IET Communications

Research Article

Multi-layer neural network for received signal strength-based indoor localisation

ISSN 1751-8628
Received on 17th May 2015
Revised on 19th November 2015
Accepted on 3rd January 2016
doi: 10.1049/iet-com.2015.0469
www.ietdl.org

Huan Dai ✉, *Wen-hao Ying*, *Jiang Xu*

Suzhou Key Laboratory of Internet of Thing, Changshu Institute of Technology, No. 99, 3rd South Ring Road, Suzhou, People's Republic of China

✉ *E-mail: daihuanjob@163.com*

Abstract: In received signal strength (RSS)-based indoor wireless localisation system, radio pathloss model or radio map must be readily obtainable. However, the unpredictability of wireless channel makes it difficult to achieve high accuracy localisation in practice. In this study, the authors employed a multi-layer neural network (MLNN) for RSS-based indoor localisation without using the radio pathloss model or comparing the radio map. The proposed MLNN localisation system integrate the RSS signals transforming section, the raw data denoising section and the node locating section into a deep architecture. Furthermore, a boosting method is designed to promote location accuracy of the MLNN effectively. Experiment results demonstrate the feasibility and suitability of the proposed algorithm.

1 Introduction

The popularity of wireless communication equipment has greatly promoted indoor location-based services [1] such as the indoor object search, navigation and tracking. However, due to the complex indoor environment and high equipment costs, it is usually not easy to achieve satisfactory location accuracy in practice [2]. Thus, obtaining real-time and accurate location information based on commonly available mobile devices without extra hardware installation or modification becomes a valuable and challenging topic.

As a convenient and economical localisation solution, received signal strength (RSS)-based algorithms have been intensively studied in the past decades. Comparing with other ranging-based algorithms, (e.g. time difference of arrival [3], time of arrival [4], angle of arrival [5] or ultra-wideband [6]), the RSS-based algorithms can be implemented without any additional hardware because RSS can always be easily obtained by many wireless devices such as mobile phone, PC, Pad, radio-frequency identification [1] and wireless sensor network [7, 8].

RSS-based localisation solutions are usually classified into propagation-model-based methods [8–10] and conventional fingerprinting methods [11, 12]. The former ones first convert the RSS into distance between the transmitter and the receiver with radio pathloss model, and then compute the location information. For example, Dai *et al.* [9] adopted the power measurements and the geometric constraints associated with planarity to estimate the path loss exponent. Angelo and Fabio [10] proposed a Bayesian formulation of RSS ranging problem alternative to the common approach of inverting the path loss formula. However, the RSS signals are easily interfered by various environmental factors (such as scattering, shadowing and the heights of transmitter and receiver) and the radio propagation characteristics (for example the transmission power and the pathloss exponent change with different environment such as air humidity and equipment energy). As a result, the distance estimation is error prone using the propagation model.

The latter methods prebuild a radio map in the first step and then estimated the location of target by comparing online RSS map with offline radio map. Zheng *et al.* [12] employed multiple frequencies and powers between the device and each landmark to obtain a larger RSS fingerprint. The compressive sensing theory is used to reconstruct the RSS radio map from measurements with a small number of fingerprints [13]. In other fingerprinting solutions [14], the probability of each potential location is analysed based on Bayesian theory and kernel functions, assuming that the received RSS signals are independent at every instant time. However, the independence may not be held in real environments and thus obtaining an explicit formulation of RSS distribution is challenging. Meanwhile, these statistical methods often involve high computational complexity, making it hard to implement in practice due to the resource limitations of wireless communication equipments.

In the two kinds of methods above, the second steps are the process of calculating the locations of the unknown nodes, which are usually adopted by expert systems with machine learning techniques. For example, Patwari *et al.* [8] and Fang *et al.* [14] use maximum likelihood estimation (MLE) and Kamol and Prashant [11] Zheng *et al.* [12] and Feng *et al.* [13] employ *k*-nearest neighbour and the Euclidean distance to estimate the locations of the unknown nodes, besides, decision trees [15], Bayesian inference [16] and support vector regression [17] are also implemented for the second steps of indoor localisation. The study on the utilisation of neural network for localisation purposes has been made by several research teams [18–20]. Neural network with one hidden layer has been applied for localisation problem [18], however the input of neural network requires other data handling methods and the location performance is not very satisfactory. To obtain best matches in terms of architectural structures, Campos *et al.* [19] applies two combined techniques that improve localisation accuracy: unsupervised clustering and majority voting committees of back-propagation artificial neural networks. In [20], the authors utilised the generalised regression neural network (GRNN) in RSS-based indoor localisation algorithm for wireless sensor network. Recently, owing to the success of the approaches for training multiple hidden layers [21, 22], the expected benefits of multi-layer neural network are shown in various domains such as visual classification [23] and audio recognition [24]. In this paper, we try to address the RSS-based indoor localisation issue by using multi-layer neural network (MLNN) [25], which integrates the RSS signal transforming, the raw data denoising and the unknown node locating into a deep architecture. Moreover, a boosting method is designed for MLNN-based on the actual localisation situations. Like the conventional fingerprinting techniques, the proposed algorithm also needs a database (offline RSS map) of RSS signals and relative positions, however, the database is used to train MLNN for obtaining network parameters, not to compare with online RSS map.

