Research Article

Deep Belief Networks for Fingerprinting Indoor Localization Using Ultrawideband Technology

Junhai Luo and Huanbin Gao

School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Junhai Luo; junhai_luo@uestc.edu.cn

Received 4 September 2015; Revised 27 December 2015; Accepted 31 December 2015

Academic Editor: Iqbal Gondal

Copyright © 2016 J. Luo and H. Gao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the increasing requirement of localization services in indoor environment, indoor localization techniques have drawn a lot of attention. In recent years, fingerprinting localization techniques have been proved to be effective in indoor localization tasks. Due to the complexity and variability of indoor environment, some traditional geometric localization techniques based on time of arrival (TOA), received signal strength (RSS), or direction of arrival (DOA) may cause big position errors. Unlike common geometric localization methods, fingerprinting localization techniques estimate the position of target by creating a pattern matching model or regression model for the measurement. Therefore, a suitable learning model is the key of a fingerprinting location system. This paper presents a fingerprinting based localization technique using deep belief network (DBN) and ultrawideband (UWB) signals in an office environment. Some location-dependent parameters extracted from channel impulse response (CIR) are used as signatures to build the fingerprinting database. The construction of DBN which is based on the fingerprinting database is also discussed in this paper. Experiment results show that, with appropriate fingerprinting database and model structure, the location system can get desired accuracy.

1. Introduction

In recent years, indoor localization technique has received a lot of attention. Some concepts of outdoor location systems are used in indoor localization, but they are always subject to big position errors. Most of the accuracy problems are caused by the complexity and variability of indoor environments such as multipath reflections.

UWB refers to the signals that can spread over a bandwidth larger than 500 MHz which can be obtained by generating a series of extremely short duration pulses with large bandwidths [1]. UWB has many attributes such as low power, large bandwidth, and high time resolution, which makes it ideal for indoor location task. Additionally, the high time resolution of UWB results in better multipath resolvability at the same time, which is quite crucial in the complex indoor environment. According to the UWB impulse sequence acquired at the receiver, a lot of useful position information related to the environment can be exacted.

Compared to the geometric localization technique which is widely used in outdoor localization tasks, fingerprinting localization technique provides a different method to determine the position of a target in an indoor environment [2–4]. It can be divided into two phases: the off-line phase and the online phase. In the off-line phase, a large fingerprinting database is obtained from the CIR as a training database of a particular machine learning model such as k-nearest neighbor (kNN) [5, 6], neural networks (NNs) [7–9], and support vector machine (SVM) [10]. After the chosen model is trained successfully, it can be used as the regression scheme which maps the signatures to the position coordinates. In the online phase, the constructed regression model can be used to locate a target node according to the obtained signatures. Considering that the signatures in fingerprinting database are of great complexity, DBN learning model is proposed in this paper.

DBN is an effective machine learning model proposed in 2006 [11]. It has been widely used in many different fields such as image [12], speech [13], and language processing [14]. It is a multilayer model which imitates the mode in which the human brain represents information. Unlike normal multilayer network models such as kNN, the training phrase of
DBN is divided into two parts, the pretraining part and the fine-tune part. In the pretraining part, the network is trained layer by layer with a greedy unsupervised method, when the network will find the hidden structure of the input data and initialize the parameters [15–19]. In the fine-tune part, the network is trained by a back propagation supervised method with initialized parameters obtained from the previous part. DBN can conquer many problems that a multilayer model may suffer from, such as overfitting and local optimum. It is proved that DBN can acquire a desired accuracy in many complex classification and regression problems.

The main contributions of this work are as follows:

(i) Proposing a fingerprinting location algorithm using DBN to match the coordinates of a target to its signatures.

(ii) Using the CIR parameters independent of transmission time as signatures of fingerprinting localization, which makes the algorithm free from the dependence of synchronization.

(iii) Applying SAE to pretrain DBN using unlabeled samples which can be acquired easily.

The remaining part of this paper is organized as follows: In Section 2, we present some related works in recent years. In Section 3, we discuss the indoor environment and channel model. The architecture of the indoor environment is presented and the transmission characteristic of UWB signals is discussed. In Section 4, a fingerprinting algorithm using DBN is proposed. For the fingerprinting localization technique, we discuss the offline and online phrase, respectively. For DBN, the details of pretraining part and fine-tune part are described and studied. In Section 5, we explain the experiment setups and show the experiment results and analysis. Finally, we put the conclusion of this paper in Section 6.

