



Article A Narrow-Down Approach Based on Machine Learning for Indoor Localization

Sahibzada Muhammad Ahmad Umair 匝 and Tughrul Arslan *

Scottish Microelectronics Centre, School of Engineering, University of Edinburgh, King's Buildings, Alexander Crum Brown Road, Edinburgh EH9 3FF, UK; s2135300@ed.ac.uk * Correspondence: t.arslan@ed.ac.uk

Abstract: Over the past decade, the demand and research for indoor localization have burgeoned and Wi-Fi fingerprinting approach has been widely considered because it is cheap and accessible. However, most existing methods lack in terms of positioning accuracy and high computational complexity. To cope with these issues, we formulate a two-stage, coarse and accurate positioning narrow-down approach (NDA). Furthermore, a three-step source domain refinement (SDR) scheme that involves outlier removal, stable AP's weight enhancement, and a data averaging technique by applying the K-means clustering algorithm is also proposed. The collaboration of SDR scheme with the training data selection, area division, and overlapping schemes reduces the computational complexity and improves coarse positioning accuracy. The effect of the proposed SDR scheme on the performance of the support vector machine (SVM) and random forest algorithms is also presented. In the final/accurate positioning phase, a set of lightweight neural networks (DNNs), trained on different sub-areas, predict the user's location. This approach significantly increases positioning accuracy while reducing the online computational complexity at the same time. The experimental results show that the proposed approach outperforms the best solutions presented in the literature.

Keywords: narrow-down approach (NDA); area division and overlapping (ADO); source domain refinement (SDR); support vector machine (SVM); random forest (RF); distributed neural networks for indoor localization (DNLoc); Wi-Fi fingerprint; Internet of Things (IoT)

1. Introduction

With the advancement in technology, the indoor positioning and tracking of smart devices has gained much popularity in the Internet of Things environment [1]. The demand for indoor localization is higher than that of outdoor localization. Globally, people spend 80 to 90% of their time indoors and about 70% of smartphone usage is in closed areas [2]. A recent report published on American lifestyle states that Americans, on average, spend 93% of their life indoors [3]. Consequently, research on indoor localization is gaining importance day by day. The global positioning system (GPS) is the most popular technique used for outdoor positioning and is commonly used in transport vehicles and smartphones [2]. In indoor environments, GPS exhibits poor performance because of the absorbance and distraction of signals from construction materials. The interpretation of accuracy is not the same between outdoors and indoors. A few meters of inaccuracy indoors is more effective than outdoors, which may lead to a different room or a different building. Traditional location-based systems (LBSs) are not efficient enough to meet the navigational challenges of these GPS-denied environments. No absolute real-time solution has been proposed yet that would be cost-effective, less time-consuming, more accurate, and generic. The Wi-Fi fingerprinting approach is the most popular because of its cost-effectiveness and accessibility [4]. The existing WiFi-RSSI-based positioning algorithms can be classified into geometric-related techniques and fingerprinting-based techniques [5], wherein fingerprintbased methods have superiority because wireless signal variances can be captured more



Citation: Umair, S.M.A.; Arslan, T. A Narrow-Down Approach Based on Machine Learning for Indoor Localization. *Algorithms* **2023**, *16*, 529. https://doi.org/10.3390/a16110529

Academic Editor: Frank Werner

Received: 28 September 2023 Revised: 2 November 2023 Accepted: 9 November 2023 Published: 17 November 2023



Copyright: © 2023 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 (https:// creativecommons.org/licenses/by/ 4.0/). accurately. However, the dynamics of routers, random signal interference, and moving obstacles cause uncertainty in measuring the wireless signal strength at reference points (RPs), which in turn degrades the accuracy of indoor positioning. Mostly machine learning (ML)-based algorithms, like K-nearest neighbor (KNN) [6], naive Bayesian (NB) [7], support vector machine (SVM) [8], random forest (RF) [9], and deep neural network (NN) [10], have been used to find the user's indoor location from the fingerprints. Indoor localization methods can be divided into two categories [11]. (1) Classification methods: The whole localization area is divided into sub-areas and then a classification algorithm finds the sub-area where the target resides. (2) Regression methods: A regression-based algorithm finds the user's exact location by utilizing Wi-Fi-based RSS (received signal strength) vectors. In this article, we aim to reduce the training and response time while increasing positioning accuracy. We propose a narrow-down approach that consists of coarse and accurate positioning phases. To deal with the first problem, in the coarse positioning phase, we propose a source domain refinement (SDR) scheme that reduces 80% of training data for classification. Furthermore, we also divide the whole localization area into sub-areas, as proposed by Jingxue Bi in [12]. To reduce the response time, we select a support vector machine whose response time is far better than the random forest algorithm for classification. We reduce the propagation delay of the regression-based algorithm by making it lightweight and by training it on each sub-area. The combination of SVM, SDR, and the group of distributed neural networks, namely DNLoc, alongside the concept of area division and localization, dramatically increases indoor localization accuracy. The main contributions of this paper are:

- This study proposes a narrow-down approach (NDA), which comprises the coarse and accurate positioning phases.
- The contribution is to select specific reference points (RPs) to train the classification
 algorithm, while the key considerations are to reduce the offline storage as we do
 not use all the RPs for training, and the chosen training points for the classifier
 are distant enough to share minimum RSSI characteristics. This strategy increases
 classification accuracy.
- We also propose a three-step source domain refinement (SDR) scheme to reduce the computational complexity of training data and enhance the classification accuracy at the same time.
- A very lightweight DNN-based multivariate regression (DNN-MVR) model, trained independently on each sub-cluster, is presented. The proposed methods are evaluated on a public dataset to show their reliability and robustness.

