

Article

Side-Information-Aided Preprocessing Scheme for Deep-Learning Classifier in Fingerprint-Based Indoor Positioning

Yue Liu, Rashmi Sharan Sinha , Shu-Zhi Liu and Seung-Hoon Hwang * 

Division of Electronics and Electrical Engineering, Dongguk University-Seoul, Seoul 04620, Korea; tiancaiyue@dongguk.edu (Y.L.); rashmisinha@dongguk.edu (R.S.S.); shuzhiliu@dongguk.edu (S.-Z.L.)

* Correspondence: shwang@dongguk.edu; Tel.: +82-2-2260-3994

Received: 17 April 2020; Accepted: 8 June 2020; Published: 12 June 2020



Abstract: Deep-learning classifiers can effectively improve the accuracy of fingerprint-based indoor positioning. During fingerprint database construction, all received signal strength indicators from each access point are combined without any distinction. Therefore, the database is created and utilised for deep-learning models. Meanwhile, side information regarding specific conditions may help characterise the data features for the deep-learning classifier and improve the accuracy of indoor positioning. Herein, a side-information-aided preprocessing scheme for deep-learning classifiers is proposed in a dynamic environment, where several groups of different databases are constructed for training multiple classifiers. Therefore, appropriate databases can be employed to effectively improve positioning accuracies. Specifically, two kinds of side information, namely time (morning/afternoon) and direction (forward/backward), are considered when collecting the received signal strength indicator. Simulations and experiments are performed with the deep-learning classifier trained on four different databases. Moreover, these are compared with conventional results from the combined database. The results show that the side-information-aided preprocessing scheme allows better success probability than the conventional method. With two margins, the proposed scheme has 6.55% and 5.8% improved performances for simulations and experiments compared to the conventional scheme. Additionally, the proposed scheme, with time as the side information, obtains a higher success probability when the positioning accuracy requirement is loose with larger margin. With direction as the side information, the proposed scheme shows better performance for high positioning precision requirements. Thus, side information such as time or direction is advantageous for preprocessing data in deep-learning classifiers for fingerprint-based indoor positioning.

Keywords: fingerprint-based indoor positioning; preprocessing; side information; dynamic environment; deep-learning classifier

1. Introduction

With the dual development of wireless communications and smart mobile devices, location-based services (LBSs) have developed rapidly and shown broad market prospects [1]. LBSs have penetrated most aspects of daily life and working environment, affording great convenience to people. In general, an LBS can be divided into outdoor and indoor positioning services according to the positioning location. With the maturity of the outdoor positioning technology, research in high-precision positioning has gradually shifted toward indoor locations [2]. At present, the leading indoor positioning technologies involve infrared, wireless local area network (WLAN), ultra-wideband, radio frequency identification, ultrasonic, and Bluetooth technologies [3]. WLAN refers to Wi-Fi networks that communicate over a wireless medium in a local area and has the advantages of full area coverage and low installation

cost [4]. Therefore, Wi-Fi-based indoor positioning has been widely studied and applied. Wi-Fi-based positioning techniques can be divided into two types: the spatio-temporal attribute method and the received signal strength indicator (RSSI)-based scheme. The latter is also known as fingerprint recognition and is one of the most popular indoor location techniques. In the fingerprint-based indoor positioning system, a k-nearest neighbor (KNN) approach is commonly used, which is simple for operation and does not require many data points for training. However, for the KNN, the output completely relies on the nearest neighbors. That is, it is sensitive to the noise and the multipath loss [5]. When an error occurs in measuring the nearest neighbor, it is necessary to manually impute the missing values, which complicates the positioning process and deteriorates its accuracy. To solve the problem, a deep-learning method was proposed in the fingerprint-aided positioning and its performance was evaluated in [6,7]. In [8], the convolutional neural network (CNN) was proposed to show the applicability of complex image classification to the emerging fields. Thus, many studies have adopted the CNN classifier to achieve improved positioning results. As described in [9], a CNN-based indoor fingerprint positioning framework was introduced to improve the accuracy of fingerprint positioning. In our past research [10], we compared the performance of our CNN framework with other CNN frameworks for fingerprint positioning; our results showed that our constructed framework provided better performance for fingerprint positioning than conventional methods. The fingerprint positioning system with a deep-learning classifier is shown in Figure 1. The Wi-Fi-based indoor fingerprinting positioning system has two phases: the online and offline phases. First, in the offline phase, a fingerprint database was built by collecting RSSI via smartphones, and the processed database was then inputted to the CNN model to train the classifier. The trained classifier is then passed to the online phase. Second, in the online phase, the Wi-Fi fingerprint at each unknown position is collected and stored in the database to be matched with the fingerprint for the test point by the trained classifier. The trained classifier subsequently returns the decision on the user's location, and the decision results are passed to the user's smartphone.

