sensors

MDPI

Article

A Large-Scale Multi-Hop Localization Algorithm Based on Regularized Extreme Learning for Wireless Networks

Wei Zheng [1], Xiaoyong Yan [2,*], Wei Zhao [1] and Chengshan Qian [3]

[1] School of Computer Engineering, Jinling Institute of Technology, Nanjing, 211169, China; zhengwei@jit.edu.cn (W.Z.); zhaowei@jit.edu.cn (W.Z.)

[2] School of Modern Post & Institute of Modern Posts, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

[3] School of Computer & Software, Nanjing University of Information Science and Technology, Nanjing 210044, China; qianchengshan@nuist.edu.cn

* Correspondence: xiaoyong_yan@njust.edu.cn; Tel.: +86-25-8618-8528

Received: 20 November 2017; Accepted: 18 December 2017; Published: 20 December 2017

Abstract: A novel large-scale multi-hop localization algorithm based on regularized extreme learning is proposed in this paper. The large-scale multi-hop localization problem is formulated as a learning problem. Unlike other similar localization algorithms, the proposed algorithm overcomes the shortcoming of the traditional algorithms which are only applicable to an isotropic network, therefore has a strong adaptability to the complex deployment environment. The proposed algorithm is composed of three stages: data acquisition, modeling and location estimation. In data acquisition stage, the training information between nodes of the given network is collected. In modeling stage, the model among the hop-counts and the physical distances between nodes is constructed using regularized extreme learning. In location estimation stage, each node finds its specific location in a distributed manner. Theoretical analysis and several experiments show that the proposed algorithm can adapt to the different topological environments with low computational cost. Furthermore, high accuracy can be achieved by this method without setting complex parameters.

Keywords: larger-scale wireless multi-hop localization; regularized extreme learning; machine learning

1. Introduction

An important issue of concern after monitoring an event or information is the location of the event or the collection of information in a variety of wireless network applications [1–3], such as the area of the enemy's movement on the battlefield, the location of forest fires, etc., which is the foundation of further measures and decision-making. Due to cost, lack of indoor signals and other reasons, it is difficult to lay GPS localization modules on wireless nodes. Therefore, the wireless network needs a localization algorithm which conforms to its own characteristics, and obtains the exact location of the unknown node through communication between the nodes.

Depending on the different focus, localization algorithms can be divided into various types [4,5], such as, on the basis of whether it is necessary to measure the distance information between nodes, localization algorithms can be divided into range-based and range-free; or according to the localization process, localization algorithms can be divided into centralized computing and distributed computing. In the practical application, various demands invoke different algorithms, in the light of their quality, stability, and efficiency. The range-free algorithm is mainly based on the connectivity of the network and the information fed back by the neighboring nodes. The unknown node calculates its own estimated position. The range-free algorithm has advantages of low network construction cost and small communication cost, and distributed computing has advantages of small computational burden and insensitivity to the network scale. If the complex environment and the node size of the monitoring area are taken into account, distributed computing can better adapt to the real needs. Therefore, the range-free algorithm is highly suitable for large-scale wireless network localization [6]. However, as with many key technologies, the range-free multi-hop localization algorithm is still plagued by technical problems in practical application. The most problematic is that multi-hop localization can only be achieved in an isotropic network with high node density and uniform distribution, while in the anisotropy network such as nonuniform distribution of nodes or sparse, the localization effect is very poor [4,7–9].

Anisotropy indicates that the quality of physical properties has different values when measured along axes in different directions. Anisotropy is most easily observed in the wireless network. As shown in Figure 1, we use d_{AB} (the dashed line) to represent the physical distance between nodes A and B in the network, and use h_{AB} (the straight line with arrows) to indicate the approximate length of the shortest hop-counts between nodes A and B. When the nodes are isotropic and uniformly distributed in the network, we can see that $d_{AB} \propto h_{AB}$; however, when there is an "obstacle" that affects the signal transmission in the network, $d_{AB} << h_{AB}$. Obviously, if no "obstacle" is present in the network, the physical distance between the nodes is proportional to the hop-count between them, that is $d_{AB} \propto h_{AB}$. When there is an obstacle in the network, this proportional relationship is no longer true. This paper focuses on handling anisotropic network problems in large scale multi-hop localization.

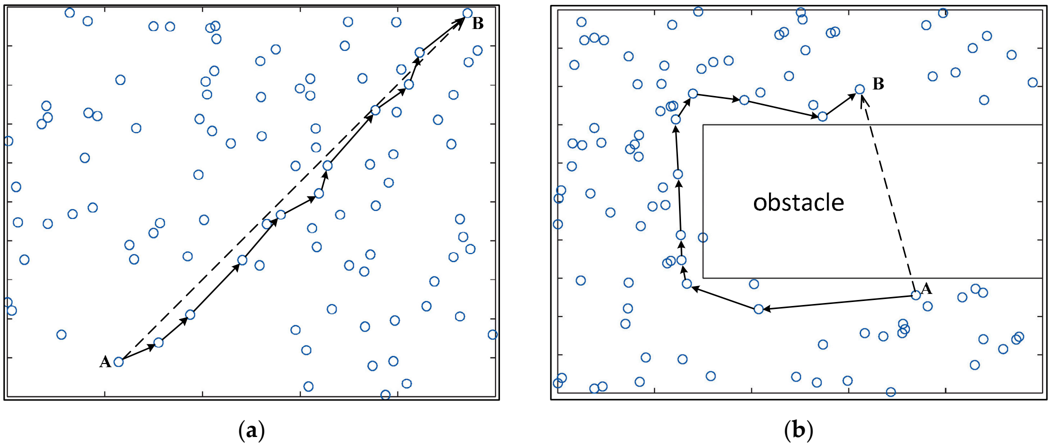


Figure 1. Different distribution of nodes (**a**) Isotropic network; (**b**) Anisotropic network.

For the problems existing in the multi-hop localization, this paper presents a novel multi-hop localization method, called Multi-hop Localization though Regularized Extreme Learning Machine (ML-RELM), which is more suitable for practical applications because of its relatively lower computational complexity, higher localization accuracy, and stronger adaptability. The ML-RELM method aims to improve the performance of the localization algorithm and to adapt it to different environments by using the topology of the wireless network and its own characteristics.

2. Related Work

The most well-known multi-hop distributed localization algorithm is the DV-hop proposed by Niculescu and Nath [10]. In the DV-hop method, the distances between the locating nodes to the anchor nodes can be represented by the product of network average hop distance and the hop-counts between them, then to obtain the location information via trilateration. For example, the average hop distance of anchor i can be calculated by Equation (1):

$$c_i = \frac{\sum_{i \neq j} d_{ij}}{\sum_{i \neq j} h_{ij}} \tag{1}$$

where d_{ij} is the physical distance between anchor node i and anchor node j, and h_{ij} is the corresponding shortest hop-count. The DV-Hop algorithm is simple and easy to implement. The whole localization process does not rely on additional hardware devices. However, its adaptability to the environment is poor, therefore the applications are limited.

By virtue of machine learning methods, modeling and designing the localization mechanism has become a hot topic in recent years [11,12]. Based on the correspondence between the distributed characteristics and the measured information of the known nodes, a mapping model can be learned. Then, this model is used to estimate the position of the unknown nodes. Compared with the previous work, such model can effectively find out the intrinsic network topology, the relevance of observations and other information hidden in large-scale data, which then effectively improve the localization performance. Lim et al. [9] proposed a Proximity Distance Map (PDM) localization algorithm using Truncated Singular Value Decomposition (TSVD) technology to model hop-counts and physical distance [13]. The PDM method first records the physical distance and the measured distance between the known nodes with the matrix respectively. Secondly, it transforms the two matrices linearly through the TSVD method to obtain an optimal linear transformation model. Thirdly, it takes the shortest hop-count from unknown nodes to anchor nodes into the model. Finally, the physical distance of unknown nodes to anchor nodes is estimated. The TSVD is essentially a linear multivariate rule by which the estimated distance obtained is the weighted sum of the estimated values of the known nodes associated with the unknown nodes, and thus the estimated values are closer to the true values. In addition, TSVD drops the smaller singular values. To a certain extent, this operation reduces the impact of noise in the conversion process, so as to enhance the stability of the algorithm. However, the PDM method ignores the problem of scale conversion between the hop-counts matrix and the physical distances matrix, which can cause data loss. In addition, it does not offer a choice for TSVD clip coefficients. Accordingly, an excessive coefficient may easily cause useful information lost, while a too small clip coefficient would retain too much high-frequency noise.