We organize the remaining article as follows. Section 2 discusses the related works. Section 3 explains the system design. Section 4 demonstrates the experimental evaluations. Conclusion remarks are discussed in Section 5.

2. Related Work

In the literature, many Wi-Fi RSSI-based machine learning (ML) approaches have been proposed for indoor localization. For example, Nafisa et al. [3] proposed a zone-based indoor localization system using neural networks with a slight modification in traditional counter propagation network (CPN). The proposed scheme reduces the number of empty clusters and performs better than the basic CPN by increasing 1% in accuracy. However, it is lacking in finding the exact user's location coordinates. A hidden Markov model-based indoor localization scheme is proposed in [4], but the random forest algorithm outperforms the proposed method. Zhang et al. [5] proposed a Wi-Fi RSSI-based indoor robot positioning system that is pluggable to existing Wi-Fi network infrastructures. They integrated the deep neural network with fuzzy forests to increase accuracy. If we use both the RSSI's value and direction to train the random forest algorithm, then its accuracy can be increased [13]. Minhui et al. [14] proposed an algorithm to divide the whole localization area into sub-areas by using the Gaussian mixture model. RF algorithm was utilized to predict the corresponding area and the final location was estimated using an adaptive KNN algorithm. Xiang

et al. [15] utilized a deep learning framework alongside a logistic regression algorithm and Pinto et al. [16] utilized the K-means clustering algorithm to divide the localization area into different sets of log-distance propagation models, while Bayesian inference improves the positioning accuracy. A random forest algorithm using a software-defined network (SDN) framework is presented in [17]. The proposed model uses cross-validation for training and performing indoor localization. Dong et al. [18] proposed a novel adaptive cluster splitting (ACS) and access point (AP) reselection scheme in each sub-cluster splitting process. In the online phase, a decision tree-based exhaustive search algorithm finds the user's location. Saddam et al. [19] proposed an algorithm consisting of clustering and searching. A measuring device determines the user's location, based on the strongest AP, in a radio map. An AP's similarity-based clustering approach is proposed in [20]. Li et al. [21] proposed a heterogeneous knowledge transfer framework for fingerprinting-based indoor localization. After removing the redundant knowledge in the source domain, the authors derived a cross-domain mapping to construct a homogeneous feature space, where they combined the mapping and weights learning into a joint objective function and solved it using a threestep iterative optimization algorithm. However, they utilized online fingerprint knowledge to train a model that makes this approach less realistic. Xiansheng et al. [22] presented a robust model by fusing derivative fingerprints of RSS with multiple classifiers (DIFMICs). This model outperforms many machine learning-based models proposed in the literature for indoor localization. Li et al. [23] proposed a probabilistic model to intelligently estimate the user's location by evaluating the label's credibility. Zhang et al. [24] presented a hybrid localization model by joining the convolutional neural network (CNN) and Gaussian process regression (GPR) algorithms. The hybrid model improved in performance by 45.8% and the GPR algorithm further increased the localization accuracy. Soro et al. [25] proposed a wavelet scattering framework (WSF)-based neural network for an indoor localization method that is not affected by the handset orientation, and Li et al. [26] proposed a hybrid fingerprint quality evaluation model (HFQEM) that can find the location by evaluating the hybrid fingerprint quality in different sub-areas. The authors of [27] present a sequence learning problem, where a recurrent neural network (RNN) with a regression output are used to estimate three-dimensional positions. The authors of [28] propose a convolutional neural network (CNN) model based on RSSI fingerprint datasets. This model contains four convolutional layers and two fully connected (FC) layers. The proposed model can complete a test with an average location error of approx. 1.44 m and an accuracy of 94.45%. The authors of [29] propose a lightweight combination of extreme learning machine (ELM) and CNN. The Conv1D layer is used to extract spatial characteristics of the radio map, and the Pooling1D layer reduces the dimensionality. The result shows that the proposed model is approx. 58% faster than the benchmark.