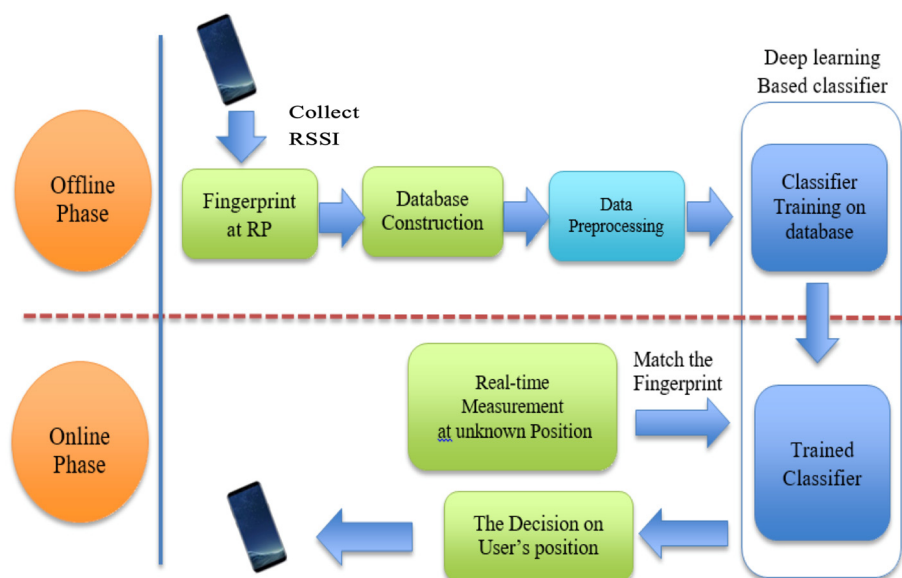


Figure 1. The fingerprint positioning with deep learning classifier.

Currently, the critical issues in Wi-Fi fingerprint-based indoor positioning systems are related to dynamic environments, such as fading of multipath propagation signals caused by obstacles, pedestrians near the user location, and addition/deletion of Wi-Fi access points (APs) [11]. Therefore, it is necessary to consider methods to deal with positioning errors arising from dynamic environments [12]. Note that several environmental factors, such as human activity, temperature, and humidity, are not easily controlled in actual experiments [13]. In [14], it was noted that a database built in a static

environment might not be suitable for positioning tests in a dynamic environment. Therefore, it would be necessary to build the fingerprint database under various conditions. For the deep-learning classifier, a large-scale database is usually constructed by collecting RSSI values under various conditions, so that the impact of the dynamic factors is reduced. However, in [15,16], where RSSI values were collected at different times (morning/afternoon) and directions (forward and reverse), it was shown that building a large-scale database may not always be suitable for deep-learning classifiers in dynamic environments. The authors of [9] observed that it was not necessary to learn all the features of a reference point (RP) to improve positioning accuracy and, hence, more advantageous to selectively learn the key features of an RP in the CNN model. In [17], the authors proposed to divide the database structure for the location area according to information that is beneficial for positioning accuracy. Therefore, we believe that it would be helpful for the deep-learning classifier to decide the position to separately store the RSSI features. In [18,19], it was shown that design of a suitable classifier and correct fingerprint feature learning for fingerprint positioning was essential. In [20], it was shown that, in a dynamic environment, the fingerprint recognition points could be improved by analysis of the time and space for the location. Further, it was shown in [7] that the preprocessing technique could improve positioning accuracy performance. In order to reduce the impact of the dynamic environment on the data fingerprint database and to improve the positioning accuracy, we subdivided the database according to different side information and then selected the database suitable for the current environment for positioning according to the positioning conditions. Therefore, in the preprocessing stage of the database generation, the collected data were classified and processed according to different conditions to construct multiple types of databases for training the classifier. In this study, we propose a side-information-aided preprocessing scheme that utilises information such as time or direction to train the deep-learning classifier. In other words, the side information was used to segment and construct the multiple databases for training classifiers to evaluate the performance of the system. This not only reduces the impact of dynamic RSSI values from the same APs for feature learning but also improves the positioning accuracies. To verify its usefulness, we categorise four types of data sets, namely morning/afternoon and forward/backward. Database preprocessing is an effective method to improve fingerprint positioning. In a previous work, we built a large-scale database by collecting a large quantity of data and used data augmentation preprocessing to improve the positioning accuracies [6]. However, we observed that the fingerprints measured at different times and directions were different, so we proposed a preprocessing method to classify the database based on the side information and in combination with the data augmentation method to effectively improve the positioning success rates. The positioning results were analysed for both simulations and real-time experiments. The remainder of this paper is organised as follows. In Section 2, the system model is described. The simulation and real-time experiment results are discussed in Section 3. Finally, Section 4 summarises the conclusions of our work.

2. System Model

2.1. Environment Setup

The user divides the collection space into several smaller grids, where the size of each grid is determined as 1 m^2 and 4 m^2 centred at the collector. The size of the grid can affect the accuracy of positioning. That is, the denser the grid area, the higher the positioning accuracy is and the longer the time required to collect the signal samples is. The environment setup for a floor map with 74 reference points is shown in Figure 2. For the experiments, the RSSI data collection and experiments were carried out on the seventh floor of the Engineering Building of Dongguk University, Seoul, Korea. The $52 \times 32\text{ m}$ area was divided into 74 reference points (RPs), and the interval between any two RPs was 2 m. For the hardware, we adopted the Dell Alienware model P31e as the positioning server. Meanwhile, a Samsung SM-E310K smartphone was used to measure the RSSI values of the available APs on each RP. In terms of the software, the RSSI database, data preprocessing process,

deep-learning classifier and online experiment programme were constructed using Python and the TensorFlow platform.



Figure 2. Environment setup for floor map with 74 reference points.

2.2. CNN Model and Data Augmentation

In our work, the deep-learning code was designed for comma-separated value (CSV) input files; hence, we converted the collected data files into CSV files for training and testing. The training and testing CSV files contained 257 columns. In column 257, a total of 74 RPs were used as labels. As shown in Figure 3, the NN represents the RSSI value. For each RP, the AP has a unique media access control (MAC) address. The programme sets each MAC address that can be detected into a separate column. At the same time, the corresponding RSSI values are placed under the MAC addresses. In the process of filling all the RSSI values of the RPs into the corresponding MAC addresses, if the RSSI values of certain MAC addresses do not exist in a certain RP, the corresponding MAC columns are filled with zeros. When the programme arranges all the RSSI values of the 74 RPs into the corresponding MAC columns, if the total number of columns in the CSV file is less than 256, the remaining columns are filled with zeros. If the total number of columns is greater than 256, the MAC address column with the lowest RSSI value is deleted. If the total number of columns in the final CSV file is equal to 256, the reference data set is complete.