2. Related Works

Recently, in [1–6], there has been research working on the improvement of fingerprinting localization techniques. Fingerprinting localization is used in different kinds of localization systems with different hardware, such as UWB [1], RFID [2], WIFI [3], and WSN [4]. Two main aspects of fingerprinting localization are the construction of database and the training of matching model. The fingerprinting database was constructed by analyzing the CIR of signals. The signatures of samples may consist of CIR parameters based on RSS, TOA, or AOA. As for the matching model, different matching models such as kNN [5, 6], NN [7–9], and SVM [10] were used to match the target position to the CIR parameters of a particular signal channel.

The concept of deep learning was first established in 2006 in [11]. As a typical deep learning model, DBN solves the training problem which may occur in a deep neural network. It is widely used in many different areas in the recent years, such as graphics processing and language recognition [12–14]. DBN is advanced model which can fit the complex nonlinear relationship between attributes in many issues [15, 16]. Pretraining phase of DBN is the key reason for its performance. Two main methods to pretrain DBN are restricted Boltzmann machine (RBM) [17] and SAE [18, 19]. They can be used to initialize the parameters of DBN and discover the hidden structure of the input data.

3. Indoor Localization Environment and Channel Model

We consider an office scenario as the indoor environment of the location finding system in this paper. The Beacon Node (BN) is the signal receiver whose position is constant, while the Blind Node (BLN) is the signal emitter whose position needs to be estimated. There are several BNs and a BLN in this test area. And the task is to find the position of BLN according to the received signals of BNs.

The environment we consider here contains different kinds of objects which may influence the transmission of signals in unpredictable ways. The indoor environment is divided into several parts by different kinds of obstructions, thus making the UWB signals propagate in mixed line-of-sight (LOS) and non-line-of-sight (NLOS) conditions. As a result, the BNs receive the signals from different paths. Because of the complexity of the environment, no existing decay model can be used to analyze the signal propagation. In addition, the multipath effect of the signals makes them more unpredictable. As for UWB signal, what BN receives is a series of impulses acting as bases of the localization system in this work.

The indoor channel model in this paper can be described as the time domain response of CIR:

\[ r(t) = N(t) \sum_{\tau=1}^{N(t)} g_{\tau}(t) e^{i(t - \tau(t))} + n(t), \]  

(1)

where \( e(t) \) is the original impulse from a BLN, while \( g_{\tau}(t) \) and \( \tau(t) \) are the path gain and delay of the \( \tau \)th path, respectively, \( N(t) \) is the total number of detected multipath components, and \( n(t) \) is the noise depending on the environment itself. Besides, if the channel is considered stationary, the domain result turns out to be

\[ r(t) = N(t) \sum_{\tau=1}^{N(t)} g_{\tau} e^{i(t - \tau(t))} + n(t). \]  

(2)

The parameters of CIR vary according to the indoor environment and the positions of the BNs and BLN. Each impulse with the path gain \( g_{\tau} \) and delay \( \tau \) represents a propagation path of signal with a specific transmission time and decay rate. It is difficult to reconstruct the propagation of UWB impulse according to the received sequence. As a result, traditional geometric localization techniques will be invalid in such kind of situation.

4. Fingerprinting Localization Using Deep Belief Network

Due to the existence of NLOS, reverberation, and multipath effect in the scenario, geometric localization technique may
consequently achieve low accuracy in this situation because of its dependency on LOS situation. As a result, fingerprinting technique is implemented in this work. Actually, compared to traditional localization techniques, the advantage of fingerprinting technique is that the localization task is no longer tied to any defined channel model to infer the geometry or distance information of the BLN. Instead, fingerprinting technique only depends on the database extracted from the acquired sequence samples.

In order to map the sequence to the position of a BLN, a regression model is needed in the fingerprinting localization problems. Some deep model can be used to learn complex relationship between attributes and labels in different research fields. However, the performance always turns out to be unsatisfactory because of the poor local optima situation which may happen in the training phase when the number of layers is increased. Taking this problem into consideration, we apply DBN as a regression model to the fingerprinting localization algorithm in this paper.

As a result, a fingerprinting localization algorithm using DBN is presented in this paper. The localization algorithm can be divided into two phases: the off-line phase and the online phase.