Recently, inspired by the PDM method, Lee and others proposed multi-hop range-free localization methods based on SVM [14,15], namely, Localization through Support Vector Regression (LSVR) and Localization through Multi-dimensional Support Vector Regression (LMSVR) [7,8]. These two methods suggest that there is a nonlinear relationship between the hop-counts and the physical distances. They use a regression based on the kernel technology SVM to predict the distance between nodes. However, according to Occam's razor principle, the more complex the model, the higher the probability of over-fitting [16]. Therefore, the localization method based on SVM algorithm has the over-fitting phenomenon with the increase of the number of anchor nodes, which resulted in the reduction of the localization accuracy. In addition, SVM-based methods are prone to excessive computational complexity.

The Xiao team from Hong Kong Polytechnic University put forward another strategy. They proposed a series of distributed multi-hop localization algorithm, such as Pattern-Driven localization Scheme (PDS) [17], Reliable Anchor-based Localization (RAL) [18], Anchor Supervised (AnSuper) [19], and Selective Multilateration (SM) [20]. Based on the anchor node information, the family of algorithms divides the deployment area into a number of uniformly distributed set of nodes. Taking the SM algorithm as an example, it uses the angle between the anchor nodes to measure the distance between the anchor nodes and divide the area. However, the SM method also requires human intervention in the selection of the number of divided regions and the regional center nodes. Figure 2 shows the localization results of the SM algorithm, which selects different regional centers and the number of regions. The figure randomly distributes 400 nodes (as shown in Figure 2a), of which 40 are anchor nodes (shown in Figure 2b). Figure 2c,e,g are the anchor nodes partition map, and Figure 2d,f,h are the corresponding localization results. Figure 2d,h are centered on node 12 and the nodes are divided into three and two areas, respectively. Figure 2f is centered on node 17 and the area is divided into three regions. It is easy to find, with different nodes as the center and divided into different numbers of regions obtained by localization, that results are not the same, which confirms the instability of the algorithm.

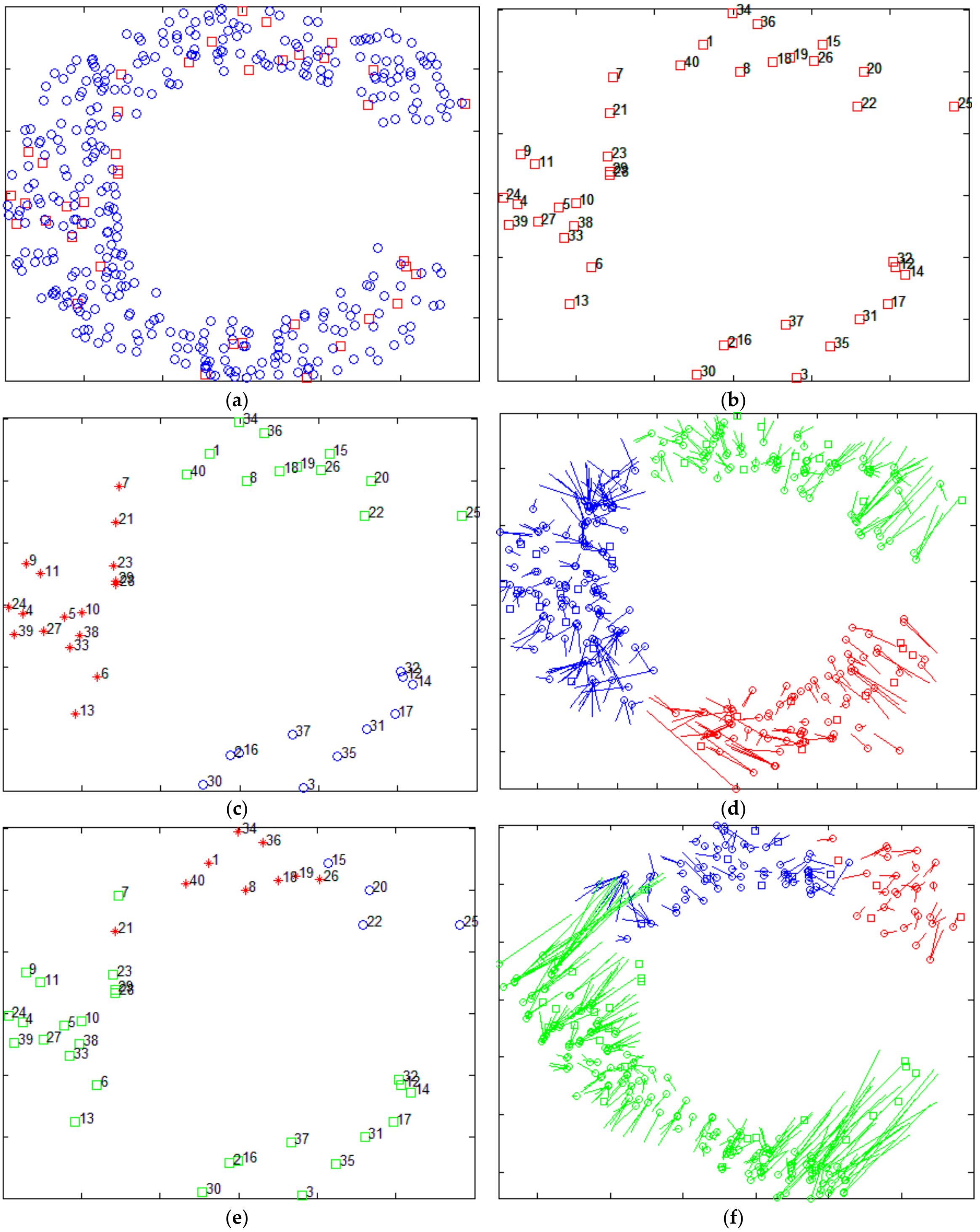
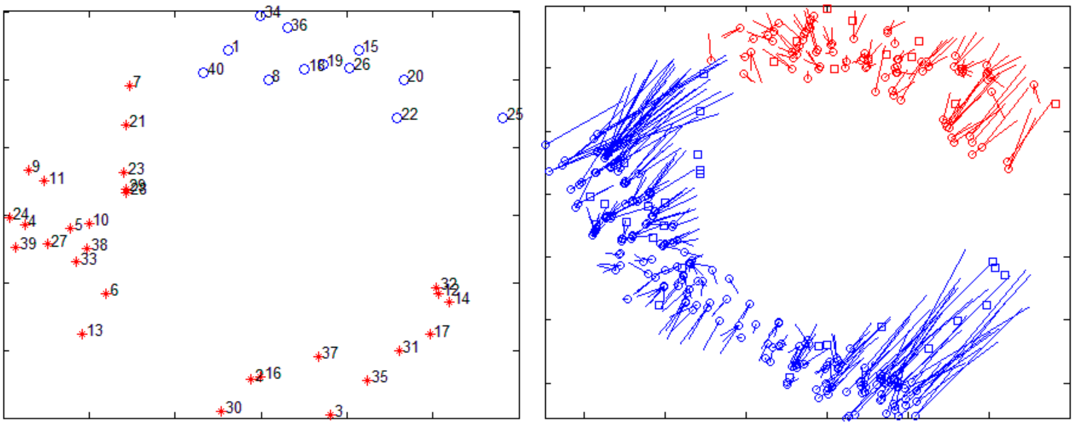


Figure 2. *Cont.*



(**g**) (**h**)

Figure 2. The localization results of SM for different regional center nodes and for different number of regions (**a**) The distribution of nodes; (**b**) The distribution of anchor nodes; (**c**) The center node is #12, area is divided into three parts; (**d**) The localization result of SM corresponds to the Figure 2c; (**e**) The center node is #17, area is divided into three parts; (**f**) The localization result of SM corresponds to the Figure 2e; (**g**) The center node is #12, area is divided into two parts; (**h**) The localization result of SM corresponds to the Figure 2g.

