A Deep Neural Network Based on Stacked Auto-Encoder and Dataset Stratification in Indoor Location

Jing Zhang[1] and Ying Su[1*]

[1]Department of Communication, Shanghai Normal University, Shanghai 200234, PRC
{jannety,yingsu}@shnu.edu.cn

Abstract: Indoor location has become the core part in the large-scale location-aware services, especially in the extendable/scalable applications. Fingerprint location by using the signal strength indicator (RSSI) of the received WiFi signal has the advantages of full coverage and strong expansibility. It also has the disadvantages of requiring data calibration and lacking samples under the dynamic environment. This paper describes a deep neural network method used for indoor positioning (DNNIP) based on stacked auto-encoder and data stratification. The experimental results show that this DNNIP has better classification accuracy than the machine learning algorithms that are based on UJIIndoorLoc dataset.

Keywords: Indoor Location, Deep Neural Network, Machine Learning Algorithm.

1 Introduction

The application of indoor location involves the integration of multiple interdisciplinary works. This location information meets the users' needs with convenient navigations in large indoor situations. In the e-commerce application, specific recommendations with accurate location information can be presented to users. In the emergency rescue, fast effective actions can be taken with the obtained exact location of the rescued target. Also in hospitals, doctors can manage patients and medical supplies through location tracking. And in social activities, groups can be easily built based on the people's indoor locations, thus enriching the interactions among peoples [1].

In an indoor environment, the traditional GPS positioning is generally not suitable due to its large signal attenuation. Thus the indoor positioning introduces a variety of sensors and equipments. In the coverage of one wireless heterogeneous network, the fingerprint location is implemented with the received signal strength indicator (RSSI) values from at least three wireless transmitters. The use of WiFi signal is a popular solution, which complies with IEEE 802.11 standard and has high bandwidth and transmit rate. The advantage of using WiFi for positioning is its full WiFi signal coverage in most cities, providing the readiness of the infrastructure. In order to improve the positioning accuracy, the acoustic ranging and WiFi are combined to test the rela-

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

tive distance information between reference nodes. Accelerometers and compasses are also used to help obtain accurate maps with large indoor areas [2].

The choice of the location of an access point is the key to improving the accuracy of positioning when applying the fingerprint matching for the location. Some literatures have analyzed the impact of the number of reference nodes on the accuracy, and proposed appropriate selection strategies. Some select reference nodes via gradient descent search during the training phase, RSSI surface fitting, and hierarchical clustering [3-4]. Even some publishes propose a goal-driven model by combining multi-target reference node deployment with genetic algorithm [5]. After the selection of RSSI signals, the process algorithm is another factor that affects the accuracy of positioning. There are two types of intelligent location algorithms: neighbor selection and machine learning. K-Nearest Neighbor (KNN) is a common fingerprint matching algorithm. The accuracy can be improved through the minimum circle clustering and the adaptive weighted KNN matching [6]. Artificial neural network (ANN) is also introduced to support indoor and outdoor positioning with its particle swarm optimization to be used to optimize the neurons [7].

In large indoor stereo positioning scenarios, the problem becomes more complex due to the existence of multiple floors and multiple buildings. How to deal with the random fluctuation of the signal, the noise of the multipath effect, and the dependence on the equipment to be used are the main challenges. This paper [8] proposes a fingerprint based pipeline processing method, which firstly identifies the user's building, and then establishes the association among access points. Through the voting of these various networks, the buildings and their floors are correctly calculated [8]. The KNN algorithm mentioned above would make proper adjustment to the building specifically selected via grid search [9]. In addition, with the popularity of mobile devices, big data has become accessible. Due to the improvement of GPU performance and rich algorithm libraries, the deep learning can address these challenges effectively. Deep neural network (DNN) simply needs to be for the comparisons of location precisions within larger positioning areas. It needs fewer parameter adjustments but provides greater scalability [10-12].

In this research, the UJIIndoorLoc dataset was selected as the original records [13-14]. Taken into the consideration of the cases like the possible classification errors of the buildings or the floors, the loss happening in the training stage will be the same as in the validation stage. Thus the dataset was segmented as layers based on the RSSI values from 520 wireless access points (WAP) and the three attributes as buildingID, floorID, and SpaceID. This paper proposes the use of deep neural network for indoor positioning (DNNIP) after the comparisons with the support vector machine (SVM), random forest and gradient boosting decision tree. The DNNIP approach ensures the accuracy and reduces the adjustment of parameters.

The rest of this paper is organized as follows: Section 2 introduces the hierarchical method for the dataset. Section 3 describes the DNNIP network structure and training algorithm. Then the experimental results are represented. The last section has the conclusion.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

2 Fingerprint Localization and Datasets Stratification

Indoor fingerprint positioning consists of two stages as shown in Fig. 1. During the offline stage, the wireless map is created by measuring the RSSI values of different access points or reference nodes. The map includes not only the RSSI values, but also the details of the reference nodes placed: the building number, the specific floor, and the space details. While at the online stage, the location of the user is obtained by matching the RSSI value of the user with the wireless map through a location algorithm.