	A	B	C	D	E	F	G
1	MAC ₁	MAC ₂	MAC ₃	MAC ₄	MAC ₅	MAC ₆	MAC ₇
2	NN ₁₁	NN ₁₂	NN ₁₃	NN ₁₄	NN ₁₅	NN ₁₆	NN ₁₇
3	NN ₂₁	NN ₂₂	NN ₂₃	NN ₂₄	NN ₂₅	NN ₂₆	NN ₂₇
4	NN ₃₁	NN ₃₂	NN ₃₃	NN ₃₄	NN ₃₅	NN ₃₆	NN ₃₇
5	NN ₄₁	NN ₄₂	NN ₄₃	NN ₄₄	NN ₄₅	NN ₄₆	NN ₄₇
6	NN ₅₁	NN ₅₂	NN ₅₃	NN ₅₄	NN ₅₅	NN ₅₆	NN ₅₇

II	IU	IV	IV
MAC ₂₅₄	MAC ₂₅₅	MAC ₂₅₆	256
NN ₁₂₅₄	NN ₁₂₅₅	NN ₁₂₅₆	L ₁₁
NN ₂₂₅₄	NN ₂₂₅₅	NN ₂₂₅₆	L ₁₂
NN ₃₂₅₄	NN ₃₂₅₅	NN ₃₂₅₆	L ₁₃
NN ₄₂₅₄	NN ₄₂₅₅	NN ₄₂₅₆	L ₁₄
NN ₅₂₅₄	NN ₅₂₅₅	NN ₅₂₅₆	L ₁₅

Figure 3. Comma-separated values (CSV) file format used as the input file [6].

The CNN model used in this work is shown in Figure 4, with five layers of network structure [6]. This five-layer network is used to predict the 74 classes. For each RP, the RSSI values of the 256 APs will be recorded. The RSSI values of these different APs are collated into 16×16 images. As shown in the input in Figure 4, different RSSI values are represented by different brightnesses. The greater the RSSI value, the higher the brightness is. That is, the RSSI value of 0 represents black. In our CNN model, the first layer has an input grayscale image of size $16 \times 16 \times 1$, rectified linear unit (ReLU) and dropout. Because of the small size of the input data set, the first layer does not use maximum pooling, and the second layer consists of a 16×16 convolution with ReLU, which then consists of a max pooling layer of size 8×8 , with a total of 18,496 parameters. This produces an output for the third 8×8 convolutional layer with ReLU and a 4×4 max pooling layer. This output feeds directly into a fully connected (FC) layer with 3072 nodes, which results in the next hidden FC layer with 1024 nodes. Finally, a softmax layer with 74 nodes is used to calculate the output, which is the number of RPs in the setup [6]. In turn, 74 RPs are used as input to the CNN to analyse its data characteristics. The input RSSI files for the 74 RPs will produce different images for input into the CNN model. Data augmentation is a useful technique which is employed for training general-purpose deep learning classifiers. It can speed up convergence or act as a regularizer, thereby avoiding overfitting and increasing generalization [21]. The data augmentation usually involves applying a set of transformations to the data space or feature space, or both. The most common augmentation generates new samples, which creates a larger number of data sets to prevent overfitting, to normalize the model, to balance classes in the database, and even to synthesize more representative new samples for use cases or tasks at hand. In this work, we plan to use data augmentation schemes to expand the existing data so that deep-learning algorithms could more effectively extract the necessary features. As more data input is beneficial for the CNN model to learn data features, the data is augmented. In [6], two data augmentation schemes were discussed for deep-learning architectures. In our work, an augmentation scheme that randomly changes the original data value within a difference range based on the difference between the calculated average value and original data is used in the data augmentation method. In our work, an augmentation scheme randomly changes the original data value within the determined range using mean and uniform random numbers to increase the number of the RSSI values in the reference data set. To identify the potential range for the uniform random number generator, the mean RSSI value for each RP is evaluated and then the current RSSI value is subtracted from it. The global repetition number is declared as N. The input data set is established in the form of CSV files. For example, let the global N be 60, the current RSSI be 33, and the mean value be 45. The potential range for the uniform random numbers becomes $12 (= 45 - 33)$, which generates any numbers within the range of 33 to 45 sixty times. If the range is equal to zero, the current RSSI value is repeated sixty times.

