GAN+: Data Augmentation Method Using Generative Adversarial Networks and Dirichlet for Indoor Localisation

Seanglidet Yean[1], Palak Somani[2], Bu Sung Lee[2] and Hong Lye Oh[2]

[1] *Singtel Cognitive and Artificial Intelligence Lab (SCALE@NTU), School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Ave, Block N4, Singapore 639798*

[2] *School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Ave, Block N4, Singapore 639798*

Abstract

Indoor localisation is indispensable to many service applications ranging from hospitals, malls, to parking lots. Several machine learning techniques have been explored with limited success. This is partly due to insufficient training data. This paper presents a data augmentation workflow, GAN+, which uses Dirichlet distribution coupled with a Generative Adversarial Network. It aims to improve the performance of the model by synthetically producing additional training samples without the need for any extra human effort. The proposed technique was tested using the multi-building dataset, UJIndoorLoc [1]. Experimental results show that the dataset generated by GAN+ can achieve an average location error that is more than ten times lower than that of the original dataset. Therefore, the proposed data augmentation scheme validates the feasibility of using GAN's in the domain of indoor localisation.

Keywords

Data Augmentation, Generative Adversarial Network, Convolutional Neural Network, Deep Neural Network, Received Signal Strength, Indoor Localisation

1. Introduction

Indoor localisation of mobile nodes has received immense interest recently due to the increase in the number of location-based services ranging from warehousing to disaster management, and also due to the advancements in mobile devices.

For an outdoor setting, GPS is the most common positioning technology used. However, due to the complicated indoor environment, Non-Light-of-Sight (NLos), multipath and signal delay issues, GPS is not able to delivery satisfactory accuracy indoors. Many wireless localisation solutions and technologies have been proposed. Some of these techniques involve the use ultrasonic, infrared, Bluetooth and ultra-wideband. However, these technologies are not ideal for the purpose of indoor localisation. While Ultra-wideband, Bluetooth and Ultrasonic requires special equipment, Infrared is easy influenced by the object in the room. Received Signal Strength (RSS), on the other hand, can be obtained during a packet reception with no impact

IPIN 2021 WiP Proceedings, November 29 – December 2, 2021, Lloret de Mar, Spain

seanglid002@e.ntu.edu.sg (S. Yean); palak001@e.ntu.edu.sg (P. Somani); ebslee@ntu.edu.sg (B. S. Lee); hloh@ntu.edu.sg (H. L. Oh)

0000-0001-5456-4037 (S. Yean); 0000-0001-7828-7900 (B. S. Lee); 0000-0001-8007-3151 (H. L. Oh)

on throughput nor energy consumption, as well as has no additional hardware requirements [2, 3, 4]. Thus, the fingerprinting using RSS has grown to become one of the most popular techniques in indoor localisation. Channel state information (CSI) has also been used in some cases since it provides a more detailed fingerprint [5]. Nonetheless, CSI requies specific wireless network interface cards (NIC), such as MIMO NIC, Intel WiFi Link 5300. Therefore, RSS is still the most popular choice.

An RSS-based fingerprinting method comprises of two stages. In the “offline stage”, also called the training stage, a radio map or site survey is built from radio signatures obtained at several reference locations within the target area. At each reference location, RSS values of the WiFi signal strengths transmitted by multiple Access Points (APs) are collected and stored. Later in the “online stage”, a localisation model can estimate the real-time location of the user by comparing the new fingerprint readings with the ones stored within the database. The offline stage is time-consuming as we need to gather multiple readings at each designated location. The drawbacks of the fingerprint method are a result of the high complexity and computation required to process the large amount of prior data [6].

Indoor localisation using machine learning has grown to become one of the most popular techniques. Many traditional machine learning methods have been used to build the localisation model. These include Support Vector Machines [7], k-Nearest Neighbors [8], as well as Random-Forest [9]. However, these methods are not able to effectively learn complex features from the training data and have several other limitations. These methods require hands-on feature engineering to improve performance and adapt to the environment complexity. In contrast, deep learning has been successfully utilised in countless fields and has resulted in much better performance. In [10], G. Flix et al, propose using deeper neural networks to increase position estimate performance. Recent work has also focused on tackling the problem of multi-floor indoor localisation. In [11] and [12], a DNN architecture consisting of a stacked auto-encoder and a feed-forward multilabel classifier was proposed. Despite improvement over traditional methods, deep learning-based models have not been able to improve the accuracy significantly. One reason is due to the notable influence that factors like fading and shadowing have on the RSS values. In [13], a time-series of RSS readings has been used to estimate a node’s location. Additionally, CNN’s have also been used to leverage the temporal dependency between time-series readings. In this paper, we examine the benefits of using our proposed augmentation scheme in training and use the CNN time-series methodology suggested in [13], while predicting the location.