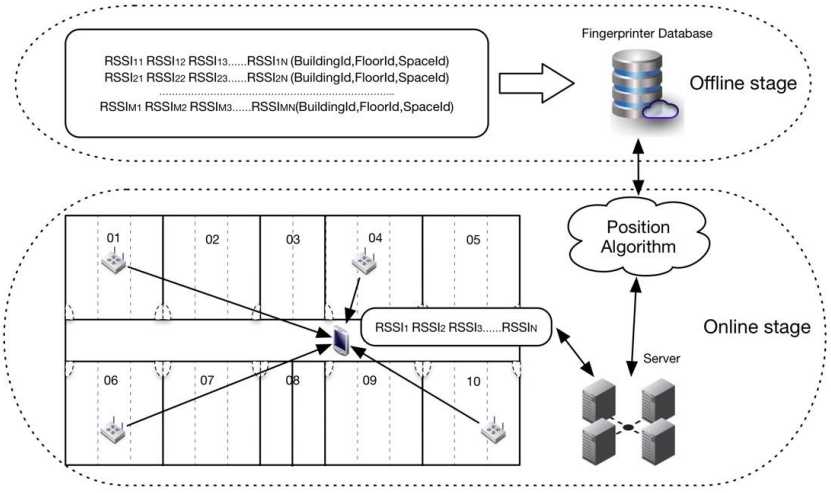

Fig. 1. Fingerprint positioning stages

2.1 Open Dataset of UJIIndoorLoc

The dataset of UJIIndoorLoc was collected from three buildings of Jaume I University, with four, four and five floors respectively. Data from 933 different reference points was measured against 20+ users who are using 25 different types of mobile terminals. The whole data set is divided into training subset and test subset with 19,937 records used for training and 1,111 records for testing. Each record contains 529 dimensions, including 520 RSSI values and 9 additional attributes, like BuildingID, FloorID, and SpaceID.

2.2 Dataset Stratification

The data preparation is based on the location stratification. Each sub-dataset is constructed by the building number, floor number and space number. The format of each sample is BuildingID, FloorID, SpaceID, WAP001, …, WAP520. 90% of the whole training dataset of UJIIndoorLoc are classified as training dataset and test dataset, represented as DNNIP_Train and DNNIP_Test. The test dataset is used for testing, while the data of the original test set are transformed as validation set. Each dataset is then divided into three sub-datasets based on the BuildingID. And each sub-dataset is

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

further divided according to the FloorID. Finally, the spaceID is added. The final bottom sub-dataset reflects the minimum positioning range. The hierarchical sub-datasets are illustrated in Fig. 2. From these sub-datasets, the building information is firstly discovered and predicted, and then the floors are classified. then each sample will be further classified by SpaceID.

