RADIO MAP ESTIMATION WITH NEURAL NETWORKS AND ACTIVE LEARNING FOR INDOOR LOCALIZATION

Sıla Güler[1][0000−0001−7866−6433], F. Serhan Daniş[2,3][0000−0002−8813−9220], and Ali Taylan Cemgil[2][0000−0003−4463−8455]

[1] Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands
s.guler@tue.nl
[2] Boğaziçi University, 34342 Istanbul, Turkey
[3] Galatasaray University, 34349 Istanbul, Turkey

Abstract. Accurate estimation of the radio frequency maps is key in practical indoor localization, but this requires dense sampling of the electromagnetic field, which is also known as fingerprinting. To decrease the time-consuming fingerprinting process, we both aim to estimate probabilistic radio frequency maps (RM) using artificial neural networks (ANN), and automatically pick the most informative fingerprint positions following an active learning strategy based on uncertainty sampling, aided by Gaussian Processes (GP). Once we have an estimate of the RM of the environment, the problem can be formulated as a tracking problem with a Hidden Markov Model (HMM), and the RM can be used as the observation densities of the HMM. The trajectories of sequential positions are approximated with a sequential Monte Carlo filter. The results indicate that fingerprint measurements can be reduced by 30% in return of 8.1% decrease in the localization performance.

Keywords: Indoor localization, neural networks, fingerprint reduction, radio map estimation, uncertainty sampling

1 INTRODUCTION

Indoor localization has become a widespread research area recently due to the use cases of marketing, healthcare [3] and asset tracking. Indoor localization approaches are divided into three categories: geometry-based, propagation-based, and fingerprint-based. Geometry-based approaches [5,10] uses triangulation or trilateration principles to calculate the position of the object. However, they are not capable of modeling the multi-path and shadowing effects. The propagation-based approaches model the attenuation of the signal power, but as environments vary, these approaches require to model the power of the signals with different parameters for each environment. Recently, fingerprint-based localization approaches has become common in this research area.

Fingerprint-based approaches have two stages. The first stage is the fingerprint collection at specified positions in the environment. A fingerprint is a statistic of the signal strength at a coordinate. It can be in the units of Received Signal Strength Indication (RSSI) or Signal-to-Noise Ratio (SNR). The second stage is the position estimation based on the data driven fingerprints. The number of collected fingerprints and the position estimation methodology are two important factors that affect the localization accuracy. Accordingly, we analyze the studies using different position estimation and fingerprint reduction methodologies.

One of the earliest fingerprint-based approach is RADAR [2]. They estimated the positions using kNN using a previously created RM of WiFi of 2.94-meter and 1-meter resolutions. In order to reduce the fingerprint measurement cost, they modeled the signal propagation with the Wall Attenuation Model and calculated the fingerprints at the other positions in their test environment. By using the calculated fingerprints with the nearest neighbor technique, they achieved 4.3-meter localization accuracy.

In [1] and [16], they used RSSI values and Channel Impulse Response as inputs for the neural network training for estimating the positions. Both achieved accuracies under 0.6 meters. Similarly, in [17], they used Auto Encoders consisting of two deep neural networks functioning as encoder and decoder and achieved 1.82-meter mean accuracy. In [13], they computed propagated signal power

CEUR-WS.org/Vol-2498/short4.pdf

distributions for each position and stored them in a probabilistic RM. They used both maximum likelihood estimation (MLE) and neural networks in localization stage and achieved 0.27-meter and 1-meter positioning error respectively. In [14], they generated a magnetic field map with the Gaussian Processes. They used Sequential Monte Carlo Method for the localization and achieved a 4.87-meter median position accuracy.

In [7], they focus on fingerprint reduction by extracting the hidden structure and redundancy of fingerprint matrices by using Singular Value Decomposition (SVD). The missing values were imputed via the kNN algorithm. They recovered the original fingerprints from only 5% of the collected data with an error rate less than 14%. In [9], they suggested an artificial fingerprint generation method based on signal propagation model. They generated two different RMs with the measured and the artificially generated fingerprints. They used the kNN approach to evaluate the localization performance based on these maps. They reduced the positioning error by 63% with the artificially generated fingerprints.

