotted in Figure 12. In the figure, the green stars are the estimated positions of the target while the red crosses represent the actual positions. As can be seen from the figure, the proposed scheme could estimate the 3D position accurately.

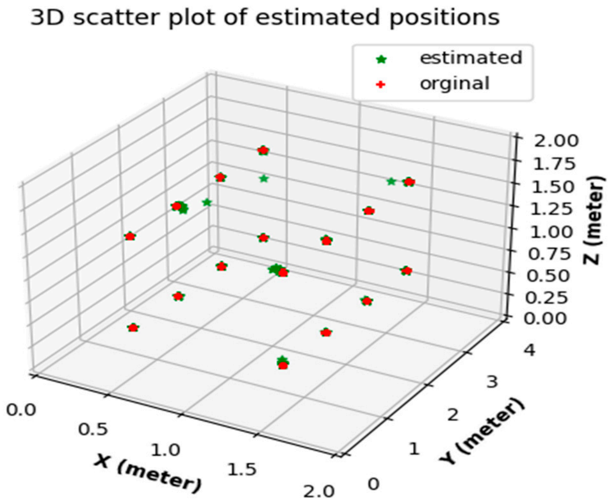

Figure 12. Results of 3D position estimation for all 16 classes.

Next, we compared the classification accuracy of the proposed scheme with that of the conventional CSP algorithm. Figure 13 shows the comparison of the results obtained using the proposed scheme and the CSP algorithm for all the classifications. As shown in the figure, the 1D CNN model outperformed the CSP algorithm in the case of every position category.

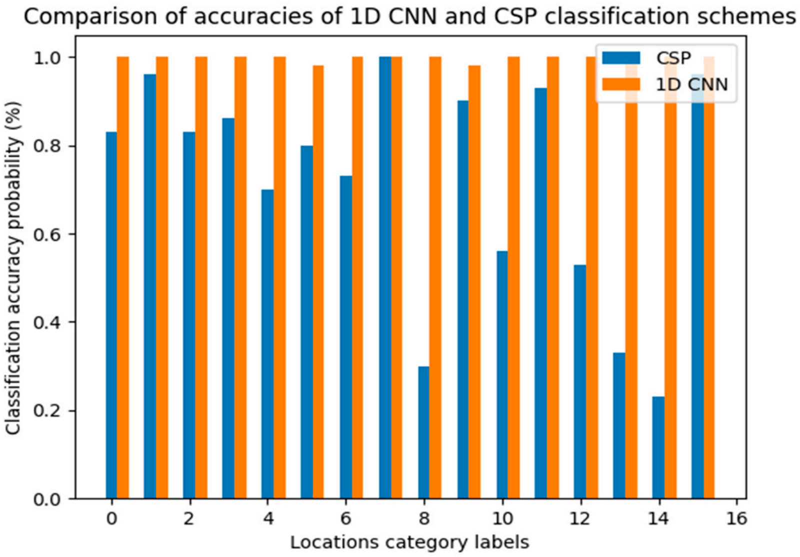

Figure 13. Comparison of classification accuracies of 1D CNN model and CSP algorithm.

Figure 14 shows the cumulative distribution functions (CDFs) of the estimated coordinate errors for the 1D CNN and CSP schemes. As can be seen from the figure, the 1D CNN scheme significantly outperforms the CSP scheme. The localization errors of the two schemes are also compared in Table 2. As per the test results, the mean error of the proposed 3D localization scheme based on the 1D CNN is 0.25 m, while that for the CSP scheme is approximately 2 m.

