


Communication

# Multiple Fingerprinting Localization by an Artificial Neural Network

Jaehyun Yoo 

School of AI Convergence, Sungshin Women's University, Seoul 02844, Korea; jhyoo@sungshin.ac.kr;  
Tel.: +82-2-920-7695

**Abstract:** Fingerprinting localization is a promising indoor positioning methods thanks to its advantage of using preinstalled infrastructure. For example, WiFi signal strength can be measured by pre-existing WiFi routers. In the offline phase, the fingerprinting localization method first stores of position and RSSI measurement pairs in a dataset. Second, it predicts a target's location by comparing the stored fingerprint database to the current measurement. The database size is normally huge, and data patterns are complicated; thus, an artificial neural network is used to model the relationship of fingerprints and locations. The existing fingerprinting locations, however, have been developed to predict only single locations. In practice, many users may require positioning services, and as such, the core algorithm should be capable of multiple localizations, which is the main contribution of this paper. In this paper, multiple fingerprinting localization is developed based on an artificial neural network and an analysis of the number of targets that can be estimated without loss of accuracy is conducted by experiments.

**Keywords:** WiFi fingerprinting localization; artificial neural network; multiple targets estimation



**Citation:** Yoo, J. Multiple Fingerprinting Localization by an Artificial Neural Network. *Sensors* **2022**, *22*, 7505. <https://doi.org/10.3390/s22197505>

Academic Editors: Raffaele Bruno, Leopoldo Angrisani, Nikos Fotiou and Ismail Butun

Received: 13 September 2022

Accepted: 30 September 2022

Published: 3 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



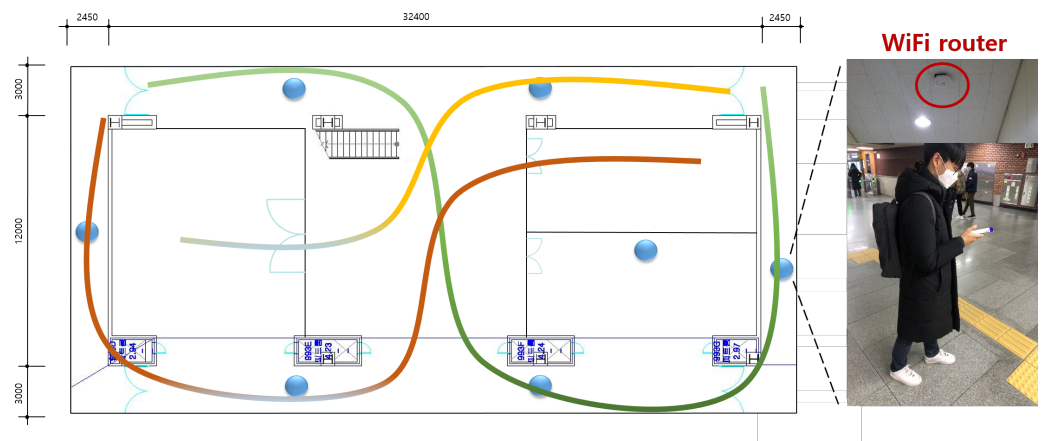
**Copyright:** © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

There are different methods for indoor localization based on RF signals. In the Time of Arrival (TOA), Time Difference of Arrival (TDOA), and Angle of Arrival (AoA) methods [1], a target is estimated by calculating the time differences, distances, and angles, respectively, between a target and transmitters using the received signals from deployed beacons. On the other hand, the WiFi fingerprinting approach estimates a target location based on pattern recognition through machine learning methods. This paper focuses on WiFi fingerprinting localization. Figure 1 illustrates multiple fingerprinting indoor localization using a WiFi router and its measurements, with different colors representing different users' trajectories.