Deep learning models however are still not able to provide satisfactory performance largely due to the scarcity of good quality data for training the model. It is not only the quantity but also the quality of the training data that is critical to the success of any deep learning model. Specifically, in the case of indoor localisation, datasets are usually created by manual data collection at several locations within the target building. This is a tedious task and is extremely time-consuming and labor-intensive. Hence, only limited datasets are available in most cases, with insufficient data needed to efficiently train a deep learning model. Furthermore, it is important to note that data collection is not a one-time process in the case of indoor localisation. This is largely due to the fact that the training database needs to be frequently updated and sometimes built a new during any change within the environment (for example – adding or shifting objects/furniture in a room to a new location), consequently causing the fingerprints to change as well. This is thus a cumbersome process that needs to be done frequently over time. To conquer the training data problem, data augmentation has been used to synthetically produce more training data [13, 14].

In this paper,

- We propose GAN+, a data augmentation technique that can be leveraged to get acceptable performance even with an extremely small training database.
- To ensure the quality of the generated data, a filtering technique is proposed to make sure that outlier data records generated (few in number) are not part of the augmented dataset.
- The proposed technique is tested on a public dataset, UJIndoorLoc by benchmarking against Deep Neural Network and a Time-series Convolutional Neural Network.

The remainder of this paper is divided as follows: We summarize some of background about GANs and its related work in indoor localisation within section 2. In section 3, we explain our proposed augmentation schemes. In section 5-6, we go through the experiment setup details, provide the experiment results and discussion. In section 7, we end with a conclusion and future work section.

2. Related Work

Both the outdoor as well as the indoor localisation problem have been extensively studied over the past few years [15, 16, 17]. Many different technologies have been used as a means of location discovery including WiFi [18], Bluetooth [19] as well as visual landmarking [20].

Data augmentation techniques have been widely used in the machine learning domain to increase the number of data samples to train the model. Sinha et al. [13] proposed two data augmentation schemes for RSS values. Original data samples were used to create new samples based on a mean distribution and uniform random numbers. In [21], Rizk et al. showed the benefits of data augmentation in cellular-based localisation using deep learning. Their proposed method showed accuracy improvements in both outdoor as well as indoor localisation. In [22], Hilal et al. proposed a data augmentation technique for room-level indoor localisation. The proposed technique aims to augment data by injecting different scenarios the signal might suffer from in reality.

In the interest of expanding the diversity and improve the quality of the augmented data, the deep-learning based data augmentation approach was introduced. In particular, Generating Adversarial Networks (also known as GANs) can be used to generate new data similar to existing training data and are hence also referred to as generative models. It was first introduced by Goodfellow et al [23] and has been used in a wide variety of applications, whether it be developing new molecules for oncology [24], generating human face images [25], generating new hypothetical inorganic materials[26], or even increasing resolution [27].

A GAN is made up of two components namely the generator Artificial Neural Network (ANN), and the discriminator ANN. These two components compete with one another. The aim of the Generator to generate new data samples while that of the Discriminator is to differentiate these samples (fake) from the original ones. The discriminator network is thus in charge of evaluating the standard of the samples being produced by the generator network. Samples either produced by the generator ANN, or those from the original dataset are fed into the discriminator. It then aims to trace back the origin of the data sample as accurately as possible. The generator on the other hand, grasps a mapping from the latent space to the distribution of the information it tries to clone. This is to ensure that when a noise vector is later passed to it, a sample from the estimated distribution can be produced. The performance of the generator is in fact assessed by that of the discriminator. Its objective is to create data samples that are as close as possible to the original data samples, thereby tricking the discriminator into believing that the fake samples are authentic. By training both the networks parallelly, they're performance both improves with time, and thus the name Generative Adversarial Networks. The discriminator becomes an expert at differentiating original and fake samples, while the generator becomes better at producing samples that are progressively much more similar to the original.

In [28], Li et al. proposed the use of DCGAN to expand the amount of training data collected. They transform the CSI data collected at every reference point into amplitude feature maps. These feature maps are then converted to images and they use Deep Convolution GAN to generate new images from these original amplitude maps. Nonetheless, solutions using CSI demand additional hardware modifications within cellular devices to capture this data thereby making it harder to apply as opposed to RSS-based methods.

In this paper, we use a Dirichlet distribution-based augmentation scheme alongside GANs to generate new RSS fingerprint data. As per our knowledge, there is currently no paper that makes use of GANs in generating synthetic RSS fingerprint data for augmenting data within the domain of indoor localisation. Furthermore, the RSS augmentation technique proposed in this paper can be easily extended to any deep-learning-based fingerprinting system.