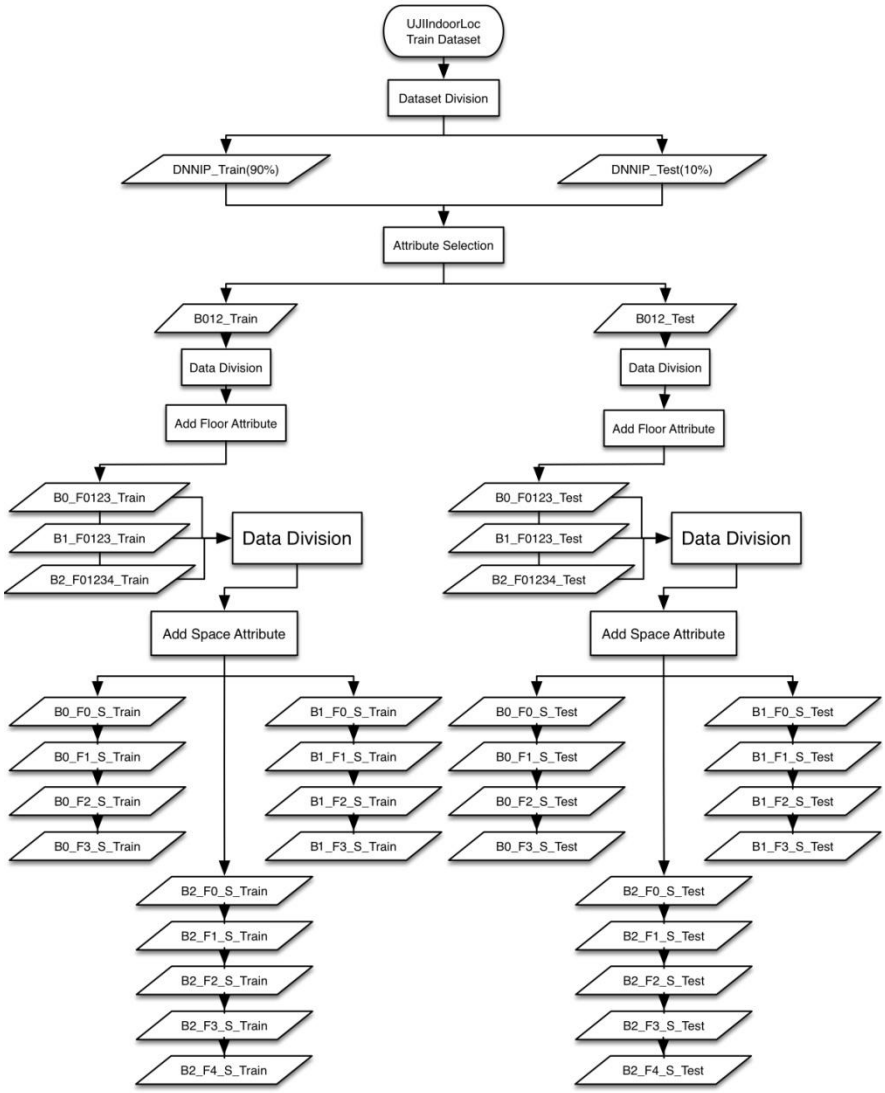

Fig. 2. Dataset preparation

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

3 DNNIP with Auto-Encoders

3.1 Network Structure of DNNIP

DNNIP has multiple auto-encoders (AE) to learn features from large samples. The Stacked AE (SAE) makes the output vector value as close as possible to the input vector value. The output of the hidden-layer neurons is of the most research interest as these post-encoding parameters carry all of the input information in a compact feature form. When the number of AE increases, the feature representation becomes more abstract. Therefore, DNNIP is more suitable for the complex classification tasks.

The reason for applying SAE into DNNIP is that if a DNN is directly trained to perform the classification task through random initialization, the underlying error is almost zero and the error gradient can easily disappear. When using the AE structure for the unsupervised learning from the training data, the pre-trained network can make the training data to a certain extent such that the initial value of the whole network is within an appropriate state. Therefore, through the supervised learning the classification of the second stage can be made easier and the convergence speed can be accelerated. Also the activation function of a rectified linear unit (ReLu) is applied to the SAE neurons, i.e., $f(z) = \max(0, z)$. It is well known that in order to use the traditional backward-propagation learning algorithm, the activation function of neurons normally uses a sigmoid function. In DNN, however, the sigmoid function would appear soft saturated, and the error gradient would disappear easily.

As shown in Fig. 3, at the top of the DNNIP network, the classifier consists of a dropout layer. For strengthening the learning of redundancy, the dropout layer randomly deletes the connections between layers during the training process to achieve better generalization and avoid over-fitting. The final output layer is the Softmax layer that outputs the probability of classifications.

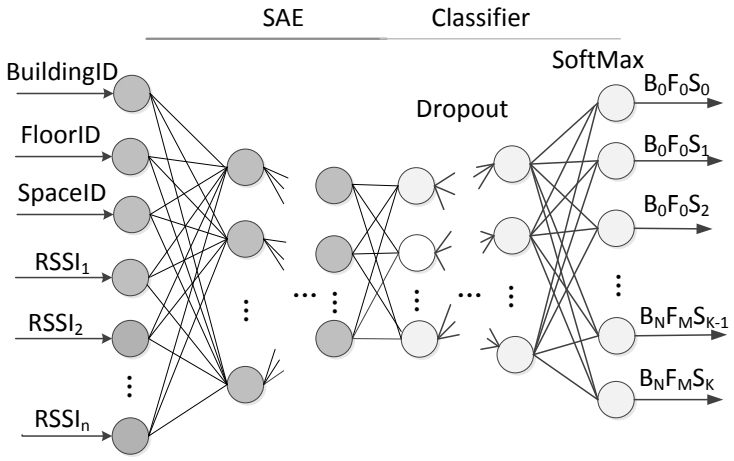

Fig. 3. The network structure of DNNIP

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

3.2 Auto-Encoder and Training Algorithm

Every AE is a fully connected multi-layer forward network, and it maps the input to the hidden layer and then maps to the output. Basically the AE reconstructs the input with which the hidden layer maps the input to become low dimensional vectors. When the number of AEs increases, the number of hidden-layer neurons would decrease. The hidden layer can learn the features of $h = f_\theta(x)$, where θ represents the connection parameters, in which the weight w, the threshold b and the activation functions are included. Moreover, the input of the next AE is the output of the previous AE from its hidden layer.

Fig. 4 shows the first two AEs of a DNNIP. The input layer of the first AE has 523 neurons to accept 523 dimensional RSSI values plus the three attributes - BuildingID, FloorID and SpaceID. The output layer also has 523 neurons, representing the reconstructed input to the next AE.