The Wi-Fi indoor positioning method requires a database of signal strength fingerprints collected from surrounding areas for each predefined location. In an indoor environment with many obstacles, such as walls blocking radio waves, more fingerprint data are required for higher positioning accuracy. In considering the resolution of the Wi-Fi signal strength according to the distance, fingerprint data are usually collected at intervals of 1 to 2 m to form a training database. When providing the positioning service, the current location is predicted by comparing the currently received signal strength set with the previously built database. Because this is the same as the training and test phases of machine learning approaches, *k*-nearest neighbors, support vector machine, Gaussian processes, and deep neural networks can be used for fingerprinting localization [2–5].

When applying a deep neural network for WiFi indoor positioning, a wireless signal strength set is configured as an input of the neural network, a location is set as an output terminal, then a hidden network is learned as a positioning model. In the learning phase, the model is trained using the signal strength fingerprint data collected over the entire positioning region, and in the test phase the predicted position is obtained by inputting the query signal strength set to the learned neural network model.



**Figure 1.** Multiple fingerprinting indoor localization.

Most indoor localization focuses on estimating the location of a single target [6–12], with only a few addressing multi-target localization. In [13,14], multi-targeting was achieved using UHF RFID and microphone devices, respectively. These methods require a number of devices to be located in a room, which is not feasible for general localization purposes. In [15,16], a WiFi fingerprint was used for multiple targets. In [15], an extraction method of signal reflection corresponding to multiple users was proposed. This technique requires a controlled environment in which measurements of near-static WiFi devices are available. In [16], graph-based optimization was used to fuse WiFi and dead reckoning for a single target. When this filtration method is expanded to multi-user scenarios, the number of users and measurements strongly affects the time complexity.

Existing neural network-based positioning technology is inefficient in terms of real-time processing speed because it has no choice but to perform estimation calculations as many times as requested when a large number of users request positioning service. When indoor positioning service is provided from a server to local smart-devices in a centralized way, the service quality depends on the time required to estimate the location, not on the learning processing time. Therefore, a new positioning method for simultaneous multiple users is needed.

As it becomes necessary to infer many locations at the same time, the dimensionality of the machine learning input data increases. Because a deep neural network has excellent reasoning ability for learning high-dimensional data, it is suitable for multi-location estimation problems. This paper handles preprocessing and standardization of WiFi received signal strength indicator (RSSI) measurements, which are significant for obtaining good learning performance.

To evaluate the proposed multi-target estimation approach based on an artificial neural network for predicting 3D information, including both location and floor, experimental data were collected in a five-story building. We analyzed the number of targets that could be estimated without loss of significant accuracy. The experimental results show that fifteen targets can be accurately estimated with similar performance in the single localization.

The rest of this paper is organized as follows. Section 2 describes the proposed multi-target WiFi fingerprinting localization approach. Section 3 reports our experimental results. Section 4 presents our conclusions.

## 2. WiFi Fingerprinting Indoor Localization

Suppose that there are a total of  $d$  WiFi routers in which the indoor positioning service is to be established. In a multi-story building, the location is expressed as a three-dimensional vector such as ( $x$  coordinate,  $y$  coordinate, floor). By storing RSSI measurements received from  $d$  routers at collecting locations, a fingerprint database consisting of (location,  $d$ -dimensional RSSI set) pairs is constructed, in which router locations are not necessarily known. To include environmental variations in the data, RSSIs are measured several times at a location. Table 1 shows the fingerprint database format.

**Table 1.** Example of WiFi RSSI fingerprint database.

	AP 1	AP 2	...	AP $d$
Position 1	−85 dBm	−68 dBm	...	Null
Position 2	Null	−51 dBm	...	−92 dBm
⋮	⋮	⋮	⋮	⋮
Position $N$	Null	Null	...	−45 dBm

Because certain signals from distantly located routers are not observable, the fingerprint database has many elements filled with empty spaces (null). Because null spaces cannot be processed in machine learning, these are intentionally converted to a  $-100$  dB value.

The following Sections 2.1 and 2.2 present feature extraction for WiFi RSSI measurements and standardization for labelled location data, respectively. Section 2.3 presents deep learning-based multi-target localization.