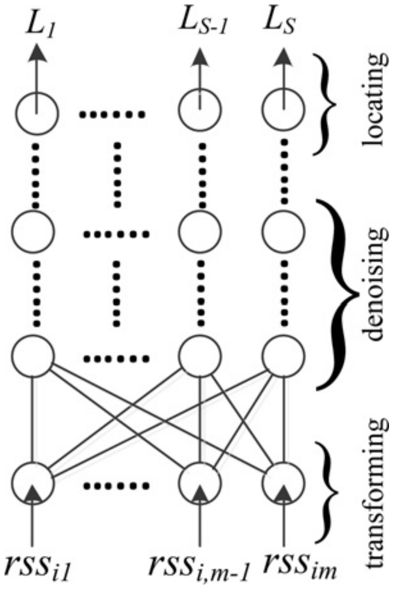


Fig. 1 *Proposed MLNN for RSS-based localisation*

2 MLNN for RSS-based location

A two-dimensional indoor space covered by wireless network is assumed to be rectangular for explanation convenience. Similar to the fingerprinting method [11], the indoor space is divided into S square grid ($l \times l$), which is represented by (x_i, y_i, L_i), where i = 1…S, L_i is the label of the square grid, and x_i, y_i are the centroid coordinate of the L_i square grid.

m anchor nodes $\{X_1,\ldots, X_m\}$ with known locations are deployed in the aforementioned space, and each anchor node constantly launches wireless signals. The unknown nodes can receive $\mathbf{rss_i} = (\mathrm{rss}_{i1},\ldots, \mathrm{rss}_{im})$, measured from aforementioned m anchor nodes. The unknown nodes do not know their own location. The localisation task is to estimate the square grid (x_i, y_i, L_i) which the unknown node belongs to, based on the received ($\mathrm{rss}_{i1},\ldots, \mathrm{rss}_{im}$).

2.1 Proposed network

MLNN is a feed-forward, artificial neural network that has more than one hidden layer between its inputs and outputs. For RSS-based indoor localisation, we design an MLNN composed of three main sections: transforming section, denoising section and locating section, as shown in Fig. 1.

In the transforming section, for one unknown node, taking its received $\mathbf{rss}_i$ as input, and map it to a hidden representation $\boldsymbol{h}^\mathbf{1}$ through a deterministic mapping function $\boldsymbol{h}^1 = f_{\theta 1}(\mathbf{rss}_i) = s(\boldsymbol{W}_1 * \mathbf{rss}_i + \boldsymbol{b}_1)$, parameterised by $\theta_1 = \{\boldsymbol{W}_\mathbf{1}, \boldsymbol{b}_1\}$. W_1 is an $m \times m$ weight matrix, $\boldsymbol{b}_1$ is a bias vector and s is the logistic function $1/[1 + \exp(-x)]$. Under such a mapping, the raw $\mathbf{rss}_i$ can be converted into a new transformed representation.

In the denoising section, there are $N(N \geq 1)$ layers for mapping the $\boldsymbol{h}^1$ by $\boldsymbol{h}^{k+1} = f_{\theta k}(\boldsymbol{h}^k) = s(\boldsymbol{W}_{k+1} * \boldsymbol{h}^k + b_{k+1})$, $k \in (1, \ldots, N)$. The representation $\boldsymbol{h}^\mathrm{k}$ of the kth layer used as input for the $(k + 1)$th. This section denoises the $\mathbf{rss}_i$ error by the multi-layer architecture.

In the locating section, the locating result $\boldsymbol{L}$, corresponding to square grid labels (L_1, $L_{2,}$,…, L_S), can be calculated by

$$\boldsymbol{L} = f_{\theta(N+2)}\left(\boldsymbol{h}^{N+1}\right) = s\left(\boldsymbol{W}_{N+2} * \boldsymbol{h}^{N+1} + b_{N+2}\right) \tag{1}$$

Let Sgn($\boldsymbol{L}$) be a 0–1 multi-class vector with S elements, of which only one element equals to 1, and others equal to 0. If the ith element equals to 1, the unknown node belongs to the ith square grid. Obviously, the problem of localisation has been transformed into that of classification.

2.2 MLNN training

After setting up the MLNN, the next task is to train the MLNN for parameters $\theta_k = \{\boldsymbol{W}_k, b_k\}$. Two-stage training procedure (pre-training each layer with unsupervised learning algorithm and then fine-tuning the whole network) is applied for training the multi-layer architectures [25]. Before MLNN training, the training database must be generated, as shown in Fig. 3, each training data contains the square grid (x_i, y_i, L_i) which the unknown node belongs to and (rss_{i1}, …, rss_{im}) received from above mentioned m anchor nodes by the unknown node.

As shown in Fig. 2, we pre-train the transforming section and denoising section with auto-encoder block, block by block, starting with the first layer, and then use the output $\boldsymbol{h}^k$ (a new representation for the $\mathbf{rss}_i$ between the unknown node and m anchor nodes) as input for the next layer. The operation continues until the last layer of denoising section. The parameters $\theta_k = \{\boldsymbol{W}_k, b_k\}$ of each block are optimised to minimise the average reconstruction error

$$\theta_k = \underset{\theta_k}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} L\left(\left(\boldsymbol{h}^k\right)^{(i)}, g\left(f\left(\left(\boldsymbol{h}^k\right)^{(i)}\right)\right)\right) \tag{2}$$

where $f(\boldsymbol{x})$ is a deterministic mapping function $s(\boldsymbol{Wx} + b)$, $g(\boldsymbol{y})$ is a ‘reconstruction’ function $s(\boldsymbol{W}'\boldsymbol{y} + b')$, $\boldsymbol{W}' = \boldsymbol{W}^\mathrm{T}$, L is a loss function