The latest study based on the number of packets between nodes and estimated distance residuals, uses Multi-objective Optimization to carry out the node position estimation [21,22]. But the bionic methods employed in the localization algorithm use pseudo-random number, so the results are not same each time. In addition, the optimal solution requires multiple iterations, which affects the efficiency of the algorithm.

The Extreme Learning Machine (ELM) is a new learning method of single-hidden-layer feedforward neural networks (SLFNs) proposed by Huang [23]. In the ELM, the input weights (connecting the input nodes to the hidden nodes) and the hidden layer nodes are randomly selected, and the output weights (connecting to the output node of hidden layer node) are determined by calculating the Moore-Penrose (MP) generalized inverse of the hidden layer output matrix. This learning algorithm ensures the generalization performance of the network, and significantly enhances the convergence as well. Moreover, the additional consumption on the tunes for learning rates, and the curses of local minimizers can be avoided [24]. Because of satisfactory precision and high efficiency, ELM has a very wide application. In this paper, a new distributed multi-hop method, ML-RELM, is proposed by combining ELM and distributed multi-hop localization algorithm. This method constructs the optimal relation (model) by the information of the actual distance and the number of inter-node hops, and then uses this model to predict the distance between the unknown node and the known node.

3. Regularized Extreme Learning

Huang et al. [23] proposed in 2006 a learning method based on Single Hidden Layer Feedforward Neural Networks (SLFNs) that is an Extreme Learning Machine. The main idea is to randomly assign the input weight and hidden layer deviation, to set the hidden layer of the number of nodes, and then through the least squares method to determine the value of the output weight. ELM algorithm overcomes the shortcoming of local minimization and over-fitting in traditional gradient-based neural network algorithms (such as BP algorithm). Thus, the ELM is a quick speed and less intervention single hidden layer feedforward neural network. It has excellent ability of regression. In addition, ELM does not need to manually set and adjust the parameters of the classifier, the completion of the entire process without iteration. Comparing to traditional learning methods, it is easier to deploy and implement. The basic principle of the ELM is shown in Figure 3.

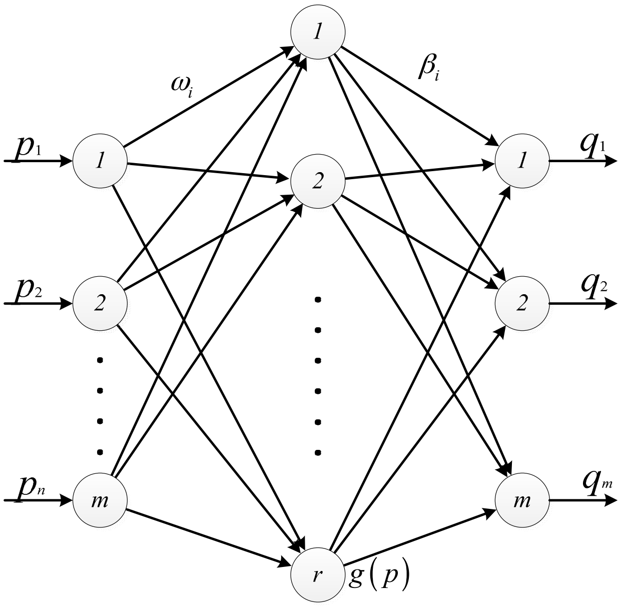


Figure 3. The schematic diagram of ELM.

Figure 3 shows a single hidden layer feedforward neural network. Among the network there are *m* input nodes, *k* hidden layer nodes, *m* output nodes; $\boldsymbol{\omega}_i$ is the input weight vector of the *i* hidden layer nodes, which represents the weight input of the *j* input node to the *i* hidden node; b_i is the deviation of the *i* hidden layer node (or the influence factor); $\boldsymbol{\beta i}$ is the weight value of the *i* hidden node output to the *j* output node; $g(p)$ is the activation function. For the activation function, Huang et al. have proved that it is an infinitely differentiable non-constant function. ELM has the characteristic of the full rank, thus it guarantees the accuracy of $\boldsymbol{\omega}$ and $\boldsymbol{b}$ randomly assigned.

For any given input data set $(\boldsymbol{U},\boldsymbol{V}) = \{(\boldsymbol{u}_1,\boldsymbol{v}_1),(\boldsymbol{u}_2,\boldsymbol{v}_2) \ldots (\boldsymbol{u}_n,\boldsymbol{v}_n)\}$, the input of the *i* sample is $\boldsymbol{u}_i = [u_{i1},u_{i2}, \ldots ,u_{im}]^{\mathrm{T}} \in R^m$, and the output is $\boldsymbol{v}_i = [v_{i1},v_{i2}, \ldots ,v_{im}]^{\mathrm{T}} \in R^m$. When the *i* sample is input to the ELM, the input of the *j* hidden layer node is shown as Equation (2):

$$\begin{aligned} \boldsymbol{\omega}_j\boldsymbol{u}_i &= \left(\omega_{j1}, \omega_{j2}, \cdots, \omega_{jn}\right)\begin{pmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{im} \end{pmatrix} \\ &= \omega_{j1}u_{i1} + \omega_{j2}u_{i2} + \cdots + \omega_{jm}u_{im}. \end{aligned} \tag{2}$$

After the processing of the *j* hidden layer node, the output is $g\left(\boldsymbol{\omega}_j\boldsymbol{u}_i + b_j\right)$. At this time, the output matrix with *r* hidden nodes is:

$$\boldsymbol{G}(\boldsymbol{u}_i) = (g(\boldsymbol{\omega}_1\boldsymbol{u}_i + b_1), g(\boldsymbol{\omega}_2\boldsymbol{u}_i + b_2), \cdots, g(\boldsymbol{\omega}_r\boldsymbol{u}_i + b_r))^T \tag{3}$$

According to the hidden layer output matrix $\boldsymbol{G}(\boldsymbol{X}) = (\boldsymbol{G}(\boldsymbol{u}_1), \ldots, \boldsymbol{G}(\boldsymbol{u}_r))$ and the output weight matrix $\boldsymbol{\beta}$, the output matrix $\boldsymbol{V}$ can be approximated as:

$$\begin{aligned} &\boldsymbol{G}(\boldsymbol{u}_i)^T\boldsymbol{\beta} = \boldsymbol{G}(\boldsymbol{u}_i)^T(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m) \\ &= (g(\boldsymbol{\omega}_1\boldsymbol{u}_i + b_1), g(\boldsymbol{\omega}_2\boldsymbol{u}_i + b_2), \cdots, g(\boldsymbol{\omega}_r\boldsymbol{u}_i + b_r))\begin{pmatrix} \beta_{11} & \cdots & \beta_{m1} \\ \vdots & \ddots & \vdots \\ \beta_{1r} & \cdots & \beta_{mr} \end{pmatrix} \\ &= (o_{i1}, \cdots, o_{im}), \end{aligned} \tag{4}$$

where element β_{ij} of the output matrix $\boldsymbol{\beta}$ represents the impact of the *j*-th node on the physical distance of the *i*-th node. The diagonal element β_{ii} of the output matrix $\boldsymbol{\beta}$ can be regarded as the scaling factor when the hop-count is converted to distance. Therefore, the distance from the unknown node to the anchor node obtained by the extreme learning method can be seen as the weighted sum of the hop-counts with all anchor nodes.

The activation function of the ELM is infinitely differentiable and can infinitely approximate to the output value, which means the estimated output value with the zero error. Then there are suitable, that make $\sum_{j=1}^{m} \left| o_i - t_{ij} \right| = 0$. So the least squares method can be used to obtain the output matrix $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} = \boldsymbol{G}^{\dagger}\boldsymbol{V}, \tag{5}$$

where $\boldsymbol{G}^{\dagger}$ is the Moore-Penrose (MP) generalized inverse matrix of the hidden weight output matrix $\boldsymbol{G}$, and $\boldsymbol{V}$ is the output matrix.