In the online phase, the signature of a BLN is then normalized and input to the learned DBN model. The values of the last layer neurons represent the normalized coordinates of BLN. After transforming the normalized coordinates into realistic coordinates, the estimate of BLN position is acquired.

The off-line phase is the key step of the localization algorithm. It can be divided into two parts: the building process of fingerprinting database and the training process of DBN. In the building process of fingerprinting database, signatures and coordinates of BLNs are extracted and normalized. In the training process of DBN, a DBN model is built and trained in order to match the signature of a BLN to its position.

4.1. The Building Process of Fingerprinting Database. The database is built according to the impulse samples acquired at the BNs and the coordinates of the BLN. From the impulse response, we can get several parameters, namely, the number of multipath components \(N\), the power of the \(i\)th path \(P_i\), and the total power \(P\). \(P_i\) can be defined according to the time domain response in Formula (2):

\[
P_i = \int_{-\infty}^{+\infty} g_i e^{(t - \tau_i)} + n(t)^2 \, dt.
\]

\(P\) can be defined as follows:

\[
P = \sum_{i=1}^{N} P_i.
\]

The fingerprinting database can be expressed as follows:

\[
S = \{(s^1, p^i), (s^2, p^i), \ldots, (s^k, p^k), \ldots, (s^M, p^M)\},
\]

where \((s^k, p^k)\) represents the \(k\)th sample of the fingerprinting database and \(M\) is the number of samples in the database. Besides, the first component of a sample \(s^k\) is a vector serving as the signatures of the \(k\)th sample, given by

\[
s^k = (s^k_1, s^k_2, \ldots, s^k_n),
\]

where \(s^k_i\) is a vector that represents the normalized signatures extracted from the \(i\)th BN. It is worth mentioning that, unlike many other location-dependent parameters, the parameters chosen in this paper are mainly based on RSS, which free the BNs from the dependence of synchronization and make the deployment of the system easier. In this paper, \(N, P, P_1, P_2\) are considered as signatures. Therefore, \(s^k_i\) turns out to be

\[
s^k_i = \left( \frac{N^k_i}{N_{\text{max}}}, \frac{P^k_i}{P_{\text{max}}}, \frac{P_{1,i}^k}{P_{1,\text{max}}}, \frac{P_{2,i}^k}{P_{2,\text{max}}} \right),
\]

where \(N_{\text{max}}, P_{\text{max}}, P_{1,\text{max}}, P_{2,\text{max}}\) are the maximum values of \(N, P, P_1, P_2\), respectively. The second component \(\hat{p}^k\) is the position of the \(k\)th sample BLN represented by its normalized coordinates:

\[
\hat{p}^k = \left( \frac{x^k, y^k, z^k}{x_{\text{max}}, y_{\text{max}}, z_{\text{max}}} \right),
\]

where \(x^k, y^k, z^k\) are the coordinates of a BLN, while \(x_{\text{max}}, y_{\text{max}}, z_{\text{max}}\) are the maximum values of the coordinates, respectively.

To be precise, \(s^k\) is the input pattern while \(\hat{p}^k\) is the output pattern. The task of the off-line phase is to match the output position \(\hat{p}^k\) to the input signature by constructing and training a regression model.

4.2. The Training Process of Deep Belief Network. In this work, DBN is used as a regression model of the localization algorithm. It is an advanced model which can achieve better results even when the network is deep or the number of hidden neurons is large. The excellent performance of DBN should be attributed to the extra unsupervised training using only unlabeled data the result of which is used to initialize DBN. In this way, the local optima problem is solved obviously.

The training phase of DBN used in our indoor location algorithm can be divided into two parts. One is the global-training procedure of the entire network using labeled samples. The other is the layer-wise pretraining procedure using a greedy unsupervised training method to discover features from unlabeled samples.