### 2.1. WiFi RSSI Feature Extraction

Principal component analysis (PCA) [17] is a data transformation method for reducing the dimensionality of data and extracting the features data by minimizing information loss. It can be understood as an unsupervised learning method, as it does not require label information such as location; thus, in this paper, it is used to extract the features of WiFi fingerprint measurement data.

Suppose  $D = \{(x_i, y_i)\}_{i=1}^N$  are fingerprint training datasets, where  $x_i$  is a vector of RSSI measurements at a location  $y_i$  provided by

$$x_i = [s_i^1, s_i^2, \dots, s_i^d]^T \in \mathcal{R}^d, \quad (1)$$

$$y_i = [x_i, y_i, f_i]^T \in \mathcal{R}^3, \quad (2)$$

where  $s_i^j$  is a scalar indicating the RSSI value at the  $i$ th location received from the  $j$ th WiFi router over a total of  $d$  routers,  $x_i$  and  $y_i$  are the 2D location, and  $f_i$  is floor level.

The original data  $x$  are transformed to the feature dataset  $\bar{x}$  by the transformation matrix  $Q$ , as follows:

$$\bar{x} \leftarrow Q \cdot x, \quad (3)$$

$$\text{where } Q \in \mathcal{R}^{l \times d}, \bar{x} \in \mathcal{R}^l, x \in \mathcal{R}^d, l \ll d. \quad (4)$$

In this paper, finding an optimal  $Q$  is achieved by the PCA algorithm, which solves a generalized eigenvalue problem. Let  $\{\rho_i\}_{i=1}^d$  be the generalized eigenvectors associated with the generalized eigenvalues  $\{\lambda_i\}_{i=1}^d$  of the following generalized eigenvalue problem:

$$A\rho_i = \lambda_i I\rho_i, \quad i = 1, \dots, d, \quad (5)$$

where  $I$  is an identity matrix and  $A$  is a scatter matrix provided by

$$A = \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T, \quad (6)$$

with the mean of all the RSSI samples  $\mu$ :

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i. \quad (7)$$

Then, the optimal matrix  $\hat{Q}$  is calculated by solving the optimization problem presented by

$$\hat{Q} = \operatorname{argmax}_Q \frac{|Q^T A Q|}{|Q^T I Q|}, \quad (8)$$

where the condition of the eigenvalue problem is that the generalized eigenvectors are orthogonal:

$$\rho_i^T \rho_j = 0, \quad \text{for } i \neq j, \quad (9)$$

and the generalized eigenvectors are normalized:

$$\rho_i^T \rho_i = 1, \quad \text{for } i = 1, \dots, d. \quad (10)$$

Finally, when the eigenvalues are sorted in descending order such that

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d, \quad (11)$$

the transformation matrix  $Q$  is provided by

$$Q = \left( \sqrt{\lambda_1} \rho_1 | \sqrt{\lambda_2} \rho_2 | \dots | \sqrt{\lambda_l} \rho_l \right) \in \mathcal{R}^{l \times d}. \quad (12)$$

It should be noted that the last small values of the eigenvalues and eigenvectors do not affect the transformation. Thus, we choose a proper  $l$  to cut down the original dimensionality from  $d$  to  $l$ .

## 2.2. Label Standardization

In a WiFi fingerprint localization system, the two-dimensional coordinate values in the position vector  $y_i$  are generally expressed in meters and the floor values are expressed as integers, ensuring that the scale of the data values between position and floor is different. In addition, the location data are not evenly distributed, as the two-dimensional location is concentrated in the hallway due to the nature of the indoor space. This characteristic can have a negative effect on the accuracy of all general machine learning algorithms, including artificial neural networks. In this paper, standardization is used to convert the original location label data  $y_i = [x_i, y_i, f_i]^T$  to  $\bar{y}_i$ , as follows:

$$\bar{y}_i \leftarrow (y_i - \mu_s) / \sigma_s, \quad (13)$$

where  $\mu_s$  and  $\sigma_s$  are the mean and standard deviation of  $\{y_i\}_{i=1}^N$ , respectively.

