CNNLoc: A Deep-Learning Based WiFi Fingerprinting Framework for Indoor Localization

Xudong Song*†, Xiaochen Fan†, Xiangjian He†, Chaocan Xiang‡§, Qianwen Ye†¶, Zumin Wang *
* College of Information Engineering, Dalian University, Dalian, China
† School of Electrical and Data Engineering, University of Technology Sydney, Sydney, Australia
‡ College of Computer Science, Chongqing University, Chongqing, China
§ Army Logistics University of PLA, Chongqing, China
¶ Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China
{xudong.song, xiaochen.fan, xiangjian.he, qianwen.ye}@uts.edu.au {xiang.chaocan}@gmail.com {wangzumin}@dlu.edu.cn

***Abstract*—With the ubiquitous deployment of wireless systems and pervasive availability of smart devices, indoor localization is empowering numerous location-based services. With the established radio maps, WiFi fingerprinting has become one of the most accessible and practical approaches to localize a mobile user. However, most fingerprint-based localization algorithms are computation-intensive, with heavy dependence on both offline training phase and online localization phase. In this paper, we propose CNNLoc, a Convolutional Neural Network (CNN) based indoor localization framework with WiFi fingerprints for multi-building and multi-floor localization. We propose a novel classification model by combining Stacked Auto-Encoder (SAE) with one-dimensional CNN. The SAE can be used to extract key features more precisely from sparse Received Signal Strength (RSS) data, and the CNN can be trained to effectively achieve high success rates in the localization phase. We evaluate CNNLoc with state-of-the-arts as benchmarks on the UJIIndoorLoc dataset and Tampere dataset. CNNLoc shows its excellence in both building-level and floor-level classifications and outperforms the existing solutions with 100% success on building success rate and an average success rate over 95% on floor-level localization.**

***Keywords*-Indoor Localization; Deep Learning; Convolutional Neural Network; WiFi Fingerprinting**

I. Introduction

Indoor Location-Based Services (ILBSs) have become an essential component for various indoor applications, such as location based wireless advertising, information retrieval and pedestrian navigation [1]. With explosive demand on high-accuracy and low-cost localization, indoor positioning has attracted a lot of interests from industrial community to research literature. Knowing the floor level of a mobile user is particularly useful to a variety of location based applications in a multi-building and multi-floor environment. Various indoor localization techniques have been proposed using different types of modalities, such as, WiFi, visible light, acoustic, Bluetooth, cellular network and their combinations [2]. The majority of localization techniques utilize Received Signal Strength (RSS) of Wireless Access Points (WAPs) [3] to identify locations of mobile users. The RSS fingerprints observed by mobile users can be leveraged to deduce their locations with the pre-constructed fingerprint dataset.

* Zumin Wang is the corresponding author.

In general, there are two phases in fingerprinting localization, *i.e.*, the offline phase (training phase) and online phase (positioning phase). In the offline phase, the fingerprint dataset (radio map) is constructed by collecting RSS fingerprinting data at pre-known reference points of interested areas, shown in Fig. 1. In this way, the RSS fingerprints are labelled with the location information for training and matching purposes [4]. In the online phase, mobile users can simply send queries to the system with current RSS measurements. By finding the best match over the dataset, the localization system will return the closest fingerprint to the requester as well as the corresponding locations.

Accordingly, literature studies are confronting with two major issues in fingerprint-based localization, *i.e.*, dataset construction and localization accuracy. In addressing the first issue, numerous recent studies embrace crowdsourcing based techniques [4] to empower automatical fingerprint collection, avoiding labour-intensive site surveys. Meanwhile, despite the efforts to improve the accuracy and efficiency, each localization algorithm is tested with its own dataset, so the reported results of different algorithms are impossible to compare [4]. This is due to each work not only uses its own dataset, but also conducts unrepeatable experiments under different environmental settings [5]. Therefore, some open-source RSS fingerprint datasets are constructed and released as benchmark datasets for indoor localization, including UJIIndoorLoc [6], Tampere University dataset [4] and IPIN dataset. [7]

A key challenge in locating indoor target based on a WiFi fingerprinting dataset is how to achieve high-accuracy and low-cost localization under the fluctuation of signal and noise from multi-path effects. Traditional approaches, including probabilistic , K-nearest-neighbor (KNN) and Support Vector Machine (SVM), are computation-intensive and time-consuming with complex filtering and parameter tuning. Recently, Deep Neural Network (DNN) based localization approaches [8], [9] has been proposed with the rising of deep-learning. Nevertheless, the performance of DNN-based methods is still subject to the sufficiency of input training data. Since DNN is fully connected, its complexity of computation is directly related to the depth (i.e., number of layers) of the neural network, affecting the accuracy of localization results.