4.2.1. Global-Training of DBN. The framework of DBN is illustrated in Figure 1. It consists of \(n\) layers of units. The first layer is the observation layer which represents the input signatures of a given BLN, while the last layer is the output layer that consists of three nodes which indicate the three coordinates of the BLN, respectively. The intermediate two layers between input and output layers are called the hidden layers, which are used to find the potential data correlations of the signature layer. Considering that the model has \(n\) layers, the parameters of the network can be illustrated as

\[
(W, b) = (W^1, b^1, W^2, b^2, \ldots, W^i, b^i, \ldots, W^{n-1}, b^{n-1}),
\]
Require: \((W, b) = \text{the parameters of DBN}; S = \text{the total fingerprinting dataset}\)
Ensure: \(\text{train DBN with a fingerprinting dataset}\)
\[
\text{pre-train } (W, b) \\
\text{divide } S \text{ into baths} \\
\text{for each bath } b \in S \text{ do} \\
\quad \text{calculate weight decay} \\
\quad \text{calculate cost function} \\
\quad \text{update } (W, b) \\
\text{end for}
\]

Algorithm 1: Global-training of DBN.

\[
\begin{align*}
\text{Position vector} & \quad L_n \\
\text{Hidden units} & \quad L_{n-1} \\
\vdots & \quad \cdots \\
\text{Hidden units} & \quad L_2 \\
\text{Fingerprint vector} & \quad L_1
\end{align*}
\]

Figure 1: The framework of DBN.

where \((W^l, b^l)\) represents the parameters between layer \(l\) and layer \(l+1\) and \(W^l\) is the weight matrix between layer \(l\) and layer \(l+1\), while \(b^l\) is the bias vector. Therefore, the activation phase between layer \(l\) and layer \(l+1\) can be described as

\[
a^{l+1}_i = f \left( \sum_{j=1}^{s_l} W^l_{ij} a^l_j + b^l_i \right),
\]

where \(W^l_{ij}\) is the weight connecting the \(i\)th neuron in layer \(l+1\) and the \(j\)th in layer \(l\), \(s_l\) is the number of neurons that layer \(l\) contains, \(a^l_j\) denotes the activation value of the \(i\)th unit in layer \(l\), and \(f(\cdot)\) is the activation function. In this paper, we choose the sigmoid function as the activation function:

\[
f(z) = \frac{1}{1 + \exp(-z)}.
\]

The task of the training phase is to learn the parameters between layers using the back propagation algorithm. Concretely, the parameters are adjusted to minimize a cost function:

\[
J(W, b) = \frac{1}{m} \sum_k^m J(W, b; s_k^k, p_k^k) + \frac{\lambda}{2} \sum_{l=1}^{n-1} \sum_{j=1}^{s_l} (W^l_{ij})^2,
\]

where the first additive term is the mean value of the cost function of \(m\) samples, while the second additive term is the weight decay term used to constrain the change rate of weights. And the cost function of a single fingerprint sample is defined as

\[
J \left( W, b; s^k, p^k \right) = \frac{1}{2} \left\| h_{W,b} \left( s^k \right) - p^k \right\|^2,
\]

where \(h_{W,b}(\cdot)\) denotes the function used to map the input signature to a predicted position. The cost function will be used to update DBN parameters with gradient descent algorithm which can be described as follows:

\[
\left( \hat{W}, \hat{b} \right) = (W, b) + \alpha \frac{\partial J(W, b; s^k, p^k)}{\partial (W, b)},
\]

where \(\alpha\) is the learning rate and \(m\) is the size of a batch. \((\hat{W}, \hat{b})\) represents the updated parameters.

The procedure of global-training is listed in Algorithm 1. The model is trained with the fingerprinting database which is divided into batches. The cost function of every sample is calculated and cumulated in a batch with size \(m\). The parameters of DBN are updated batch by batch with gradient descent algorithm in order to avoid the overfitting situation.

4.2.2. Pretraining of DBN. Since poor local optimization phenomenon depresses the performance of deep model, it is always caused by inappropriate values of initial parameters. In DBN, pretraining algorithm is introduced to initialize the parameters. Instead of initializing the parameters randomly, we use pretraining algorithm to set the parameters connecting the input and hidden layers. There are two models that can be used to initialize a feed-forward neural network. One is the restricted RBM, and the other is SAE. Considering the fact that DBN is used as a regression model in this paper, we apply SAE as the unsupervised training model.