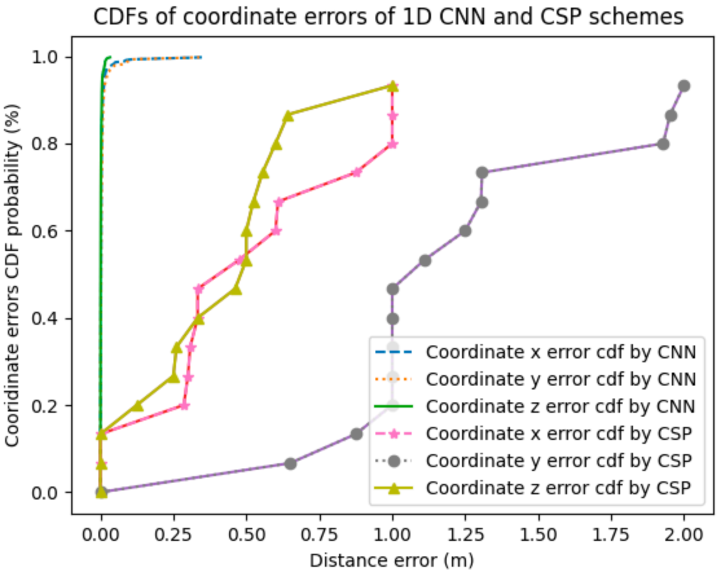

Figure 14. CDFs of estimated coordinate errors of 1D CNN and CSP schemes.

Table 2. Comparison of positioning errors of 1D CNN and CSP schemes.

-	Maximum Error/m	Mean Error/m
1D CNN	0.75	0.25
CSP	2.95	2.01

The reason the proposed scheme exhibits higher accuracy may be the independence of the data obtained from the beacons. During the experiments, we deployed 8 BLE beacons at different locations, and the Raspberry Pi receiver collected 32 distinct RSS values from each BLE transmitter beacon for one location. This means that the proposed scheme exploits the time-series data of each beacon using the 1D CNN while ensuring that the data from the multiple beacons remain independent.

6. Conclusions

In this paper, we proposed an indoor 3D localization scheme based on both fingerprinting and a 1D CNN. In the proposed scheme, instead of using the conventional fingerprint matching method, the 3D positioning problem is transformed into a classification problem, and a 1D CNN is used with the RSS time-series data from the BLE beacons to determine the target locations. By using a 1D CNN with the time-series data from multiple beacons, the inherent drawback of RSS-based fingerprinting, namely, its susceptibility to noise and randomness, could be overcome, resulting in enhanced positioning accuracy. To evaluate the proposed scheme, we developed a 3D positioning system, including BLE signal reception and uploading process, introduced multiple signal preprocessing methods and performed comprehensive tests in real scenarios. One the one hand, we evaluated our proposed 1D CNN model itself. On the other hand, in terms of the positioning accuracy, the results showed that the proposed scheme significantly outperforms the conventional CSP classification algorithm. The accuracy of the proposed scheme in 3D location classification was almost 100%, while that of the conventional CSP scheme was only 70%. Moreover, the mean error of the proposed 3D localization scheme based on the 1D CNN was 0.25 m while that of the CSP scheme was approximately 2 m. Our proposed scheme can be used in small-area indoor environment and improves the practicality of 3D positioning. In future work, we plan to investigate the coverage problem of Bluetooth signals, hoping to find the optimal coverage solution, and we will discuss the impact on the computational complexity of the model and stability as the number of nodes varies.

Author Contributions: Conceptualization, methodology, and writing—original draft preparation, S.Y.; validation and data curation, C.S.; writing—review and editing and supervision, Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (NRF-2019R1F1A1049677 and NRF-2021R1F1A1049509). The present research has been conducted by the Research Grant of Kwangwoon University in 2021.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Abbreviation and description in this manuscript are summarized in Table A1.

Table A1. Abbreviation and description.