3. System Design

Wi-Fi RSSI-based indoor localization is a two-stage process [13]. In the first stage, the localization area of interest is composed of reference points (RPs) and test points (TPs) [30]. A database containing offline FPs is created where each entry (RSS vector/FP) is associated with a reference point (RP). A localization algorithm like DNN, RF, or SVM is trained on the offline database. In the second stage, the already-trained localization algorithm tries to find the user's location by matching online FPs with the database. Distinct from the traditional ML approaches, we process the dataset by using a three-step SDR-scheme subarea overlapping technique and also utilize lightweight neural networks that are trained on each sub-area independently. The architecture of the proposed model is illustrated in Figure 1, which mainly consists of five components: area division and overlapping, training data selection for classification and regression, source domain refinement (SDR) scheme, coarse positioning phase, and accurate positioning. For the coarse positioning phase, specific points are selected as reference points (RPs). A three-step domain source refinement (DSR) scheme is applied to these RPs to obtain a refined dataset and a classifier is trained

on this refined dataset to find the relevant sub-area where the user can reside in. For the accurate positioning phase, the training dataset is divided into several sub-areas to train a deep neural network (DNN) on each sub-area independently. We will explain the aforementioned components in the following sections.



Figure 1. Architecture of the system's framework.

3.1. Area Division and Overlapping

The continuous radio propagation makes it difficult to divide whole areas into distinct clusters. An overlapping between adjacent clusters is required to reduce classification error [12]. We utilized the same concept of area overlapping as given in [12]. The whole area is divided into sub-areas as shown in Figure 2. In the classification/coarse positioning phase, the uncertainty in finding the relevant sub-area is the highest at the intersection of different sub-areas. For example, let the actual position of some object, e.g, "T", be in area 1, but because of the sharp boundary, the classifier might predict it in area 2. This problem would cause an error later, as the regression algorithm, in the accurate positioning phase, tries to find the object's localization coordinates in the wrongly predicted area. To reduce this uncertainty, it is worthwhile to increase the margin or overlapping between interconnected areas. Furthermore, if there is no overlapping, the reference points lying at the boundary would have to belong to only one sub-area but not the other one, which in turn reduces the localization accuracy in that particular area. To avoid these problems, we need to introduce an overlapping among sub-areas. Neighboring sub-areas would share all the RPs lying in the overlapped area. In Figure 2, the red squares show the RPs and both red diamonds and green circles represent the test points. The classification algorithm will run on the preprocessed dataset to find the relevant sub-areas of the online fingerprints.



Figure 2. Area division.

3.2. Training Data Selection

To train the coarse positioning algorithm (e.g., classifier). Instead of using all the RPs for training purposes, we selected only those RPs lying in the black dotted rectangles, as shown in Figure 3. There are two key considerations for this particular selection. First, the reduction in the offline computational complexity without compromising the classification accuracy. Second, distant training points among neighboring areas have a low risk of sharing similar features. This strategy increases classification accuracy. Figure 4 depicts the training points used to train regression-based algorithms in the accurate positioning phase. We select all the RPs lying in any particular area to train their corresponding DNN-MVR model. We did not reduce training data in the final positioning phase because DNN is more sensitive to overfitting; however, to reduce the training time, we divided the whole area into sub-areas and used a set of single-layered lightweight neural networks.



Figure 3. RPs selected to train classifier.



Figure 4. RPs used to train algorithm for regression.

3.3. Source Domain Refinement (SDR)

This section deals with the offline data SDR scheme to remove the redundant knowledge from the source domain and make it more efficient for finding the user's localization area. The tree-step refinement plan consists of the following steps.

3.3.1. Data Averaging Technique

In order to reduce the training data in the source domain, which in turn reduces the classifier's computational complexity and offline data storage, we use the K-means clustering algorithm to make small groups of similar fingerprints in the source domain. K-means is an unsupervised clustering algorithm that is used to cluster given data into K number of clusters [31]. The algorithm iteratively assigns the data points to one of the K clusters based on how near the point is to the cluster's centroid. After making K clusters, we calculate the mean vector of each group. Those mean vectors from each group are stored in a source file and then used to train the classification algorithms. Assuming that we have $(x_0^n)_{n=1}^{N_0}$ RSS vectors in the source domain and we want K clusters, the K means algorithm tries to pick K points as the initial centroids from the dataset. If each cluster's centroid is denoted by c_k , then each data point x is assigned to a cluster based on

$$Argmin_{(c_k \in C)} = euclidean(c_k, x)^2, \tag{1}$$

where "C" is the total number of clusters.

3.3.2. Outlier Removal Scheme

In order to increase the classification accuracy and reduce overfitting, we presented an outlier removal technique. Any distant observation from the other observations can be considered as an outlier [32]. Assuming that total "N" RSS vectors are obtained in step 1 and the total number of sub-areas is "A", in order to distinguish the outliers from other fingerprints in each area, we calculate the mean vector δ^a of each area "a", where $a \in A$. The Euclidean distance γ_i^a of each fingerprint x_i^a from its a^{th} area's mean vector δ^a is

$$\gamma_i^a = Euclidean(x_i^a - \delta^a). \tag{2}$$

Here, all the RSS vectors showing the value of Euclidean distance above a certain threshold, Th_a , would be considered as an outlier. The value of Th_a is chosen arbitrarily.