SAE is an unsupervised learning model, whose aim is to reconstruct the input vector from a neural network with a hidden layer. The structure of SAE is shown in Figure 2. The first layer is the visible layer, while the last layer is the reconstructed result of it. The training phase is to train a neural network with three layers whose label vector is the input vector itself. Namely, it will fit a function \(h_{W,b}(x) = x\). If the model can reconstruct the input vector at the output layer, the hidden layer can be regarded as another expression of the input vector. And the parameters of SAE are initialized.
In this paper, the signatures acquired in the indoor environment are always not accurate. The impulse sequences received by the BNs contain noises. As a result, reconstructing the fingerprint vector at the output layer in Figure 2 is not enough. In order to reduce the influence of noises, the average fingerprints are acquired by averaging the signature values of repeated measurements. The input vector in Figure 2 is acquired when the real fingerprint vector is the input vector in Figure 1. The output vector in Figure 2 is acquired when the average fingerprint vector is the output vector in Figure 1. Using \((W^1, b^1)\) to initialize the corresponding parameters in DBN can reduce the influence of noises in real fingerprints.

In this paper, the input vector consists of signatures from several receivers in the indoor environment. Considering the fact that the dimension of our input vector is small, we need to increase the number of hidden units in order to find the interesting structure in the input data. However, too many hidden units may cause poor performance of the pretraining phase. Consequently, a sparsity constraint is required for the hidden units during the unsupervised learning in order to limit the average activation of a hidden unit. Assuming that samples are input to SAE, the average activation of the \(i\)th unit in hidden layer can be defined as

\[
\tilde{\rho}_i = \frac{1}{m} \sum_{k=1}^{m} \alpha_i^k,
\]

where \(\alpha_i^k\) is the value of the \(i\)th unit in hidden layer when the \(k\)th sample is input to the model. The sparsity constraint term can be acquired as

\[
J_{\text{sparsity}}(W^1, b^1) = \sum_{i=1}^{L_1} \rho \log \frac{\rho}{\tilde{\rho}_i} + (1 - \rho) \log \frac{1 - \rho}{1 - \tilde{\rho}_i},
\]

where \(\rho\) is the sparsity parameter which is close to 0. The value of \(\rho\) depends on the number of hidden units. The sparsity constraint term \(J_{\text{sparsity}}(W^1, b^1)\) is minimized only when \(\tilde{\rho}_i\) equals \(\rho\). Consequently, the cost function can be modified as

\[
J_{\text{SAE}}(W, b) = J(W, b) + \beta J_{\text{sparsity}}(W^1, b^1),
\]

where \(\beta\) is the weight of sparsity constraint term. Accordingly, after training SAE, the learned parameters can be used to initialize DBN. The parameters among the first three layers, namely, \((W^1, b^1)\) and \((W^2, b^2)\), are pretrained by corresponding SAE bottom-up in a greedy method. Afterwards, the fine-tune method based on back propagation algorithm is applied to the entire DBN to map the input signatures to the coordinates of the BLNs.

The procedures of pretraining are listed in Algorithm 2. The parameters of DBN are pretrained layer by layer with SAE model. Every SAE is trained with the fingerprinting database which is divided into batches. The cost function of every sample is calculated and cumulated in a batch with size \(m\). The parameters of DBN are updated batch by batch with gradient descent algorithm.

5. Experiment and Results

5.1. Setups of Experiment. As shown in Figure 3, the localization system is deployed in an office scenario with several obstructions. There are three fixed BNs distributed in the localization area. When the BLN emits a UWB impulse in this area, we will acquire three different impulse sequences in the BNs, respectively.

Python 2.7 is used as a simulator in this paper. The propagation process of UWB signal in indoor environment is simulated by using the 3D ray tracing algorithm [20]. The BLN is abstracted to a point which emits rays in a certain position in the 3D space. The rays may go through the obstructions or reflect on the surfaces. Every ray is traced until its power is too low. The BNs are abstracted to spheres called receive balls which record the rays that pass through them. The rays received by a BN can be converted to the impulse sequences which are needed in the experiment.

In the off-line phase, a large amount of fingerprinting samples needs to be collected at first. Gathering fingerprinting samples is a time-consuming work, owing to the fact that collecting the signatures as well as the corresponding position...
Require: DBN = the structure of DBN; \( S \) = the total fingerprinting dataset

Ensure: pre-train DBN with a fingerprinting dataset

\[
\text{for each hidden layer} \in \text{DBN do}
\]

\[
\begin{align*}
&\text{create SAE model} \\
&\text{randomly initialize parameters of SAE} (W, b) \\
&\text{divide} S \text{ into batches} \\
&\text{for each batch} \in S \text{ do} \\
&\quad\text{calculate sparsity constraint term} \\
&\quad\text{calculate weight decay} \\
&\quad\text{calculate cost function} \\
&\quad\text{update} (W, b) \\
&\text{end for} \\
&\text{initialize the parameters of DBN with} (W^1, b^1) \\
\end{align*}
\]

\textbf{Algorithm 2: Pretraining of DBN.}

is inconvenient in realistic work. However, relatively speaking, the unlabeled fingerprinting samples without position information can be accessed easily.