The SAE part is reconstructed as follows: Given the required input data $x = \{x^{(i)} | i = 1, \cdots, m\}$. Let the number of AEs be N_k. By feeding x to train the first AE, the network parameters $(w^{(1,1)}, b^{(1,1)}, w^{(1,2)}, b^{(1,2)})$ along with the output of the hidden layer $a^{(1,2)}$ are obtained, as shown in Fig 5(a). Afterwards, this $a^{(1,2)}$ is then used as the input of the second AE, as shown in Fig 5(b). Another group of parameters $\left(w^{(2,1)}, b^{(2,1)}, w^{(2,2)}, b^{(2,2)}\right)$ and $a^{(2,2)}$ can be obtained. Repeat this procedure until the N_k number of AE trainings are reached. At this point, multiple groups of network parameters $\{(w^{(k,1)}, \mathrm{b}^{(k,1)}, a^{(k,2)}) | k = 1, \cdots, N_k\}$ are available.

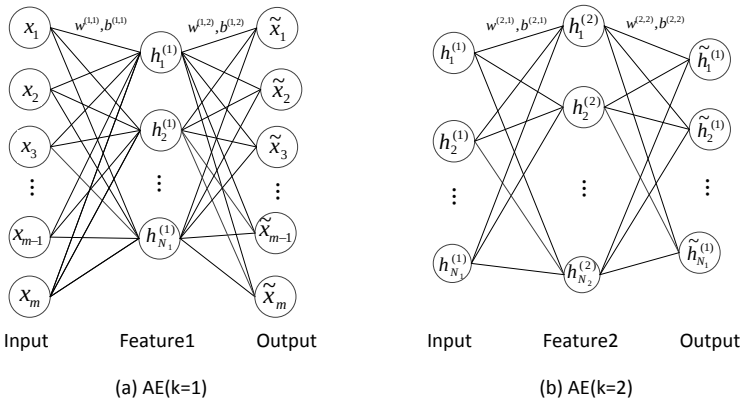

Fig. 4. Stacked AEs

The training of the AE network uses the backward propagation algorithm. This training is to find parameters $(w, b) = (w^{(k,1)}, b^{(k,1)}, w^{(k,2)}, b^{(k,2)})$ by minimizing the loss function,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

$$\min_{\arg w,b} L(w,b) = \left[\frac{1}{m}\sum_{i=1}^{m} J\left(w,b;x^{(i)},y^{(i)}\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(w_{ji}^{(k,l)}\right)^2 + \beta\sum_{j=1}^{s_l} KL(\rho||\bar{\rho}_j). \tag{1}$$

where $J(\cdot)$ is the mean squared error (MSE) between the input and the output, which is further formulated as in (2). In (1), $x^{(i)} \in R^{s_l \times 1}$ and $y^{(i)}$ denote the input and output of the ith sample, respectively; m, n_l and s_l represent the number of training samples, the number of network layers and the number of neurons in layer l, respectively; $w_{ji}^{(k,l)}$ denotes the connection weight between the ith neuron on the layer l and the jth neuron on the l+1 layer of the kth AE; $KL(\cdot)$ represents the sparseness constraint formulated in (3); λ, β and ρ represent the weight coefficients, respectively.

$$J\left(w,b;x^{(i)},y^{(i)}\right) = \frac{1}{2}\left\|h_{w,b}\left(x^{(i)}\right) - y^{(i)}\right\|^2, \tag{2}$$

where $h_{w,b}\left(x^{(i)}\right)$ is the SAE output vector,

$$h_{w,b}\left(x^{(i)}\right) = a^{(k,3)} = f\left(w^{(k,2)}a^{(k,2)} + b^{(k,2)}\right), \tag{3}$$

where $b^{(k,2)}$ and $a^{(k,2)}$ represent the offset vector and output vector of the kth AE, respectively,

$$a^{(k,2)} = f\left(w^{(k,1)}x^{(i)} + b^{(k,1)}\right), \tag{4}$$

and

$$KL(\rho||\bar{\rho}_j) = \rho lg\frac{\rho}{\bar{\rho}_j} + (1-\rho)lg\frac{1-\rho}{1-\bar{\rho}_j}. \tag{5}$$

where $\bar{\rho}_j$ is the average output of the jth neuron, $\bar{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(k,2)}x^{(i)}]$; wherein $a_j^{(k,2)}$ is the output of the jth hidden neuron.