3.3.3. Stable AP's Weight Enhancement

APs showing less signal fluctuations are more stable and hence are more reliable in the online signal matching process [18]. It has been observed that if we add some bias to the

stable AP's RSSI measurements of each area independently, it will increase the difference among the fingerprints of different areas. Consequently, the classification accuracy will be increased. However, the values of the bias terms are arbitrary, and large values can change the originality of the RSS vectors significantly, which decreases the accuracy. The stability of an AP is directly proportional to its frequency of occurrence. The procedure is explained below. Assuming that total "L" APs are detected in D_{R1} dataset, a total number of sub-areas are "A", and N_a represents the total number of training samples present in the a_{th} sub-area, where $a \in A$:

- 1. We define a detection vector $d_a^l = [d_a^{l,1}, d_a^{l,2}, \dots, d_a^{l,N_a}]$, where $d_a^{l,n} \in \{0,1\}$, is the detection indicator for the l^{th} AP of n^{th} sample in the sub-area "a". When the value of a particular RSSI feature is above a threshold, Th_a , the corresponding AP is detected, and the value of $d_a^{l,n}$ would be considered as 1 or otherwise, 0. The detection vector d_a^l is calculated for each AP in each sub-area.
- 2. For the current sub-area "a", the sum S_a^l for the l^{th} AP's detection indicators $d_a^{l,n}$ can be calculated as:

$$S_{a}^{l} = \sum_{n=1}^{N_{a}} d_{a}^{l,n}$$
(3)

And the distinction vector $K_a \in R^{(L \times 1)}$ can be written as, $K_a = [S_a^1, S_a^2, ..., S_a^L]$, which we can normalize by dividing the whole vector K_a by the maximum entry in the vector K_a :

$$G_a = \frac{K_a}{max(K_a)} \tag{4}$$

3. Sort G_a in descending order, where each entry is the stability indicator of the corresponding AP in sub-area "a". Now, select those APs whose stability indicator is greater than a threshold Th'_a and add a small bias b_a into the RSSI measurements of the selected APs. Remember that the values of Th'_a and b_a are arbitrary.

3.4. Sub-Clustering Algorithms

3.4.1. Support Vector Machine (SVM)

SVM is a supervised learning discriminant technique that solves the convex optimization problem analytically, and unlike generative ML approaches, it does not suffer from the multi-local minima [33]; in other words, it always returns the same solution. Owing to a better data generalization performance, the regularization of non-linear datasets, theoretical guarantees regarding overfitting, relatively easier implementation, and a higher transparency in operation than neural networks [34], we choose SVM as a classifier. In multi-class support vector machine (MCSVM) problems, mainly two approaches, one against-one (OAO) and one-against-all (OAA), are used to classify data. The latter approach is faster than the former [34], and hence is why we adopt this approach. It constructs "K" SVM models for k classes. The nth SVM is trained with nth class samples, treating them as positive samples, and the rest as negative. The dataset is not linearly separable, so we use the Gaussian RBF kernel function for its better performance than the linear and polynomial kernel functions [35].

3.4.2. Random Forest (RF) Classifier

Random forest (RF) is an ensemble tree-based supervised learning algorithm. The building blocks of RF are decision trees (DTs) and the final prediction of the RF is obtained by combining the majority vote of different DTs [36]. RF exhibits great performance with heterogeneous feature space and high dimensional data. RF is insensitive to overfitting, however, it is computationally demanding with high-dimensional data and large forest size [37]. DT makes predictions by applying feature-based splits, and it depends on the impurity of the dataset. Features with the lowest impurity or Gini index are treated as root nodes.

3.4.3. Multi-Variate Regression-Based DNN Algorithm

A deep neural network (DNN) model consists of multiple stacked hidden layers and it is able to approximate any arbitrary function (i.e. linear/non-linear) to any degree of accuracy [38]. The training process is involved in finding the optimal weights, so that the loss function is minimal [39]. We develop our multivariate regression approach based on the lightweight deep neural network (MVR-DNN). A quick response, less training time, high localization accuracy, and generalization are the key considerations. The process of finding the optimal network size is very complicated, and no reliable method exists in the literature to find the proper size of the neural networks [40]. Researchers use their intuition and experience to find the best architecture to solve their problems. A small network requires less memory to store weights, involves less computational work, and shows a fast response as there are very short propagation delays [41]. In contrast, larger networks exhibit poor generalization. Cybenko et al. [42] proposed that a single hiddenlayer network is enough to approximate a non-linear decision boundary. We utilized a single hidden layer to construct our proposed model and utilized rectified linear function (Relu) as the activation function since they help reduce the vanishing gradient problem [43]. We have applied the mini-batch gradient descent (GD) optimization technique to boost the training process, as batch gradient descent (BGD) can be very slow since they are involved in redundant computations for large datasets. Although, stochastic gradient descent resolves this redundancy, it falls prey to local minima. Mini-batch gradient descent has two advantages [44]: (1) it shows a more stable convergence and (2) it makes computing the gradient very efficient. Common mini-batch sizes range between 50 and 256. Our proposed model is a supervised training algorithm, where the training samples are split according to their measuring space. We trained our proposed model on each sub-area's dataset separately; consequently, each sub-area has its own trained model, and this strategy works well to reduce computational complexity and increase localization accuracy, as shown in the Section 4.4. Each model consists of an input layer, a hidden layer(s), and an output layer. Each neuron in the input layer represents an input variable from the training samples, so the number of input layer neurons is equal to the length of the input feature vector (i.e., RSS Vector). The output layer has two nodes because the response of the DNN-based MVR model is the actual 2D position of the object.