Abbreviation	Description
1D CNN	One-Dimensional Convolutional Neural Network
2D CNN	Two-Dimensional Convolutional Neural Network
3D	Three-Dimensional
BLE	Bluetooth Low Energy
CSP	Common Spatial Pattern
Conv.	Convolution Layer
CDF	Cumulative Distribution Function
FC	Fully Connected
GPS	Global Positioning System
GPU	Graphical Processing Units
ID	Identity
IQR	Interquartile Range
IL	Inner Limitation
ISM	Industrial Scientific Medical
RSS	Received Signal Strength
Q1	First Interquartile Point
Q2	Second Interquartile Point
Q3	Third Interquartile Point
RFID	Radio Frequency Identification
ReLU	Rectified Linear Unit
UAV	Unmanned Aerial Vehicles

References

1. Basiri, A.; Lohan, E.S.; Moore, T.; Winstanley, A.; Peltola, P.; Hill, C.; Amirian, P.; Figueiredo e Silva, P. Indoor location based services challenges, requirements and usability of current solutions. *Comput. Sci. Rev.* **2017**, *24*, 1–12. [CrossRef]
2. Alkhawaja, F.; Jaradat, M.; Romdhane, L. Techniques of indoor positioning systems (IPS): A survey. In Proceedings of the 2019 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 26 March–11 April 2019; pp. 1–8.
3. Brena, R.F.; García-Vázquez, J.P.; Galván-Tejada, C.E.; Munoz-Rodriguez, D.; Vargas-Rosales, C.; Fangmeyes, J. Evolution of indoor positioning technologies: A survey. *J. Sens.* **2017**. [CrossRef]
4. Zafari, F.; Gkelias, A.; Leung, K.K. A survey of indoor localization systems and technologies. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2568–2599. [CrossRef]
5. Gu, F.; Hu, X.; Ramezani, M.; Acharya, D.; Khoshelham, K.; Valaee, S.; Shang, J. Indoor localization improved by spatial context—A survey. *ACM Comput. Surv.* **2019**, *52*, 1–35. [CrossRef]