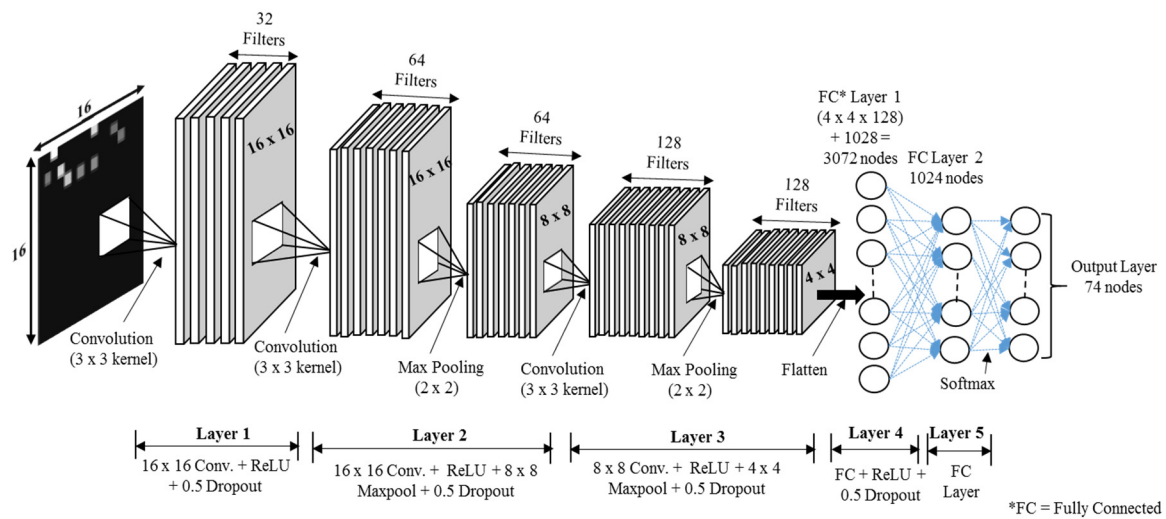


Figure 4. CNN architecture used in this study [6].

2.3. RSSI Database Setup

The Wi-Fi fingerprint-based indoor positioning system is divided into two phases: online and offline phases. The offline phase is used for database generation and training of the CNN classifiers. In the online phase, the trained CNN classifier is applied to the actual positioning experiment. In the offline phase, the RSSI fingerprint of each reference point is mapped using the smartphone and computer to build the database. The database in the offline phase consists of two types of databases, namely the trial database and training database. The trial database is used to perform the offline test simulation. The training database is used for training the deep-learning classifier. During the data collection stage, the RSSI values are collected for different times and directions. The specific data collection plan is as follows. In the morning, we conduct forward and backward data collections, respectively. We do the same thing in the afternoon. Forward data collection refers to collecting the RSSI value of each RP, from RP1 to RP74, in sequence. Backward data collection is the opposite of forward data collection, which collects the RSSI value of each RP from RP74 to RP1 sequentially. For data collection in each direction, the RSSI values of each RP are collected five times. The sampling time for each RP measurement is 5 s, which means 25 s for a total of five measurements. The data collection is performed for seven consecutive days. In total, about 28 RSSI data files are obtained. Since time and direction are employed as side information for the preprocessing stage, the collected data files are classified into four data sets: morning, afternoon, forward, and backward. Each data set includes 14 data files. The data set structure is shown in the following table. In Table 1, M and A indicate the time of data collection for morning and afternoon, respectively, and F and B indicate the direction of data collection for forward and backward, respectively. Finally, the numbers represent the days for which data were collected. Note that we combine all 28 data files to form a conventional data set to generate the conventional databases.

Table 1. Types of data set.

Data Set	Forward	Backward	Number of Data Files
Morning	MF1, MF2, ..., MF7	MB1, MB2, ..., MB7	14
Afternoon	AF1, AF2, ..., AF7	AB1, AB2, ..., AB7	14
Number of Data Files	14	14	-

Here, we divide the data files in each set into two groups, and one group forms the training database while the other group forms the trial database. For each data set, except for the conventional data set, we select 13 data files to perform the data augmentation in [6] for the training database and

one rest data file for the trial database. For the conventional database, we select 24 data files, except for four data files used to generate the trial database, to perform data augmentation and then create a conventional training database. The specific division method and database types are shown in Table 2. During the training database construction phase, we performed four repeated measurements on the 74 RP points in one day (forward and backward measurements in the morning and afternoon, respectively) and measured the RSSI values of each RP five times for each measurement. Each data file thus contains 74 kinds of RPs, and each RP is measured five times. Therefore, each file contains 370 RPs. For each training database except the conventional database, there are 13 data files, and then, after the data augmentation scheme, each training database contains 288,600 RPs whose size is approximately 250 MB. The conventional training database contains 24 data files, and, after data augmentation, it contains a total of 532,800 RPs and the size is 350 MB.

Table 2. Types of the databases and structures.

Data Set	Database	
	Training	Test
Morning	13 data files	1 data file
Afternoon	13 data files	1 data file
Forward	13 data files	1 data file
Backward	13 data files	1 data file
Convention	24 data files	4 data file

2.4. Side-Information-Aided Preprocessing Scheme

The basic idea is that, in the offline phase, we input the training database constructed with side information and the trial database into the CNN model to train the classifier and select the classifier with the highest success rate among all simulation results. Then, we pass this classifier to the online phase. In the online phase, real-time measurement information on unknown user location is passed to the trained classifier. The trained classifier returns the user's location decision, which is then passed on to the user's smartphone. In [15] and [16], we found that when using conventional methods to test positioning performance, there were large differences between the positioning results obtained during the morning and afternoon. There was also a gap between the forward and backward test results that cannot be ignored. Therefore, we believe that the difference in test time and direction can affect the positioning results. We subdivide the fingerprint database according to the side information of time and direction and then use different databases to perform positioning tests under the corresponding test conditions. The specific test plan is shown as follows. The plan is divided into simulation, that is, an offline phase, and experiment, which is an online phase. In the simulation (offline phase), we use a mobile phone to collect the RSSI value of each RP and store it as a text file. We call the collected text files as data sets and then collate the collected data sets into CSV files. We call these CSV files as databases of two types: training databases and trial databases. Both training and trial databases are preprocessed using the proposed preprocessing method. Since we choose time and direction as side information to perform the preprocessing, here, we have created different databases based on morning and afternoon, as well as forward and backward. The two types of databases constructed with side information are simultaneously passed to the CNN model for feature learning. The specific process involves taking the training database and trial database as inputs and then performing feature learning in the CNN model. In our work, the Python-based CNN classifier produces a metafile after each epoch, which is a set of weights and biases. After multiple epochs, many metafiles are generated, and we select the metafile with the highest success rate for real-time positioning experiments (online phase). Based on the four kinds of side information, we perform four corresponding simulation tests. At the same time, we also use the conventional method for simulation. In the experiment plan (online phase), we use the classifier trained in the simulation phase in the software used for the experiment and later use the mobile phone and computer for location testing. In the actual measurements, RSSI values are collected