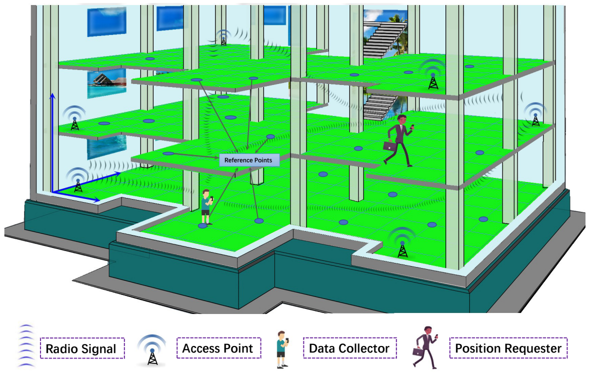

Fig. 1: Fingerprinting based indoor localization scenario

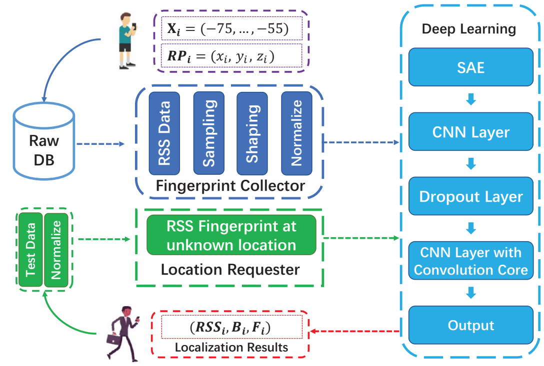

Fig. 2: System architecture of CNNLoc

To address the above issues, we explore the possibility of using a CNN for indoor localization. By leveraging the CNN, the convolution can replace the general matrix multiplication in neural networks thus reducing the computation complexity. We propose a **CNN**-based indoor **Loc**alization with WiFi fingerprinting (CNNLoc). Compared with the existing indoor localization approaches, the main contributions of this work are summarized as follows.

1) We propose an innovative deep-learning based model for multi-building and multi-floor indoor localization. Our model leverages a stacked auto-encoder to reduce the data dimension and combines a one-dimensional CNN to increase the network depth and improve localization accuracy.
2) We present a novel algorithm to extract a verification dataset from the training dataset, resolving the uncertainty brought by conventional random selection especially when the dataset volume is small.
3) We evaluate CNNLoc on two open-source datasets and make a complete comparison with state-of-the-art approaches as benchmarks. CNNLoc demonstrates its excellency with high accuracy, and outperforms all of the benchmarks with 100% success on building success rate and an average success rate over 95% on floor-level localization.

The remainder of this paper is organized as follows. We review the related indoor localization work in Section II. In Section III, based on open-source datasets, we present the system architecture of CNNLoc. In Section IV, we devise an innovative algorithm for extracting a verification dataset and introduce the data preprocessing and model pre-training. In Section V, we optimize our model through experiments and compare CNNLoc with several benchmarks in terms of localization accuracy under two public datasets. At last, we conclude this work with a discussion on future work in Section VI.

II. Related Work

WiFi based Indoor localization generally falls into two main categories: device-free and device-based localizations [3]. In device-free localization, the target entities do not carry any wireless devices and the deployed system captures the presence and motion of each target entity via their reflection on the WiFi signals. Meanwhile, device-based localization tracks a WiFi-enabled mobile device through various measurements, including Time-of-Arrival (ToA), Angle-of-Arrival (AOA), Received Signal Strength (RSS) and Channel State Information (CSI). WiFi fingerprinting has become a major approach as it can achieve applicability in various indoor environments. In particular, machine learning has recently become attractive for WiFi-fingerprinting based indoor localization. In [10], a weighted KNN algorithm was proposed to assign different weights to WAPs and it can achieve room-level localization. In [11], a DNN-based indoor fingerprinting scheme was employed to address the labour-intensive and time-consuming issues in achieving reliable and accurate localization. In [12], a deep CNN was proposed to train the weights of AOA images derived from CSI information. In [13], a CNN-based WiFi fingerprinting method was presented, and it outperformed the DNN-based methods. In this paper, we leverage deep learning by integrating a CNN with SAE for more accurate and efficient localization in a multi-building and multi-floor environment.

