Neural Network-Based Accuracy Enhancement Method for WLAN Indoor Positioning

Yubin Xu, *Member*, *IEEE*, Yongliang Sun, *Student Member*, *IEEE*
Communication Research Center, Harbin Institute of Technology
Harbin, China
ybxu@hit.edu.cn, syl_peter@163.com

***Abstract*—As the need for location-based services (LBS) in indoor environments increases, high accuracy positioning technologies are required, which makes fingerprinting-based positioning methods using wireless local area network (WLAN) develop from single fingerprinting algorithm into multi-algorithm integration. A neural network-based accuracy enhancement (NNAE) method for indoor positioning using WLAN is proposed in this paper. The method takes the advantages of the fingerprinting algorithms based on pattern matching and distance dependence. It uses a neural network-based pattern matching algorithm to estimate the positioning errors and then the estimated positioning errors are used to correct the positioning results calculated by a distance dependent algorithm. The experimental results show that the proposed NNAE method outperforms classical fingerprinting algorithms and effectively enhances the positioning accuracy.**

Keywords-neural network; accuracy enhancement; indoor positioning; WLAN

I. INTRODUCTION

With the popularization of global positioning system (GPS)-enabled mobile devices, location-based services (LBS) have been well developed in outdoor environments in the past few years. However, because satellite signals are blocked by buildings, GPS is not suitable for positioning in indoor environments [1]. In order to offer the LBS like pedestrian navigation, object positioning and proximity marketing in the indoor environments such as shopping malls, hospitals and underground car parks, many indoor positioning technologies have been developed by using ultrasonic, radio frequency identification (RFID), infrared, ultra wideband (UWB), and wireless local area network (WLAN) [2, 3]. For instance, Cricket system is an ultrasonic and RFID-based positioning system developed by Massachusetts Institute of Technology [4]. Active Badge system uses infrared technology to offer room-level positioning service [5]. Ubisense's positioning system provides accurate positioning service for manufacturing industries based on UWB [6]. RADAR positioning system is a WLAN indoor positioning system based on received signal strength (RSS) presented by Microsoft Research and it proves that RSS can be used for positioning inside buildings [7]. Considering general demands and construction cost, the positioning method based on fingerprinting by using WLAN has been favored [8], which also has better performance under non-line-of-sight (NLOS) conditions than time difference of arrival (TDOA) and angle of arrival (AOA) [9, 10].

The fingerprinting-based method has two steps: the off-line step and on-line step. In the off-line step, specific locations are chosen as reference points (RPs). Their Cartesian coordinates and RSS data from different access points (APs) at the RPs are stored in a database called radio-map. In the on-line step, when RSS data are measured by a user in real-time, the RSS data are used to calculate the positioning results through matching the relationship between the RSS data and location coordinates of the RPs in the radio-map [11, 12]. The fingerprinting algorithms are generally composed of pattern matching and distance dependent algorithms [13]. The pattern matching algorithms use artificial intelligence technologies like neural network (NN) [14], support vector machine (SVM) [15] and adaptive neural-fuzzy inference system (ANFIS) [16]. These algorithms must be trained with the radio-map in the off-line step to map the nonlinear relationship between RSS data and location coordinates. In the on-line step, the trained nonlinear function is used for calculating the positioning results. Regarding the distance dependent algorithms, such as nearest neighbor, K nearest neighbors (KNN) and weighted KNN (WKNN) [13, 17], RSS distances between the stored RSS data and measured RSS data in real-time are calculated. Different amounts of the RPs are chosen for calculating the positioning results through averaging the chosen RPs' location coordinates.