As a summary of the data preprocessing, the PCA transforms the RSSI measurement data from  $x$  to  $\bar{x}$ , and standardization transforms the location data from  $y$  to  $\bar{y}$ . As a result, the new feature database  $\bar{D} = \{\bar{x}_i, \bar{y}_i\}_{i=1}^N$  is obtained.

## 2.3. Multi-Target Estimation Based on Artificial Neural Network

An artificial neural network consists of several layers. In a layer, neural nodes have a linear combination composed of learning parameters. When the combined signal propagates to the next layer, a nonlinear model such as a rectified linear unit (ReLU) activation function decides the output of the layer. As the signal propagates to multiple layers, the nonlinearity increases. The success of deep learning in finding optimal node parameters depends on determining the appropriate number of hidden layers and neural nodes.

Figure 2 shows the structure of a deep neural network for general single-location learning. In this case, the input layer of the neural network uses the  $l$ -dimensional RSSI vector,  $\bar{x} \in \mathcal{R}^l$  in (3).

Figure 3 is a neural network structure for multi-location recognition. It is a parallel extension of the single-location neural network shown in Figure 2. As shown in Figure 3, as the number of positions to be simultaneously estimated increases, more neural network nodes and a higher number of learning parameters are required. When the input data and output data are expanded, as shown in Figure 3, the dimensionality of the data increases.

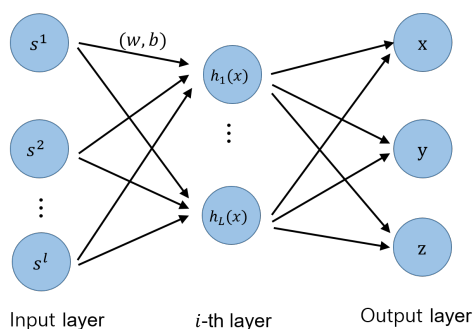


Figure 2. Deep neural network structure for single-position learning.

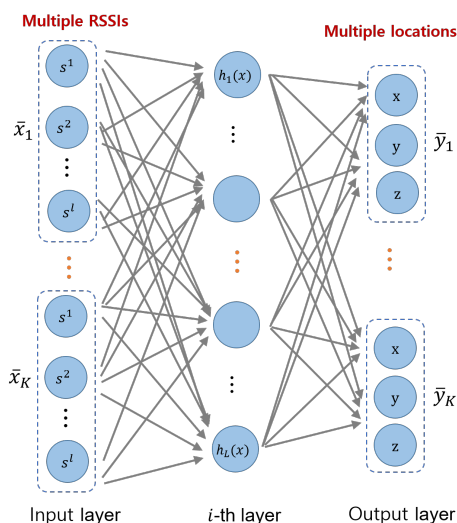


Figure 3. Extended deep neural network structure for multi-position learning.

General machine learning techniques including deep neural network require large amounts of training data for high dimensionality data [18]. The datasets  $\bar{D} = \{\bar{x}_i, \bar{y}_i\}_{i=1}^N$  have undergone preprocessing for single positioning. Given a fixed amount of data  $\bar{D}$ , in order to generate sufficient data for multi-location learning (that is, to make up the input/output dataset represented in Figure 3), an additional rendering operation is required.