Furthermore, for multi-hop localization, the element of the hop-count matrix is mostly integer, which can cause the linear correlation between the vectors of the hop-count matrix, and the ill-conditioned problem may occur when solving the generalized inverse matrix $\boldsymbol{G}^{\dagger}$ (i.e., $\boldsymbol{G}^{\dagger} = (\boldsymbol{G}^{T}\boldsymbol{G})^{-1}\ \boldsymbol{G}^{T}$). Therefore, we add a positive matrix $k\boldsymbol{I}$ ($k > 0$) as the regular term to the matrix $(\boldsymbol{G}^{T}\ \boldsymbol{G})^{-1}$, so that the result $(\boldsymbol{G}^{T}\boldsymbol{G} + k\boldsymbol{I})^{-1}$ will be not singular. It effectively improves ELM's ability to deal with regression problems of linearly related variables, where k is a regular parameter, and $\boldsymbol{I}$ is the unit matrix.

4. Distributed Multi-Hop Localization Based on Regularized Extreme Learning

The classical large-scale multi-hop localization methods assume that the network is isotropic and evenly distributed, so the hop-counts can correspond to the physical distance. Unfortunately, in practice, networks often present anisotropy. If a network is anisotropic, the hop-counts between nodes may not match the physical distances well. Using fixed coefficients (such as the DV-hop algorithm) to match the hop-counts and the physical distances will produce a greater error. Therefore, this match should be related to the direction and position. In order to solve this problem, we use regularized extreme learning to construct the model between the hop-counts and the physical distances, representing the anisotropy of the network. With the help of regularized extreme learning, the model describes the optimal linear transformation between hops and physical distances. With the help of the proposed algorithm, an unknown node can get a more accurate distance conversion, resulting in better position estimation.

We aim to design a learning machine which transforms the hop-counts to the physical distances. Therefore, we utilize the supervised information (hop-counts and physical distances between anchors) to train a model, including hop-counts and physical distances of the anchor nodes. Then the learned model can be used to predict the physical distances via given hop-counts (from the unknown node to the anchor node). The pipe line of our algorithm is summarized in Figure 4.

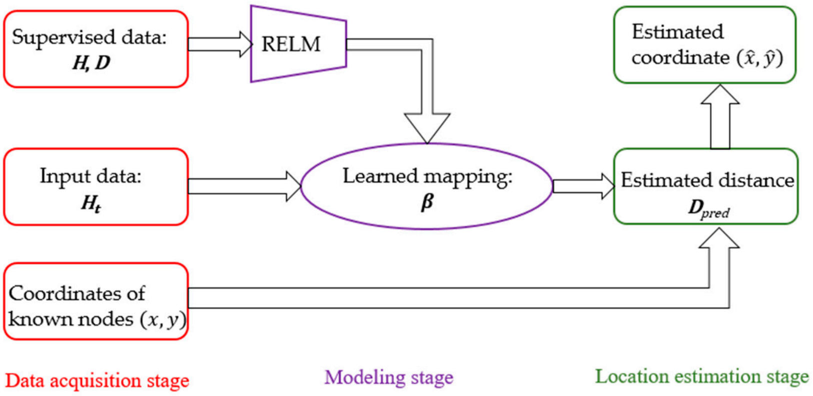


Figure 4. The framework of proposed algorithm. The mapping is firstly trained by REML, using supervised data consist of the known hop-counts and physical distances. After that, in the testing phase, the physical distances of the unknown node are predicted by the learned mapping.

The localization of nodes based on ML-RELM method can be divided into three stages: data acquisition, modeling and location estimation. In the data acquisition stage, the program collects the hop-counts and physical distance information between nodes and provides them to the modeling stage. In the modeling stage, the mapping relation between the hop-counts and the physical distances are trained by the hop-counts and physical distances information collected in the data acquisition stage. In the location estimation stage, the distance from the unknown node to the known node is estimated by using the mapping model obtained in the training stage, and the location of the unknown node can be calculated. Details are as follows.

4.1. Data Collection

Suppose that n nodes are distributed in a two-dimensional plane. The first m ($m << n$) anchor nodes $\{S_i\}_{i=1}^{m}$ are the known self-coordinate information, and the remaining $n-m$ nodes $\{S_i\}_{i=m+1}^{n}$ are the normal nodes waiting to acquire the location information. When the program begins, any anchor node in the area firstly uses the distance vector routing protocols to broadcast a "HELLO" packet to the remaining nodes. After a period of time, the shortest hop-counts vector $\boldsymbol{h}_i = [h_{i1},h_{i2}, \ldots ,h_{im}]^{\mathrm{T}}$ of the anchor node S_i to the remaining anchor nodes can be obtained, and the shortest hop-counts vector $\boldsymbol{h}_k = [h_{k1},h_{k2}, \ldots ,h_{km}]^{\mathrm{T}}$ from the unknown node $\{S_i\}_{i=m+1}^{n}$ to the anchor node can be obtained. Thus, the shortest hop-counts matrix corresponding to the anchor node and the unknown node is $\boldsymbol{H} = [\boldsymbol{h}_1,\boldsymbol{h}_2, \ldots ,\boldsymbol{h}_m]$ and $\boldsymbol{H}_t = [\boldsymbol{h}_{m+1},\boldsymbol{h}_{m+2}, \ldots ,\boldsymbol{h}_n]$, respectively. The straight distance of the anchor nodes S_i to S_j can be expressed by Equation (6):

$$\begin{aligned} d(S_i,S_j) &= d_{ij} = \|cor(S_i) - cor(S_j)\|_2 \\ &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \end{aligned} \tag{6}$$

where $cor(S_i) = (x_i, y_i)$ is the coordinate of node S_i, and d_{ij} is the physical distance of the i-th node to the j-th node.

4.2. The Proposed Model

According to [9], it is clear that in an ideal case $d \propto \boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}$, when each $S_i \in \boldsymbol{B}$ collects the m pair of data pairs $\{\boldsymbol{h}_k, d_{i,k}\}_{k=1}^{m}$. Therefore, we assume that the input dimension of ELM is m, and the optimal relationship between the physical distance and the optimal relationship between the physical distance and the hop count can be obtained by minimizing the mean square error, that is:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{m} \|e_i\|_2^2 = \sum_{i=1}^{m} \|\boldsymbol{d}_i - \boldsymbol{\beta}_i \boldsymbol{G}(\boldsymbol{h}_i)\|_2^2 \tag{7}$$

where $\boldsymbol{d}_i = [\boldsymbol{d}_{i1},\boldsymbol{d}_{i2}, \ldots ,\boldsymbol{d}_{im}]^{\mathrm{T}}$ is a vector which presents the physical distance between the i-th anchor node and the remaining $\boldsymbol{m} - 1$ anchor nodes.

Based on the rule of minimum norm solution (ensure both $\|\boldsymbol{d}_i - \boldsymbol{\beta}_i \boldsymbol{G}(\boldsymbol{h}_i)\|_2^2$ and $\|\boldsymbol{\beta}_i\|_2^2$ minimum at the same time), Equation (7) exists a minimal norm least squares solution as follows:

$$\boldsymbol{\beta} = \boldsymbol{G}^{\dagger}\boldsymbol{D}, \tag{8}$$

where $\boldsymbol{G}^{\dagger}$ is the augmented inverse of the Moore-Penrose of the hidden layer response matrix and the i-th column of $\boldsymbol{G}$ is the output of the i-th hidden node with respect to the input hop vector $\boldsymbol{h}_i$. $\boldsymbol{G}^{\dagger}$ has a variety of calculation methods. In the extreme learning machine, the orthogonal method is often used for the calculation of $\boldsymbol{G}^{\dagger}$: when $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$ is not singular, $\boldsymbol{G}^{\dagger} = (\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}$; when $\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}}$ is not singular, $\boldsymbol{G}^{\dagger} = \boldsymbol{G}^{T}(\boldsymbol{G}^{T}\boldsymbol{G})^{-1}$.

Since the vector between the hops matrix is prone to correlation, we add a small positive number to the diagonal of $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$ or $\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}}$ when calculating $\boldsymbol{G}^{\dagger}$, making the whole system more stable. Then the hops-distance optimal relation model $\hat{\boldsymbol{\beta}}$ is obtained, e.g., the feedback of single hidden layer is completed. Finally, the program broadcasts the mapping model to each node in the network through distance vector protocol.

The model obtained by RELM actually stores the distance feature of all anchor nodes in all directions, so the model can accurately describe the anisotropic relationship between the hop-count and the physical distance of the nodes. Algorithm 1 summarizes the steps of the model building by RELM algorithm.