However, the performance of distance dependent algorithms is sensitive to the RPs' location distribution and the amount of the chosen RPs used for calculating the positioning coordinates. Although the pattern matching fingerprinting algorithms have no such problem, when the radio-map is updated dynamically, the nonlinear function must be trained again. It is a time-consuming process and is hardly implemented in the on-line step. Thus, this paper proposes a NN-based accuracy enhancement (NNAE) method for WLAN indoor positioning. It offers a solution to enhance the positioning accuracy from a novel perspective of error estimation, which is different from most researches. The method utilizes a NN to map the nonlinear relationship between the RSS data and positioning errors and corrects the positioning results with the errors estimated by the NN, which can effectively eliminate the negative effect caused by the RPs' location distribution and the amount of the chosen RPs. Even if the radio-map is updated dynamically when the nonlinear

This work was supported by National High-Tech Research & Development Program of China under Grant 2008AA12Z305 and National Natural Science Foundation of China under Grant 61071105 and 61101122.

relationship is changed accordingly, the proposed method still can calculate the positioning results by using the distance dependent algorithm.

The rest of the paper is organized as follows: Section II introduces the related algorithms that are applied in this paper. In Section III, the proposed NNAE method for indoor positioning is described in detail. The experimental process, results and analysis are given in Section IV. Finally, the paper is concluded in Section V.

II. RELAED WORK

A. Distance Dependent Algorithm

As a distance dependent fingerprinting algorithm, KNN algorithm has been widely used because of its simplicity and performance. It calculates the RSS distances between the RSS data in the radio-map and measured in real-time and chooses K RPs according to the RSS distances for positioning. The algorithm is given by

$$\begin{cases} d_i = \left(\sum_{j=1}^{N} \left| s_j - S_{i,j} \right|^q \right)^{1/q}, i = 1, \cdots, M \\ D = \{d_1, \cdots, d_M\} \\ \boldsymbol{L}_l = (x_l, y_l), d_l \in MinK(D), l = 1, \cdots, K \\ (\hat{x}, \hat{y}) = \frac{1}{K} \sum_{l=1}^{K} (x_l, y_l) = \frac{1}{K} \sum_{l=1}^{K} \boldsymbol{L}_l \end{cases} \quad (1)$$

where N and M are the amounts of the APs and RPs, respectively. s_j and $S_{i,j}$ are the RSS data received from j th AP in real-time and the RSS data from j th AP at i th RP in the radio-map, respectively. d_i and D are the RSS distance of i th RP and the set of the RSS distances, respectively. $MinK(D)$ is the set of first K minimum RSS distances. $\boldsymbol{L}_l$ and $(\hat{x}, \hat{y})$ are the coordinates of chosen RP and positioning coordinates, respectively. K and q represent the amount of the chosen RPs and the distance type, respectively. When K is equal to 1, the KNN algorithm turns to the nearest neighbor algorithm and only one RP with the minimum RSS distance is chosen. Its location coordinates are viewed as the positioning coordinates.

Regarding the WKNN algorithm, the location coordinates of K chosen RPs are allocated different weights according to their RSS distances. The positioning coordinates are determined by (2) as follows

$$(\hat{x}, \hat{y}) = \frac{\sum_{l=1}^{K} \left((x_l, y_l) / d_l \right)}{\sum_{l=1}^{K} \left(1 / d_l \right)} = \frac{\sum_{l=1}^{K} \left(\boldsymbol{L}_l / d_l \right)}{\sum_{l=1}^{K} \left(1 / d_l \right)}. \quad (2)$$

B. Neural Network

As an artificial intelligence technology, NN has been applied in the fields like nonlinear mapping, parallel process and data fusion. A multi-layer perceptron trained by back-propagation (BP) algorithm has been a powerful tool for these applications. BP algorithm, which was presented by D. E. Rumelhart *et al.* [18], achieves a minimum error through optimizing weights and thresholds. Because a three-layer perceptron has ability to approximate to any nonlinear function [19], a three-layer network structure is applied in this paper. When NN is applied as a fingerprinting algorithm for WLAN indoor positioning, its inputs are the RSS data from different APs and outputs are the positioning coordinates [14, 20]. However, in this paper, the outputs of the NN are changed into the positioning errors and network structure applied in the paper is shown in Fig. 1.