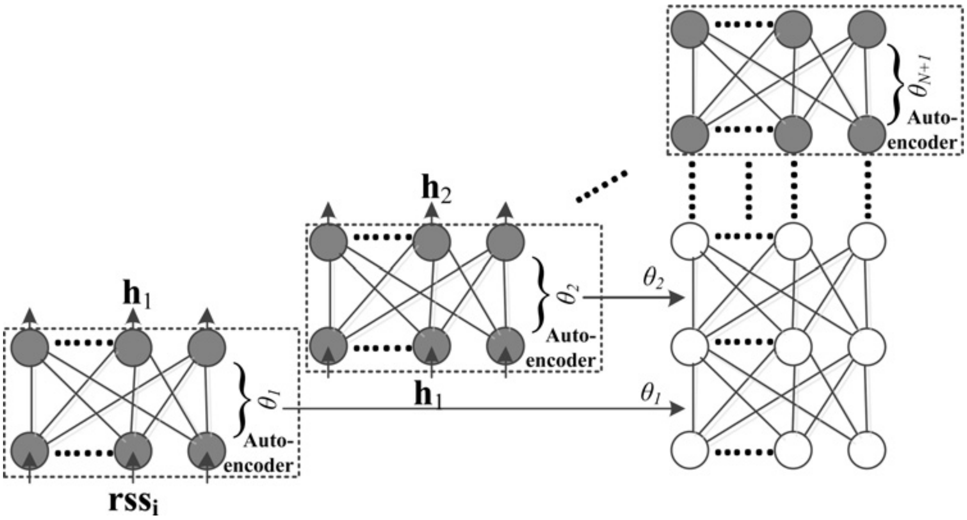


Fig. 2 *Pre-training consists of learning a stack of blocks, each having only one layer of* $\mathbf{rss}_i$ *representation*

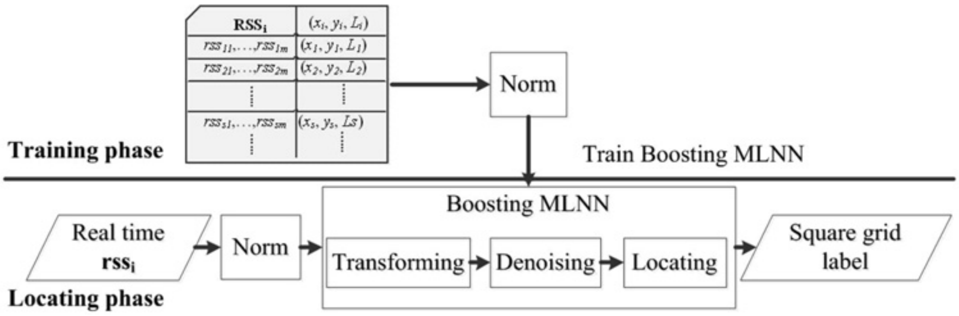


Fig. 3 *Structure of boosting MLNN localisation system*

$L(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2$ and n is the number of inputs. The aforementioned optimisation can typically be carried out by stochastic gradient descent.

To effectively obtain new representation $\boldsymbol{h}^k$ containing less errors, for each block, we corrupt the input $\boldsymbol{h}^k$ to get an incomplete version $\tilde{\boldsymbol{h}}^k$ by means of a stochastic mapping $\tilde{\boldsymbol{h}}^k \sim q_D(\tilde{\boldsymbol{h}}^k|\boldsymbol{h}^k)$. In the experiments, a proportion v elements of $\boldsymbol{h}^k$ are chosen at random, and their values are forced to be 0, while the others are retained. As a result, defining the joint distribution

$$q^0(\boldsymbol{h}^k, \tilde{\boldsymbol{h}}^k, \boldsymbol{h}^{k+1}) = q^0(\boldsymbol{h}^k) q_D(\tilde{\boldsymbol{h}}^k|\boldsymbol{h}^k)\delta_{f_{\theta_k}(\tilde{\boldsymbol{h}}^k)}(\boldsymbol{h}^{k+1}) \tag{3}$$

where $q^0(\boldsymbol{h}^k)$ represents the empirical distribution associated to all training inputs, $\delta_u(v)$ puts mass 0 when $u \neq v$. Thus $\boldsymbol{h}^{k+1}$ is a deterministic function of $\tilde{\boldsymbol{h}}^k$, and $q^0(\boldsymbol{h}^k, \tilde{\boldsymbol{h}}^k, \boldsymbol{h}^{k+1})$ is parameterised by θ_k. The aforementioned objective function becomes

$$\theta_k = \arg\min_{\theta_k} \frac{1}{n}\sum_{i=1}^{n} L\left(\left(\boldsymbol{h}^k\right)^{(i)}, g\left(f\left(\left(\tilde{\boldsymbol{h}}^k\right)^{(i)}\right)\right)\right) \tag{4}$$

After pre-training, fine-tune all the parameters of MLNN with respect to a supervised training criterion against the labels of the unknown node. In fine-tuning phase, the back propagation through the MLNN slightly adjusts the weights found in pre-training phase. The pre-training finds a region of the weight space that allows the fine-tuning to make rapid progress, and meanwhile reduces overfitting. The detailed training steps are supported in [25]. The training database D is illustrated in Fig. 3.

2.3 Network boosting

During the research, we found that MLNN can reach normal location accuracy for most square grids, but location accuracy is relatively low in some square grids, especially, when square grid length l becomes smaller. This phenomenon will greatly affect the whole localisation result. We design a boosting method to solve this problem. The detailed algorithm is shown in Fig. 4.