III. System Design

In this section, we first present the system architecture of CNNLoc, then we introduce the offline phase and online phase with detailed model design of CNNLoc, respectively.

A. System Architecture

In this subsection, we present the system architecture of CNNLoc, as shown in Fig. 2. The localization process of CNNLoc consists of training phase and testing phase. The first step of training phase it to build a fine-grain fingerprint database and preprocess the RSS data into normalized fingerprint. More importantly, the deep learning model is built with SAE layers and CNN layers, then we train this model with the preprocessed data. In online positioning phase, the CNNLoc will compute the fingerprint data of location requester and send back the localization results to the requester. Next, we first specify the data input and output and present the floor

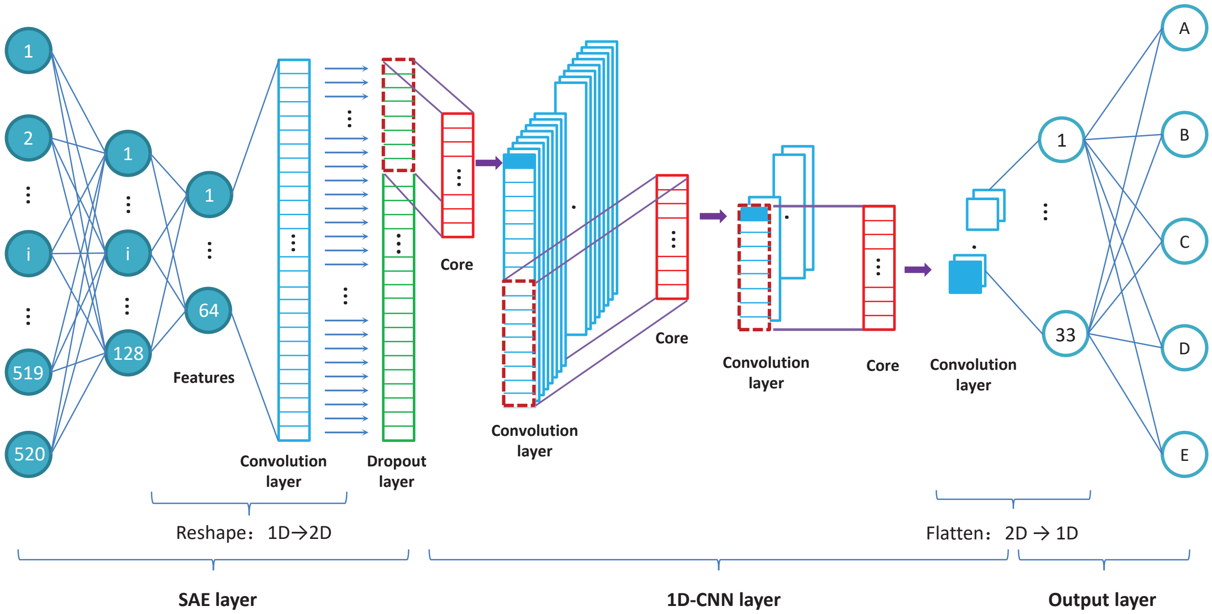

Fig. 3: Floor model with SAE and CNN layers

model and building model in detail. In addition, we explain the cost function and earlystopping methods used in CNNLoc model.

B. Input and Output Specification

TABLE I: Feature comparison of two datasets

Features	UJIIndoorLoc	Tampere
Number of training samples	19938	697
Number of test samples	1111	3951
Number of WAPs	520	992
Default value of missed RSS	100	100
Floor representation	floor number	floor height

To specify the input and output structure of proposed model, we employ UJIIndoorLoc [6], the biggest open-access indoor localization database in literature studies. To evaluate the performance of CNNLoc, we also adopt the recently released Tampere dataset and the details are shown in Section V. As shown in Table I, the UJIIndoorLoc database contains 21049 fingerprint samples and covers 3 buildings with 4 or 5 floors. Accordingly, we determine the input of CNN model as $\mathbf{X} = (x_1, ...x_i, ...x_{520})$, where $\mathbf{x}$ is a 1-D vector with length of 520 and x_i denotes the RSS measurement on WAP i. In training phase, each RSS measurement is associated with an unique label of building number or floor number. For example, $(\mathbf{X}, b, f)$ denotes the RSS fingerprint $\mathbf{X}$ with building label b and floor label f. For training purpose, the dataset is divided into three parts, *i.e.*, the training set, the verification set and the test set.