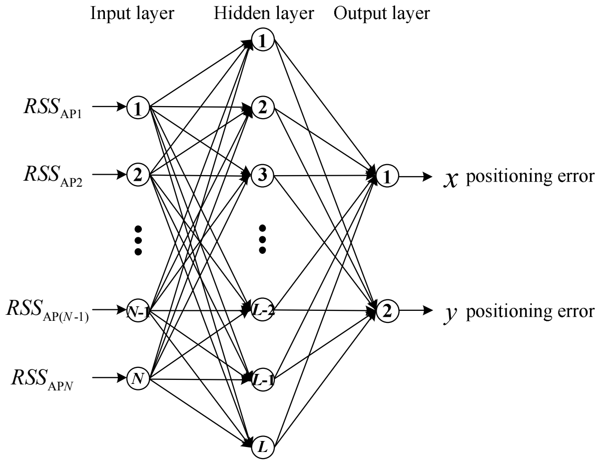

Figure 1. The proposed structure for error estimation.

The aim of the BP algorithm is to optimize weights and thresholds in order to minimize $E(k)$, which is the sum of square errors of the neurons in the output layer given by

$$E(k) = \frac{1}{2} \sum_{j=1}^{2} (d_{j,k} - y_{j,k}^{(3)})^2 \quad (3)$$

where $d_{1,k}$ and $d_{2,k}$ are the expected positioning errors in x and y axis for k th training sample, respectively. $y_{1,k}^{(3)}$ and $y_{2,k}^{(3)}$ are the actual positioning errors in x and y axis for k th training sample, respectively.

The output $y_{j,k}^{(l)}$ of j th neuron in l th layer for k th training sample is calculated by

$$\begin{cases} u_{j,k}^{(l)} = \sum_{i=1}^{I} \left(\omega_{i,j}^{(l-1,l)} x_{i,k}^{(l)} - \theta_j^{(l)} \right) \\ y_{j,k}^{(l)} = f(u_{j,k}^{(l)}) \\ \begin{cases} l = 2, \ j = 1, \cdots, L; \ I = N \\ l = 3, \ j = 1, 2; \ I = L \end{cases} \end{cases} \quad (4)$$

where $x_{i,k}^{(l)}$ is the input of j th neuron in l th layer from i th neuron in $(l\text{-}1)$ th layer for k th training sample. $\omega_{i,j}^{(l-1,l)}$ is the weight from i th neuron in $(l\text{-}1)$ th layer to j th neuron in l th layer. $\theta_j^{(l)}$ is the threshold of j th neuron in l th layer. N, L and 2 are the amounts of neurons in the input layer, hidden layer and output layer, respectively. $f(\cdot)$ is the activation function.

The BP algorithm is an iterative process for optimizing the weights and thresholds. Using the basic BP algorithm, the weights and thresholds of the network are updated by

$$\begin{cases} \omega_{i,j}^{(l,l-1)}(k+1) = \omega_{i,j}^{(l,l-1)}(k) + \eta\delta_{j,k}^{(l)} y_{j,k}^{(l-1)} \\ \theta_j^{(l)}(k+1) = \theta_j^{(l)}(k) \text{-} \gamma\delta_{j,k}^{(l)} \\ \delta_{j,k}^{(l)} = \begin{cases} \sum_{m=1}^{2} \delta_{m,k}^{(l+1)} \omega_{i,m}^{(l,l+1)} f'\left(u_{j,k}^{(l)}\right), \ l=2 \\ [d_{j,k} - y_{j,k}^{(l)}] f'\left(u_{j,k}^{(l)}\right), \ l=3 \end{cases} \\ \begin{cases} l=2, \ i=1,\cdots,N, \ j=1,\cdots,L; \\ l=3, \ i=1,\cdots,L, \ j=1, 2; \end{cases} \end{cases} \tag{5}$$

where η and γ are the learning rates of the weight $\omega_{i,j}^{(l-1,l)}$ and threshold $\theta_j^{(l)}$, respectively. The learning rates are generally set from 0.01 to 0.1 in order to keep the network stable, but a method of adaptive learning rate has been applied to balance the training time and performance.

III. PROPOESD METHOD

The proposed NNAE method is also composed of the off-line step and on-line step and the process of estimating positioning errors is integrated into the two steps. Compared with the classical fingerprinting algorithms, the proposed method chooses extra location points called training points and records the RSS data at these training points for training the NN in the off-line step. In the on-line step, the positioning errors are estimated by the trained NN and used for correcting the positioning coordinates calculated by a distance dependent algorithm. The positioning process of the proposed NNAE method is described in Fig. 2.