4. Experiment Evaluations

To make a comparison with the literature, we evaluated our model on an open-sourced public database containing Wi-Fi RSSI measurements.

4.1. Experimental Setup

The experiments were simulated on a Microsoft Windows 10 education OS with a 1.10 GHz Intel (R) Core (TM) i7-10710 CPU and 10 GB RAM by utilizing Scikit-learn version 1.0.2. with Python bindings.

4.2. Data Description

The proposed scheme is evaluated on the dataset using Wi-Fi RSS measurements collected on the third and fifth floor of the library environment, as shown in Figure 5, at Universitat Jaume I in Spain [30]. The total area of both floors is about 308.4 m². All the samples were collected by a trained person using a Samsung Galaxy S3 smartphone over the span of 15 months. During the first month, 15 offline datasets were collected, and we utilized them to train our model. The model performance is evaluated on the 75 online databases that were collected during the whole 15 months. Each floor contains 24 reference points and 106 test points.



Figure 5. Library environment; photo taken from [30]; dataset available at the Zenodo repository under the open-source MIT license (https://doi.org/10.3390/data3010003, accessed on 15 February 2023).

4.3. Evaluation Metrics

Our proposed scheme follows the narrow-down approach (NDA) that consists of two stages. To measure the performance of the classification model, we use "Accuracy" as a metric:

$$AccuracyScore = \frac{tp+tn}{tp+fn+tn+fp'}$$
(5)

where tp is true positive, tn is true negative, fp is false positive, and fn is false negative. In the second stage, DNLoc finds the final location of the targeted object in the alreadypredicted area. To measure the performance of the model, we use the average distance error:

$$AED = \frac{1}{N_T} \sum_{p=1}^{N_T} \|(y'_{test} - y_{test})\|_2$$
(6)

where N_T represents the total number of online testing samples. y'_{test} and y_{test} are predicted and actual labels of the test data, respectively, and both comprise 2D positioning components.

4.4. Experiment Results

4.4.1. Classification Performance

We evaluated our classifiers on the third floor for 15 months. Table 1 shows the training time and average response time of both classifiers using the full training dataset and refined dataset.

Table 1. Classifier's training and response time (s).

Classifier —	Training	Deemen of Time	
	Unprocessed Data	Refined Data	Kesponse Time
SVM	0.15	0.01	0.0009
RF	9.7	4.7	0.0019

Figure 6a,b depict the compression ratio (CR) of the SDR scheme on the training data for classification. We divide the training data, by using the K-mean clustering algorithm, where each cluster contains five samples on average. We utilized only the mean RSS vector of each cluster; hence, the compression ratio is 80%. The overall impact of data compression on the classification accuracy is negative, as depicted in Figure 7a,b. To improve the classification accuracy, we apply the three-step SDR scheme on the reduced

dataset. Figure 8a,b illustrate that the proposed SDR scheme improves accuracy by reducing overfitting as can be seen in the later months. It is seen that the refined dataset improves the classification accuracy by approximately 4% for RF and 2.23% for SVM. Figure 9a,b show the comparison of full, reduced, and SDR-scheme-refined datasets in terms of accuracy.

lumber of Sample:





(b) Number of Samples

(a) Number of Features **Figure 6.** Data compression using SDR scheme.

5

Accuracy (%)

80

75 L



Figure 7. Impact of data compression on accuracy (a) SVM; (b) RF.



Figure 8. Impact of SDR scheme on the classification accuracy (a) SVM; (b) RF.



Figure 9. Classification accuracy of full, reduced, and SDR-refined dataset (a) SVM; (b) RF.

4.4.2. Regression Performance

In the regression/final positioning phase, the DNLoc algorithm finds the user's final location in the already-predicted area and we compare its performance in terms of accuracy and online computational time with the best papers published on the same dataset like TransLoc [21], DFMIC [22], SmrtLoc [23], and aslo with some other papers in the literature like ViVi [45], KAAL [46], UFL-ECLS [47], EMSS [48], and MSSE [49]. We utilized both third and fifth floor datasets over the span of 15 months. Table 2 shows the number of samples used to train and test the proposed MVR-DNN model in each area. The training time is also presented in the last column. Table 3 shows the AED and percentile error of the DNLoc model. Table 4 shows the response time and AED comparison with that of the literature. Figure 10 depicts the AED of DNLoc as compared to other Wi-Fi RSSI-based best techniques in the literature and shows that DNLoc outperforms the other methods, reporting an AED of 2.09 m. TransLoc [21], SmartLoc [23], EMSS [48], Wi-Fi-FAGOT [47], VIVI [45], and KAAL [46] incur an AED of 2.21 m, 2.588 m, 3.9 m, 2.68 m, 2.7 m, 3.23 m, and 3.25 m, respectively. Figure 11 shows the cumulative distribution function (CDF) of different techniques juxtaposed with DNLoc. DNLoc reduces the 85th percentile of DIFMIC, Wi-Fi-FAGOT, KAAL, MMSE, and MUCUS by 15.7%, 37.64%, 45.8%, 48.9%, and 66.8% respectively, demonstrating that DNLoc outperforms all other methods.