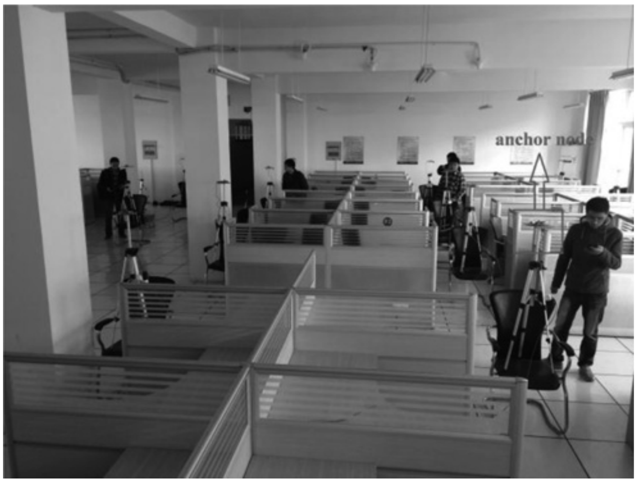


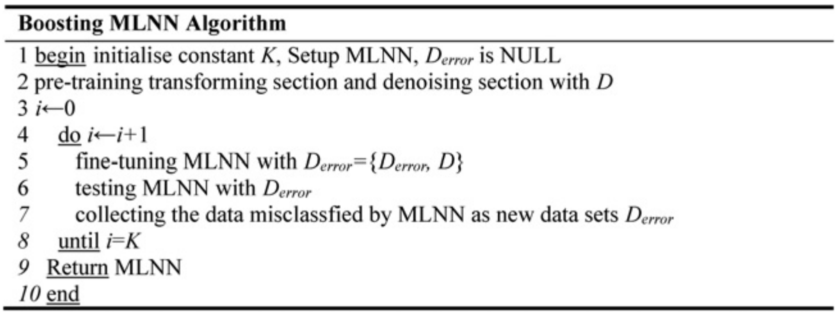
Fig. 5 *Deployment of anchors and experimental environment*

2.4 Algorithm implementation

There are two phases when using boosting MLNN for RSS-based localisation. In the training phase, the training database must be generated, and the network is trained by the normalised RSS data and square grid labels. In the locating phase, the label of square grid, in which the unknown node is located, can be calculated by the real time $\mathbf{rss}_i$ which are received by the unknown node. The detailed structure is illustrated in Fig. 3. To perform the significant numeric calculation, $\mathbf{rss}_i$ are normalised by min–max normalisation.

3 Experiment

Experiments are conducted in our rectangular laboratory of 144 m^2, 3.8 m height, as shown in Fig. 5. The lab contains more than ten computers, necessary equipment and furniture, and six rectangular concrete columns of about 0.53 m on each side. In experiments, we chose wireless fidelity (WiFi) technology for APs with two



```
Boosting MLNN Algorithm
begin initialise constant $K$, Setup MLNN, $D_{error}$ is NULL
pre-training transforming section and denoising section with $D$
$i \leftarrow 0$
  do $i \leftarrow i+1$
    fine-tuning MLNN with $D_{error}=\{D_{error}, D\}$
    testing MLNN with $D_{error}$
    collecting the data misclassfied by MLNN as new data sets $D_{error}$
  until $i=K$
 Return MLNN
end
```

Fig. 4 *Steps of boosting MLNN algorithm*

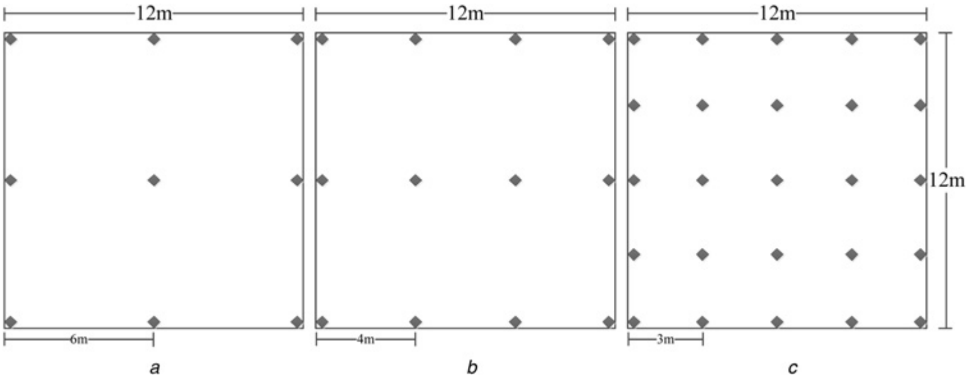


Fig. 6 *Deployment of AP nodes*

a 9 AP nodes
b 16 AP nodes
c 25 AP nodes

reasons: (i) the IEEE 802.11 network protocol supported devices such as mobile phone or Pad are growing fast, making it possible to avoid using extra receiving devices, (ii) with more and more WiFi APs installed indoor, its application for future indoor location may become more convenient. WiFi APs made by integrated chip ESP8266, are used as anchor nodes, and transmit the IEEE 802.11 protocol in the 2.4 GHz frequency band.

The properties of the proposed algorithm are individually impacted by different parameters, for example, the number of AP nodes, the number of K, the length l of square grid and the number of training sample. In order to assess the effects of parameters on the proposed algorithm performance, several scenarios are presented. As shown in Fig. 6, in the lab, AP nodes are deployed in three different ways. The grey diamonds are AP nodes. The distance between AP nodes and ground is 160 cm, as illustrated in Fig. 5.

According to the requirements of different location accuracy, the 144 m^2 indoor space is divided into three square grids, (2m × 2 m), (1.5m × 1.5 m) and (1m × 1 m), illustrated in Fig. 7. We use mobile phone to collect the database (300 training samples and 100 testing samples for each square grid) in different deployments of AP nodes and different divisions of square grid. When collecting signals, there are students walking in the lab randomly. The mobile phone receives a group of $\mathbf{rss}_i$ once in a second.

To evaluate the influence of different parameters in the proposed algorithm from the view of classification, we utilised the successful location rate (SLR) in different square grid size ($l \times l$), which is defined as

$$\mathrm{SLR}_{l\times l} = \frac{N_\mathrm{s}}{N_\mathrm{all}}, \tag{5}$$

where N_s denotes the number of successful localisation samples. N_all is the number of all localisation samples. As shown in Fig. 8*a*, when the number of hidden layer is larger than 3, the SLRs remain stable. When the number of nodes per hidden layer is equal to or greater than the input m, the changes of SLRs are small, as illustrated in Fig. 8*b*. The SLR against proportion v elements of $\boldsymbol{h}^k$ is shown in Fig. 8*c*. In these simulations, the boosting MLNN is trained with 300 training samples and tested with 100 test samples. According to Fig. 8, in next experiments, we take denoising section with four fixed layers, the number of nodes per hidden layer equals to m, namely the number of input, the proportion v equals to 10% in the denoising section.