C. Floor Classification Model

In this section, we introduce the one-dimensional convolutional neural network (1D-CNN) classification model for multi-level indoor localization. The 1D-CNN model consists of self-encoding layers, a dropout layer, convolutional layers and an output layer. The architecture of floor model is illustrated in Fig. 3 and we introduce the details of 1D-CNN model as follows.

SAE: For sparse data, the Stacked Auto-Encoder (SAE) can effectively reduce the data dimension and still preserve the necessary feature information [11]. In our SAE model, we leverage unsupervised learning to compress the input data to 64 features. The outputted features are further connected to CNN model for classification.

1D-CNN: Due to the small variations of RSS fingerprints between adjacent floors, a major challenge in improving localization accuracy is how to distinguish users at these floors. For example, a benchmark approach proposed in [6] can only achieve the success rate of floor classification at 89.92%. In order to obtain better prediction results, we connect a 1D-CNN model to the pre-trained SAE to further supervise the entire model. With the calculation of SAE, the input data of CNN is refined as a feature vector with length of 64. To avoid overfitting, we add a Dropout layer between the SAE and CNN. To enable convolutional calculation, we further convert the feature vector into a two-dimensional vector data. The 1D-CNN operate three convolutional operations, the output is a set of 33 convolutional layers. To this end, the two-dimensional layers are flatten into one-dimensional features connected to output layer for classification.

D. Building Classification Model

While building classification is prior to the floor classification, we use the same self-coding layer as floor model, which is further connected to a fully connected hidden layer and a building classification layer.

E. Cost Function

We adopt cost function 1 to calculate the degree of inconsistency between the localization results and groundtruth. The localization result is more accurate when the value of cost function is smaller. The cost function we use is called quadratic cost function, as shown in Equation 1.

$$C = \frac{1}{2n} \sum_{x} \|y - a\|^2 \tag{1}$$

Here, n is the total number of input samples, $\mathbf{X}$ represents the input vector of training samples, $\mathbf{y}$ represents the vector of groundtruth, $\mathbf{a}$ represents the corresponding localization output vector.

F. EarlyStopping Strategy

We further adopt EarlyStopping strategy to monitor the performance of CNNLoc on verification set. During each training session, the trained model will be evaluated by verification dataset. For example, if the localization accuracy on verification dataset shows no improvement within last p training sessions, the training process will be terminated. In CNNLoc model, we use the patience parameter p to control the training process with EarlyStopping. In this way, we can improve the training efficiency and localization performance simultaneously.

IV. Data pre-processing and Model pre-training

Before start to train the deep learning model proposed in previous section, we pre-process the data and pre-train the model in three folds. First, we devise a novel algorithm to extract a verification set from the training set. Second, we normalize the input data and restore it into original representation, thereby improving the accuracy of the result. Third, the SAE model is pre-trained before we start to train the whole CNNLoc model.

A. Extracting the verification set

CNNLoc has several setting that we can use to control the behavior of the learning process, and these settings are called $hyperparameters$. If learned on the training set, the hyperparameters would always choose the maximum possible model capacity, resulting in overfitting. Therefore, we need to extract a disjoint set called verification set out of training set and use this verification set to update the hyperparameters accordingly. However, as most conventional methods extract the verification set by random selection, the performance of trained model are not always stable. To solve this problem, we adopt uniform sampling [14] to extract the verification set from the training set evenly, thus improving the stability experimental results.