Figure 10. Comparison of AED with different methods for 15 months.

Floor	Sub-Area	Offline Samples	Online Samples	Training Time (s)
3	1	1800	9000	45.7
3	2	1800	9000	33.4
3	3	1800	9000	42.1
3	4	1800	9000	20.15
5	1	1800	9000	23.6
5	2	1800	9000	59.2
5	3	1800	9000	37.3
5	4	1800	9000	66.27
Total/Average	8	14,400	72,000	327.72

 Table 2. Number of samples and training time for DNLoc.



Figure 11. CDFs of different methods compared with DNLoc in the library environment.

Table 3. AED and percentile errors of DNLoc.

Floor 3					
Sub-Area	AED (m)	25th Percentile (m)	50th Percentile (m)	75th Percentile (m)	95th Percentile (m)
1	2.02	1.2500	1.9000	2.6500	3.7900
2	2.316	1.51	2.20	2.93	4.0
3	1.94	1.21	1.862	2.577	3.6073
4	2.34	1.4000	2.1900	3.0200	4.2700
			Floor 5		
1	1.99	1.1500	1.8460	2.5900	3.4780
2	2.38	1.4320	2.2000	3.0800	4.3500
3	1.79	1.0000	1.6000	2.2500	3.1100
4	1.95	1.1000	1.7300	2.3400	3.2800

Methods	25th Percentile	50th Percentile	75th Percentile	AED	Response Time
MSSE [49]	1.58	3.01	4.86	3.34	10.4
KAAL [46]	1.65	3.28	4.68	3.26	12.6
ViVi [45]	1.79	3.39	4.37	3.21	41.2
Wi-Fi-FAGOT [47]	1.55	2.47	3.88	2.79	228
SmartLoc [23]	1.23	2.29	3.46	2.58	281
DNLoc	1.2565	1.9410	2.68	2.09	100

Table 4. Positioning error measures (in meters) and average response time (in milliseconds) for different methods.

5. Discussion and Conclusions

A narrow-down approach is presented for indoor localization that involves coarse and accurate positioning phases. Training points' selection, area division, and overlapping strategies are presented to reduce the uncertainty in finding the actual user's location. In the coarse positioning phase, the SDR scheme involves data averaging, outlier removal, and stable APs' weight enhancement techniques, which are presented. The SDR scheme compressed 80% of classification training data and increased classification accuracy. In the final/accurate positioning phase, a set of MVR-DNN models, trained on each subarea, is utilized to find the final user's location. Our experimental results establish the superiority of the proposed model over the existing machine learning approaches. It is worthwhile to find the proper size of clusters as the classification accuracy may vary with it. The selection of the optimal size and parameters of the neural networks, the effect of environmental dynamics, data collection overhead, and possible changes in signal strengths over time, which are dependent on a person's movements, as well as object displacements, an analysis of the bias effect to increase the weight of the RSS vectors of RPs, and a better area classification strategy could all be potential future research topics. Our proposed model is simple, generic, and flexible that can be applied to find the user's location coordinates accurately in any indoor environment.

Author Contributions: Conceptualization, methodology, validation and writing—original draft preparation by S.M.A.U. The investigation, resources, writing—review and editing by T.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data were obtained from a third party. The data are open-sourced and publicly available.

Acknowledgments: The authors acknowledge the support provided by the University of Edinburgh for the experiments.

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

References

- Sadowski, S.; Spachos, P.; Plataniotis, K.N. Memoryless Techniques and Wireless Technologies for Indoor Localization with the Internet of Things. *IEEE Internet Things J.* 2020, 7, 10996–11005.
- Sabanci, K.; Yigit, E.; Ustun, D.; Toktas, A.; Aslan, M.F. WiFi Based Indoor Localization: Application and Comparison of Machine Learning Algorithms. In Proceedings of the 2018 XXIIIrd International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), Tbilisi, Georgia, 24–27 September 2018; pp. 246–251.
- Anzum, N.; Afroze, S.F.; Rahman, A. Zone-Based Indoor Localization Using Neural Networks: A View from a Real Testbed. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–7.
- Belmonte-Fernandez, O.; Sansano-Sansano, E.; Caballer-Miedes, A.; Montoliu, R.; García-Vidal, R.; Gascó-Compte, A. A Generative Method for Indoor Localization Using Wi-Fi Fingerprinting. Sensors 2021, 21, 2392.
- Zhang, L.; Chen, Z.; Cui, W.; Li, B.; Chen, C.; Cao, Z.; Gao, K. WiFi-Based Indoor Robot Positioning Using Deep Fuzzy Forests. *IEEE Internet Things J.* 2020, 7, 10773–10781.