As illustrated in Fig. 9, the change of SLR becomes smooth when training number exceeds 100 and number of APs equals to 16 or 25, which means that boosting MLNN needs enough training samples and the number of APs directly affects the location accuracy. In these simulations, the length l of square grid equals to 2, and the parameter K of boosting MLNN equals to 3.

To evaluate the location performance of the boosting MLNN, the SLR against the parameter K of boosting MLNN is illustrated in Fig. 10. When $K = 1$, in other words, only using the MLNN, the SLR is low, especially when the training number is not large enough. The surface becomes smooth with increasing K, which indicates that the boosting method can help MLNN to improve the localisation performance. In these simulations, the length l of square grid equals to 2, and the number of AP nodes equals to 25.

Multi-layer neural network provides good results in visual classification [23] and audio recognition [24], and from the analysis results above, it can also perform well in localisation problem. In this paper, the proposed MLNN localisation system first transforms and denoises the raw RSS signals and then classifies the labels of the unknown nodes. The results indicate that multi-layer structure and auto-encoder pre-training method are the reasons for well performance in localisation problem.

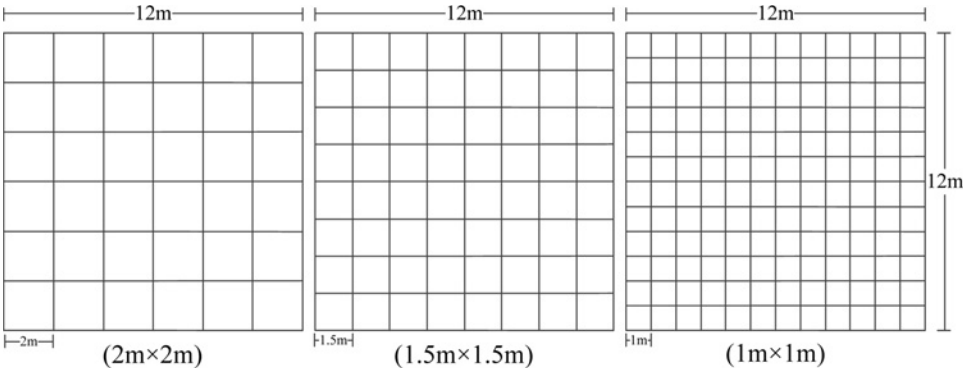


Fig. 7 *Divisions of square grid*

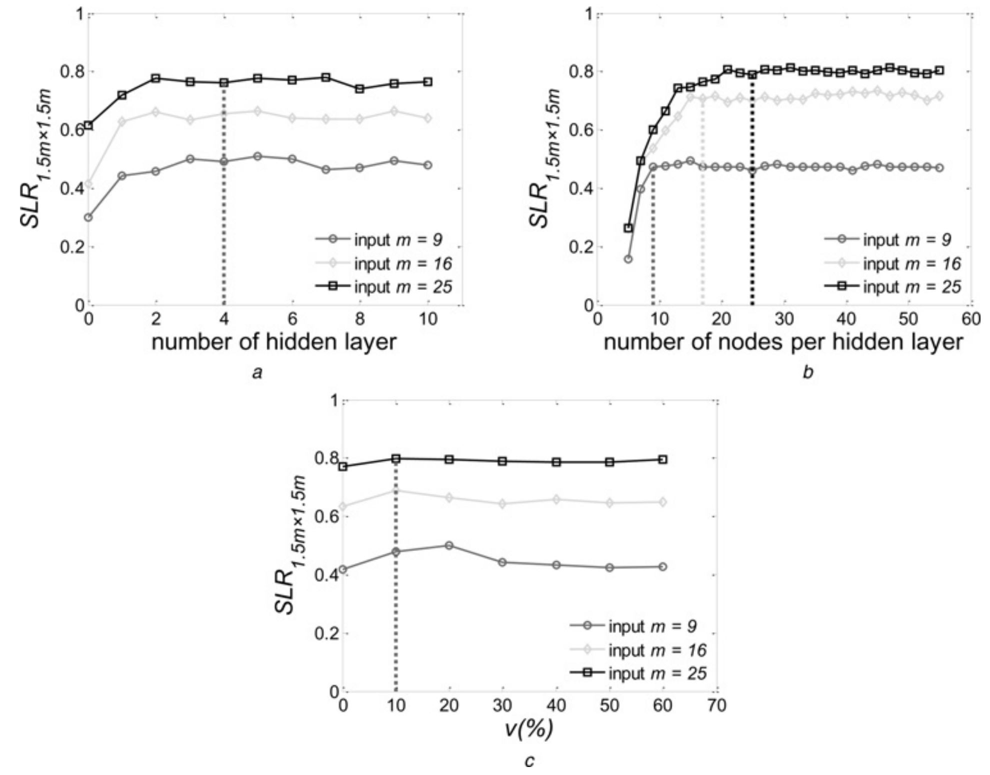


Fig. 8 *Successful location rate against different parameters*

a SLR against number of hidden layer
b SLR against number of nodes per hidden layer
c SLR against proportion of random elements of $\boldsymbol{h}^{12}$

Moreover, the boosting method can improve effectively the SLR of the system.