at each unknown RP point, and the collected values are inputted to a trained classifier for comparative analysis to determine user position. For each experiment, we obtained five RSSI measurements at the same RP point and then the trained classifier returned five positioning decisions. Since we chose time and direction as side information to perform the preprocessing, we also strictly followed the side information requirements when conducting experiments. For example, in the morning, the classifier trained on the morning database was used to perform the positioning experiment in the front–rear direction. To verify the feasibility of this preprocessing, we used the conventional method to conduct experiments for the same time and direction.

We use a database constructed with side information to perform the positioning tests and analyse the positioning performances from simulated and real-time experimental results. We call this scheme the side-information-aided preprocessing scheme. The simulation and experimental environments of the side-information-aided preprocessing scheme are shown in Table 3. For the real-time experiment in each case, we conducted four tests over two days. We also used a conventional training database to simulate and perform a two-day real-time experiment, which we call the conventional scheme. We compare the results under the two schemes to verify the feasibility of the proposed design.

Table 3. Simulation and experiment environments.

	Simulation		Experiment	
	Training Database	Test Database	Trained Classifier	Test Timing, Direction
case 1	Morning	Morning	Morning	Morning (Forward/Backward)
case 2	Afternoon	Afternoon	Afternoon	Afternoon (Forward/Backward)
case 3	Forward	Forward	Forward	Forward (Morning/Afternoon)
case 4	Backward	Backward	Backward	Backward (Morning/Afternoon)

3. Numerical Results

3.1. Simulation Results

For the side-information-aided preprocessing scheme, we performed simulation tests as follows. We inputted the training database into the CNN model with the help of the side information, trained the classifier through data feature learning, and inputted the corresponding trial database into the CNN model to verify the positioning performance of the trained classifier. Additionally, the simulation for the conventional scheme was performed. We inputted a conventional training database to the CNN model to train the classifier and then used the trial database. The positioning performances were compared under two schemes. In the simulation, we employed the success probability to describe the positioning accuracy. When using the trial database as the input, the success probability is the probability of correct position prediction with an error margin of two RPs. As mentioned earlier, the interval between two consecutive RPs is 2 m, so the precision of success was 4 m in this work. Here, we used three different margins (0, 1, and 2) to analyse the simulation results. The 0 margin indicates that the decision obtained at a certain RP position is entirely correct; the 1 margin means that the positioning decision result is within the range of (RP−1, RP+1), and the 2 margin indicates that the positioning decision value is within the range of (RP−2, RP+2). Since there are 74 RPs in the database, the total number of samples is 74. The success probability of the classifier is thus given as follows [7]:

$$\text{Success probability [\%]} = \frac{\text{the number of success count}}{\text{total number of samples}} \times 100 \quad (1)$$

In the CNN model, the simulation results with the highest success probabilities were selected for comparisons and analyses. The simulation results refer to the success probabilities calculated by matching the 74 RPs in the trial database with the 74 RPs in the training database individually. The simulation results, in terms of success probabilities, are shown in Table 4. Here, we call the side-information-aided preprocessing scheme as the proposed scheme. It can be seen from all the results in Table 4 that, irrespective of 0, 1, or 2 margins, all types of databases preprocessed with side information can achieve better performance than conventional databases. With the margin of two RPs, the success probabilities of the proposed scheme for cases 1 and 2 (time as the side information for preprocessing) are higher than those of the conventional scheme by 6.55% and 4.97%, respectively. For example, according to Table 4, with margin 2, the positioning success probability of case 1 is 92.64% and that for the conventional case is 86.09%; thus, the difference between the two cases is 6.55%. Meanwhile, the success probabilities of the proposed scheme for cases 3 and 4 (direction as the side information for preprocessing) are higher than those for the conventional scheme by 3.07% and 4.42%, respectively.

Table 4. Simulation results.

Margin	Proposed Scheme				Conventional Scheme
	Case 1	Case2	Case3	Case4	
0	46.38%	45.58%	49.86%	50.03%	41.62%
1	79.89%	78.65%	80.76%	80.49%	73.85%
2	92.64%	91.06%	89.16%	90.51%	86.09%

The principal component analysis (PCA) plot for training and test data is shown in Figure 5. In the Figure 5, graphs a–d correspond to four different positioning cases for the side-information-aided preprocessing scheme. The green points in the figure represent the reference points in the training database, and the red points represent the unknown location points (test data set). We analyze the numerical characteristics of the unknown points in the test database, and then compare them with the RP characteristics in the trained classifier. As shown in Figure 5, the green point closest to the red point is considered the final positioning decision.