Algorithm 1. Localization Model Building by RELM.

```
1. Input: the shortest hop-counts matrix $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_m]$; the physical distance matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m]$,
the count of hidden nodes $r$.
2. For $i$ from 1 to $r$
3.      assign input weights $\boldsymbol{\omega}_i$ randomly;
4.      assign hidden layer bias $b_i$ randomly;
5. End
6. For $i$ from 1 to $m$
        calculate the output of hidden layer $\boldsymbol{G}(\boldsymbol{h}_i)$;
7. End
8. Calculate the inverse matrix $\boldsymbol{G}^{\dagger} = \left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + k\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}$.
9. Calculate $\hat{\boldsymbol{\beta}} = \boldsymbol{G}^{\dagger}\boldsymbol{D}$.
10. Output: the hidden layer weights $\hat{\boldsymbol{\beta}}$.
```

4.3. Location Estimation

For any unknown node, it can obtain the hop-counts vector $\boldsymbol{h}_k \subset \boldsymbol{H}_t$ and the mapping model $\hat{\boldsymbol{\beta}}$ of the anchor node in the data collection stage to estimate the linear distance from S_k to the anchor node, that is:

$$\boldsymbol{D}_{pred} = \boldsymbol{G}(\boldsymbol{H}_t)\hat{\boldsymbol{\beta}}. \tag{9}$$

Then, the unknown nodes combine the coordinates of the anchor nodes and the distance between them to obtain the final coordinates. That is solving the following equations group for$k \in \{m+1, m+2, \ldots, n\}$:

$$\begin{cases} (\hat{x}_k - x_1)^2 + (\hat{y}_k - y_1)^2 = {d_{k1}}^2 \\ \cdots \\ (\hat{x}_k - x_m)^2 + (\hat{y}_k - y_m)^2 = {d_{km}}^2 \end{cases}, \tag{10}$$

in which $d_{kj} = \boldsymbol{D}_{pred}(k, j)$ is the physical distance predicated by Equation (9). Above equations group is usually solved via the maximum likelihood estimation (MLE) of multiple-source location [25]. Specifically, the optimal solution to Equation (10) is expressed as:

$$\hat{\boldsymbol{c}}_k = (\hat{x}_k, \hat{y}_k) = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b}, \tag{11}$$

where the two projection matrices are:

$$\boldsymbol{A} = 2\begin{bmatrix} (x_1 - x_m)^T, & (y_1 - y_m)^T \\ \cdots & \cdots \\ (x_{m-1} - x_m)^T, & (y_{m-1} - y_m)^T \end{bmatrix}, \tag{12}$$

and:

$$\boldsymbol{b} = \begin{bmatrix} (x_1 - x_m)^2 + (y_1 - y_m)^2 + d_{km}^2 - d_{k1}^2 \\ \vdots \\ (x_{m-1} - x_m)^2 + (y_{m-1} - y_m)^2 + d_{km}^2 - d_{k1}^2 \end{bmatrix} \tag{13}$$

respectively. Finally the details of location estimation are shown in Algorithm 2.

Algorithm 2. Location estimation.

```
1. Input: the shortest hop-counts for unknown nodes to anchor nodes $\boldsymbol{H}_t = [\boldsymbol{h}_{m+1}, \boldsymbol{h}_{m+2}, \cdots, \boldsymbol{h}_n]_{m\times(n-m)}$;
   the coordinates of known nodes $\{cor(S_i) = (x_i, y_i)\}_{i=1}^{m}$.
2. Employ the learned weights of Algorithm 1 to predicate the distances $\boldsymbol{D} = \boldsymbol{G}(\boldsymbol{H}_t)\hat{\boldsymbol{\beta}}$.
3. **For** $k$ from $m+1$ to $n$
4.     calculate matrix $\boldsymbol{A}$ using Equation (12);
5.     calculate matrix $\boldsymbol{b}$ using Equation (13);
6.     calculate $\hat{\boldsymbol{c}}_k = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b}$;
7. **End**.
8. **Output:** estimated coordinates of unknown nodes $\{c\hat{o}r(S_k) = \hat{\boldsymbol{c}}_k = (\hat{x}_k, \hat{y}_k)\}_{k=m+1}^{n}$
```

5. Performance Evaluation

To analyze and evaluate the performance of the ML-RELM algorithm, we conducted simulation and realistic scenario experiments. The simulation experiments are carried out on a PC (equipped with Windows 7 OS, Intel Core I7 and 8 GB RAM), and all the codes are implemented in MatLab. Realistic scenario experiments are implemented in indoor and outdoor environments respectively.

In order to reduce the one-sidedness of the experimental results in the simulation experiments, each simulation experiment is carried out 50 times. Each node will be randomly re-distributed in the experimental area, taking 50 times average of the Root Mean Square (RMS) [26] as the basis for evaluation:

$$\text{RMS} = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}\left((\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2\right)} \tag{14}$$

where $(\hat{x}_i, \hat{y}_i)$ is the estimated coordinate of the node. (x_i, y_i) is the actual coordinate of the i node, and N_t is the number of nodes that can be located. In addition, our performance comparisons are implemented among those schemes, including the DV-hop, PDM, LSVR in this section. Since the PDM method needs to set the threshold of TSVD to discard the eigenvalue; the LSVR method needs to set the kernel parameters, the penalty coefficient C and the width of insensitive loss function ε. For the sake of fairness, in the experiment, the TSVD method abandons the eigenvectors whose eigenvalues are less than or equal to 2. The C and ε of the LSVR method follows the settings in reference [8], and the kernel function is chosen to be the Gaussian kernel function. The kernel function is related to the distance of the training samples, so the kernel parameter is set to 40 times the mean value of the training samples.

5.1. Simulation Experiments

Two strategies of the node deployment are considered in the experiment: regular deployment and random deployment. We assume that the nodes are uniformly deployed in the detected region. Our work focuses on anisotropy network. The letter-shaped topology is most commonly used to compare and verify the localization accuracy [4,7,8,18,21]. In order to give a fairly comparison with the classical methods, such as DV-hop, PDM and LSVR, we adopt the same network topologies in the simulation.

5.1.1. Regular Deployment

In this group of experiments, as shown in Figure 5, 405, 487 and 525 nodes are regularly deployed in the 600 × 600 C-shaped, Z-shaped and S-shaped areas.

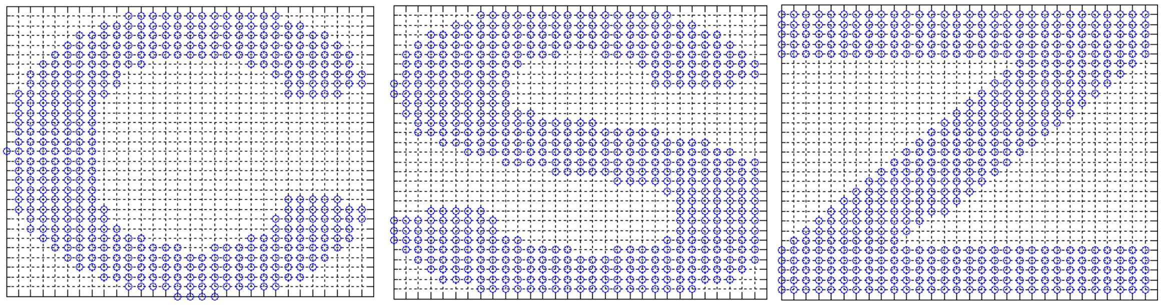


Figure 5. The regular distribution of nodes.

Experiments select 30 to 60 nodes as anchor nodes. First, we examine the final localization results of the two ML-RELM methods (the number of anchor nodes in Figure 5 is 40), where the circle represents the unknown node, and the square represents the anchor node. Then, as shown in the Figure 5, we use a straight line to connect the real coordinates of the unknown node and its estimated coordinates. The longer the line, the greater the estimation error of nodes.

It is obvious to observe from Figure 6 that the DV-hop method uses fixed hop-counts vs distances coefficients, and when there is an obstacle, the hop-count and the distance may deviate on converting. Although the PDM method can solve hop-distance deviation to a certain extent, it cannot deal with significantly different scales. For LSVR, the optimization of multiple parameters easily results in over-fitting (unknown nodes estimate an unknown trend to a curve). Nevertheless, estimation accuracy from the proposed ML-RELM is higher than from other localization methods.