Let us suppose that the number of multiple locations to be estimated is  $K$ ; then, the composition of input data and output data of the neural network is as follows:

$$X_i = [\bar{x}_1^T, \bar{x}_2^T, \dots, \bar{x}_K^T]^T, \tag{14}$$

$$Y_i = [\bar{y}_1^T, \bar{y}_2^T, \dots, \bar{y}_K^T]^T. \tag{15}$$

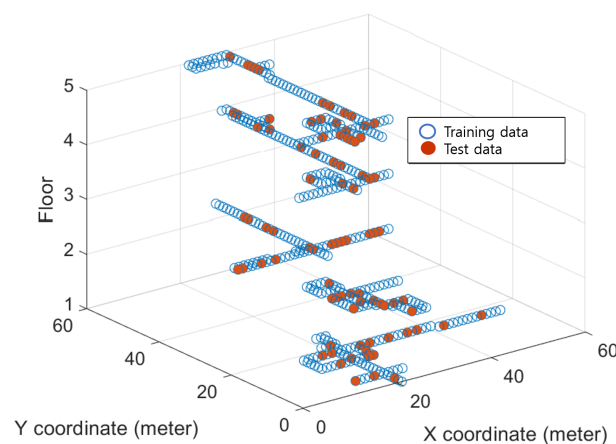
In (14) and (15), the  $j$ th component  $\bar{x}_j^T$  and  $\bar{y}_j^T$  are sampled from the original  $\bar{D} = \{\bar{x}_i, \bar{y}_i\}_{i=1}^N$ . It should be noted here that according to the sorting order of the pairs  $(\bar{x}_j^T, \bar{y}_j^T) \forall j = 1, \dots, K$ , data  $X_i$  and  $Y_i$  can be heterogeneous. For example,  $X_1 = [\bar{x}_1^T, \bar{x}_2^T, \dots, \bar{x}_K^T]^T$  and  $X_2 = [\bar{x}_K^T, \bar{x}_{K-1}^T, \dots, \bar{x}_1^T]^T$  are different sets unless all elements are equal. Therefore, an appropriate amount of new training data to be used for multi-location learning can be generated by repeating and aligning random extractions at  $\bar{D}$  while keeping the condition of  $\bar{y}_j^T \neq \bar{y}_k^T$  for any  $j \neq k \in \{1, \dots, N\}$ . As a result, new training datasets for the multiple targets  $F = \{X_i, Y_i\}_{i=1}^c$  are made, in which parameter  $c$  is defined by a developer. Algorithm 1 summarizes the algorithm used to produce multi-target training data  $F$  from  $\bar{D}$ .

**Algorithm 1** Rendering multi-target fingerprint training dataset**Input:**  $\bar{D} = \{\bar{x}_i, \bar{y}_i\}_{i=1}^N$ , single target fingerprint training data**Output:**  $F = \{X_i, Y_i\}_{i=1}^c$ , multi-target fingerprint training data**Initialization:** Set the desired number of multi-target training data  $c$  and the multi-target  $K$ , and define  $X_i = Y_i = \phi$  as empty sets.

- 1: Initialize generator with  $\theta_G$ , discriminator with  $\theta_D$  and classifier with  $\theta_C$ .
- 2: **for**  $i = 1, 2, \dots, c$  **do**
- 3:   **for**  $j = 1, 2, \dots, K$  **do**
- 4:     Sample  $(\bar{x}_j, \bar{y}_j)$  randomly from  $\bar{D}$  subject to  $\bar{x}_j \notin X_i$  and  $\bar{y}_j \notin Y_i$ .
- 5:     Add the samples by  $X_i \leftarrow X_i \cup \bar{x}_j$  and  $Y_i \leftarrow Y_i \cup \bar{y}_j$
- 6:   **end for**
- 7:   (Here, the  $i$ th data in Equations (14) and (15) are made up.)
- 8: **end for**

**3. Experimental Results****3.1. Setup**

Figure 4 shows the experimental five-story building with a size of  $60 \times 60$  m used to evaluate the developed localization. For Wi-Fi fingerprint data collection, RSSI measurements and the corresponding location data were collected along the corridor at intervals of 1.5 m, and a total of 2207 learning data and 100 test data measurements were collected distributed among the five floors. The RSSI data dimensionality is determined by the number of routers  $d$ , the total number of routers found in this test building is  $d = 508$ , and the amount of data is  $N = 2207$ . The 2D location data were processed according to the Universal Transverse Mercator coordinate system, and the floor values are integers from 1 to 5.