3. Methodology

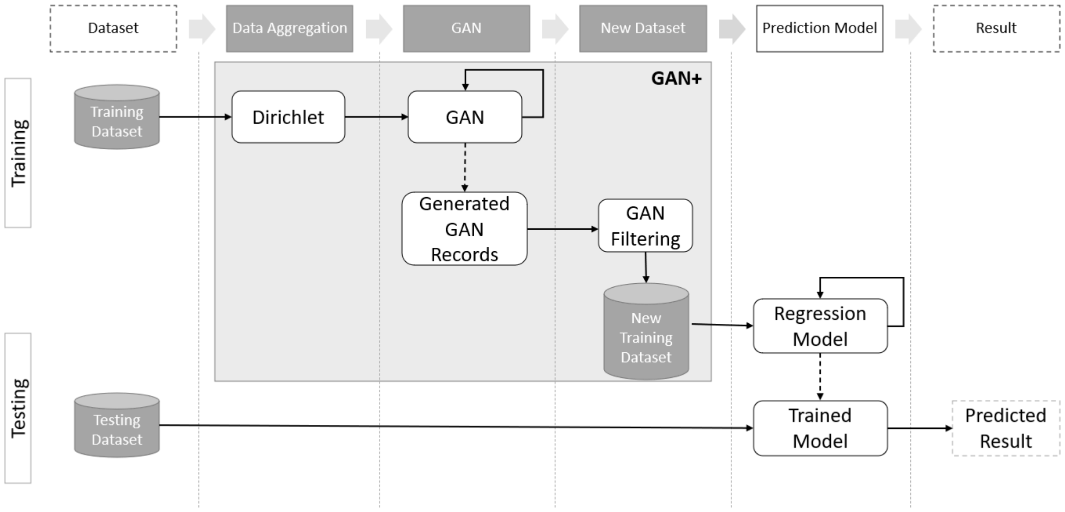


Figure 1: System WorkFlow

In this section, we explore 2 data augmentation schemes: data aggregation using Dirichlet distribution and data augmentation with a Generative Adversarial Network (GAN). We propose the GAN+ framework, which combines the Dirichlet and GAN augmentation techniques, to generate augmented RSS achieving acceptable performance even with a small training database. Figure 1 illustrates the system's workflow overview using the proposed GAN+ technique.

In the case of indoor localisation, a mobile device or a laptop is used to manually collect snapshots of the signal strengths from the neighboring WiFi access points. It is known that the RSS at a specific location generally varies over time. Some of the causes of this effect include indoor objects obstruction, hardware changes, the direction of equipment, humidity, attack, and even temperature [29]. The environmental factors can also lead to a temporary loss of signals received from some network devices. Additional radio bandwidth will also change RSS in a bandwidth-constrained system [30] [31]. GANs can be used to create new entries by tweaking the original data in meaningful ways to simulate the effects of the above conditions. GAN could effectively diversify the training RSS values by constructing new data samples from existing samples. This also ensures that the model is also able to generalize well even for non-line-of-sight (NLOS) conditions.

In this paper, we aim to train a separate GAN at each unique location identified by its longitude and latitude. Due to the lack of sufficient training data at each unique location, other augmentation schemes are explored prior to the use of GAN. Experiments with two such augmentation schemes (prior to using GAN) have been performed in this paper.

3.1. Data Aggregation - Uniform random numbers (DA-Uniform)

As mentioned in section 2, data augmentation techniques aim to replicate the original data to increase both density as well as diversity of data to represent different real-life scenarios. Sinha et al in [13] use mean value and uniform random numbers to create new samples of RSS. Given a particular sample RSS_i, the generated RSS value for each access point ($RSS_{i,ap}$) is randomly selected within the range of $RSS_{i,ap}$ and the mean value of RSS_i.

3.2. Data Aggregation - Dirichlet Method (DA-Dirchlet)

Dirichlet-distribution is well-known in the area of text mining and text network analysis to fit a topic model. It allows a mixture of topics and words to overlap (rather than being repeated in discrete group). Instead of augmenting each records (by row) with uniform random numbers, dirichlet-distributed random variable provides a multivariate generalization of a Beta distribution for **each access point (by column)**.

A Dirichlet distribution[32], represented by a vector $\vec{\alpha}$, has the probability density given by equation (1).

$$\rho(\overrightarrow{x}) = Z \prod_i x_i^{\alpha_i - 1} \tag{1}$$

where random variable x_i is distributed according to the dirichlet distribution $Dir(\alpha)$ ($x_i \sim Dir(\alpha)$) if density function $\rho(\overrightarrow{x})$ holds. $\alpha_i = \{\alpha_1, \alpha_2, ...\alpha_n\} > 0$ is a vector that holds the parameter of the distribution while Z is the generalised multinomial coefficient and n is the number of samples.

The dirichlet augmentation scheme requires at least 2 records from a given location (let the number of records at the unique location be N) (shown in Algorithm 1). Additionally, let the total number of access points be t. In this scheme, new RSS fingerprints are generated by assigning random weightage to each of these N records consisting of t measurements, such that the weights sum up to 1. The reading of an access point in the new sample can be obtained by performing a simple weighted average on the respective access point reading across the N records. In our study, we generate 100 new samples from all the training records at each unique location (identified by its longitude-latitude).