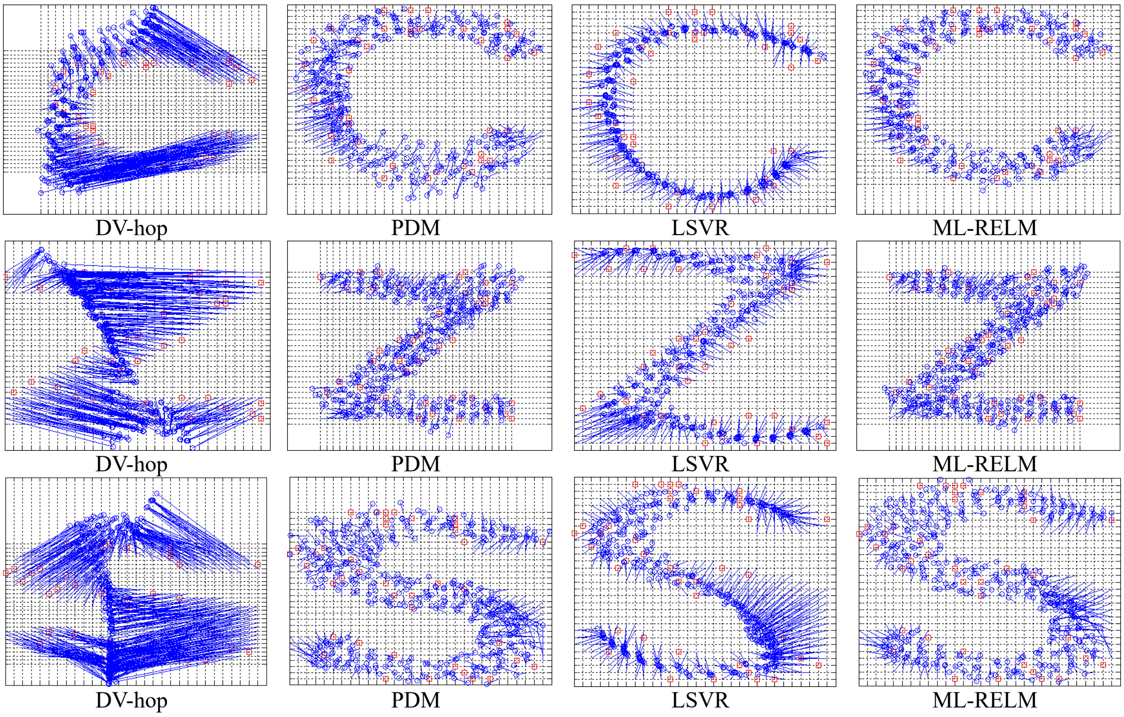


Figure 6. The localization results of regularly distributed nodes.

For each deployment area in the regular deployment network, Figure 7 shows the average RMS value for 50 experiments with 40 anchor nodes. The experimental results illustrated in Figure 7 show that the classical multi-hop localization method in the C-shaped, Z-shaped and S-shaped regions has low DV-hop performance, and the RMS values are 214.22, 169.75 and 246.65, respectively. The PDM and LSVR methods have better ability to obtain the relationship between the hop-counts and distances, so their performance is improved by 80.9% and 79.2% compared with the DV-hop method. ML-RELM has improved the prediction and fitting ability by using ELM as the core algorithm, so the ML-RELM localization performance is improved by 21.8% and 28.3% respectively compared with the PDM and LSVR methods.

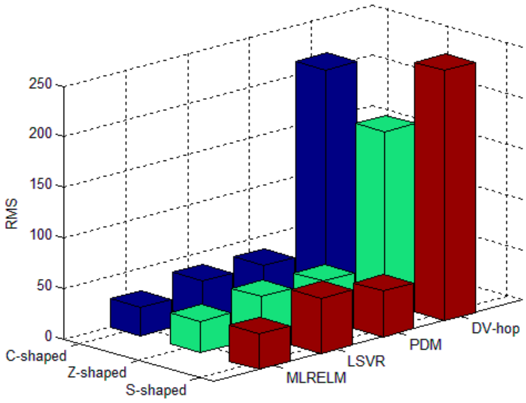


Figure 7. Average RMS for different algorithms in the regular deployment environment.

The ELM machine is very efficient, which only needs to calculate the weights (the computation complexity is $O\left(\min\left(L^3, N^3\right)\right)$). So the ELM is often in a very short period of time able to complete the operation. Table 1 shows that the running time of our proposed ML-RELM algorithm is compared with that of multiple multi-hop distributed localization algorithms in a variety of regular deployment environments of 40 anchor nodes. Clearly, ML-RELM is proposed to be more efficient than LSVR, DV-hop and PDM.

Table 1. Running time in the regular deployment.

CPU TIME (m)		DV-Hop	PDM	LSVR	ML-RELM
40 anchors, 365 unknown nodes (C-shaped)	**Average**	1.0917	1.1244	1.5707	1.1027
	Median	1.0782	1.1081	1.5586	1.0858
	Worst	1.2618	1.3289	1.679	1.2232
	Standard deviation	0.0517	0.0697	0.0445	0.0456
40 anchors, 447 unknown nodes (Z-shaped)	**Average**	2.2876	2.3103	2.7801	2.2796
	Median	2.3745	2.3779	2.8577	2.3926
	Worst	2.9239	2.7247	3.1042	2.5051
	Standard deviation	0.1730	0.1746	0.1933	0.1870
40 anchors, 485 unknown nodes (S-shaped)	**Average**	2.4172	2.4079	2.8511	2.3901
	Median	2.3745	2.3779	2.8577	2.3926
	Worst	2.9239	2.7247	3.1042	2.5051
	Standard deviation	0.1534	0.1063	0.1024	0.0696

Bars in Figure 8 represent RMS values and standard deviations of the four algorithms varying with the number of anchor nodes in the three regular deployment areas of the C, Z, and S shapes. It is easy to see that DV-hop is not suited to non-convex environment (there are obstacles in the network) localization. We also notice that the LSVR algorithm is superior to PDM when the number of anchor nodes is less than 40, and the PDM algorithm is better than LSVR. This is due to the increase in the number of anchor nodes, resulting in the over-fitting, making the performance of LSVR degraded (shown in Figure 6).

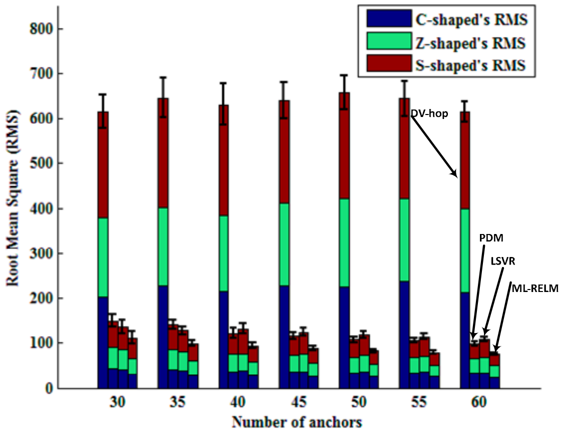


Figure 8. Comparison of different algorithms with different number of anchor nodes in the regular deployment.

5.1.2. Random Deployment

In this group of experiments, 400 nodes are randomly deployed in 600 × 600 two-dimensional square areas, due to the reasons of the block, the deployment shaped to C, S and Z-type (shown in Figure 9).

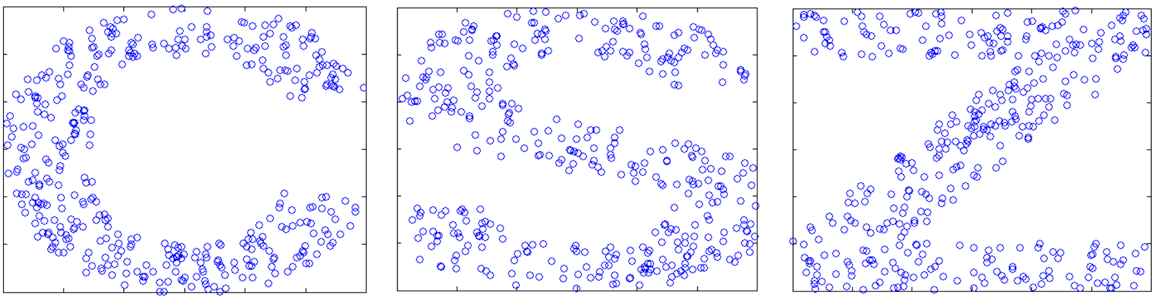


Figure 9. The random distribution of nodes.