**Figure 4.** Experimental WiFi fingerprint data distribution along hallway of a multi-story building.

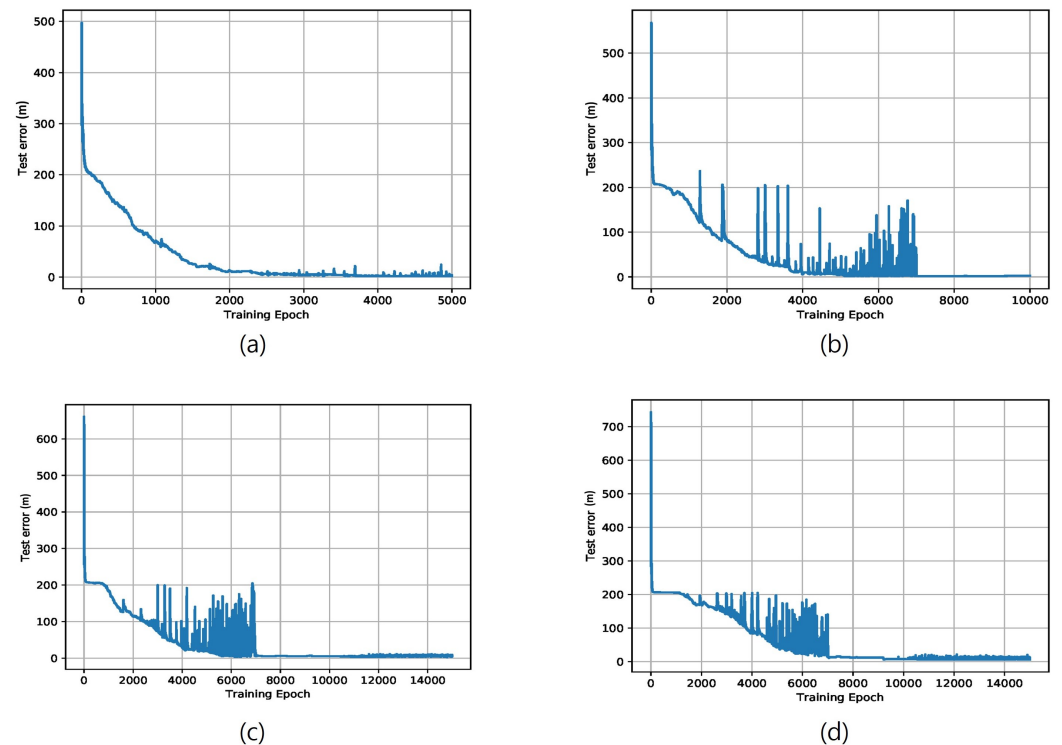
The deep neural network was trained and tested using Tensorflow. The training parameters of the neural network were initialized through the Xavier method, the activation function was defined as the ReLU function, the gradient descent method was performed through the Adam-Optimizer, and the loss function of the training cost function was defined as the root mean square error.

Through feature extraction using PCA, the original data with  $d = 508$  were reduced to feature data with  $l = 14$ . The reason for this setup is that the eigenvalues of the transformed feature data after the fourteenth dimension of the PCA-driven feature data are relatively small.

**3.2. Result**

The structure of the proposed deep neural network for indoor positioning consists of one hidden layer and 220 nodes. This setup was used because when the number of hidden layers was set two or more, the learning performance was degraded.

Figure 5 shows the test performance according to the training epoch when the number of positions to be estimated is 10, 15, 20, and 25, respectively. The test error on the  $y$ -axis is defined as the sum of the Euclidean distance on the two-dimensional coordinates and the 1 m penalty for the three-dimensional coordinates (i.e., floor), and is calculated as the average of 100K number of measurements. Because the output value of the deep network is not limited to within the real space, however, the test error may be very large in the early stages of learning. As shown in Figure 5, over all training results, the test error decreases according to the increment of the training epoch. However, as the number of simultaneous estimated positions increases, the converged test mean error increases.