As shown in Fig. 2, in the off-line step, after establishing the radio-map, some training points are chosen. At each training point, the RSS data $\boldsymbol{RSS}_{\mathrm{Tr}} = [r_1, r_2, \cdots, r_N]$ from different APs and location coordinates $(x_{\mathrm{Tr}}, y_{\mathrm{Tr}})$ are recorded for training the NN. Using the RSS data $\boldsymbol{RSS}_{\mathrm{Tr}} = [r_1, r_2, \cdots, r_N]$ at the training point, the positioning coordinates $(\hat{x}_{\mathrm{Tr}}, \hat{y}_{\mathrm{Tr}})$ of the training point are calculated by a distance dependent algorithm. With the positioning coordinates $(\hat{x}_{\mathrm{Tr}}, \hat{y}_{\mathrm{Tr}})$ and location coordinates $(x_{\mathrm{Tr}}, y_{\mathrm{Tr}})$ of the training point, the positioning errors $(\delta x_{\mathrm{Tr}}, \delta y_{\mathrm{Tr}})$ of the training point are calculated by

$$\begin{cases} \delta x_{\mathrm{Tr}} = x_{\mathrm{Tr}} \text{ - } \hat{x}_{\mathrm{Tr}} \\ \delta y_{\mathrm{Tr}} = y_{\mathrm{Tr}} \text{ - } \hat{y}_{\mathrm{Tr}} \end{cases}. \tag{6}$$

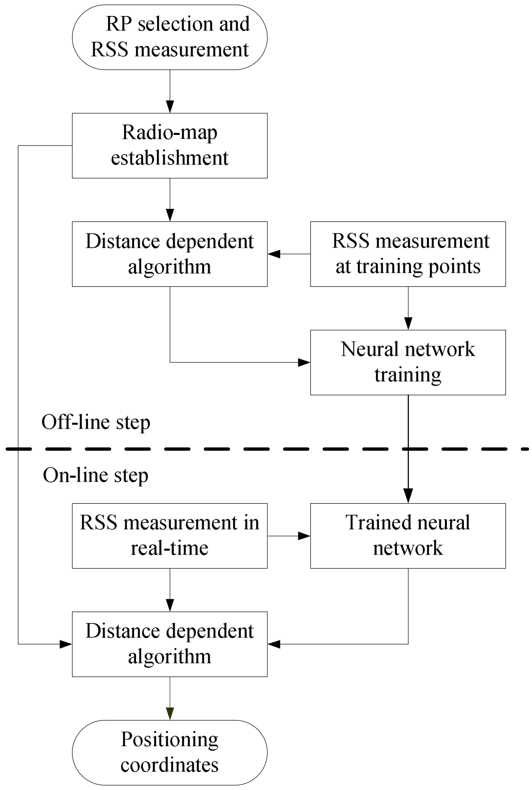

Figure 2. The positioning process of the proposed NNAE method.

After all the positioning errors are calculated by the distance dependent algorithm, the positioning errors and RSS data of these training points are separately used as the outputs and inputs of the NN for its training. Although the training process is time-consuming, the training process and the workload at the training points are completed in the off-line step, which can barely have influence on the real-time LBS.

In the on-line step, when the real-time RSS data $\boldsymbol{RSS}_{\mathrm{Te}} = [s_1, s_2, \cdots, s_N]$ from different APs are measured at a test point (TP) on an experimental trajectory, they are not only inputted into the same distance dependent algorithm to calculate the positioning coordinates $(\hat{x}_{\mathrm{Te}}, \hat{y}_{\mathrm{Te}})$, but also inputted into the trained NN to estimate the positioning errors $(\delta x_{\mathrm{Te}}, \delta y_{\mathrm{Te}})$. Then the final positioning coordinates $(\hat{x}_{\mathrm{P}}, \hat{y}_{\mathrm{P}})$ after the correction are calculated by

$$\begin{cases} \hat{x}_{\mathrm{P}} = \hat{x}_{\mathrm{Te}} + \delta x_{\mathrm{Te}} \\ \hat{y}_{\mathrm{P}} = \hat{y}_{\mathrm{Te}} + \delta y_{\mathrm{Te}} \end{cases}. \tag{7}$$