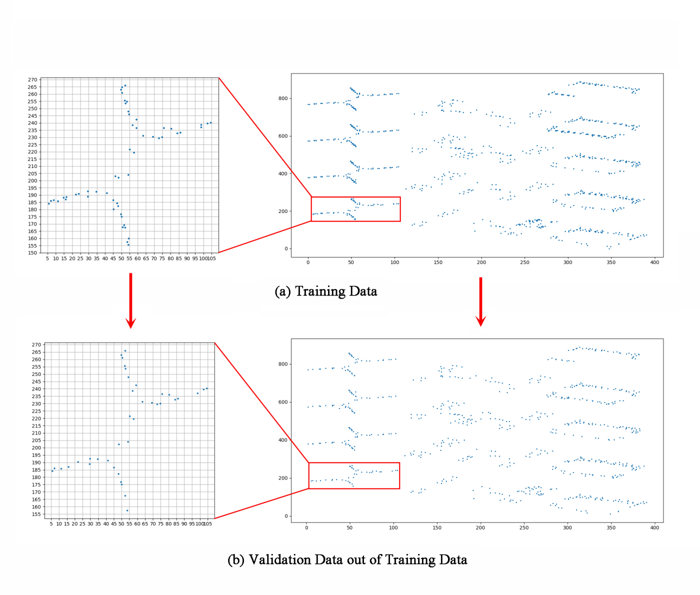

Fig. 4: Extract the verification dataset from the training dataset

Methodology: The uniform sampling algorithm for UJIIndoorLoc dataset is presented in Algorithm 1. The input is the all-dataset AD, side length of grid L and number of samples per grid N. The output is the training set T and the verification set V.

First, the all-dataset AD is divided into sub-dataset SD by building and floor. Second, we create a grid with a cell length of L for each floor to cover all the coordinates in SD. The set of center coordinates for each cell in the SD is denoted as C. Third, for each coordinate center G in set C, we select a set of data I in the SD covered by the cell whose center is G. Finally, if I not empty, for each N data closest to the coordinates of the center point G are selected as verification set V the remaining part as training set T in the set of data I.

As shown in Fig. 4, subgraph(a) shows the relative position of AD, and subgraph(b) is the relative position of the verification set V, which is extracted from AD. In this experiment, the parameters L, N is set to 5 and 3 respectively. It is worth mentioning that this dataset collects multiple data at each position, so the points shown in the figure are the result of the superposition, and the number of real points is more than the number of displays.

B. Dataset preprocessing

We preprocess UJIIndoorLoc dataset for training CNNLoc model first. In this dataset, the value of input RSS data ranges from -104 dBm to 0 dBm, and we convert these RSS values from (-104, 0) into (0, 1) using Function 5. For any WAP is not detected in one measurement, its RSS value is marked as 100 dBm, and we denote the RSS value on any undetected WAP 0. In [15], it is indicated that the different data representation of RSS fingerprints can influence the success rate and error rate during localization. Specifically, the authors compare the linear representation (Equation 3), exponential representation (Equation 4) and the exponential representation (Equation 5). As a result, exponential and powed data tend to represent

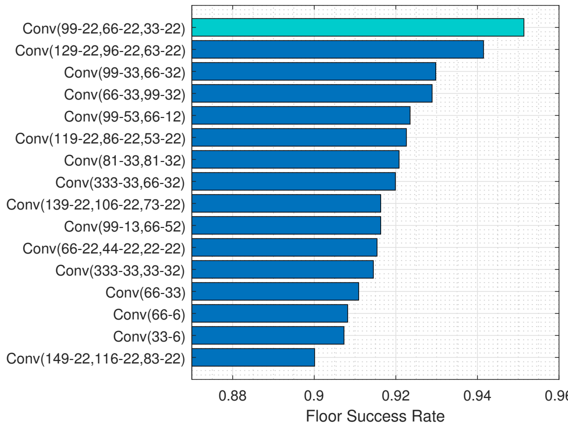

Fig. 5: Effect of different CNN floor classification

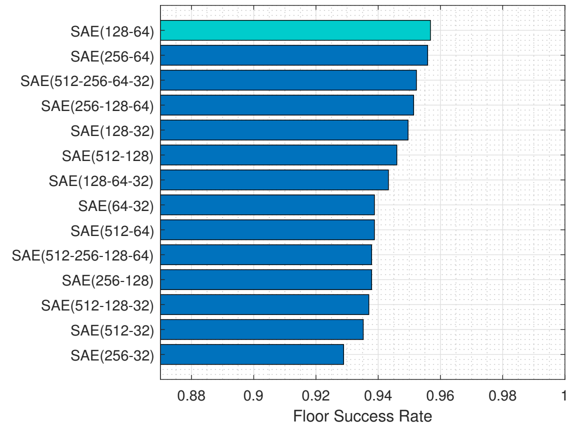

Fig. 6: Effect of different SAE floor classification

Algorithm 1 The verification set is extracted from the training data.