In this work, we hypothetize that both neighboring fingerprints and their coordinates are related to each other. Modeling these relations with Gaussian Processes (GP) and Artificial Neural Networks (ANN), we propose a novel reduction strategy to be applied to any fingerprint-based localization problem. Our reduction strategy combines two methods: ANNs are used to estimate the RM of fingerprints in the positions where no fingerprint data is originally available, and GPs are used to model the uncertainty of fingerprints. For evaluation, we compare the localization error with and without these methods. We approach our problem as a tracking problem based on HMM and use SMC to approximate the original trajectory. We calculate the localization error by comparing the estimated trajectory with the ground truth.

2 METHODOLOGY

2.1 Radio Frequency Map Estimation

The relationship between an RSSI distribution at a position and its 2D position coordinates is a complex relationship, as it bases on the degenerative effects on signals, such as path loss, shadowing, or even the architecture of the environment. ANNs are good at modeling complex relationships and multi-class classification problems where the output layer represents a probability distribution.

Five different sets of input features, $\mathcal{S}_i$, are constructed with different combinations of the following features: the distances between a BLE transmitter (i.e. beacon) and a bluetooth receiver (i.e., dongle), fingerprints in the form of RSSI histograms on the neighboring positions and their distances to the beacon. In Table 1, PT stands for the distance of the position to the beacon, PF_i for the distance of the position to the i^{th} closest fingerprint, TF_i for the distance between the i^{th} closest fingerprint and the beacon and $\mathbf{H}_i$ for the histogram on the i^{th} closest fingerprint, with indices for the rank of the proximity.

Table 1. Input Features

Feature Sets	Features
$\mathcal{S}_1$	PT
$\mathcal{S}_2$	$PF_1, TF_1, \mathbf{H}_1$
$\mathcal{S}_3$	$PF_{1,2}TF_{1,2}, \mathbf{H}_{1,2}$
$\mathcal{S}_4$	$PF_{1,2,3}, TF_{1,2,3}, \mathbf{H}_{1,2,3}$
$\mathcal{S}_5$	$PT, PF_{1,2,3,4}, TF_{1,2,3,4}, \mathbf{H}_{1,2,3,4}$

We train a Single Layer Neural Network (SNN) using these five feature sets. For every set we select different number of units in the hidden layer ranging from 20 to 520. As an SNN may not be successful enough to approximate the complex function between our input features and output, we also train a Deep Neural Network (DNN) with two hidden layers. We feed the features to the input layers and we estimate the discrete probability distribution of RSSI values ranging from -100 to -20.

We compare two distributions using a cross-bin comparison metric that is also called the squared Earth Mover's Distance (EMD^2) [12]. It corresponds to the square sum of element-wise difference between the cumulative distribution functions (CDF) of two distributions. In an ordered multi-class classification problem it becomes equivalent to Mallow's distance [8]:

$$\mathrm{EMD}^2(p,q) \propto ||F_P(p) - F_Q(q)||_2^2 \tag{1}$$

where p and q are original and predicted distributions respectively, and $F_P(p)$ and $F_Q(q)$ are their cumulative distribution functions.

For intermediate and top layers of ANNs, we select the sigmoid function as the activation function to model complex relations of features. The weights and biases are initialized with the Glorot Initializer and as zero respectively. Weights and bias values are optimized via the RMSProp algorithm [6].

2.2 Active Learning with Gaussian Processes

We use GP to model the mapping functions between the positions and the fingerprints together with the uncertainty in these positions. GP is composed of random variables, f, any finite set of which have a joint Gaussian distribution. The positions and the RSSI values are the observations, and the functions are the latent variables (see Fig. 1). When the functions for the given observations are known, the function value, f_*, for a new position, x_*, can be calculated.