**Figure 5.** Positioning test error by deep neural network according to the number of estimated positions: (a)  $K = 10$ , (b)  $K = 15$ , (c)  $K = 20$ , (d)  $K = 25$ .

Sometimes, a high test error occurs intermittently during the learning optimization process, which shows the sensitivity of the multi-position neural network model to changes in the neural network's learning parameters.

As the size of the learning rate increases during the learning process, the change rate of the updated parameter values is large. In general, a relatively large learning rate is set in the early stages of learning, and a small rate is set in the later stages to guarantee learning convergence. For our experiment, convergence was secured by changing the learning rate from 0.01 to 0.001 after around 7000 epochs, meaning that the test error plots after 7000 epochs are almost changeless, as in Figure 5.

Table 2 shows the number of training epochs, the duration of training, the number of training data items, and the average error. The mean error/standard deviation error indicated in the fifth column of Table 2 is the average of only the test errors from the last epoch of the learning phase to the 100th time. According to the number of estimated positions increasing or decreasing, the number of training epochs, the training duration, and the amount of data all increase or decrease. This result is obvious in that when the number of multi-targets  $K$  increases, the average error increases. In this experiment, when the number of position estimates was  $K = 10$  and  $K = 15$ , good performance was obtained without much difference from single position estimation accuracy. When  $K > 15$ , however, the estimation performance becomes too inaccurate.

**Table 2.** Experimental results.

Positions K	Learning Iterations	Learning Time (s)	# of Training Data	Ave. Test Error ± Std (m)
1	3500	170.49	2575	2.29 ± 0.002
10	5000	394.43	67,080	2.60 ± 1.55
15	10,000	994.16	67,080	2.68 ± 0.03
20	15,000	1818.18	77,400	6.07 ± 1.48
25	15,000	2474.04	103,200	9.93 ± 2.86

As a baseline to compare accuracy and test time,  $k$ -NN [19] is implemented, in which  $k = 10$  after fine tuning; this approach uses the same PCA-driven feature data as the input. For a single target, the  $k$ -NN experiment achieved an error of 2.69 m, which is similar to the performance of proposed method when the target number was 15. The  $k$ -NN took 70 ms for test time. Because the test-time complexity of  $k$ -NN is  $\mathcal{O}(nm)$ , in which  $n$  and  $m$  are the number of training and test samples, respectively, it consumes time in proportional to increment of the number of targets. On the other hand, the proposed algorithm took 50~55 ms regardless of variations in the number of targets because of the consistent neural network structure.

#### 4. Conclusions

In this paper, WiFi-based indoor positioning using a deep neural network for multi-location learning and estimation is proposed. WiFi wireless signal strength fingerprint original data were pre-processed using PCA to extract feature data and location through standardization for use as input data for a neural network. The input data were further rendered for multi-location learning in an artificial neural network structure. A positioning experiment performed in a five-story building confirmed that accurate learning results were derived when estimating less than 20 locations at the same time.