Comparing with other indoor location algorithms (MLE [8], fingerprint method [11] and GRNN [20]), we use the average location error (Avg) and standard deviation (SD) of errors to evaluate the performance of the algorithms, which are defined as

$$\mathrm{Avg}(m) = \frac{\sum_{i=1}^{N_{\mathrm{all}}} \left\| \tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i \right\|_2}{N_{\mathrm{all}}}, \tag{6}$$

$$\mathrm{SD}(m) = \sqrt{\frac{1}{N_{\mathrm{all}} - 1} \sum_{i=1}^{N_{\mathrm{all}}} \left(\left\| \tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i \right\|_2 - \mathrm{Avg} \right)^2}, \tag{7}$$

where $\|\bullet\|_2$ denotes the Euclidean distance, $\tilde{\boldsymbol{x}}_i$ is the estimated position of the ith unknown node, $\boldsymbol{x}_i$ is the actual position of the ith unknown node and N_{all} is the number of all localisation samples. In GRNN, fingerprinting and boosting MLNN simulations, any estimate of unknown node's location is limited to the central point of the square grid. Smaller Avg, or similar Avg with smaller SD means better performance.

The average location errors and SDs under different number of static APs for MLE, GRNN, fingerprinting and boosting MLNN methods are shown in Table 1. In the area, there are altogether 64 square grids of size 1.5 m × 1.5 m, in each of which 300 training samples and 100 testing samples are collected for simulations. In the boosting MLNN simulations, parameter $K = 3$. In the MLE simulations, 400 locations are computed with 400 samples, and the

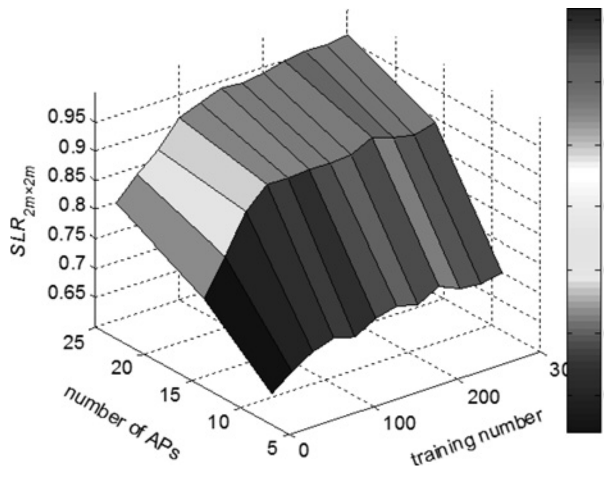


Fig. 9 *Successful location rate against the training number and the number of APs*

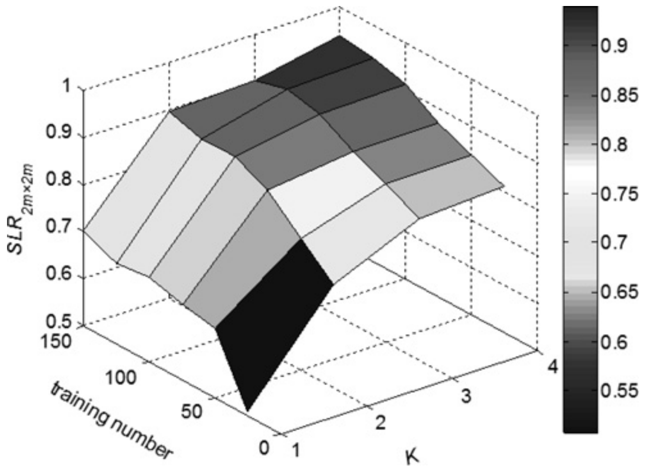


Fig. 10 *Successful location rate against the training number and the parameter K*

Table 1 Average location errors and SDs under different number of static APs

Algorithm	Avg, *m*, SD, *m*		
	9 static APs	16 static Aps	25 static APs
MLE	(2.57, 1.81)	(1.73, 1.56)	(1.09, 1.14)
GRNN	(1.91, 1.61)	(0.99, 1.15)	(0.63, 1.09)
fingerprint	(1.96, 1.30)	(1.02, 1.27)	(0.61, 1.05)
boosting MLNN	(1.13, 1.19)	(0.67, 0.91)	(0.46, 0.71)

average result is the estimated target location. The measured RSS match the log-normal shadowing model with rssi_0(dBm) = −45, $n = 2.3$, using $d_0 = 1$ m. In the GRNN simulations, the GRNN model is trained by the 64 × 300 training samples and tested by the 64 × 100 testing samples, parameter σ = 0.3. In the fingerprinting simulations, 64 fingerprint entries are recorded as the fingerprint database and each entry is the mean of 300 training samples on one square grid. About 100 testing samples are compared with the fingerprint database and then the average result is the target location. Results indicate that the performance of the boosting MLNN is better than other algorithms under different number of static APs.

The path estimation can now be carried out based on the above mentioned trained location models. The dotted line in each subfigure of Fig. 11 shows the path of a mobile phone taken moving around in the lab, covering 20 square grids. In each square grid, ten $\mathbf{rss}_i$ samples are collected. In the MLE simulations, for each square grid, ten locations are computed with the ten $\mathbf{rss}_i$ samples, and the average result is the target estimated location. In the GRNN and boosting MLNN simulations, the average result of the ten $\mathbf{rss}_i$ samples is used for the input of GRNN and boosting MLNN methods. In the fingerprinting simulations, the average result of the ten $\mathbf{rss}_i$ samples is compared with above mentioned fingerprint database. The results of path estimation with different algorithms are shown in Fig. 11. The average location errors and SDs are (1.13, 0.71), (0.67, 0.98), (0.59, 0.96) and (0.33, 0.68).