6. Bahl, P.; Padmanabhan, V.N. RADAR: An in-building RF-based user location and tracking system. In Proceedings of the IEEE INFOCOM 2000, Conference on Computer Communications, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064), Tel Aviv, Israel, 26–30 March 2000; pp. 775–784.
7. Jiang, H.; Peng, C.; Sun, J. Deep belief network for fingerprinting-based RFID indoor localization. In Proceedings of the 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–5.
8. Bianchi, V.; Ciampolini, P.; De Munari, I. RSSI-based indoor localization and identification for ZigBee wireless sensor networks in smart homes. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 566–575. [CrossRef]
9. Luo, R.C.; Hsiao, T.J. Indoor localization system based on hybrid Wi-Fi/BLE and hierarchical topological fingerprinting approach. *IEEE Trans. Veh. Technol.* **2019**, *68*, 10791–10806. [CrossRef]
10. Wang, Y.; Shang, Y.; Tao, W.; Yu, Y. Target positioning algorithm based on RSS fingerprints of SVM of fuzzy kernel clustering. *Wirel. Pers. Commun.* **2021**, 1–19. [CrossRef]
11. Mirowski, P.; Milioris, D.; Whiting, P.; Ho, T.K. Probabilistic radio-frequency fingerprinting and localization on the run. *Bell Labs Tech. J.* **2014**, *18*, 111–133. [CrossRef]
12. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **2021**, *151*, 107398. [CrossRef]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
14. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In Proceedings of the European Conference on Computer Vision (ECCV 2014), Zurich, Switzerland, 5–12 September 2014; pp. 818–833.
15. Abdeljaber, O.; Avci, O.; Kiranyaz, S.; Gabbouj, M.; Inman, D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* **2017**, *388*, 154–170. [CrossRef]
16. Han, Y.; Qi, W.; Ding, N.; Geng, Z. Short-time wavelet entropy integrating improved LSTM for fault diagnosis of modular multilevel converter. *IEEE Trans. Cyber.* **2021**. [CrossRef]
17. Jia, B.; Zong, Z.; Huang, B.; Baker, T. A DNN-based WiFi-RSSI Indoor Localization Method in IoT. In Proceedings of the International Conference on Communications and Networking in China (ChinaCom 2020), Hangzhou, China, 20–21 November 2020; pp. 200–211.
18. Wang, Y.; Gao, J.; Li, Z.; Zhao, L. Robust and accurate Wi-Fi fingerprint location recognition method based on deep neural network. *Appl. Sci.* **2020**, *10*, 321. [CrossRef]
19. Jain, C.; Sashank, G.V.S.; Markkandan, S. Low-cost BLE based Indoor Localization using RSSI Fingerprinting and Machine Learning. In Proceedings of the Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Online Conference, 25–27 March 2021; pp. 363–367.
20. Sun, D.; Wei, E.; Yang, L.; Xu, S. Improving Fingerprint Indoor Localization Using Convolutional Neural Networks. *IEEE Access* **2020**, *8*, 193396–193411. [CrossRef]
21. Sun, D.; Wei, E.; Ma, Z.; Wu, C.; Xu, S. Optimized CNNs to Indoor Localization through BLE Sensors Using Improved PSO. *Sensors* **2021**, *21*, 1995. [CrossRef] [PubMed]
22. Tasaki, K.; Takahashi, T.; Ibi, S.; Sampei, S. 3D Convolutional Neural Network-Aided Indoor Positioning Based on Fingerprints of BLE RSSI. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Auckland, New Zealand, 7–10 December 2020; pp. 1483–1489.
23. Frattasi, S.; Della Rosa, F. *Mobile Positioning and Tracking: From Conventional to Cooperative Techniques*; John Wiley & Sons: Hoboken, NJ, USA, 2017; Volume 3, ISBN 9781119068815.
24. Pušnik, M.; Galun, M.; Šumak, B. Improved Bluetooth low energy sensor detection for indoor localization services. *Sensors* **2020**, *20*, 2336. [CrossRef] [PubMed]
25. Davis, S.J. Indoor Wireless RF Channels. Available online: http://www.wirelesscommunication.nl/reference/chaptr03/indoor.htm (accessed on 29 May 2021).
26. Zhao, X.; Ruan, L.; Zhang, L.; Long, Y.; Cheng, F. An Analysis of the optimal placement of beacon in Bluetooth-INS indoor localization. In Proceedings of the 14th International Conference on Location Based Services (LBS 2018), Zurich, Switzerland, 15–17 January 2018; pp. 50–55.
27. Upton, G.; Cook, I. *Understanding Statistics*; Oxford University Press: Oxford, UK, 1996; ISBN 9780199143917.
28. Xiao, L.; Behboodi, A.; Mathar, R. A deep learning approach to fingerprinting indoor localization solutions. In Proceedings of the 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, Australia, 22–24 November 2017; pp. 1–7.
29. Njima, W.; Ahriz, I.; Zayani, R.; Terre, M.; Bouallegue, R. Deep CNN for indoor localization in IoT-sensor systems. *Sensors* **2019**, *19*, 3127. [CrossRef] [PubMed]
30. Wang, F.; Feng, J.; Zhao, Y.; Zhang, X.; Zhang, S.; Han, J. Joint activity recognition and indoor localization with WiFi fingerprints. *IEEE Access* **2019**, *7*, 80058–80068. [CrossRef]
31. Wang, Y.; Wu, K.; Ni, L.M. Wifall: Device-free fall detection by wireless networks. *IEEE Trans. Mobile Comput.* **2016**, *16*, 581–594. [CrossRef]
32. Wang, H.; Zhang, D.; Wang, Y.; Ma, J.; Wang, Y.; Li, S. RT-Fall: A real-time and contactless fall detection system with commodity WiFi devices. *IEEE Trans. Mobile Comput.* **2016**, *16*, 511–526. [CrossRef]

33. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
34. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
35. Feasybecaon Home Page. Available online: https://www.feasybeacon.com/bluetooth-ble-proximity-beacons.html (accessed on 29 May 2021).