Input: All-dataset AD,
Side length of grid L,
The number of samples per grid N
Output: Training set T,
Verification set V

for all Getting Sub-dataset SD of each floor of each building from AD **do**
  Create a grid with a cell length of L for each floor to cover all the coordinates in the SD. C ←The set of center coordinates for each cell
  for all $G \in C$ **do**
    I ←A set of data in the SD covered by the cell whose center is G
    if I not empty **then**
      NP ←(N data closest to G center in I)
      Delete NP from I
      $V.append(NP)$
      $T.append(I)$
    end if
  end for
end for
return T,V

the RSS values as they really are. Furthermore, they can better tame the fluctuations existing RSS signals, so we adopt exponential and powed data for WiFi fingerprint representation in this work.

$$Positive_i(x) = \begin{cases} 0, & RSS_i \ is \ None \\ RSS_i - min, & otherwise \end{cases} \tag{2}$$

where i is the WAP identifier, RSS is the actual intensity level provided by i-th WAP, min is the lowest RSS value minus 1 considering all the of fingerprints and WAPs of the database.

$$0-1Normalized_i(x) = \frac{Positive_i(x)}{-min} \tag{3}$$

Equation 3 corresponds to the positive values representation, where intensity values are normalized in the positive [0, 1] range.

$$Exponential_i(x) = \frac{\exp(\frac{Positive_i(x)}{\alpha})}{\exp(\frac{-min}{\alpha})} \tag{4}$$

$$Powed_i(x) = \frac{(Positive_i(x))^\beta}{(-min)^\beta} \tag{5}$$

where α is set to 24 and the exponent β is set to the mathematical constant e.

C. Model pre-training

As the input data for WiFi fingerprint positioning is very sparse, we import SAE model before CNN layers to compress the dimensions of the input data. In specific, we pre-train the SAE network [9] to obtain appropriate parameters before we train the CNNLoc model. Ultimately, the whole CNNLoc model (consisting of encoder of SAE and CNN layers) will be further fine-turned.

V. Performance Evaluation

In this section, we evaluate the proposed CNNLoc by comparing its performance with state-of-the-art approaches. Two public open-source datasets, UJIIndoorLoc dataset [6] and Tampere dataset [4], are employed for experimental studies. We implement the CNNLoc on a Computing Cluster with 16GB GPU, and the deep learning model is trained on Keras-2.2.2 (with Tensorflow-gpu-1.10.0) using Python-3.6.6.

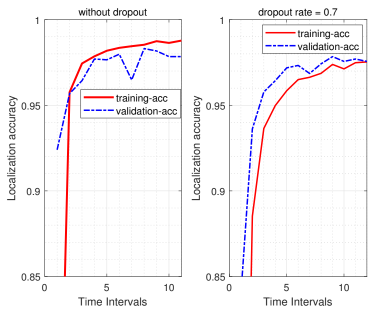

Fig. 7: Floor success rate without/with dropout layer

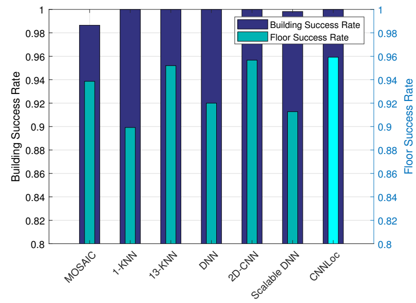

Fig. 8: Localization accuracy compared with state-of-the-arts

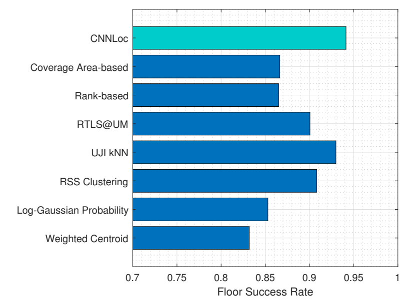

Fig. 9: Localization accuracy compared with other approaches

TABLE II: General parameter setting of the model

Parameter	Values
SAE activation function	Rectified Linear Unit (ReLU)
SAE Optimizer	Adam (lr=0.001)
SAE loss	Mean Squared Error (MSE)
1D-CNN activation function	ReLU
1D-CNN Optimizer	Adam (lr=0.001)
1D-CNN loss	MSE
1D-CNN Output layer activation function	Softmax
Earlystopping parameter patience	3
Batch size	66

A. Model Optimization

In this section, we explore how to optimize the CNNLoc model through experiments. The parameters used in the optimization experiments are shown in Table II.