Algorithm 1: Data Augmentation using Dirichlet Distribution (DA-Dirichlet)

```
Initialization new_reading = [0 for i in range(t)]
Generate a Dirichlet distribution of size N (all values sum up to 1) called 'D'
for each access point 't' do
    for each reading 'n' do
        new_reading[t] += (reading of reference point 't' in reading 'n')*(weight of 'n'
          in 'D')
    end
end
Result: new_reading
```

3.3. Data Augmentation - Generative Adversarial Network (DA-GAN)

From DA-Dirichlet, a sufficiently large database is generated for a GAN to effectively learn from. Noise from the physical environment often causes mismatch between the online measuring data and the offline recorded data in the location fingerprint system. The previously stored fingerprinting map thus no longer reflect the statistical features of the current RSS, causing the performance of the system to decrease. GANs can be used to make the offline recorded data more robust thereby substantially improving model performance in real-life scenarios. This is because GANs start from a completely random noise vector and produce outputs that can effectively mimic these constantly varying RSS signals observed over time.

The proposed GAN architecture is depicted in the Figure 2. Upon data aggregation via DA-Dirichlet and data normalisation to $[0, 1]$, DA-GAN is trained at each unique location. Binary Cross Entropy loss (l) is used while training the generator and discriminator of the GAN. The discriminator predicts whether the input is real or fake according to Equation (2). Given y as the ground truth and p as the prediction, we get:

$$l = -[y.log(p) + (1 - y).log(1 - p)] \tag{2}$$

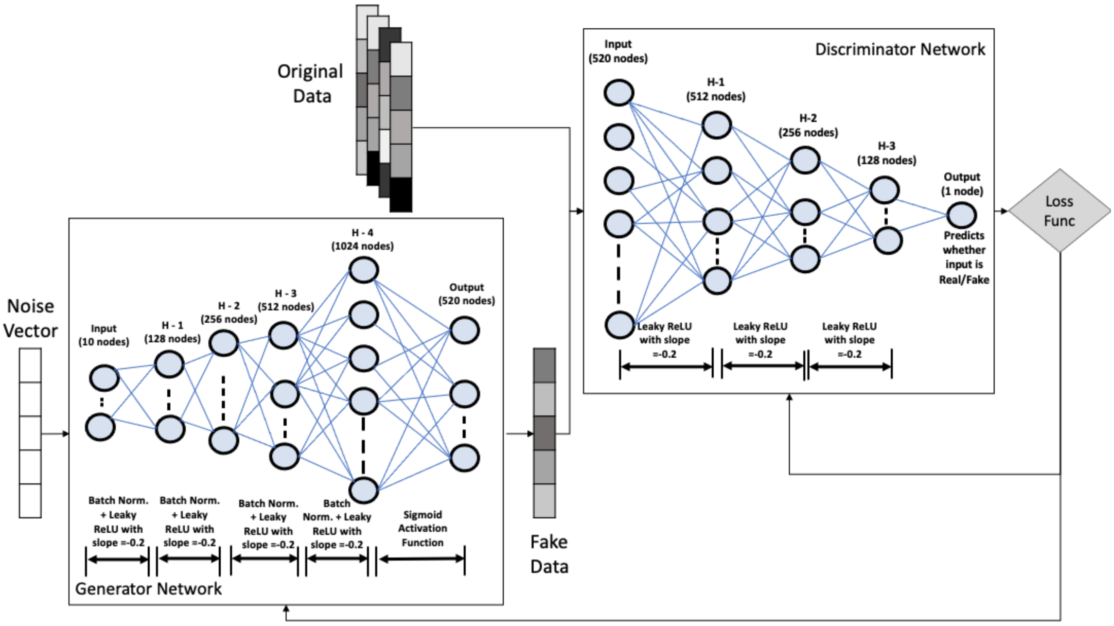

Original Data
Noise Vector
Input (10 nodes)
H - 1 (128 nodes)
H - 2 (256 nodes)
H - 3 (512 nodes)
H - 4 (1024 nodes)
Output (520 nodes)
Batch Norm. + Leaky ReLU with slope =-0.2
Batch Norm. + Leaky ReLU with slope =-0.2
Batch Norm. + Leaky ReLU with slope =-0.2
Batch Norm. + Leaky ReLU with slope =-0.2
Sigmoid Activation Function
Generator Network
Fake Data
Discriminator Network
Input (520 nodes)
H-1 (512 nodes)
H-2 (256 nodes)
H-3 (128 nodes)
Output (1 node)
Predicts whether input is Real/Fake
Leaky ReLU with slope =-0.2
Leaky ReLU with slope =-0.2
Leaky ReLU with slope =-0.2
Loss Func

Figure 2: DA-GAN Architecture

3.4. DA-GAN-Filtering Method

It is important to note that not all the outputs generated from the GAN are of good quality. Since the input to a GAN is a completely randomized vector, the output produced may not always resemble the original data. In fact, some of the outputs generated might have a significantly different structure than the original data. These points act as outliers in the augmented dataset affecting the model learning and performance. These data outliers can thereby spoil and mislead the training process resulting in longer training times, less accurate models, and ultimately poorer results. Thus, we propose a technique to identify and minimise such data points from being included in the augmented dataset. The proposed DA-GAN Filtering technique is robust, unique, simple to implement, and can easily be extended and applied on any dataset.