For the sample $(x^{(i)},y^{(i)})$, the error gradient on the neuron n of the output layer is

$$\delta_n^{(k,3)} = \frac{\partial}{\partial z_n^{(k,3)}}\left\{\frac{1}{2}\left\|h_{w,b}\left(x^{(i)}\right) - y^{(i)}\right\|^2\right\} = -\left(y_n^{(i)} - a_n^{(k,l)}\right)f'\left(z_n^{(k,3)}\right), \tag{6}$$

The gradient residual of the neuron n on the hidden layer k is

$$\delta_n^{(k,2)} = f'\left(Z_n^{(k,2)}\right)\cdot\left[\sum_{m=1}^{s_l}(w_{mn}^{(k,2)}\delta_n^{(k,3)} + \beta\cdot(-\frac{\rho}{\bar{\rho}_m} + \frac{1-\rho}{1-\bar{\rho}_m})\right], \tag{7}$$

In (6) and (7), the vector $z^{(k,l+1)} = w^{(k,l)}a^{(k,l)} + b^{(k,l)}, l = 1,2$; $z_n^{(k,l)}$ is the nth element of $Z^{(k,l)}$.

To find the partial derivatives of the cost function yields

$$\nabla_{w^{(k,l)}}J\left(w,b;RSSI^{(i)},y^{(i)}\right) = \frac{\partial J\left(w,b;RSSI^{(i)},y^{(i)}\right)}{\partial w^{(k,l)}} = \delta^{(k,l+1)}(a^{(k,l)})^T, \tag{8}$$

$$\nabla_{b^{(k,l)}}J\left(w,b;RSSI^{(i)},y^{(i)}\right) = \frac{\partial J\left(w,b;RSSI^{(i)},y^{(i)}\right)}{\partial b^{(k,l)}} = \delta^{(k,l+1)}. \tag{9}$$

After obtaining the loss function and the partial derivative of the parameters, the gradient descent algorithm is to get the optimal parameters of the AE network. The training process is as follows.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

1) For $l = 1$ to N_k, let the matrix $\Delta w^{(k,l)} = 0$, the vector $\Delta b^{(k,l)} = 0$, initialize the learning step α, $0 < \alpha < 1$.
2) For $i = 1$ to m, calculate
$$\Delta w^{(k,l)} := \Delta w^{(k,l)} + \nabla_{w^{(k,l)}} J\left(w, b; RSSI^{(i)}, y^{(i)}\right),$$
$$\Delta b^{(k,l)} := \Delta b^{(k,l)} + \nabla_{b^{(k,l)}} J\left(w, b; RSSI^{(i)}, y^{(i)}\right).$$
3) Update the parameters
$$w^{(k,l)} = w^{(k,l)} - \alpha\left[\left(\frac{1}{m}\Delta w^{(k,l)}\right) + \lambda w^{(k,l)}\right],$$
$$b^{(k,l)} = b^{(k,l)} - \alpha\left[\frac{1}{m}\Delta b^{(k,l)}\right].$$
4) Repeat 2) until the algorithm converges or reaches the maximum number of iterations and output $(w^{(k,l)}, b^{(k,l)}, w^{(k,l+1)}, b^{(k,l+1)})$.

In the training process, the value of the cost function is determined by all the training samples. It has irrelevance with the training sequence. Inside the SAE, the output of the hidden layer $a^{(k,2)}$ is the feature of the whole training dataset.

3.3 Classification

After the SAE's unsupervised training process is completed, the decoder layer of each layer, i.e., the output layer is disconnected. The trained SAE is then connected to the classifier, as can be seen in Fig. 3. Generally the number of buildings is denoted by N, the number of floors in building i by M_i, and the number of spaces in floor j by K_j. Then the outputs of the classifiers are separated into $\sum_{i=1}^{N}\sum_{j=1}^{M_i} K_j$ classes.

4 Experimental Results Comparison

During the experiments, the Pandas library is selected for data processing, Keras library for DNN, TensorFlow for numerical computation, and Scikit-learn library for the typical machine learning algorithm computations. The relevant parameters of simulation are shown in Table 1.

Table 1. The DNNIP related parameters.

DNNIP Parameters	Values
SAE hidden layers	64,128,256
SAE activation function	ReLu
SAE optimizer	ADAM
SAE loss	MSE
Classifier hidden layers	128-128
Classifier optimizer	ADAM
Classifier loss	Categorical Cross Entropy
Classifier dropout rate	0.18

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

Ratio of training data to overall data	0.90
Number of epochs	20
Batch size	10

To evaluate the performance of the DNNIP against other indoor positioning algorithms that include SVM, random forest algorithm and gradient-promotion decision tree, we calculate the accuracy according to (10) based on the definition of accuracy for classification problems used in statistical learning, and then select the true positive (TP), false positive (FP), false negative (FN) as well as true negative (TN) to ultimately measure the correct rate (CR).

$$CR = \frac{TP+TN}{TP+TN+FP+FN}. \quad (10)$$

In fact, TP and TN represent the correct classification numbers, while FP and FN indicate the wrong classification numbers.