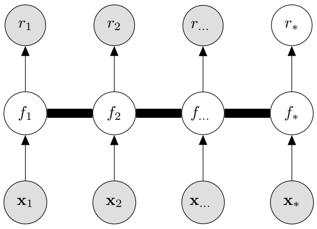


Fig. 1. Graphical Model of Gaussian Process Regression [11]

We assume that the distances between the positions makes a difference in RSSI distributions. We use a squared exponential function as the covariance function with the kernel $k(\mathbf{x}_i, \mathbf{x}_j)$. The hyper-parameters of the kernel, σ_f, σ_n, and l are optimized with gradient ascent by maximizing the log marginal likelihood as in [11]. Each function for each input value is treated as a random variable and they have a joint Gaussian distribution.The predictive distribution of RSSI values is computed as follows:

$$\begin{gathered} f_*|\mathbf{x}_*, \mathbf{x}, \mathbf{r}, \sim \mathcal{N}(\mu_* \ , \Sigma_*) \\ \mu_* = m(\mathbf{x}) + K(\mathbf{x}_*, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1} \ (\mathbf{r} - m(\mathbf{x})) \\ \Sigma_* = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1} K(\mathbf{x}, \mathbf{x}_*) \end{gathered} \tag{2}$$

where $\mathbf{r}$ and $\mathbf{x}$ are the RSSI observations and positions respectively. The function value can be predicted only by calculating the correlation between existing positions and a new coordinate. The output value $\mathbf{r}_*$ differs from f_* only in the sense that it has an additional noise term in its covariance, $\Sigma_* + \sigma_n^2 I$.

Predictive variance, Σ_*, is used for iterative active fingerprint selection. In the initial step, a random fingerprint is selected for training. At each step, the fingerprint that has the greatest variance is added to the training set. Retraining with the newly added fingerprint, predictive variance for each of the remaining positions is recalculated and this procedure continues until we reach a target number of fingerprints.

2.3 Tracking with Radio Frequency Map

We model the tracking problem as HMM, as in Fig. 2. Measurements, $\mathbf{r}_t$, are the RSSI values and the latent variables, $\mathbf{x}_t$, are the positions. We are to estimate the current position, $\mathbf{x}_t$, given the observations so far. Filtering, $p(\mathbf{x}_t, \mathbf{r}_{1:t})$, is a recursive process:

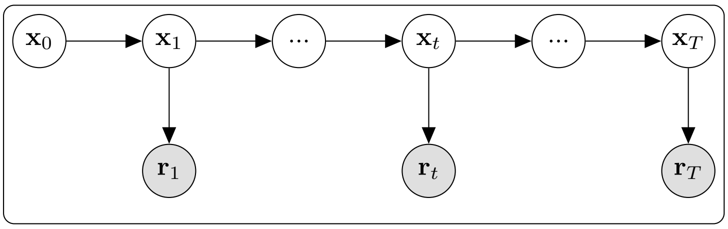


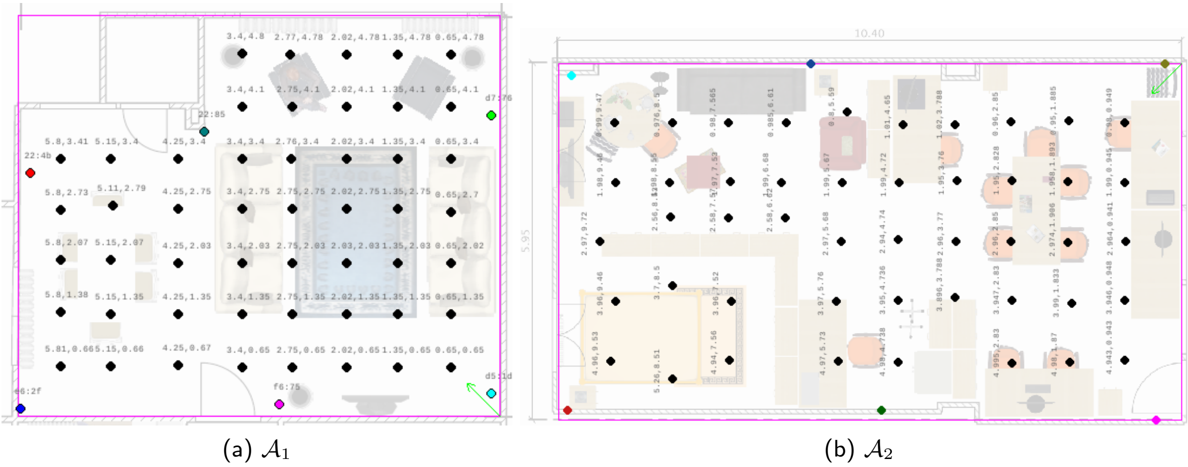
Fig. 2. Graphical Model for Tracking Problem