**Funding:** This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2022R1F1A1070968).

**Data Availability Statement:** Not applicable.

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

#### References

1. Yang, T.; Cabani, A.; Chafouk, H. A Survey of Recent Indoor Localization Scenarios and Methodologies. *Sensors* **2021**, *21*, 8086. [[CrossRef](#)] [[PubMed](#)]
2. Xu, H.; Wu, M.; Li, P.; Zhu, F.; Wang, R. An RFID indoor positioning algorithm based on support vector regression. *Sensors* **2018**, *18*, 1504. [[CrossRef](#)] [[PubMed](#)]
3. Lee, J.; Park, K.; Kim, Y. Deep Learning-Based Device-Free Localization Scheme for Simultaneous Estimation of Indoor Location and Posture Using FMCW Radars. *Sensors* **2022**, *22*, 4447. [[CrossRef](#)] [[PubMed](#)]
4. Yiu, S.; Yang, K. Gaussian process assisted fingerprinting localization. *IEEE Internet Things J.* **2015**, *3*, 683–690. [[CrossRef](#)]
5. Xie, Y.; Wang, Y.; Nallanathan, A.; Wang, L. An improved K-nearest-neighbor indoor localization method based on spearman distance. *IEEE Signal Process. Lett.* **2016**, *23*, 351–355. [[CrossRef](#)]
6. Li, S.; Welsen, S.; Brusica, V. Multi-AP and Test Point Accuracy of the Results in WiFi Indoor Localization. *Sensors* **2022**, *22*, 3709. [[CrossRef](#)] [[PubMed](#)]
7. Boschi, A.; Salvetti, F.; Mazzia, V.; Chiaberge, M. A Cost-Effective Person-Following System for Assistive Unmanned Vehicles with Deep Learning at the Edge. *Machines* **2020**, *8*, 49. [[CrossRef](#)]
8. Abbas, M.; Elhamshary, M.; Rizk, H.; Toriki, M.; Youssef, M. WiDeep: WiFi-based accurate and robust indoor localization system using deep learning. In Proceedings of IEEE International Conference on Pervasive Computing and Communications, Kyoto, Japan, 11–15 March 2021; pp. 1–10.
9. Wang, X.; Gao, L.; Mao, S. BiLoc: Bi-modal deep learning for indoor localization with commodity 5GHz WiFi. *IEEE Access* **2017**, *5*, 4209–4220. [[CrossRef](#)]
10. Zhang, W.; Sengupta, R.; Fodero, J.; Li, X. DeepPositioning: Intelligent fusion of pervasive magnetic field and WiFi fingerprinting for smartphone indoor localization via deep learning. In Proceedings of IEEE International Conference on Machine Learning and Applications, Cancun, Mexico, 18–21 December 2017; pp. 7–13.



11. Rizk, H.; Torki, M.; Youssef, M. CellinDeep: Robust and accurate cellular-based indoor localization via deep learning, *IEEE Sens. J.* **2018**, *19*, 2305–2312. [[CrossRef](#)]
12. Chen, Z.; Zou, H.; Yang, J.; Jiang, H.; Xie, L. WiFi fingerprinting indoor localization using local feature-based deep LSTM. *IEEE Syst. J.* **2020**, *14*, 3001–3010. [[CrossRef](#)]
13. Ma, Y.; Zhang, Y.; Wang, B.; Ning, W. SCLA-RTI: A novel device-free multi-target localization method based on link analysis in passive UHF RFID environment. *IEEE Sens. J.* **2020**, *21*, 3879–3887. [[CrossRef](#)]
14. Zhou, X.; Jiang, K.; He, D.; Liang, L.; Li, Y.; Shi, X. NL-Locate: An Inaudible Multi-Targets Indoor Localization for IoT Devices via Microphone Nonlinearity. *IEEE Sens. J.* **2021**, *21*, 24879–24889. [[CrossRef](#)]
15. Tan, S.; Zhang, L.; Wang, Z.; Yang, J. MultiTrack: Multi-user tracking and activity recognition using commodity WiFi. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; pp. 1–12.
16. Nowicki, M.R.; Skrzypczyński, P. A multi-user personal indoor localization system employing graph-based optimization. *Sensors* **2019**, *19*, 157. [[CrossRef](#)]
17. Peng, P.; Yu, C.; Xia, Q.; Zheng, Z.; Zhao, K.; Chen, W. An Indoor Positioning Method Based on UWB and Visual Fusion. *Sensors* **2022**, *22*, 1394. [[CrossRef](#)] [[PubMed](#)]
18. Christopher, C.B. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
19. Bi, J.; Wang, Y.; Li, X.; Qi, H.; Cao, H.; Xu, S. An adaptive weighted KNN positioning method based on omnidirectional fingerprint database and twice affinity propagation clustering. *Sensors* **2018**, *18*, 2502. [[CrossRef](#)] [[PubMed](#)]