Several distinct DNN structures are shown in Fig. 5. These networks are represented by having the numbers in parentheses, which indicate the number of neurons used in the hidden layer. The first DNN is a fully connected network with no SAE components, and it uses a dropout layer to prevent over-fitting. For each chosen structure, a variety of optimization strategies are used by starting with constant tuning and testing the dropout value to be within the range of 5% to 20%, The final settling is set on 18% as shown in Table 1. Meanwhile, the different values of the learning rate of the ADAM optimizer are tried and compared with the best value achieved through repeated adjustments.

4.1 Effect of the SAE Structure

Fig. 5 displays the accuracy performances of different DNN networks. It clearly shows that the higher the number of hidden-layer neurons, the higher the accuracy. The accuracy degree of the classifications of the buildings and floors coming out of the test set can achieve 94.2%, while the accuracy against the validation dataset can only be close to 83.8%. The reason is because of the characteristics of the validation dataset - only part of the samples contains valid RSSI values, while other samples lack RSSI values, and the default value is 0. The results demonstrate that the SAE's network structure can effectively reduce the dimensioning of the input vector from 523 to 256, 128, and 64. The simplified results can then be linked to classifiers.

Fig. 5 indicates that the SAE's identification accuracy with 256 and 128 hidden neurons superimposed on the validation data set can reach up to 98%, and the accuracy on the test set can be improved up to 89%. This proves that the SAE can learn even from a simplified representation of the input information and get better results than the DNN that does not have an AE network. The comparison of SAE (256-128) vs. SAE (128-64) would lead to a conclusion that the more neurons are put on the hidden-layer, the higher accuracy can be obtained. The SAE (256-128-64), however, achieves the similar performance as the SAE (256-128) does. This is mainly due to the fact that the more number of AEs the more complex the network would be, and thus consuming more time to adjust parameters, and causing the performance to improve slowly.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

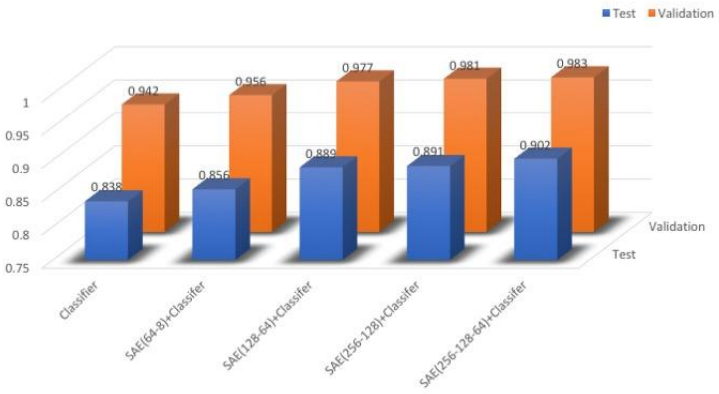

Fig. 5. Impact on the accuracy of **C**lassifications of different DNN structures

Fig. 6 shows the changes of the accuracy of the classifications of the buildings and the floors on both the test-set and the validation-set along with a training duration by SAE (256-128). The degree of the correct recognitions by the SAE achieves up 95% or higher after the 11th iteration.

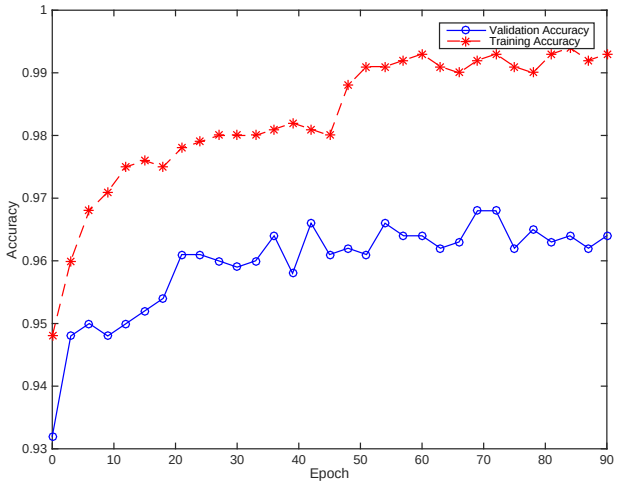

Fig. 6. Accuracy of test and validation dataset using DNNIP

4.2 Accuracy Comparison of Different Algorithms

The SAE that does not partition the dataset is labeled as DNLIP. And the algorithm of the experiment used for the building level and the floor level in the training and simulation in [11] is labeled as SVM, i.e., the random forest and decision tree algorithms. As shown in Fig. 7, DNNIP presents the highest accuracy of 88.9%, while SVM has the lowest accuracy of 82.5%. Compared with the DNLIP algorithm, the DNNIP al-

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

gorithm has some improvement. This is because that when the DNLIP algorithm classifies the buildings and floors, it gets the same loss value in the training phase and cannot obtain the correct high recognition rate.