$$p(\mathbf{x}_t, \mathbf{r}_{1:t}) = p(\mathbf{r}_t|\mathbf{x}_t) \sum_{i=1}^{t} p(\mathbf{x}_{t-i+1}|\mathbf{x}_{t-i}) p(\mathbf{x}_{t-i}, \mathbf{r}_{1:t-i}) \tag{3}$$

For the transition density, $p(\mathbf{x}_{t-i+1}|\mathbf{x}_{t-i})$, a diffusion-based motion model is used as explained in [4]. For the observation densities, $p(\mathbf{r}_t|\mathbf{x}_t)$, we use the RM estimations, matrices that store the conditional probabilities, $p(\mathbf{r}|\mathbf{x})$. When the latent variable space is too big and/or continuous, the exact method will be intractable and/or impossible to apply. We employ an SMC method, the particle (bootstrap) filter, to approximate the conditional marginal distribution, $p(\mathbf{x}_t|\mathbf{r}_{1:t})$. We evaluate the final results by measuring the filtered trajectory of positions with the ground truth trajectories (see [4] for details).

3 EXPERIMENTS AND RESULTS

We conducted our experiments on two different datasets, collected from two different areas, the first of which, $\mathcal{A}_1$, is a living room (5.28×6.35 m^2) containing six BLE beacons, logged at 50 positions each for 24 hours [4] (see Fig. 3a). The second area, $\mathcal{A}_2$, is a home simulation environment (10.40×5.95 m^2) [15], with six beacons, logged at 50 positions each for 20 minutes. (see Fig. 3b).

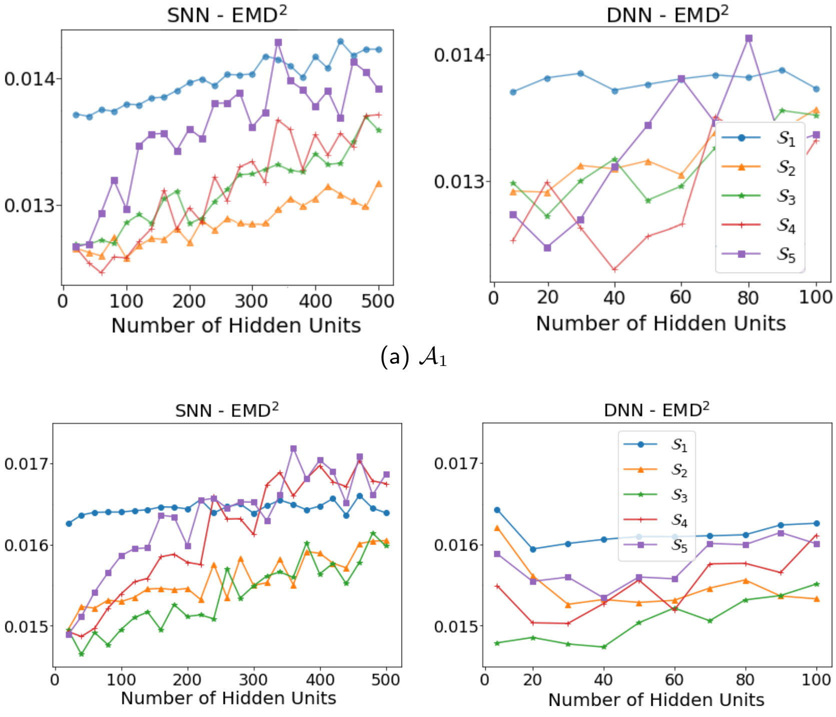

(a) $\mathcal{A}_1$

(b) $\mathcal{A}_2$

Fig. 3. Fingerprints are represented by black dots on two test areas.

We used 70% of these fingerprints for training, and the remaining for validation and test purposes. The training data were used for optimizing the weights and biases, and the validation data for selecting the best model. The test data were used only for evaluating the localization performance. All computations were performed on an Intel Core i7 based computer and 8GB of RAM. The models were constructed on Keras 2.1.5 with Tensorflow as backend.