CNN model optimization. We evaluate the structure of a CNN model with the same SAE model. To make the comparison be more reasonable, we adopt the SAE (256-128-64) from [11], where the SAE contains three hidden layers of 256, 128 and 64 neurons. We vary the CNN structure with number of layers from 1 to 3 with different filters and kernel sizes. For example, Conv(99-33,63-22) refers to two convolutional layers having 99 output filters with kernel size 33 and 63 output filters with kernel size 22, respectively. In Fig. 5, we compare performance of different CNN models and find that Conv(99-22,66-22,33-22) can achieve the best success rate of 95.14% in floor classification.

SAE model optimization. We evaluate the performance of various SAE models by using the optimized CNN model. Therefore, the Conv(99-22,66-22,33-22) is connected to the SAE models with different combinations of hidden layers. The comparison results are shown in Fig. 6 and the best SAE model is the two-layer SAE with 128 and 64 neurons in each layer.

Dropout layer optimization. To prevent convolutional neural networks from overfitting, we further adopt a dropout layer before convolutional computation. According to a fraction $rate$, the dropout layer randomly sets the number of input units to 0 at each update during the training process. We use the optimized SAE model and CNN model obtained from above evaluations and evaluate CNNLoc with different dropout rates. We list the floor classification success rates under different dropout rates in Table III. It can be observed that the success rate reaches the peak of 0.9541 when the dropout rate is 0.7. On this basis, we show the validation and training results without/with dropout (of rate=0.7) in Fig. 7. During each iteration of the training stage, we test the CNNLoc model using the training dataset and verification dataset, respectively. In Fig. 7, the blue curves show the test results using the verification dataset and the red curves show the test results using the training dataset. This figure shows that as the training time increases, the accuracy rates using the training dataset and verification dataset both rise to over 95%. However, when the CNNLoc does not have the dropout layer, the overfitting problem occurs, as the localization accuracy on the verification dataset is lower than and not approaching to that on the training dataset. In contrast, when we integrate a dropout layer to CNNLoc, the localization results on verification dataset and training dataset ultimately approach to each other. In this way, we address the overfitting problem in model training.

TABLE III: Results of different dropout rates

Dropout rate	0.4	0.5	0.6	0.7	0.8	0.9
Success rate	0.9478	0.9529	0.9520	0.9541	0.9514	0.9481

B. Evaluation on different verification datasets

To validate the proposed Algorithm 1 on various verification datasets, we further evaluate the impact of the selection of a verification dataset. Here, the parameters set for CNNLoc are shown in Table II and Table IV. Based on the trained model using a fixed or a randomly selected verification dataset, we evaluate the performance of CNNLoc using the test dataset for 5 times and present the results in Table V. It can be observed that the testing results on the model with fixed verification datasets are more stable than those of randomly extracted verification dataset.

The primary parameters used in these experiments are inherited from Table II, and added parameters are listed in Table IV. The results and their average values are listed in Table V. The experimental results show that using a randomly selected verification dataset are less stable and the average success rate is lower compared with the results using the fixed verification dataset.

TABLE IV: Parameter Values for model

Parameter	Values
SAE hidden layers	(128-64-128)
1D-CNN hidden layers	(99-22, 66-22, 33-22)
SAE/1D-CNN	Adam(lr=0.0001)
Dropout rate	0.7
Max training iterations	40

TABLE V: Comparison of results on different verification datasets

Data type	1st(%)	2nd(%)	3rd(%)	4th(%)	5th(%)	Average(%)
fixed	94.96	94.69	95.41	95.32	94.96	95.07
random	95.41	93.97	95.14	92.08	94.51	94.22

C. Experiments on UJIIndoorLoc dataset

In this subsection, we demonstrate the superiority of the proposed CNNLoc model by comparing it with the state-of-the-arts on the UJIIndoorLoc dataset. We apply the uniform extraction algorithm on UJIIndoorLoc dataset and obtain 2,198 samples for the verification set, 17,739 samples for the training set, and 1,111 samples for the testing set. The benchmark approaches include MOSAIC [16], 1-KNN [6], 13-KNN [15], DNN [11] 2D-CNN [13] and scalable DNN [9]. We illustrate the comparison results in Fig. 8 in terms of building success rate and floor success rate. CNNLoc and most of the benchmark approaches achieve 100% success rates in building localization, showing that the proposed SAE can effectively handle building classification. Meanwhile, CNNLoc outperforms other benchmarks with a highest floor success rate of 95.92%.