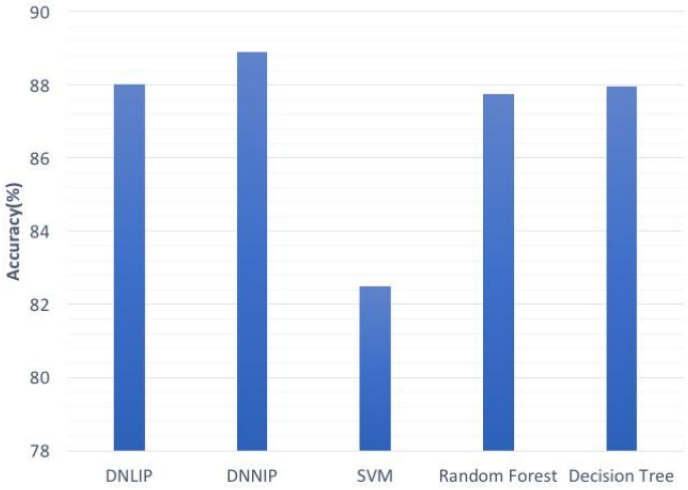

Fig. 7. Accuracy comparison of building and floor positioning

The first step of using the DNNIP method for the indoor positioning is the classification of the buildings, and the accuracy of the algorithm is shown in Fig. 8. It can be seen that the algorithms above can achieve very good accuracies. And the accuracy of the decision tree algorithm can be up to 99.2%, which is better than all other machine learning algorithms.

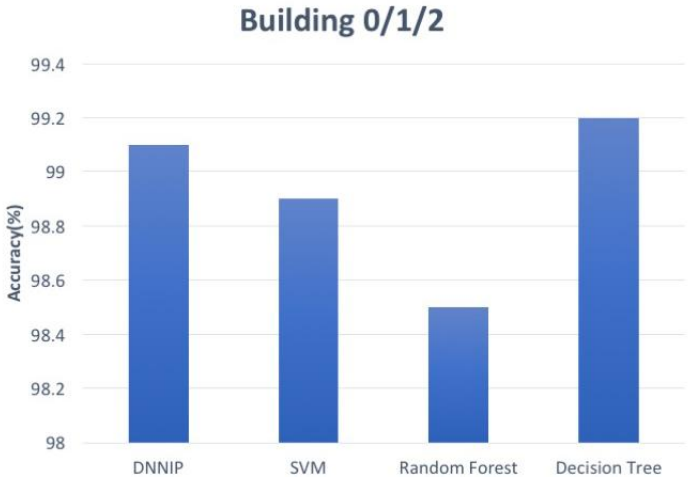

Fig. 8. Accuracy comparison of building positioning

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

The second step is to classify the floors of each building using the floor attributes. The resultant outcome is shown in Fig. 9. It indicates that when the data set is divided by the numbering of the buildings, the obtained accuracy of the proposed DNNIP is better than other machine learning algorithms in most cases. The averaged positioning accuracy of this DNNIP algorithm is 93.8%, which suggests that DNNIP has a certain capability in accuracy.

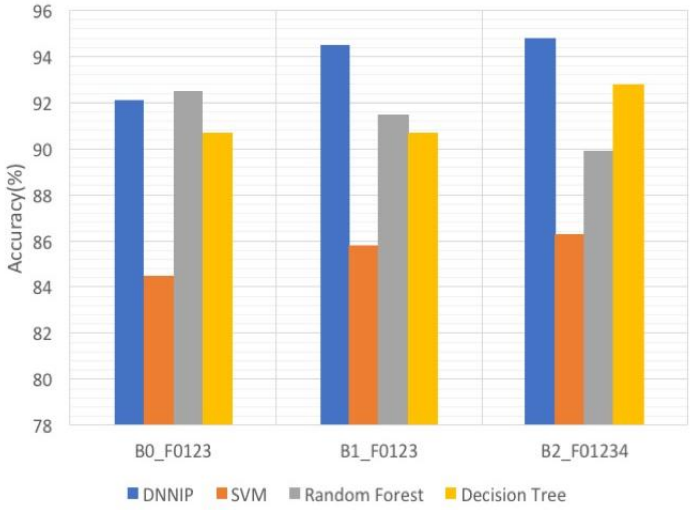

Fig. 9. Accuracy comparison of floor positioning

Finally, the accuracies of the classifications against the space among these algorithms are compared with results shown in Fig. 10. It shows that the DNNIP has the highest average positioning accuracy in some classifications. It can also be found that all the algorithms would get worse results when classifying the ground and top floors than classifying the middle floors for the same building.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