For the test area of $\mathcal{A}_1$, we trained the SNN with five feature sets (see Table 1) and 25 different hidden unit sizes. We repeated the training process ten times, constructing ten different randomly selected fingerprint configurations which are then used to compare the results of these feature sets. We evaluate the performance by measuring the average validation loss over all beacons. Fig. 3 shows that $\mathcal{S}_1$ yields much greater error than the remaining feature sets, meaning the estimations that depend solely on the distance to the beacons perform very poorly. $\mathcal{S}_4$ has the best performance with minimum EMD^2 loss with 60 units in the hidden layer. It shows that the input feature with the four nearest fingerprint data, their distances to the beacons, and their histograms performs better than the other feature sets. We trained the DNN with hidden unit sizes ranging from 10 to 100. Finally, the configuration of $\mathcal{S}_4$ and 40 units is the best architecture for each layer, as seen in Fig. 3. Repeating the same experiments for the fingerprints collected in the $\mathcal{A}_2$, we found that $\mathcal{S}_3$ and 40 hidden units in the hidden layers performs better for both SNN and DNN (see Fig. 3).



(a) $\mathcal{A}_1$

(b) $\mathcal{A}_2$

Fig. 4. Errors with respect to hidden unit sizes for two environments

To evaluate our strategy, we trained ANNs using (i) all collected fingerprints, (ii) randomly selected 70% of the fingerprints and (iii) actively selected 70% of the fingerprints. With the trained networks, we made estimations of dense RMs with 0.1-meter intervals for ten times.

We used the constructed maps as the observation distribution in the particle filter to estimate the trajectory points. We applied the SMC with 2000 particles and at 400 trajectory points in $\mathcal{A}_1$ and $\mathcal{A}_2$. In each run, the first 32 estimated points were omitted as burn-in period. We run the filtering algorithm ten times for each radio frequency map and each time particles were initialized randomly. This prevented us from getting poor performance because of the random initialization. We reported the performance of our methodology as median tracking error instead of the mean absolute error, because errors form skewed distributions, which are clearly non-Gaussian. Tracking accuracies in $\mathcal{A}_1$ and $\mathcal{A}_2$ are given in Fig. 5a.

We have three observations. The first one is that deep neural networks cause higher median error in both areas. Second one is when the training size is decreased from 100% to 70% of data, median error increases. Finally, the median error for the actively selected data in each test area is lower than the one for the randomly selected data regardless of the neural network type.

As seen in Table 2, we can reduce the number of training data by 30% while increasing the tracking accuracy by at most 8.1% in return. For $\mathcal{A}_1$, the resulting increase are 1.9% and 2.3% for SNN and DNN, respectively. For $\mathcal{A}_2$, the resulting increase are 8.1% and 3.96%.

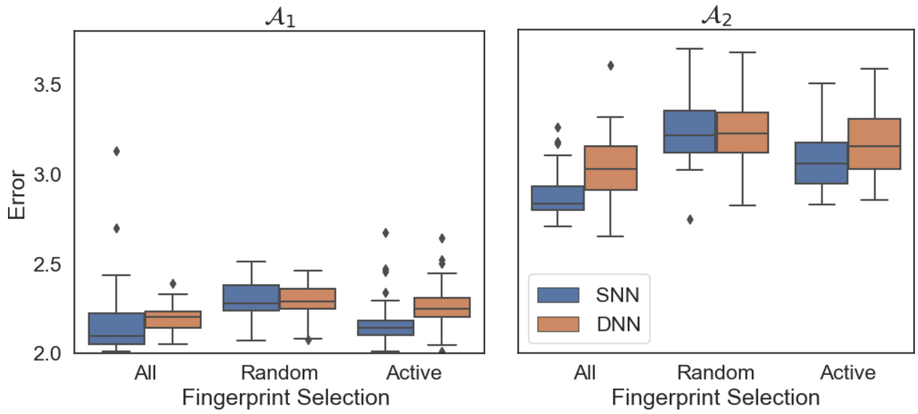


Fig. 5. Comparisons of neural network performances in both areas with different selection strategies. “All” refers to the case when all collected fingerprints are used.