As shown in Figure 10, we examine the final localization result of a set of ML-RELM methods (the number of anchor nodes is 40). It is clear from the figure that the proposed method performance is still superior to other localization methods.

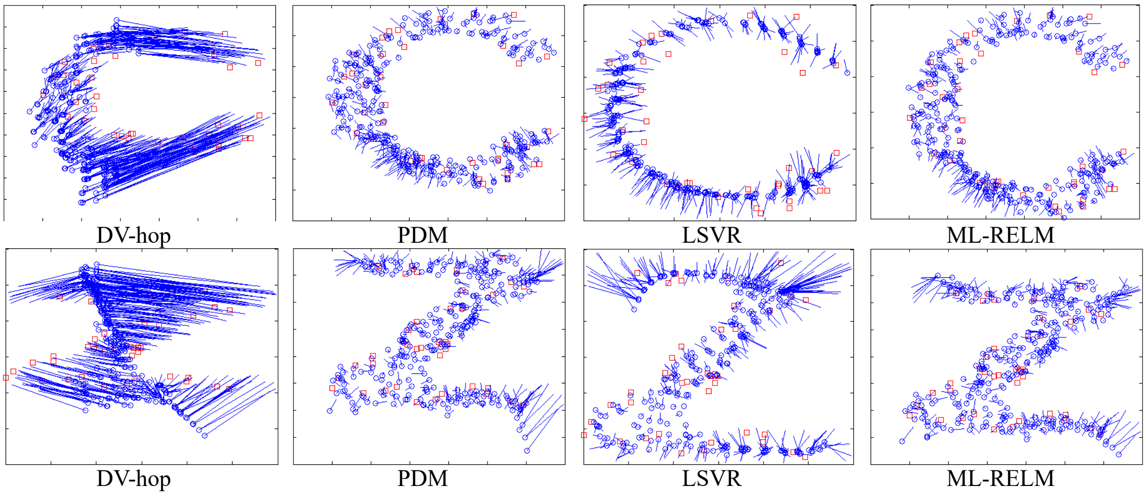


Figure 10. *Cont.*

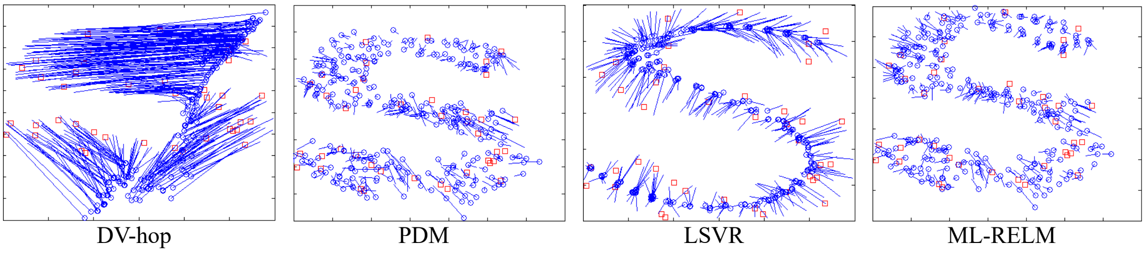


Figure 10. The localization results of randomly distributed nodes.

After 50 trials, the average RMS values of the four localization algorithms in the three type areas are shown in Figure 11. The proposed ML-RELM performance is 82.2%, 12.1%, and 20.5% higher than that of DV-hop, PDM, and LSVR respectively.

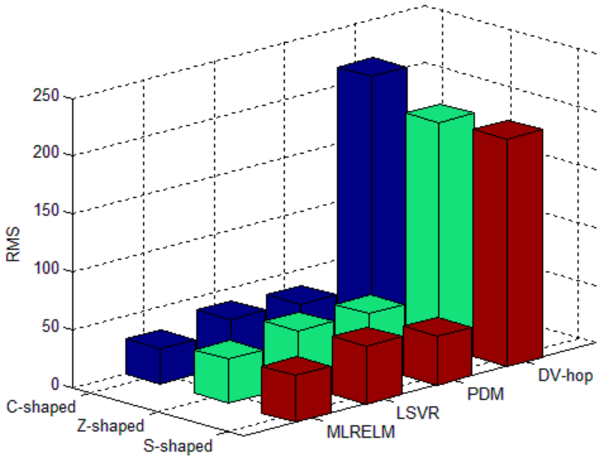


Figure 11. Average RMS for different algorithms in the random deployment environment.

Table 2 describes the running time of four algorithms in three random deployment environment. The ML-RELM method is more efficient and superior to the latest LSVR method.

Table 2. Running time in the random deployment.

CPU TIME (m)		DV-Hop	PDM	LSVR	ML-RELM
40 anchors, 360 unknown nodes (C-shaped)	**Average**	1.0561	1.09	1.5637	1.0852
	Median	1.0588	1.0854	1.5563	1.0855
	Worst	1.1127	1.1391	1.7832	1.1334
	Standard deviation	0.0245	0.0235	0.0719	0.0320
40 anchors, 360 unknown nodes (Z-shaped)	**Average**	1.0901	1.1048	1.5693	1.0913
	Median	1.0805	1.0981	1.5701	1.0839
	Worst	1.2112	1.1631	1.6103	1.1226
	Standard deviation	0.0376	0.0242	0.0189	0.0181
40 anchors, 360 unknown nodes (S-shaped)	**Average**	1.0201	1.0346	1.4501	1.0256
	Median	1.0185	1.0343	1.4793	1.0287
	Worst	1.0665	1.0657	1.5043	1.0518
	Standard deviation	0.0285	0.0185	0.1173	0.0219

Bars in Figure 12 represent RMS values and standard deviations of the four algorithms varying with the number of anchor nodes in the three random deployment areas of the C, Z, and S shapes. It is also obvious that ML-RELM has the best performance, and LSVR is degraded due to over-fitting when the anchor nodes numbers are large.

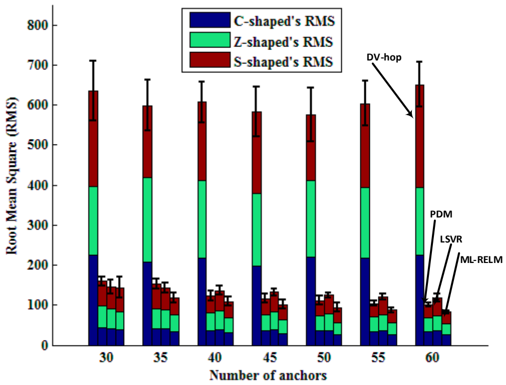


Figure 12. Comparison of different algorithms with different number of anchor nodes in the random deployment.

5.2. Realistic Scenario Experiments

In order to evaluate the performance of the proposed algorithm ML-RELM in realistic situations, we employ 20 nodes with the TI CC2530 chip which are deployed outside of buildings (as shown in Figure 13b) and inside of buildings (as shown in Figure 13e) respectively at the Jinling Institute of Technology. Figure 13a,d are satellite images of the test locations, respectively. The size of the outdoor and indoor test-beds is 30 m × 5 m and 30 m × 16 m, respectively. We set the communication radius of nodes equal to 6 m. Nodes 1#, 2#, 3#, 4#, 5# and 6# are selected as anchor nodes, and the remaining 14 nodes are normal nodes. Other parameters are similar to that in simulation experiments. Three previous methods: (1) The classic DV-hop algorithm [10]; (2) PDM [9]; and (3) LSVR [7,8] are compared.