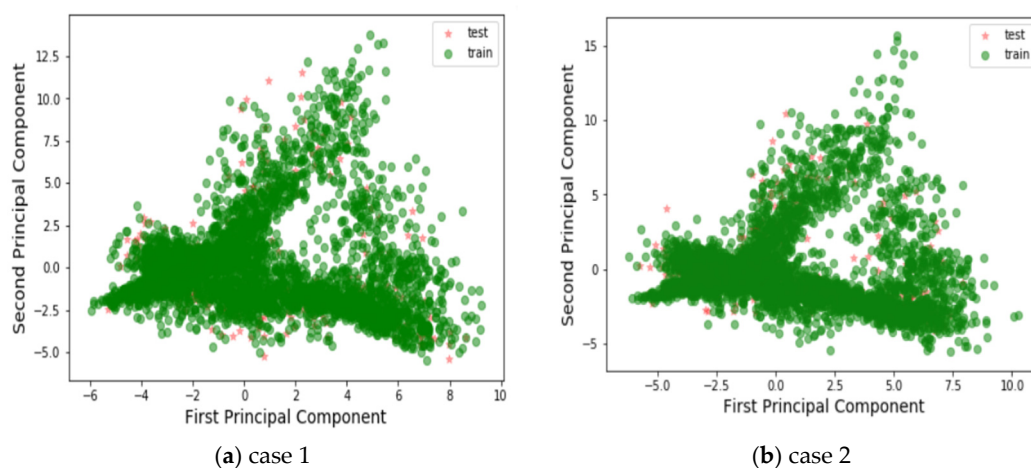


Figure 5. Cont.

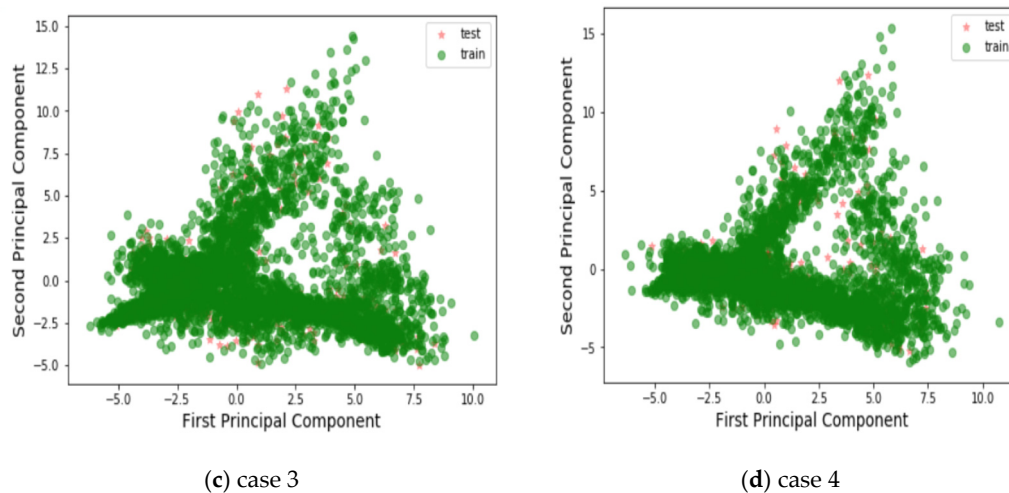


Figure 5. Principal component analysis plot. (a) Case 1- PCA plot for morning. (b) Case 2- PCA plot for afternoon. (c) Case 3- PCA plot for forward. (d) Case 4- PCA plot for backward.

3.2. Experimental Results

For real-time experiments, we use the trained classifier for location testing. The specific process involves inputting the measured RSSI value of the unknown point into the CNN model to analyse the numerical features. Then, the obtained features are compared with those in the trained classifier. The RP with the most similar features in the trained classifier is determined as the final position decision. Based on the side-information-aided preprocessing scheme, we used four different trained classifiers to perform the forward and backward positioning tests in the morning and afternoon as planned. For this scheme, we performed a total of 16 positioning experiments and conducted four experiments in each case. According to the conventional scheme, the eight-time positioning experiment was carried out using the conventionally trained classifier under the same conditions as the side-information-aided preprocessing scheme. When processing the experimental results, we stipulated that the test results with measurement errors less than or equal to 2 were successful, and the data represent the probability of successful results obtained in the measurement results. Table 5 shows the experimental results. As shown in Table 5, for each RP positioning test in an experiment, five positioning decisions were performed continuously for the same RP.

Table 5. Example of experimental results.

NO.	Decision #					# of Success Decisions		
RF#	#1	#2	#3	#4	#5	0 Margin	1 Margin	2 Margin
1	1	1	1	1	1	5	5	5
2	2	3	3	2	2	3	5	5
3	2	4	2	2	2	0	5	5
73	73	26	73	73	72	3	4	5
74	73	74	73	73	73	1	5	5
Experiment Success Rate (%)						27.57	71.08	88.11

In order to facilitate comparison between the results of the side-information-aided preprocessing scheme and the results of the conventional scheme, we merged all the experimental results into Table 6, which depicts the data results summarised according to different positioning times and directions. When analysing the data for all cases, we averaged the forward and backward results for cases 1 and 2 and averaged the morning and afternoon results for cases 3 and 4, respectively. The average results of the experiments are shown in Table 7. According to the results in Table 7, using the side-information-aided preprocessing for the classifier in the real-time experiments can achieve better positioning performance.

Specifically, with the margin of two RPs, the side-information-aided preprocessing scheme achieved a gain of 4.93% to 5.8% as compared to the conventional positioning scheme. It is worth noting that, for cases 1 and 2, the performance differences between the proposed and conventional schemes increase as the margin increases. Meanwhile, for cases 3 and 4, the differences between the two schemes decrease as the margin increases. This means that, as the requirements for positioning accuracy become looser with larger margins, the difference in terms of success probability becomes larger when using time as the side information. In other words, the proposed scheme using time as the side information achieve higher gain with lower precision. Meanwhile, under the requirement of high-precision positioning, the proposed scheme, using direction as the side information, achieves higher gain. For example, for cases 3 and 4 with a margin of 0, the proposed scheme using the forward and backward directions as the side information shows gains of 13.72% and 15.06%, respectively. This means that the proposed scheme with time as the side information enables a higher success probability when the requirement of positioning accuracy is loose. Further, with direction as the side information, the proposed scheme shows better performance for the high positioning precision requirement.