Table 2. Accuracy results of NNs in both areas. 50% corresponds to the second quartile (median) error of positioning error distribution.

$\mathcal{A}_1$

NN type	Fingerprints	50%
SNN	All	2.10
	Random	2.28
	Active	2.14
DNN	All	2.20
	Random	2.29
	Active	2.25

$\mathcal{A}_2$

NN type	Fingerprints	50%
SNN	All	2.83
	Random	3.21
	Active	3.06
DNN	All	3.03
	Random	3.22
	Active	3.15

4 CONCLUSION

We propose a novel approach to reduce the training size required to estimate probabilistic radio maps by picking training positions adaptively. We have two conclusions showing the success of our approach. First, we pick the fingerprint positions in an adaptive manner, we estimate fingerprints at remaining positions better than we do with the randomly selected fingerprints in both test areas. Second, we reduced the training size by 30% with an increase in the median error by a maximum of 8.1%. Since fingerprinting is a time-consuming procedure in indoor localization field, this study would be used to reduce the fingerprinting process of any other study in the indoor localization research area.

References

1. Altini, M., Brunelli, D., Farella, E., Benini, L.: Bluetooth indoor localization with multiple neural networks. In: IEEE 5th International Symposium on Wireless Pervasive Computing 2010. pp. 295–300 (May 2010)
2. Bahl, P., Padmanabhan, V.N.: Radar: an in-building rf-based user location and tracking system. In: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064). vol. 2, pp. 775–784 vol.2 (2000)
3. Calderoni, L., Ferrara, M., Franco, A., Maio, D.: Indoor localization in a hospital environment using random forest classifiers. Expert Systems with Applications **42**(1), 125 – 134 (2015)
4. Daniş, F.S., Cemgil, A.T.: Model-based localization and tracking using bluetooth low-energy beacons. Sensors **17**(11), 2484 (2017)
5. Fang, B.T.: Simple solutions for hyperbolic and related position fixes. IEEE Transactions on Aerospace and Electronic Systems **26**(5), 748–753 (Sept 1990)
6. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
7. Gu, Z., Chen, Z., Zhang, Y., Zhu, Y., Lu, M., Chen, A.: Reducing fingerprint collection for indoor localization. Computer Communications **83**, 56 – 63 (2016)
8. Hou, L., Yu, C.P., Samaras, D.: Squared earth mover's distance-based loss for training deep neural networks. CoRR **abs/1611.05916** (2016)
9. Li, Y., Shi, G., Zhou, X., Qu, W., Li, K.: Reducing the site survey using fingerprint refinement for cost-efficient indoor location. Wireless Networks **25**(3), 1201–1213 (Apr 2019)
10. Peterson, B.B., Kmiecik, C., Hartnett, R., Thompson, P.M., Mendoza, J., Nguyen, H.: Spread spectrum indoor geolocation. Navigation **45**(2), 97–102
11. Rasmussen, C.E.: Gaussian processes for machine learning. MIT Press (2006)
12. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision **40**(2), 99–121 (Nov 2000)
13. Saleem, F., Wyne, S.: Wlan–based indoor localization using neural networks. Journal of Electrical Engineering **67**(4), 299 – 306 (2016)
14. Solin, A., Särkkä, S., Kannala, J., Rahtu, E.: Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning. In: 2016 European Navigation Conference (ENC). pp. 1–9 (May 2016)
15. Tunca, C., Alemdar, H., Ertan, H., Incel, O.D., Ersoy, C.: Multimodal Wireless Sensor Network-Based Ambient Assisted Living in Real Homes with Multiple Residents. Sensors **14**(6), 9692–9719 (2014)
16. Yu, L., Laaraiedh, M., Avrillon, S., Uguen, B.: Fingerprinting localization based on neural networks and ultra-wideband signals. In: 2011 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). pp. 184–189 (Dec 2011)
17. Zhou, C., Gu, Y.: Joint positioning and radio map generation based on stochastic variational bayesian inference for fwips. In: 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN). pp. 1–10 (Sept 2017)