In this paper, the BLN is placed in 250 different positions successively. 10 different samples are collected when BLN is placed in a position. Considering the fact that a large amount of fingerprint samples is needed, the collected samples are used repeatedly in the training process. Besides, extra 1000 labeled samples are needed to test the performance of this localization system.

As is previously discussed, the signature of the \( k \)th sample \( f^k \) consists of several parameters of the CIR. In this experiment we consider four different parameters of the CIR for the \( i \)th BN, namely, the number of multipath components \( N^k_i \), the total power \( P^k_i \), the first-path component power \( (P1)^k_i \), and the second-path component power \( (P2)^k_i \). We will chose all or part of the parameters and discuss the choice of signatures and its effect later.

In addition, as shown in Figure 3, we deploy 3 BNs in the test area. The position of each BN is fixed. We will also discuss the influence of the BNs by changing the number of deployed BNs.

DBN model considered in this experiment is constructed of 7 layers including 5 hidden layers, and each hidden layer contains 60 hidden units. In other words, the parameters connecting the first 6 layers will be initialized layer by layer using SAE.

5.2. Results and Discussion. As is discussed above, after the off-line phase, the indoor localization system based on DBN model is established and used to predict the position of a BLN according to the impulse sequence received by the BNs. Comparing the predicted coordinates to the actual coordinates in the test database, we can get the evaluation of the localization performance. In this experiment, cumulative density function (CDF) of position error is used to evaluate the performance.

The localization performances of different matching models are compared in Figure 4. The fingerprinting databases used in DBN, kNN, SVR, and NN are the same.

The training time of DBN is the longest in these models while the position error of DBN is the lowest in this experiment. The \( k \)NN model does not have a training process and its accuracy is lowest. The complexity of SVR is lower than DBN and NN, but its accuracy is still lower than DBN and NN. DBN and NN model in the experiment share the same structure of network. Compared to NN, DBN has an extra pretraining process which increases its complexity as well as accuracy.

The choice of signatures will influence the localization result of the system. In Figure 5, different groups of CIR parameters are chosen to construct different signatures. The model using all of the four parameters acquires the best accuracy in this experiment. The parameters of CIR contain different environment information which is the key of ensuring localization accuracy.

Another factor related to the signatures is the number of BNs. The BNs deployed in different positions will receive
different impulse sequences which represent the position feature of a BLN. And more BNs mean more valuable position information that the signatures contain. In Figure 6 we take three experiments with different numbers of BNs and compare the results. The experiment with the most BNs reaches the highest position accuracy.

6. Conclusion

In this paper, we propose an indoor localization algorithm based on fingerprinting and DBN techniques. According to the UWB impulse sequences received by different BNs, we extract the CIR parameters and construct the fingerprinting database. In the off-line phase of fingerprinting localization, we apply a DBN model as a regression model to map the position of a BLN to the corresponding signatures. Greedy unsupervised training method to SAE model is used to initialize the parameters of a DBN model in the pretraining part. Supervised back propagation training method is used to fine-tune the initialized parameters of DBN afterwards. At last, CDF of position error is used to evaluate the performance of the localization algorithm. The localization accuracy is improved when DBN model is used in this algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported in part by the Program for Science and Technology Innovative Research Team for Young Scholars in Sichuan Province, China, Grant no. 2014TD0006, National Natural Science Foundation of China, Grant no. 61001086, the Fundamental Research Funds for the Central Universities, Grant no. ZYGX2011X004, the Overseas Academic Training Funds, University of Electronic Science and Technology of China (OATF, UESTC), Grant no. 20150607501, and the Program for Science and technology support in Sichuan Province, Grant no. 2014GZ0100.

References

[1] E. Bogdani, D. Vouyioukas, N. Nomikos et al., “A hybrid lateration-time-fingerprint position estimation technique for indoor UWB systems,” in Proceedings of the IEEE 19th International Workshop on Modeling and Design of Communication Links and Networks (CAMAD ‘14), pp. 350–354, IEEE, Athens, Greece, December 2014.