D. Experiments on Tampere dataset

To show the scalability of CNNLoc, we adopt a recently released WiFi RSS fingerprint database, *i.e.*, Tampere dataset [4], to evaluate the performance of CNNLoc. In Table I, we compare the key features of UJIIndoorLoc dataset and Tampere dataset. With 992 WAPs in theTampere dataset, CNNLoc is adapted to have an input layer with the length of 992 at the SAE model. As the Tampere dataset uses floor height as the floor representation instead of the floor number used in UJIIndoorLoc, we preprocess the dataset using the z-coordinates to represent the floor levels. The experimental results are shown in Fig. 9 and CNNLoc outperforms the state-of-the-art methods by 2.45% to 12.25%. Therefore, CNNLoc performs highly accurate rates in multi-building and multi-floor localization.

VI. Conclusion

In this work, we have presented CNNLoc, a deep-learning based WiFi fingerprinting framework for multi-building and multi-floor localization. By combining SAE with a 1D-CNN, CNNLoc can be used to precisely extract key features from sparse WiFi fingerprints and achieve a high success rate. We have evaluated CNNLoc on two open-source datasets, *i.e.*, UJIIndoorLoc and Tampere. The experimental results have demonstrated the superiority of CNNLoc, as CNNLoc has achieved the highest success rates in multi-building and multi-floor localization compared with the state-of-the-art approaches. As our future work, we will focus on improving location accuracy by devising new deep-learning models for indoor localization.

Acknowledgment

Xudong Song and Xiaochen Fan have equal contribution to this work.

References

[1] A. Basiri, E. S. Lohan, T. Moore, A. Winstanley, P. Peltola, C. Hill, P. Amirian, and P. F. e Silva, "Indoor location based services challenges, requirements and usability of current solutions," *Computer Science Review*, vol. 24, pp. 1–12, 2017.

[2] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2016.

[3] J. Xiao, Z. Zhou, Y. Yi, and L. M. Ni, "A survey on wireless indoor localization from the device perspective," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, p. 25, 2016.

[4] E. S. Lohan, J. Torres-Sospedra, H. Leppäkoski, P. Richter, Z. Peng, and J. Huerta, "Wi-fi crowdsourced fingerprinting dataset for indoor positioning," *Data*, vol. 2, no. 4, p. 32, 2017.

[5] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental limits of rss fingerprinting based indoor localization," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 2479–2487.

[6] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "Ujiindoorloc: A new multi-building and multi-floor dataset for wlan fingerprint-based indoor localization problems," in *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*. IEEE, 2014, pp. 261–270.

[7] R. Montoliu, E. Sansano, J. Torres-Sospedra, and O. Belmonte, "Indoorloc platform: A public repository for comparing and evaluating indoor positioning systems," in *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference on*. IEEE, 2017, pp. 1–8.

[8] W. Zhang, K. Liu, W. Zhang, Y. Zhang, and J. Gu, "Deep neural networks for wireless localization in indoor and outdoor environments," *Neurocomputing*, vol. 194, pp. 279–287, 2016.

[9] K. S. Kim, S. Lee, and K. Huang, "A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on wi-fi fingerprinting," *Big Data Analytics*, vol. 3, no. 1, p. 4, 2018.

[10] J. Niu, B. Wang, L. Cheng, and J. J. Rodrigues, "Wicloc: An indoor localization system based on wifi fingerprints and crowdsourcing," in *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3008–3013.

[11] M. Nowicki and J. Wietrzykowski, "Low-effort place recognition with wifi fingerprints using deep learning," in *International Conference Automation*. Springer, 2017, pp. 575–584.

[12] X. Wang, X. Wang, and S. Mao, "Cifi: Deep convolutional neural networks for indoor localization with 5 ghz wi-fi," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[13] J.-W. Jang and S.-N. Hong, "Indoor localization with wifi fingerprinting using convolutional neural network," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2018, pp. 753–758.

[14] H. Zheng, F. Yang, X. Tian, X. Gan, X. Wang, and S. Xiao, "Data gathering with compressive sensing in wireless sensor networks: A random walk based approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 35–44, 2015.

[15] J. Torres-Sospedra, R. Montoliu, S. Trilles, Ó. Belmonte, and J. Huerta, "Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263–9278, 2015.

[16] R. Berkvens, M. Weyn, and H. Peremans, "Localization performance quantification by conditional entropy," in *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*. IEEE, 2015, pp. 1–7.