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. *Experimental Setup*

As the plan of the experimental environment shown in Fig. 3, the size of the floor is about 24.9m by 66.4m and the experimental trajectory begins from A to B along the 3m width corridor. 9 Linksys WAP54G APs are deployed in the floor for communication requirement and 5 APs are in the experimental area. An ASUS A8F laptop equipped with an Intel PRO/Wireless 3945ABG wireless network card is used to measure the RSS data. With a software program called NetStumbler, 2 RSS samples per second can be collected by the laptop. In the off-line step, 182 RPs in two parallel lines with 1m gaps along the whole corridor are chosen and 300 RSS samples are collected within 150s at each RP. A total of 960 RSS samples along the experimental trajectory are collected for training the NN in the off-line step. In the on-line step, 96 TPs from A to B are chosen with 0.6m gaps to test the proposed NNAE method. 2 RSS samples are measured within 1s at each TP to simulate the condition under which a user moves along the corridor with a terminal device at a speed of 0.6m/s.

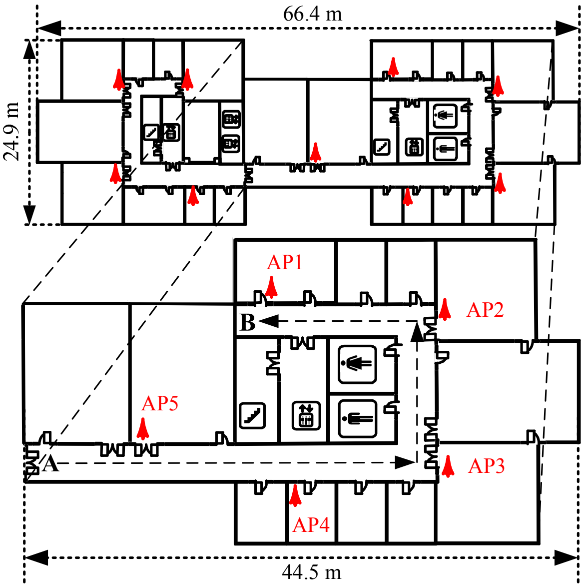

Figure 3. The plan of the experimental environment.

Regarding the parameters of the algorithms used in the paper, parameter K of the KNN and WKNN algorithm is set equal to 7 and parameter q is set equal to 2 to calculate the Euclidean distance of RSS data. So 7 RPs are chosen for calculating the positioning coordinates. The three-layer NN is trained by using the BP algorithm of gradient descent with adaptive learning rate. Sigmoid function and linear function are used as the activation functions for the hidden layer and output layer, respectively. 9 inputs are the RSS data from 9 APs and 2 outputs are the positioning errors in x and y axis. The amounts of neurons in the hidden layer and the training epochs are set equal to 20 and 10000, respectively. In order to compare the proposed NNAE method with a NN-based fingerprinting algorithm, a NN of same parameters and structure is applied as a fingerprinting algorithm in the paper. It is trained by the same data that include the RSS data and RPs' coordinates in the radio-map for positioning and the RSS data and location coordinates of the training points for error estimation.

B. Experimental Results and Analysis

For a fixed user, it is easy to get sufficient RSS data at one location to calculate the user's location coordinates. However, when the user moves in indoor environments, it is difficult to collect enough RSS data at one location to represent the radio characteristics of that location. Therefore, offering a mobile user satisfactory LBS is more difficult but more practical than a fixed user. Thus, for a mobile user, it is necessary to process the less accurate positioning results calculated by the classical fingerprinting algorithms to improve the positioning accuracy. According to the RSS data collected under the experimental condition in this paper, take the KNN algorithm as an example, the positioning results of the KNN algorithm and the proposed NNAE (KNN) method are shown in Fig. 4. Because of the influence of the RPs' locations at the beginning and end of the experimental trajectory, some RPs that are not near to the TPs are still chosen for positioning according to the RSS distances, so the performance of the KNN algorithm decreases at these locations. However, the proposed NNAE (KNN) method can offer more accurate positioning results under such condition through correcting the positioning results of the KNN algorithm by using the estimated positioning errors.