The idea comes from the fact that the outputs generated by the GAN should be within an allowable deviation range from the corresponding original data samples.

Hence, the maximum allowable threshold for a given dataset was calculated through equations (3) and (4). Given each unique location $loc(lat, lon)$, there exists a sample (RSS_i, RSS_j) combination where $i \neq j$. Each sample refers to the received signal strength $RSS_i = \{RSS_{i,1}, RSS_{i,2}, ..., RSS_{i,AP}\}$ where AP is the total number of detected access points. The differences between the two vectors $(\vec{RSS_i}, \vec{RSS_j})$ can be computed by summation of absolute differences of the two vectors at each access point ap. Equation (3) presents the local threshold (θ_{loc}) as a result of maximum summation differences of all sample combination per unique location. Finally, the maximum allowable threshold (Θ) is the average of maximum summation differences of each location, where num_loc is the total number of unique locations present in the dataset.

It is to be noted that RSS_{O_i} and RSS_{G_i} is the RSS_i of the **O**riginal and **G**enerated dataset respectively.

$$\theta_{loc} = \max_{i,j \in loc; i \neq j} \sum_{ap \in AP} \left| RSS_{O_i,ap} {-}_{O_j,ap} \right| \tag{3}$$

$$\Theta = \frac{\Sigma \theta_{loc}}{num\_loc} \tag{4}$$

In order to ensure the quality of the generated dataset, the output generated by GAN (RSS_{G_i}) is validated according to equation (5). It means that the generated output would be discarded if the variation between the GAN output sample is beyond the acceptable limit Θ. The condition is based on the calculated differences between $RS\vec{S}_G$ and $RS\vec{S}_O$.

$$\left(\min_{O \in loc} \sum_{ap \in AP} \left| RS\vec{S}_{G,ap} - RS\vec{S}_{O,ap} \right| \right) \leq \Theta \tag{5}$$

4. Experiment Setup

Table 2 provides the details of the evaluation models used to test the effectiveness of the proposed data augmentation approach (GAN+).

4.1. Dataset

We assess the performance of our proposed scheme on the UJIIndoorLoc Dataset. The UJIIndoorLoc is a publicly accessible multi-building multi-floor dataset [1]. Table 1 summarises the key details of the UJIIndoorLoc dataset. It is to be noted that the data has been splitted where the testing records are unaltered, un-augmented and different from the training records.

Table 1
UJIIndoorLoc Dataset

Number of Attributes	529
Number of Access Points	520
Number of Training Records	19937
Number of Testing Records	1111
Number of Buildings	3

Our model takes in the RSS readings received from 520 access points as input. The values range from 0 to -104 (immensely poor signal), with +100 being used for access points not detected. As part of preprocessing, readings with values=100 are converted to -110. Furthermore, these values are scaled ranging from [-110,0] to [0,1] respectively before being passed into the model.

Table 2
Evaluation Models' Architecture and Hyperparameters

	DNN	t-CNN
Input	520 (vector)	[520x10] (feature map)
Convolutional Layers	-	Layer1: 8 out channels and 10 x 3 kernel Layer2: 4 out channels and 5 x 3 kernel Pooling: 2 x 2 (stride = 2)
Hidden Layer Units	150,150	128, 128, 128
Outputs	2 (lat, lon)	2 (lat, lon)
Activation Function	ReLU (rectified Linear)	ReLU (rectified Linear)
Dropout	0.2	0.2
Optimiser	ADAM optimizer	ADAM optimizer
Loss Function	mean squared error loss	mean squared error loss
Learning Rate	0.0003	0.0003
Batch Size	128	1
Epoches	100	100

4.2. DA-GAN setup

RSS readings of all real readings are normalized from [-110,0] to [0,1] respectively while training the GAN model. Outputs from the GAN are rescaled to [-110,0] when augmenting the original dataset. Furthermore, the model was trained for 1000 epochs, and a learning rate of 0.00001 was used. Lastly, a BCEWITHLOGIT loss was used (This loss combines a sigmoid layer along with the BCELoss into a single class), alongside the ADAM optimizer. Empirically, The architecture refers to the Figure 2 in Section 3.3. For each unique location, 120 new samples were created by the generator (filtering was then performed on these 120 samples). Furthermore, a batch size of 4 was used.