Table 6. Summary of experimental results (%).

			Proposed Scheme			Conventional Scheme		
Margin			0	1	2	0	1	2
Case 1	Day 1	Forward	27.57	71.08	88.11	33.51	69.46	85.41
		Backward	35.95	77.03	93.24	25.41	67.84	85.14
	Day 2	Forward	28.65	71.62	89.73	34.32	70.54	86.49
		Backward	34.59	75.95	92.16	26.22	68.65	86.22
Case 2	Day 1	Forward	34.32	73.78	88.65	33.24	70.27	82.43
		Backward	30.00	71.62	90.27	33.78	73.78	87.03
	Day 2	Forward	34.86	75.95	89.73	31.08	64.86	83.24
		Backward	28.38	69.73	89.46	28.65	70.54	85.68
Case 3	Day 1	Morning	46.76	81.08	89.46	33.51	69.46	85.41
		Afternoon	48.65	81.35	89.19	33.24	70.27	82.43
	Day 2	Morning	42.16	76.22	91.62	34.32	70.54	86.49
		Afternoon	49.46	82.16	90.00	31.08	64.86	83.24
Case 4	Day 1	Morning	43.78	83.24	94.59	25.41	67.84	85.14
		Afternoon	42.70	82.16	93.78	33.78	73.78	87.03
	Day 2	Morning	47.57	78.38	88.11	26.22	68.65	86.22
		Afternoon	40.27	78.11	90.81	28.65	70.54	85.68

Table 7. Average results.

		Proposed Scheme	Conventional Scheme	Difference
Case 1	0 Margin	31.69%	29.87%	1.82%
	1 Margin	73.92%	69.12%	4.80%
	2 Margin	90.81%	85.82%	4.99%
Case 2	0 Margin	31.89%	31.69%	0.20%
	1 Margin	72.77%	69.86%	2.91%
	2 Margin	89.53%	84.60%	4.93%
Case 3	0 Margin	46.76%	33.04%	13.72%
	1 Margin	80.20%	68.78%	11.42%
	2 Margin	90.07%	84.39%	5.68%
Case 4	0 Margin	43.58%	28.52%	15.06%
	1 Margin	80.47%	70.20%	10.27%
	2 Margin	91.82%	86.02%	5.8%

It is generally known that multiple sampling can help reduce the multipath effect on RSSI [22,23]. Therefore, a large-scale comprehensive database construction method is used to generate the fingerprint database. However, owing to the indiscriminate learning of data with multiple information, the unique features for the side information may be ignored. Therefore, it is necessary to build various databases based on the types of side information and to select a suitable database for each test condition. When constructing the database, it was found that some RSSI values of the adjacent RPs collected from different directions were the same. For example, a certain RSSI value of RP1 collected in the forward direction is identical to a certain RSSI value of RP2 collected in the backward direction. This is not conducive to the CNN model training classifier. For the RSSI values collected in different directions, the multipath effects may also be different [14]. Therefore, we consider that the database constructed with RSSI values collected at different times or in different directions can reduce the effects of dynamic environments. In the same environment, the data collected at different times may have different characteristics. Therefore, if indoor positioning is obtained at different times, it may be best to use a database positioning built at the corresponding time. Since the RSSI value is affected by various factors, we need to consider different conditions to generate multiple databases. Through location testing, we can select the database that is best suited to the current positioning environment. When the impact of external factors on positioning cannot be changed, multiple databases can be built based on different external factors, and the trained classifier can then be used for positioning experiments. Here, we choose the classifier that is suitable for the current positioning environment based on the experimental results, which can not only reduce the influence of external factors on positioning but also improve positioning accuracy.

4. Conclusions

In this work, multiple fingerprint databases based on different types of side information were constructed and employed to determine the classifiers that would be trained by the corresponding fingerprint databases. It was shown that selection of a suitable trained classifier for a particular situation could effectively improve indoor positioning accuracies. The simulated and experimental results showed that the side-information-aided preprocessing scheme used herein allowed better positioning success probability than conventional schemes. With two margins, the proposed scheme could improve the simulated and experimental performances by 6.55% and 5.8%, respectively, in comparison to the conventional scheme. The proposed scheme, with time as the side information, also obtained a higher success probability when the positioning accuracy requirement was loose with a larger margin. Furthermore, with direction as the side information, the proposed scheme showed better performance for the high positioning precision requirement. As a future work, we plan to consider other factors, such as weather, temperature, humidity, and traffic, to establish more positioning test databases and to investigate which of these databases could better handle positioning experiments under various conditions.

Author Contributions: Y.L. and S.-H.H. contributed to the main idea of this research work. Y.L. and R.S.S. performed the simulations, experiments, and database collection. S.-Z.L. confirmed the numerical results of the work. The research activity was planned and executed under the supervision of S.-H.H., and Y.L., R.S.S. and S.-H.H. contributed to the writing of this article. All authors have read and agreed to the published version of the manuscript.

Funding: Following are results of a study on the “Leaders in INdustry-university Cooperation+” Project, supported by the Ministry of Education and National Research Foundation of Korea.

Acknowledgments: The authors would like to express their sincere gratitude to Sang Moon Lee, CTO of JMP Systems, Korea, for providing the equipment for database collection and setting up the experimental environment.

Conflicts of Interest: The authors declare no conflict of interest regarding the publication of this article.