- Yang, Z.; Wu, C.; Liu, Y. Locating in fingerprint space: Wireless indoor localization with little human intervention. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; pp. 269–280.
- 7. Xiang, P.; Ji, P.; Zhang, D. Enhance RSS-based indoor localization accuracy by leveraging environmental physical features. *Wirel. Commun. Mob. Comput.* **2018**, 2018, 8956757.
- Tran, D.A.; Pham, C. Fast and accurate indoor localization based on spatially hierarchical classification. In Proceedings of the IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Philadelphia, PA, USA, 28–30 October 2014; pp. 118–126.
- 9. Wang, Y.; Xiu, C.; Zhang, X.; Yang, D. WiFi indoor localization with CSI fingerprinting-based random forest. *Sensors* **2018**, *18*, 2869. [CrossRef]
- Zhang, W.; Liu, K.; Zhang, W.; Zhang, Y.; Gu, J. Deep neural networks for wireless localization in indoor and outdoor environments. *Neurocomputing* 2016, 194, 279–287. [CrossRef]
- 11. Dou, F.; Lu, J.; Xu, T.; Huang, C.-H.; Bi, J. A Bisection Reinforcement Learning Approach to 3-D Indoor Localization. *IEEE Internet Things J.* **2021**, *8*, 6519–6535.
- 12. Bi, J.; Huang, L.; Cao, H.; Yao, G.; Sang, W.; Zhen, J.; Liu, Y. Improved Indoor Fingerprinting Localization Method Using Clustering Algorithm and Dynamic Compensation. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 613.
- Gao, J.; Li, X.; Ding, Y.; Su, Q.; Liu, Z. WiFi-Based Indoor Positioning by Random Forest and Adjusted Cosine Similarity. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 1426–1431.
- Luo, M.; Zheng, J.; Sun, W.; Zhang, X. WiFi-based Indoor Localization Using Clustering and Fusion Fingerprint. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021; pp. 3480–3485.
- 15. Xiang, C.; Zhang, S.; Xu, S.; Chen, X.; Cao, S.; Alexandropoulos, G.C.; Lau, V.K. Robust Sub-Meter Level Indoor Localization with a Single WiFi Access Point—Regression Versus Classification. *IEEE Access* **2019**, *7*, 146309–146321. [CrossRef]
- Pinto, B.; Barreto, R.; Souto, E.; Oliveira, H. Robust RSSI-Based Indoor Positioning System Using K-Means Clustering and Bayesian Estimation. *IEEE Sens. J.* 2021, 21, 24462–24470.
- Gomes, R.; Ahsan, M.; Denton, A. Random Forest Classifier in SDN Framework for User-Based Indoor Localization. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018; pp. 0537–0542.
- Liang, D.; Zhang, Z.; Peng, M. Access Point Reselection and Adaptive Cluster Splitting-Based Indoor Localization in Wireless Local Area Networks. *IEEE Internet Things J.* 2015, 2, 573–585. [CrossRef]
- Alraih, S.; Alhammadi, A.; Shayea, I.; Al-Samman, A.M. Improving accuracy in indoor localization system using fingerprinting technique. In Proceedings of the 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 18–20 October 2017; pp. 274–277.
- Chen, W.; Chang, Q.; Hou, H.-T.; Wang, W.-P. A novel clustering and KWNN-based strategy for Wi-Fi fingerprint indoor localization. In Proceedings of the 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), Harbin, China, 19–20 December 2015; pp. 49–52.
- 21. Li, L.; Guo, X.; Zhao, M.; Li, H.; Ansari, N. TransLoc: A Heterogeneous Knowledge Transfer Framework for Fingerprint-Based Indoor Localization. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 3628–3642. [CrossRef]
- 22. Guo, X.; Elikplim, N.R.; Ansari, N.; Li, L.; Wang, L. Robust WiFi Localization by Fusing Derivative Fingerprints of RSS and Multiple Classifiers. *IEEE Trans. Ind. Inf.* 2020, *16*, 3177–3186. [CrossRef]
- 23. Li, L.; Guo, X.; Ansari, N. SmartLoc: Smart Wireless Indoor Localization Empowered by Machine Learning. *IEEE Trans. Ind. Electron.* **2020**, *67*, 6883–6893. [CrossRef]
- 24. Zhang, G.; Wang, P.; Chen, H.; Zhang, L. Wireless Indoor Localization Using Convolutional Neural Network and Gaussian Process Regression. *Sensors* 2019, *19*, 2508.
- 25. Soro, B.; Lee, C. A Wavelet Scattering Feature Extraction Approach for Deep Neural Network Based Indoor Fingerprinting Localization. *Sensors* **2019**, *19*, 1790. [CrossRef]
- Li, L.; Guo, X.; Ansari, N. A Hybrid Fingerprint Quality Evaluation Model for WiFi Localization. *IEEE Internet Things J.* 2019, 6, 9829–9840. [CrossRef]
- Khassanov, Y.; Nurpeiissov, M.; Sarkytbayev, A.; Kuzdeuov, A.; Varol, H.A. Finer-level Sequential WiFi-based Indoor Localization. In Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII), Iwaki, Japan, 11–14 January 2021; pp. 163–169.
- 28. Sinha, R.S.; Hwang, S.H. Comparison of CNN Applications for RSSI-based Fingerprint Indoor Localization. *Electronics* 2019, *8*, 989. [CrossRef]
- Thirunavukkarasu, K.; Sing, A.; Rai, P. Classification of IRIS Dataset using Classification Based KNN Algorithm in Supervised Learning. In Proceedings of the 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 14–15 December 2018.
- Mendoza-Silva, G.; Richter, P.; Torres-Sospedra, J.; Lohan, E.; Huerta, J. Long-Term WiFi Fingerprinting Dataset for Research on Robust Indoor Positioning. Data 2018, 3, 3. [CrossRef]
- 31. Sinaga, K.P.; Yang, M.-S. Unsupervised K-Means Clustering Algorithm. IEEE Access 2020, 8, 80716–80727. [CrossRef]