4.3. Evaluating Datasets

- **Original**: Training dataset of the UJIndoorLoc Dataset (explained in Section 4.1).
- **DA-Uniform**: Original + Data Aggregation method using mean value and uniform distribution of each RSS sample explained in section 3.1.
- **DA-Uniform + DA-GAN**: Original + DA-Uniform + DA-GAN trained on (Original + DA-Uniform)
- **DA-Dirichlet**: Original + Data Aggregation method using Dirichlet distribution on the access point of each unique location, proposed in section 3.2.
- **GAN+ (DA-Dirichlet + DA-GAN)**: The proposed method in section 3.

4.4. Position Estimate Methods

4.4.1. Deep Neural Network (DNN)

a simple DNN model was trained to predict the user's location. It is used as a baseline method where the input data is a single sample of RSS data. The label of this DNN regression model uses the location's the longitude and latitude from the datasets (Figure 3-A).

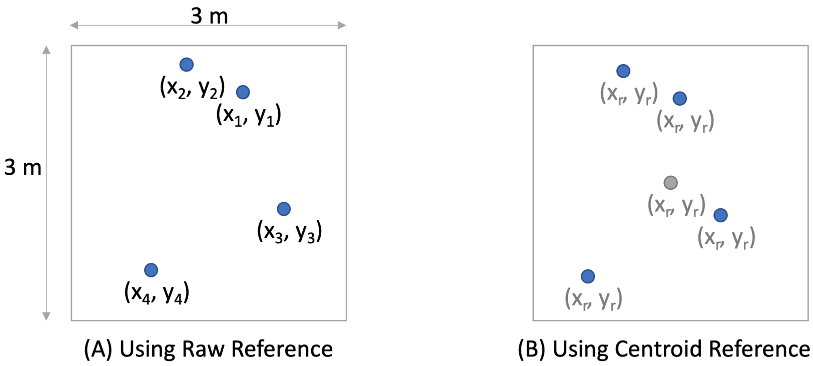


Figure 3: Reference Location between DNN and t-CNN methods

4.4.2. Time-Series Convolutional Neural Network (t-CNN)

A method proposed by [33] leveraging RSS time-series-based CNN model. This approach overcomes the problems caused by fading and shadowing by exploiting a time series of RSS readings. Using multiple consecutive RSS readings is expected to reduce the noise and randomness present in separate RSS readings and enhance the localisation performance. CNN's are thus used to leverage the temporal dependency between consecutive RSS time-series readings. In order to use the consecutive RSS time-series-based CNN model, multiple readings from the same location are needed to train the model. Since our dataset does not have sufficient number of RSS readings for the same position, a small manipulation was performed to obtain the necessary training and testing records. This involves dividing the area covered into $(3m \times 3m)$ cells and allocating each of the training records to one these cells based on their latitude and longitude (Figure 3-B). Each input data is a matrix comprising of 10 records with 520 features. The center of each cell was approximated to be the longitude and latitude of the corresponding cell sample.

5. Experiment Results and Analysis

5.1. Effectiveness of DA-GAN and its Combination

The objective of this experiment is to investigate the effectiveness of DA-GAN in comparison to other discussed augmentation schemes. Therefore, we performed an initial experiment by generating the same number of training records (80,000 training records) using each scheme. The same simple DNN architecture (Section 4.4.1) was used to evaluate the performance of each of these datasets. The error is measured in terms of the Euclidean distance.

Table 3 summarizes the results obtained. It is evident that the training records generated by the DA-GAN are of superior quality as opposed to those using the two data aggregations individually. Therefore, we are confident that DA-GAN is able to augment better data for the model.

Table 3
DNN Model's Accuracy using the Same Number of Records

Dataset	**Distance Error**
DA-Uniform	9.13
DA-Uniform + GAN	***8.90***
DA-Dirichlet	11.10
GAN+	***8.90***

5.2. Position Estimate Performance

This experiment aims to test the performance of the augmentation schemes using all records generated by each scheme respectively.

The various hyper-parameters across all models are kept constant. A learning rate of 0.0003 along with an ADAM optimizer was used. The model was trained for 100 epochs with a batch size of 128. Preprocessing to convert all RSS values between [0,1] was performed, where 1 indicates strong signal strength.

We then train the model on the entire training dataset generated by each of the augmentation techniques to assess model performance. Table 4 summarises the results obtained by using both the DNN and t-CNN model for predicting the longitude-latitude of a location. In addition, it illustrates he proportion of records generated using the various augmentation methods described. Regardless of the differences in the distance error between DNN and t-CNN, GAN+ has outperformed any other augmentation scheme. It not only illustrates the effectiveness of using GAN, but also highlights the importance of choosing data augmentation methods such as DA-Dirichlet prior to using a GAN. Additionally, DA-Dirichlet unlike DA-Uniform does not augment using one single sample, but rather, using access point (AP) readings from all samples belonging to the same location. This ensures better performance since it helps in dealing with the fluctuating nature of RSS.