[2] C. A. Bertocnini, K. E. Rudder, B. D. Nousain et al., “System and method for radio-frequency fingerprinting as a security layer in RFID devices,” U.S. Patent 8810404B2, 2014.

[3] P. Lin, Q. Li, Q. Fan, X. Gao, and S. Hu, “A real-time location-based services system using WiFi fingerprinting algorithm for safety risk assessment of workers in tunnels,” Mathematical Problems in Engineering, vol. 2014, Article ID 371456, 10 pages, 2014.

[4] X. Wang, J. Qiu, S. Ye, and G. Dai, “An advanced fingerprint-based indoor localization scheme for WSNs,” in Proceedings of the 9th IEEE Conference on Industrial Electronics and Applications (ICIEA ‘14), pp. 2164–2169, IEEE, Hangzhou, China, June 2014.

[5] Z. Gu, Z. Chen, Y. Zhang et al., “Reducing fingerprint collection for indoor localization,” Computer Communications, 2015.

[6] Y. Fang, Z. Deng, C. Xue et al., “Application of an improved K nearest neighbor algorithm in WiFi indoor positioning,” in China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III, vol. 342 of Lecture Notes in Electrical Engineering, pp. 517–524, Springer, Berlin, Germany, 2015.

[7] G. Chen, Y. Zhang, L. Xiao, J. Li, L. Zhou, and S. Zhou, “Measurement-based RSS-multipath neural network indoor positioning technique,” in Proceedings of the IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE ‘14), pp. 1–7, IEEE, Toronto, Canada, May 2014.
[8] Y. Qin, Z. Ye, Q. Feng, and J. Zhang, “A WIFI positioning system based on an ensemble neural network algorithm,” in *Proceedings of the International Conference on Electrical Engineering and Information Technology (ICEEICT ’14)*, vol. 63, p. 175, 2014.

[9] R. Zouari, R. Zayani, and R. Bouallegue, “Indoor localization based on feed-forward Neural Networks and CIR fingerprinting techniques,” in *Proceedings of the IEEE Radio and Wireless Symposium (RWS ’14)*, pp. 271–273, IEEE, Newport Beach, Calif, USA, January 2014.

[10] W. Le, Z. Wang, J. Wang, G. Zhao, and H. Miao, “A novel WIFI indoor positioning method based on genetic algorithm and twin support vector regression,” in *Proceedings of the 26th Chinese Control and Decision Conference (CCDC ’14)*, pp. 4859–4862, IEEE, Changsha, China, May-June 2014.

[11] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[12] N. Lopes and B. Ribeiro, “Towards adaptive learning with improved convergence of deep belief networks on graphics processing units,” *Pattern Recognition*, vol. 47, no. 1, pp. 114–127, 2014.

[13] O. Gahabi and J. Hernando, “Deep belief networks for i-vector based speaker recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’14)*, pp. 1700–1704, Florence, Italy, May 2014.

[14] L. Deng, M. L. Seltzer, and D. Yu, “Binary coding of speech spectrograms using a deep auto-encoder,” in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH ’10)*, pp. 1692–1695, Chiba, Japan, September 2010.

[15] V. T. Tran, F. Althobiani, and A. Ball, “An approach to fault diagnosis of reciprocating compressor valves using Teager-Kaiser energy operator and deep belief networks,” *Expert Systems with Applications*, vol. 41, no. 9, pp. 4113–4122, 2014.

[16] H. Lee, R. Grosse, and R. Ranganath, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML ’09)*, pp. 609–616, ACM, Montreal, Canada, June 2009.

[17] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, “Time series forecasting using a deep belief network with restricted Boltzmann machines,” *Neurocomputing*, vol. 137, pp. 47–56, 2014.

[18] S. Lange and M. Riedmiller, “Deep auto-encoder neural networks in rein-forcement learning,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN ’10)*, pp. 1–8, Barcelona, Spain, July 2010.

[19] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, “Auto-encoder bottleneck features using deep belief networks,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’12)*, pp. 4153–4156, Kyoto, Japan, March 2012.

[20] A. Muqaibel, A. Safaai-Jazi, A. Bayram, and S. Riad, “Ultra wideband material characterization for indoor propagation,” in *Proceedings of the IEEE International Symposium on Antennas and Propagation*, pp. 623–626, IEEE, Columbus, Ohio, USA, June 2003.