- 32. Bhatti, M.A.; Riaz, R.; Rizvi, S.S.; Shokat, S.; Riaz, F.; Kwon, S.J. Outlier detection in indoor localization and Internet of Things (IoT) using machine learning. *J. Commun. Netw.* **2020**, *22*, 236–243. [CrossRef]
- Chamasemani, F.F.; Singh, Y.P. Multi-class Support Vector Machine (SVM) Classifiers—An Application in Hypothyroid Detection and Classification. In Proceedings of the 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications, Penang, Malaysia, 27–29 September 2011; pp. 351–356.
- Prakash, J.S. Multi class Support Vector Machines classifier for machine vision application. In Proceedings of the 2012 International Conference on Machine Vision and Image Processing (MVIP), Coimbatore, India, 14–15 December 2012.
- 35. Sangeetha, R. Performance Evaluation of Kernels in Multiclass Support Vector Machines. *Int. J. Soft Comput. Eng. (IJSCE)* **2011**, *1*, 2231–2307.
- 36. Breiman, L. Random Forests. Mach. Learn. 2001, 45, 5–32. [CrossRef]
- Guo, X.; Ansari, N. Localization by fusing a group of fingerprints via multiple antennas in indoor environment. *IEEE Trans. Veh. Technol.* 2017, 66, 9904–9915. [CrossRef]
- 38. Wang, R.; Fu, B. Deep and Cross Network for AD Click Predictions. In Proceedings of the ADKDD 17, Halifax, NS, Canada, 13–17 August 2017; pp. 1–7.
- Wu, H.; Shapiro, J.L. Does overfitting affect performance in estimation of distribution algorithms. In Proceedings of the Conference on Genetic and Evolutionary Computation, ACM, Seattle, WA, USA, 8–12 July 2006; pp. 433–434.
- 40. Zou, J.; Han, Y.; So, S.S. Overview of Artificial Neural Networks. In *Artificial Neural Networks. Methods in Molecular Biology*™; Livingstone, D.J., Ed.; Humana Press: Totowa, NJ, USA, 2008; Volume 458.
- 41. Bebis, G.; Georgiopoulos, M. Feed-forward neural networks. IEEE Potentials 1994, 13, 27–31. [CrossRef]
- 42. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signal Syst.* **1989**, 2, 303–314. [CrossRef]
- 43. Rizk, H.; Torki, M.; Youssef, M. CellinDeep: Robust and Accurate Cellular-Based Indoor Localization via Deep Learning. *IEEE Sens. J.* **2019**, *19*, 2305–2312. [CrossRef]
- 44. Ruder, S. An overview of gradient descent optimization algorithms. Cornell University. [Submitted on 15 September 2016 (v1), last revised 15 June 2017 (this version, v2). Aylien Ltd., Dublin]. *arXiv* **2016**, arXiv:1609.04747.
- 45. Wu, C.; Xu, J.; Yang, Z.; Lane, N.D.; Yin, Z. Gain without pain: Accurate WiFi-based localization using fingerprint spatial gradient. *Proc. ACM UbiComp* **2017**, *1*, 29. [CrossRef]
- 46. Guo, X.; Li, L.; Ansari, N.; Liao, B. Knowledge aided adaptive localization via global fusion profile. *IEEE Internet Things J.* 2017, *5*, 1081–1089. [CrossRef]
- 47. Guo, X.; Zhu, S.; Li, L.; Hu, F.; Ansari, N. Accurate WiFi localization by unsupervised fusion of extended candidate location set. *IEEE Internet Things J.* 2018, *6*, 2476–2485. [CrossRef]
- Guo, X.; Li, L.; Feng, X.; Ansari, N. Expectation maximization indoor localization utilizing supporting set for internet of things. IEEE Internet Things J. 2018, 6, 2573–2582. [CrossRef]
- Gwon, Y.; Jain, R.; Kawahara, T. Robust indoor location estimation of stationary and mobile users. In Proceedings of the IEEE INFOCOM, San Jose, CA, USA, 25–29 October 2004; pp. 1032–1043.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.