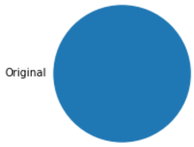
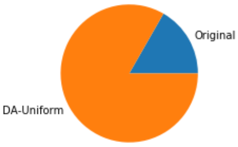
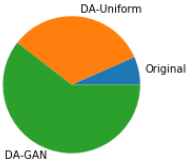
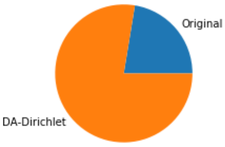
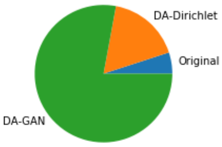
6. Discussion

Table 4 depicts the higher accuracy of t-CNN method compared to DNN. With GAN+, a centroid distance error as low as 0.3 is obtained. This error outperforms the result from the centroid distance error of the original t-CNN paper [33] of 2.77.

Furthermore, Table 5 provides the accuracy of the location estimate denoting the percentage of correct (3x3) grid cells identified in the case of the t-CNN model. It refers to the fact that the location of each cell was approximated to be the center of the cell while preparing the t-CNN dataset. The results shows that GAN+ is able to achieve a 99.6% accuracy in predicting the right cell to which the sample belongs.

Since a grid was used to approximate the location, the maximum distance error is 2.4 $(0.3 + \sqrt{18}/2)$, which corresponds to the scenario when the actual location of the sample is on one of the four vertices of the square grid. However on average, a distance error of 1.19 would exist (0.3 + 0.89) since the average distance of a point selected randomly in a square from its center is $\frac{L}{6}(\sqrt{2} + log(1 + \sqrt{2}))$ where L is the length of the side [34]. Based on these results,

Table 4
Accuracy - Distance Error

Datasets	Data Proportion	DNN	t-CNN
Original		10.620	3.700
DA-Uniform		8.910	2.003
DA-Uniform + DA-GAN		8.550	1.300
DA-Dirichlet		9.060	1.200
GAN+		8.490	0.300

we propose the data augmentation scheme illustrated in Figure 1 to be used on any indoor localisation dataset. This augmentation technique is expected to help the ML model to learn as well as generalize better, thereby giving a drastic improvement in performance.

Table 5
t-CNN's Accuracy

Datasets	Accuracy
Original	27%
DA-Uniform	66.7%
DA-Uniform + DA-GAN	84%
DA-Dirichlet	83.5%
GAN+	99.6%

7. Conclusion

In this paper, we propose a novel data augmentation scheme, called GAN+, using a Dirichlet distribution followed by a GAN. We first propose two schemes to augment the original dataset in order to ensure a sufficiently large dataset to subsequently train a GAN. The extended fingerprint database both reduces the human effort by accelerating new training base creation and improves the accuracy of the indoor localisation system. When working with machine learning, it is important to have a good quality dataset to train the algorithm. This means that the data should not only be sufficiently large, but should also be of high quality, and should be a good representation of all possible scenarios that might occur in reality. Results of our experiment highlight that GAN's are an effective way of generating such a data set for the purpose of indoor localisation. A testing error as low as 0.3 (2.4 in the worst case scenario) is obtained on the UJIIndoorLoc dataset. This is largely due to the fact that GAN+ is effective in creating new entries by altering the original dataset in such a way that it can simulate random effects (like shadowing and fading) commonly observed in indoor locations. This makes the training model more robust and also allows it to generalize better.

For future work, we plan to evaluate our proposed augmentation schemes on additional indoor localisation datasets. We also plan on investigating the use of DCGAN's (Deep Convolutional GAN's) as an augmentation method to generate additional RSS images directly from the original training images.

Acknowledgments

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU).

References

[1] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, J. Huerta, Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems, in: 2014 international conference on indoor positioning and indoor navigation (IPIN), IEEE, 2014, pp. 261–270.

[2] M. Youssef, A. Agrawala, The horus wlan location determination system, in: Proceedings of the 3rd international conference on Mobile systems, applications, and services, 2005, pp. 205–218.

[3] S.-H. Fang, T.-N. Lin, Indoor location system based on discriminant-adaptive neural network in ieee 802.11 environments, IEEE Transactions on Neural networks 19 (2008) 1973–1978.

[4] W. Njima, I. Ahriz, R. Zayani, M. Terre, R. Bouallegue, Smart probabilistic approach with rssi fingerprinting for indoor localization, in: 2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), IEEE, 2017, pp. 1–6.

[5] F. Zafari, A. Gkelias, K. K. Leung, A survey of indoor localization systems and technologies, IEEE Communications Surveys & Tutorials 21 (2019) 2568–2599.

[6] S. Xia, Y. Liu, G. Yuan, M. Zhu, Z. Wang, Indoor fingerprint positioning based on wi-fi: An overview, ISPRS International Journal of Geo-Information 6 (2017) 135.

[7] S. Dayekh, S. Affes, N. Kandil, C. Nerguizian, Cooperative localization in mines using fingerprinting and neural networks, in: 2010 IEEE Wireless Communication and Networking Conference, IEEE, 2010, pp. 1–6.