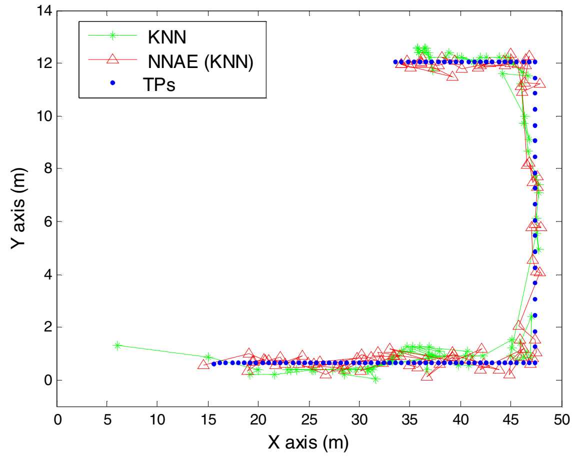

Figure 4. The positioning results of the KNN and NNAE (KNN).

According to the experimental results shown in Table I, The NNAE method greatly outperforms the classical fingerprinting algorithms and the mean errors of the KNN, WKNN, NN, NNAE (KNN), and NNAE (WKNN) are 2.54m, 2.52m, 2.42m, 1.33m, and 1.26m, respectively. The cumulative probability of the NNAE (KNN) and NNAE (WKNN) can reach 79.2% and 81.3% within the positioning error of 2m, respectively. The KNN, WKNN and NN can only reach 45.8%, 46.9% and 61.5%, respectively. More details of the results are shown in Table I and the curves of the cumulative probabilities are also shown in Fig. 5.

TABLE I. Performance Comparison of Various Methods

Method	**Mean Error (m)**	**Cumulative Probability (%)**	
		Within 2m Error	***Within 3m Error***
KNN	2.54	45.8	74.0
WKNN	2.52	46.9	75.0
NN	2.42	61.5	74.0
NNAE (KNN)	1.33	79.2	92.7
NNAE (WKNN)	1.26	81.3	96.9

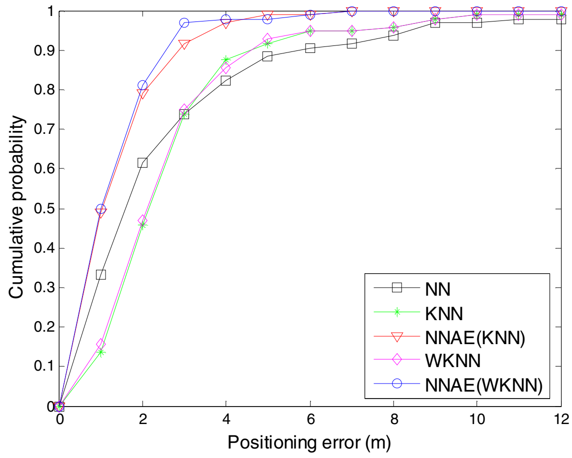

Figure 5. Comparison of cumulative probabilities.

As shown in Fig. 5 and Table I, the proposed NNAE method is apparently effective in enhancing the positioning accuracy and has better performance than the classical fingerprinting algorithms. Compared with the classical fingerprinting algorithms, although extra RSS data are collected at the training points, it is worthy to increase a small quantity of workload in the off-line step and compute only an extra nonlinear function in the on-line step to improve the positioning performance.

V. Conclusion

In order to offer mobile users satisfactory LBS, a NNAE method is proposed in this paper to enhance the positioning accuracy of classical fingerprinting-based indoor positioning system. The proposed NNAE method not only integrates the fingerprinting algorithms based on pattern matching and distance dependence, but also overcomes their disadvantages and improves the positioning performance. In the off-line step, the method trains a three-layer NN with BP algorithm to approximate to the nonlinear relationship between the RSS data and positioning errors generated by a distance dependent algorithm. In the on-line step, positioning errors are estimated by the trained NN when the real-time RSS data are measured and then used to correct the positioning results of the distance dependent algorithm. Thus, more accurate positioning results are obtained for LBS, which has been proved by the experiment in the paper. Additionally, when the radio-map is dynamically updated and the trained NN fails to work, the proposed NNAE method still can calculate the positioning coordinates by using the distance dependent algorithm.