Different scenario sizes should also be considered for localisation algorithm application. In these simulations, three scenario sizes (10 m × 10 m), (20 m × 20 m) and (30 m × 30 m) are tested for location performance comparison. In each scenario, 25 static APs are deployed accordingly as in Fig. 6*c*, and the size of the square grid is 2.5m × 2.5 m. A database of collecting 300 training and 100 testing RSS for each grid is generated with actual distance *d*

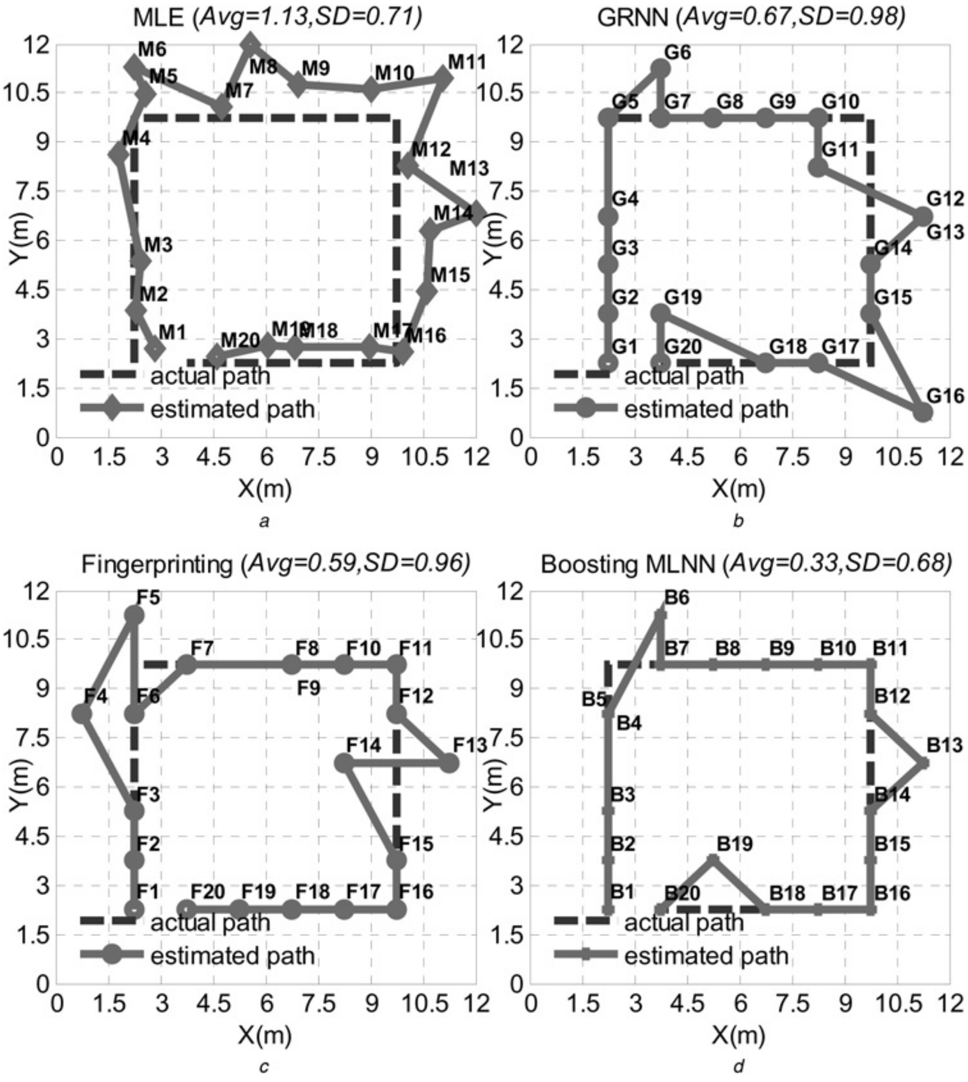


Fig. 11 *Illustration of path estimation*

a MLE
b GRNN
c Fingerprinting
d Boosting MLNN

Table 2 Average location errors and SDs under different scenario sizes

Algorithm	Avg, *m*, SD, *m*		
	10 m × 10 m	20 m × 20 m	30 m × 30 m
MLE	(0.86, 1.64)	(2.93, 2.16)	(5.82, 3.75)
GRNN	(0.31, 0.97)	(1.73, 2.77)	(3.64, 4.45)
fingerprint	(0.36, 0.86)	(1.82, 2.69)	(3.35, 3.37)
boosting MLNN	(0.23, 0.76)	(1.05, 2.11)	(2.63, 3.24)

between static AP and unknown node and log-normal model

$$\mathrm{rss}i(\mathrm{dBm}) \sim N(\overline{\mathrm{rss}i}(\mathrm{dBm}), \sigma_{\mathrm{dB}}^2) \tag{8}$$

$$\overline{\mathrm{rss}i}(\mathrm{dBm}) = \mathrm{rss}i_0(\mathrm{dBm}) - 10n_p\log_{10}(\mathrm{d}/\mathrm{d}_0) \tag{9}$$

where the random variable $\mathrm{rss}i(\mathrm{dBm}) = 10\log_{10}\mathrm{rss}i$, $\overline{\mathrm{rss}i}(\mathrm{dBm})$ is the mean power in decibel milliwatt (dBm), σ_{dB}^2 is the variance of the shadowing, and $\mathrm{rss}i_0(\mathrm{dBm})$ is the received power in dBm at a reference distance d_0. Typically, $d_0 = 1$ m, and $\mathrm{rss}i_0(\mathrm{dBm})$ is calculated by using (9). The path loss exponent n_p is a function of the environment. In our simulations, $\mathrm{rss}i_0(\mathrm{dBm}) = -45$, $\sigma_{\mathrm{dB}} = 3.92$, $n_\mathrm{p} = 2.3$ and $d_0 = 1$ m. Table 2 shows the average location error and the SD against the scenario size for the aforementioned localisation methods.

4 Conclusion

By utilising the multi-layer architecture in RSS-based indoor localisation system, the problem of localisation has been transformed into that of classification. Most importantly, the unknown node location can be directly computed without using radio pathloss model or comparing with a radio map, which is regarded as a new way to solve the RSS-based indoor location problem. The trained MLNN can 'remember' the radio information in its deep architecture based on common devices only and without extra hardware installation and modification. Experiments demonstrate that the proposed algorithm achieves higher location accuracy compared with MLE, GRNN and fingerprinting methods.