[8] H. Liu, H. Darabi, P. Banerjee, J. Liu, Survey of wireless indoor positioning techniques and systems, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37 (2007) 1067–1080.

[9] A. Olejniczak, O. Blaszkiewicz, K. K. Cwalina, P. Rajchowski, J. Sadowski, Deep learning approach for los and nlos identification in the indoor environment, in: 2020 Baltic URSI Symposium (URSI), IEEE, 2020, pp. 104–107.

[10] G. Félix, M. Siller, E. N. Alvarez, A fingerprinting indoor localization algorithm based deep learning, in: 2016 eighth international conference on ubiquitous and future networks (ICUFN), IEEE, 2016, pp. 1006–1011.

[11] K. S. Kim, S. Lee, K. Huang, A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on wi-fi fingerprinting, Big Data Analytics 3 (2018) 1–17.

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

[13] R. S. Sinha, S.-M. Lee, M. Rim, S.-H. Hwang, Data augmentation schemes for deep learning in an indoor positioning application, Electronics 8 (2019) 554.

[14] H. Rizk, M. Torki, M. Youssef, Cellindeep: Robust and accurate cellular-based indoor localization via deep learning, IEEE Sensors Journal 19 (2018) 2305–2312.

[15] M. A. Cheema, Indoor location-based services: challenges and opportunities, SIGSPATIAL Special 10 (2018) 10–17.

[16] Q. D. Vo, P. De, A survey of fingerprint-based outdoor localization, IEEE Communications Surveys & Tutorials 18 (2015) 491–506.

[17] A. Salman, S. El-Tawab, Z. Yorio, A. Hilal, Indoor localization using 802.11 wifi and iot edge nodes, in: 2018 IEEE Global Conference on Internet of Things (GCIoT), IEEE, 2018, pp. 1–5.

[18] M. Youssef, M. Mah, A. Agrawala, Challenges: device-free passive localization for wireless environments, in: Proceedings of the 13th annual ACM international conference on Mobile computing and networking, 2007, pp. 222–229.

[19] M. Terán, J. Aranda, H. Carrillo, D. Mendez, C. Parra, Iot-based system for indoor location using bluetooth low energy, in: 2017 IEEE Colombian Conference on Communications and Computing (COLCOM), IEEE, 2017, pp. 1–6.

[20] Q. Li, J. Zhu, T. Liu, J. Garibaldi, Q. Li, G. Qiu, Visual landmark sequence-based indoor localization, in: Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery, 2017, pp. 14–23.

[21] H. Rizk, A. Shokry, M. Youssef, Effectiveness of data augmentation in cellular-based localization using deep learning, in: 2019 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2019, pp. 1–6.

[22] A. E. Hilal, I. Arai, S. El-Tawab, Dataloc+: A data augmentation technique for machine learning in room-level indoor localization, arXiv preprint arXiv:2101.10833 (2021).

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets in: Advances in neural information processing systems (nips) (2014).

[24] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, A. Zhavoronkov, The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology, Oncotarget 8 (2017) 10883.

[25] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of stylegan, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8110–8119.

[26] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, J. Hu, Generative adversarial networks (gan) based efficient sampling of chemical composition space for inverse design of inorganic materials, npj Computational Materials 6 (2020) 1–7.

[27] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[28] Q. Li, H. Qu, Z. Liu, N. Zhou, W. Sun, S. Sigg, J. Li, Af-dcgan: Amplitude feature deep convolutional gan for fingerprint construction in indoor localization systems, IEEE Transactions on Emerging Topics in Computational Intelligence (2019).

[29] A. Guidara, G. Fersi, F. Derbel, M. B. Jemaa, Impacts of temperature and humidity variations on rssi in indoor wireless sensor networks, Procedia Computer Science 126 (2018) 1072–1081.

[30] W.-J. Chang, J.-H. Tarng, Effects of bandwidth on observable multipath clustering in outdoor/indoor environments for broadband and ultrawideband wireless systems, IEEE transactions on vehicular technology 56 (2007) 1913–1923.

[31] H. Hashemi, Impulse response modeling of indoor radio propagation channels, IEEE journal on selected areas in communications 11 (1993) 967–978.

[32] V.-A. Nguyen, J. Boyd-Graber, S. F. Altschul, Dirichlet mixtures, the dirichlet process, and the structure of protein space, Journal of Computational Biology 20 (2013) 1–18.

[33] M. Ibrahim, M. Torki, M. ElNainay, Cnn based indoor localization using rss time-series, in: 2018 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2018, pp. 01044–01049.

[34] nettrino (https://math.stackexchange.com/users/194498/nettrino), What

is average distance from center of square to some point?, Mathematics Stack Exchange, ???? URL: https://math.stackexchange.com/q/1033093. arXiv:https://math.stackexchange.com/q/1033093, uRL:https://math.stackexchange.com/q/1033093 (version: 2014-11-22).