VI. Acknowledgment

The authors gratefully acknowledge the reviewers for helpful comments. This work was supported by National High-Tech Research & Development Program of China under Grant 2008AA12Z305 and National Natural Science Foundation of China under Grant 61071105 and 61101122.

References

[1] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, “On indoor position location with wireless lans,” in *Proceedings of 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 2, pp. 720-724, Sept. 2002.

[2] Y. Y. Gu, A. Lo, and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13-32, First Quarter. 2009.

[3] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 6, pp. 1067-1080, Nov. 2007.

[4] R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The active badge location system”, *ACM Transactions Information Systems*, vol. 10, no. 1, pp. 91-102, Jan. 1992.

[5] N. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system,” in *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pp. 32-43, Aug. 2000.

[6] Ubisense. [Online]. Available: http://www.ubisense.net/.

[7] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in *Proceedings of 19th IEEE Computer and Communications Societies Conference*, vol. 2, pp. 775-784, Mar. 2000.

[8] Y. Zheng, O. Baala, and A. Caminada, “Efficient design of indoor positioning systems based on optimization model,” *Wireless Days*, pp. 1-5, Oct. 2010.

[9] R. Yamasaki, A. Ogino, T. Tamaki, T. Uta, N. Matsuzawa, and T. Kato, “TDOA location system for IEEE 802.11b WLAN,” in *Proceedings of Wireless Communications and Networking Conference*, vol. 4, pp. 2338-2343, Mar. 2005.

[10] C. Wong, R. Klukas, and G. G. Messier, “Using WLAN infrastructure for angle-of-arrival indoor user location,” in *Proceedings of 2008 IEEE Vehicular Technology Conference*, pp. 1-5, Sept. 2008.

[11] K. Kaemarungsi, “Efficient design of indoor positioning systems based on location fingerprinting,” in *Proceedings of International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 1, pp. 181-186, June 2005.

[12] O. Baala and A. Caminada, “Location precision in indoor positioning system,” *2006 Innovations in Information Technology*, pp. 1-5, Nov. 2006.

[13] Y. B. Xu, M. Zhou, W. X. Meng, and L. Ma, “Optimal KNN positioning algorithm via theoretical accuracy criterion in WLAN indoor environment,” in *Proceedings of 2010 IEEE Global Telecommunications Conference*, pp. 1-5, Dec. 2010.

[14] S. H. Fang and T. N. Lin, “Indoor location system based on discriminant-adaptive neural networks in IEEE 802.11 environments,” *IEEE Transactions on Neural Network*, vol. 19, no. 11, pp. 1973-1978, Nov. 2008.

[15] M. Brunato and R. Battiti, “Statistical learning theory for location fingerprinting in wireless LANs,” *Computer Networks*, vol. 47, no. 6, pp. 825–845, Apr. 2005.

[16] A. Teuber, B. Eissfeller, and T. Pany, “A two-stage fuzzy logic approach for wireless LAN indoor positioning,” in *Proceedings of 2006 IEEE/ION Position, Location, And Navigation Symposium*, pp. 730-738, Apr. 2006.

[17] Q. Tran, J. W. Tantra, C. H. Foh, A. H. Tan, K. C. Yow, and D. Y. Qiu, “Wireless indoor positioning system with enhanced nearest neighbors in signal space algorithm,” in *Proceedings of 2006 IEEE Vehicular Technology Conference*, pp. 1-5, Sept. 2006.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533-536, Otc. 1986.

[19] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.

[20] S. Outemzabet and C. Nerguizian, “Accuracy enhancement of an indoor ANN-based fingerprinting location system using particle filtering and a low-cost sensor,” in *Proceedings of 2008 IEEE Vehicular Technology Conference*, pp. 2750-2754, May 2008.