References

- Basri, C.; El Khadimi, A. Survey on Indoor Localization System and Recent Advances of WIFI Fingerprinting Technique. In Proceedings of the IEEE 5th International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Morocco, 29 September–1 October 2016; pp. 253–259.
- Sithole, G.; Zlatanova, S. Position, location, place and area: An indoor perspective. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci* **2016**, *3*, 89. [\[CrossRef\]](#)
- Zafari, F.; Gkelias, A.; Leung, K.K. A survey of indoor localization systems and technologies. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2568–2599. [\[CrossRef\]](#)
- Kitasuka, T.; Nakanishi, T.; Fukuda, A. Wireless LAN Based Indoor Positioning System WiPS and Its Simulation. In Proceedings of the 2003 IEEE Pacific Rim Conference on Communications Computers and Signal Processing (PACRIM 2003), Victoria, BC, Canada, 28–30 August 2003; pp. 272–275.
- Wang, H.P.; Jia, F. A Hybrid Modeling for WLAN Positioning System. In Proceedings of the 2007 International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 21 September 2007; pp. 2152–2155.
- Sinha, R.S.; Lee, S.M.; Rim, M.; Hwang, S.H. Data augmentation schemes for deep learning in an indoor positioning application. *Electronics* **2019**, *8*, 554. [\[CrossRef\]](#)
- Haider, A.; Wei, Y.; Liu, S.; Hwang, S.H. Pre-and post-processing algorithms with deep learning classifier for Wi-Fi fingerprint-based indoor positioning. *Electronics* **2019**, *8*, 195. [\[CrossRef\]](#)
- Chishti, S.O.; Riaz, S.; Zaib, M.B.; Nauman, M. Self-Driving Cars Using CNN and Q-Learning. In Proceedings of the 2018 IEEE 21st International Multi-Topic Conference (INMIC), Karachi, Pakistan, 1–2 December 2018; pp. 1–7.
- Song, X.; Fan, X.; Xiang, C.; Ye, Q.; Liu, L.; Wang, Z.; He, X.; Yang, N.; Fang, G. A novel convolutional neural network based indoor localization framework with WiFi fingerprinting. *IEEE Access* **2019**, *7*, 110698–110709. [\[CrossRef\]](#)
- Sinha, R.S.; Hwang, S.H. Comparison of CNN applications for RSSI-based fingerprint indoor localization. *Electronics* **2019**, *8*, 989. [\[CrossRef\]](#)
- Liu, K.; Wang, Y.; Lin, L.; Chen, G. An analysis of impact factors for positioning performance in WLAN fingerprinting systems using Ishikawa diagrams and a simulation platform. *Mob. Inf. Syst* **2017**, *2017*, 8294248. [\[CrossRef\]](#)
- Xia, S.; Liu, Y.; Yuan, G.; Wang, Z. Indoor fingerprint positioning based on Wi-Fi: An overview. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 135. [\[CrossRef\]](#)
- Çalış, G.; Becerik-Gerber, B.; Göktepe, A.B. Analysis of the variability of RSSI values for active RFID-based indoor applications. *Turk J. Eng. Environ. Sci.* **2013**, *37*, 186–210. [\[CrossRef\]](#)
- Lin, Q.; Xu, Y.; Zhou, M.; Deng, Z.-A.; Liang, Y. Characteristics of Fingerprint Location Technology in WLAN Environment. In Proceedings of the 2009 International Forum on Information Technology and Applications, Chengdu, China, 15–17 May 2009; pp. 40–43.
- Liu, Y.; Wei, Y.; Sinha, R.S.; Hwang, S.H. Influence of Test Methods on Wi-Fi Fingerprint-Based Indoor Positioning System. In Proceedings of the KICS Winter Conference, Yongpyeong, Korea, 23–25 January 2019; pp. 378–379.
- Liu, S.Z.; Haider, A.; Hwang, S.H. Impact of Test Environment on the Accuracy of Wi-Fi Fingerprint-Based Indoor Positioning System. In Proceedings of the KICS Winter Conference, Yongpyeong, Korea, 23–25 January 2019; pp. 380–381.
- Xu, J.; Zhang, H.; Zhang, J. Self-Adapting Multi-Fingerprints Joint Indoor Positioning Algorithm in WLAN Based on Database of AP ID. In Proceedings of the IEEE 33rd Chinese Control Conference, Nanjing, China, 28–30 July 2014; pp. 534–538.
- Zheng, Z.; Chen, Y.; He, T.; Sun, L.; Chen, D. Feature learning for fingerprint-based positioning in indoor environment. *Int. J. Distr. Sens. Netw.* **2015**, *11*, 452590. [\[CrossRef\]](#)
- Song, J.; Hur, S.; Park, Y.; Choi, J. An Improved RSSI of Geomagnetic Field-Based Indoor Positioning Method Involving Efficient Database Generation by Building Materials. In Proceedings of the 2016 IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcalá de Henares, Spain, 4–7 October 2016; pp. 1–8.

20. Wen, X.; Tao, W.; Own, C.M.; Pan, Z. On the dynamic RSSI feedbacks of indoor fingerprinting databases for localization reliability improvement. *Sensors* **2016**, *16*, 1278. [[CrossRef](#)] [[PubMed](#)]
21. DeVries, T.; Taylor, G.W. Dataset augmentation in feature space. *arXiv* **2017**, arXiv:1702.05538.
22. Zhao, F.; Luo, H.Y.; Lin, Q.; Ma, Y. Node localization algorithm based on kernel function and Markov chains. *J. China Inst. Commun.* **2010**, *31*, 195–204.
23. Liu, H.; Darabi, H.; Banerjee, P.; Liu, J. Survey of wireless indoor positioning techniques and systems. *IEEE Trans. Syst. Man Cybern. Part C* **2007**, *37*, 1067–1080. [[CrossRef](#)]



© 2020 by the authors. 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 (<http://creativecommons.org/licenses/by/4.0/>).