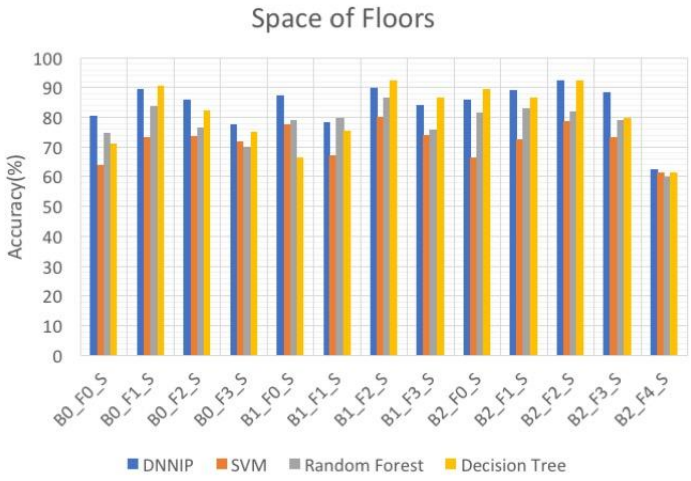

Fig. 10. Accuracy comparison of space positioning

5 Conclusion

In this paper, the DNNIP method is proposed to locate the user's position in large indoor buildings, and further used to segment and layer the original UJIIndoorLoc dataset. The DNNIP takes the network structure of the stacked auto-encoders and ReLu activation function so as to avoid the gradient disappearance in the training processes. The obtained classifying accuracy of the proposed DNNIP algorithm is the highest among all other machine learning algorithms. And after the training, this algorithm does not need to find the best match of samples within the database, which saves the time of manually adjusting the parameters. One disadvantage of the proposed DNNIP algorithm is that when being used against the hierarchical datasets for the training, the computation is much higher than that of the traditional machine learning algorithms, and more computing resources are needed when the training set is updated and adjusted.

References

1. Gu, Y. Lo, A., Niemegeers, I.: A survey of indoor positioning systems for wireless personal networks. IEEE Communications Surveys & Tutorials, 11(1):13-32 (2009).
2. Machaj, J., Brida, P., Majer, N.: Challenges introduced by heterogeneous devices for Wi-Fi-based indoor localization. Concurrency and computation: practice and experience, 1-10 (2019).
3. Miao, H., Wang, Z., Wang, J., Zhang, L., Zhengfeng, L.: A novel access point selection strategy for indoor location with Wi-Fi. In: China Control and Decision Conference (CCDC), pp. 5260-5265, IEEE, Changsha (2014).

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3

4. Wang, B., Zhou, S., Yang, L.T., Mo, Y.: Indoor positioning via subarea fingerprinting and surface fitting with received signal strength. Pervasive & Mobile Computing, 23:43-58 (2015).
5. Lin, T., Fang, S., Tseng, W., Lee, C., Hsieh, J.: A group-discrimination-based access point selection for WLAN fingerprinting localization. IEEE Transactions on Vehicular Technology, 63(8): 3967-3976 (2014).
6. Liu, W., Fu, X., Deng, Z., Xu, L., Jiao, J.: Smallest enclosing circle-based fingerprint clustering and modified-WKNN matching algorithm for indoor positioning. In: International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1-6, IEEE, Alcala de Henares (2016).
7. Gharghan, S.K., Nordin, R., Ismail, M., Ali, J.A.: Accurate wireless sensor localization technique based on hybrid PSO-ANN algorithm for indoor and outdoor track cycling. IEEE Sensors Journal, 16(2): 529-541 (2016).
8. Turgut, Z., Ustebay, S., Aydin, M.A., Aydin, Z.G., Sertbaş, A.: Performance analysis of machine learning and deep learning classification methods for indoor localization in Internet of things environment. Transactions on Emerging Telecommunications Technologies, 30(9), 1-18 (2019).
9. Ma, Y.-W., Chen, J.-L., Chang, F.-S., Tang, C.-L.: Novel fingerprinting mechanisms for indoor positioning. International Journal of Communication Systems, 29(3): 638–656(2016).
10. Félix, G., Siller, M., Álvarez, E.N.: A fingerprinting indoor localization algorithm based deep learning. In: Eighth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 1006-1011, IEEE, Vienna, Austria (2016).
11. Nowicki, M., Wietrzykowski, J.: Low-effort place recognition with WiFi fingerprints using deep learning. In: International Conference on Automation, pp. 575-584, Springer, Cham (2017).
12. Zhang, W., Liu, K., Zhang W.D., Zhang, Y., Gu, J.: Deep Neural Networks for wireless localization in indoor and outdoor environments. Neurocomputing, 194:279-287(2016).
13. Moreira, A., Nicolau, M.J., Meneses, F., Costa, A.: Wi-Fi fingerprinting in the real world-RTLS@UM at the EvAAL competition. In: International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1–10, IEEE, Banff AB (2015).
14. Torres-Sospedra, J., Montoliu, R., Martinez-Uso, A., et al.: UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In: International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 261-270, IEEE, Busan, South Korea (2014).

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_3