5 Acknowledgments

This work was supported by NSFC (grant 61300186), Suzhou key laboratory of Internet of Thing (grant SZS201407) and University Science Research Project of Jiangsu Province (grant 13KJB510001, 13KJB520001).

6 References

1 Gabriel, D., Kevin, C., Joan, C.: 'A survey of active and passive indoor localisation systems', *Comput. Commun.*, 2012, **35**, (16), pp. 1939–1954

2 Gianni, G., Sandeep, K., Gupta, G.M.: 'Understanding the limits of RF-based collaborative localization', *IEEE/ACM Trans. Netw.*, 2011, **19**, (6), pp. 1638–1651

3 Hu, J., Xie, L., Xu, J., *et al.*: 'TDOA-based adaptive sensing in multi-agent cooperative target tracking', *Signal Process.*, 2014, **98**, pp. 186–196

4 Yiyin, W., Li, L., Ma, X., *et al.*: 'Dual-tone radio interferometric positioning systems using undersampling techniques', *IEEE Signal Process. Lett.*, 2014, **21**, (11), pp. 1311–1315

5 Xu, J., Ma, M., Law, C.L.: 'Cooperative angle-of-arrival position localization', *Measurement*, 2015, **59**, pp. 302–313

6 Li, X., Pahlavan, K.: 'Super-resolution TOA estimation with diversity for indoor geolocation', *IEEE Trans. Wirel. Commun.*, 2004, **3**, (1), pp. 224–234

7 Zhao, J., Xi, W., He, Y., *et al.*: 'Localization of wireless sensor networks in the wild: pursuit of ranging quality', *IEEE/ACM Trans. Netw.*, 2013, **21**, (1), pp. 311–323

8 Patwari, N., Alfred, O.H., Perkins, M., *et al.*: 'Relative location estimation in wireless sensor networks', *IEEE Trans. Signal Process.*, 2003, **51**, (8), pp. 2137–2148

9 Dai, H., Zhu, Z.-m., Gu, X.-F.: 'Multi-target indoor localization and tracking on video monitoring system in a wireless sensor network', *J. Netw. Comput. Appl.*, 2013, **36**, (1), pp. 228–234

10 Angelo, C., Fabio, R.: 'RSS-based localization via Bayesian ranging and iterative least squares positioning', *IEEE Commun. Lett.*, 2014, **18**, (5), pp. 873–876

11 Kamol, K., Prashant, K.: 'Modeling of indoor positioning systems based on location fingerprinting', *IEEE INFOCOM2004*, 2005, **2**, pp. 1012–1022

12 Zheng, X., Liu, H.B.J., Yang, J., *et al.*: 'A study of localization accuracy using multiple frequencies and powers', *IEEE Trans. Parallel Distrib. Syst.*, 2014, **25**, (8), pp. 1955–1965

13 Feng, C., Au, W.S.A., Valaee, S., *et al.*: 'Received-signal-strength-based indoor positioning using compressive sensing', *IEEE Trans. Mob. Comput.*, 2012, **11**, (12), pp. 1983–1993

14 Fang, S., Lin, T., Lee, K.: 'A novel algorithm for multipath fingerprinting in indoor WLAN environments', *IEEE Trans. Wirel. Commun.*, 2008, **7**, (9), pp. 3579–3588

15 Yim, J.: 'Introducing a decision tree-based indoor positioning technique', *Expert Syst. Appl.*, 2008, **32**, (2), pp. 1296–1302

16 Zhou, M., Zhang, Q., Tian, Z., *et al.*: 'Integrated location fingerprinting and physical neighborhood for WLAN probabilistic localization'. Proc. Fifth Int. Conf. on Computing, Communications and Networking Technologies, 11–13 July 2014, pp. 1–5

17 Yoo, J., Kim, H.J.: 'Target localization in wireless sensor networks using online semi-supervised support vector regression', *Sensors*, 2015, **15**, pp. 12539–12559

18 Faug, S.-H., Tsung-Nan, L.: 'Indoor location system based on discriminant-adaptive neural network in IEEE 802.11 environments', *IEEE Trans. Neural Netw.*, 2008, **19**, (11), pp. 1973–1978

19 Campos, R.S., Lovisolo, L., de Campos, M.L.R.: 'Wi-Fi multi-floor indoor positioning considering architectural aspects and controlled computational complexity', *Expert Syst. Appl.*, 2014, **41**, pp. 6211–6223

20 Rahman, M.S., Park, Y., Kim, K.D.: 'RSS-based indoor localization algorithm for wireless sensor network using generalized regression neural network', *Arab. J. Sci. Eng.*, 2012, **37**, (4), pp. 1043–1053

21 Bengio, Y., Lamblin, P., Popovici, D., *et al.*: 'Greedy layer-wise training of deep networks', *Adv. Neural Inf. Process. Syst.*, 2007, **19**, pp. 153–160

22 'DeepLearnToolbox'. Available at https://www.github.com/rasmusbergpalm/DeepLearnToolbox, accessed 2012

23 Krizhevsky, A., Sutskever, S., Hinton, G.: 'Imagenet classification with deep convolutional neural networks', *Adv. Neural Inf. Process. Syst.*, 2012, **25**, pp. 1–9

24 Mohamed, A., Dahl, G.E., Hinton, G.: 'Acoustic modeling using deep belief networks', *IEEE Trans. Audio, Speech, Lang. Process.*, 2012, **20**, (1), pp. 14–22

25 Bengio, Y.: 'Learning deep architectures for AI', *Found. Trends Mach. Learn.*, 2009, **2**, (1), pp. 1–127