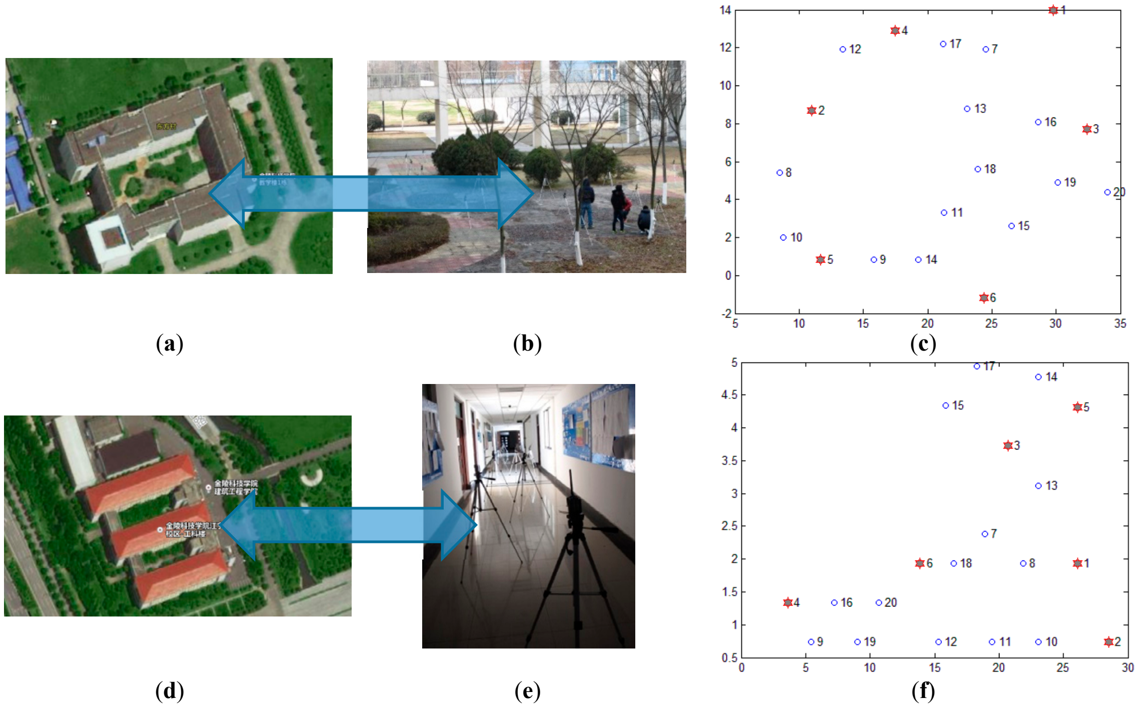


Figure 13. Nodes distribution under realistic scenarios (**a**) Satellite picture of outdoor location; (**b**) Picture of outdoor test-bed; (**c**) Topological graph of outdoor test-bed; (**d**) Satellite picture of indoor location; (**e**) Picture of indoor test-bed; (**f**) Topological graph of indoor test-bed.

As shown in Figure 14, the ML-RELM method proposed in this paper can achieve higher localization accuracy under these two conditions. The average RMS of ML-RELM is about 2.4 outdoor.

Compared with DV-hop, PDM and LSVR, the average performance of ML-RELM is improved by 17.93%, 12.78% and 51.55%, respectively. Besides, the average RMS of ML-RELM is about 2.3 indoor. Compared with DV-hop, PDM and LSVR, the average localization accuracy (measured by the RMS) is 61.27%, 39.99% and 45.58% higher than DV-hop, PDM and LSVR respectively.

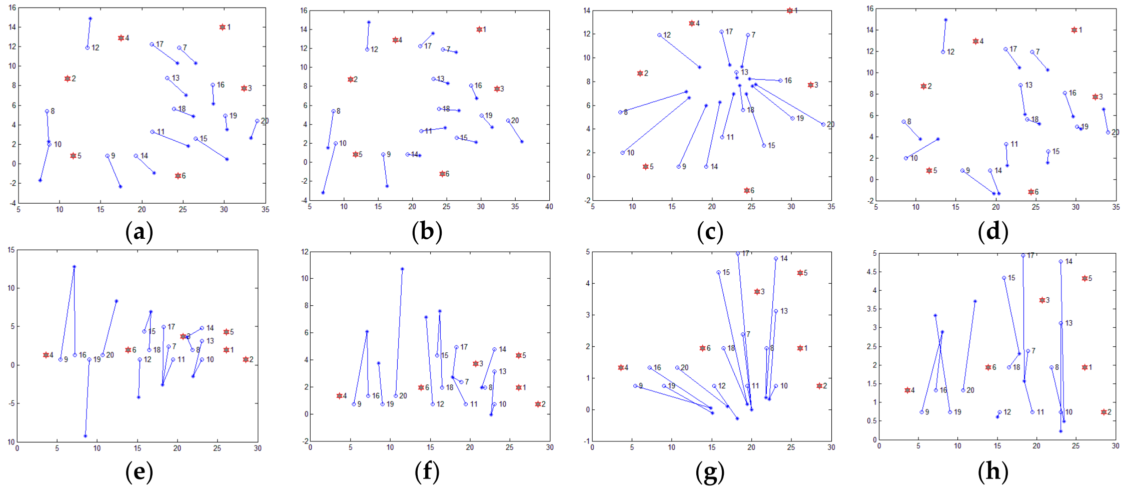
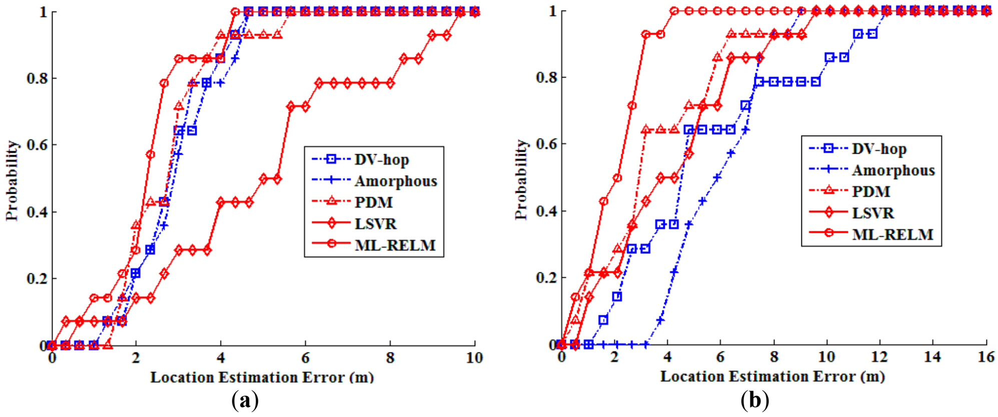

(**a**) (**b**) (**c**) (**d**)
(**e**) (**f**) (**g**) (**h**)

Figure 14. Results of location estimation in realistic scenarios (**a**) DV-hop in the outdoor, RMS = 3.1446; (**b**) PDM in the outdoor, RMS = 3.0085; (**c**) LSVR in the outdoor, RMS = 5.7151; (**d**) ML-RELM in the outdoor, RMS = 2.6829; (**e**) DV-hop in the indoor, RMS = 6.6422; (**f**) PDM in the indoor, RMS = 4.4114; (**g**) LSVR in the indoor, RMS = 5.0065; (**h**) ML-RELM in the indoor, RMS = 2.5197.

In these two group of realistic scenario experiments, we also study the Cumulative Distribution Function (CDF) of the localization errors. CDF shows the percentage of localized nodes whose localization error is smaller than some thresholds, thus tells us whether local damages have proliferated to other areas. Figure 15 shows the CDF of localization errors outdoor and indoor. We can find that when using ML-REML the localization performance is effectively improved. More than 71% nodes are localized with localization error less than 3 m outdoor, and more than 64% nodes are localized with localization error less than 3 m indoor.



(**a**) (**b**)

Figure 15. The Cumulative Distribution Function (CDF) of the errors in realistic scenarios (**a**) CDF of outdoor; (**b**) CDF of indoor.

6. Conclusions

A novel large-scale range-free localization algorithm for wireless networks is proposed in this paper. The proposed method is based on regularized extreme learning, which is a type of quick speed and less intervention single-hidden-layer feedforward neural network, one of the newest machine learning algorithms. In the proposed algorithm, each node estimates its location using the model by regularized extreme learning trained with anchor nodes. In this model, the location of the node is predicted by the information related to the shortest hop-counts of the node to all of the anchor nodes. The proposed algorithm is applied to the complex deployment environment. The experimental results show that the proposed algorithm is more efficient and accurate than the existing multi-hop localization methods in a variety of deployment environments.

Acknowledgments: The paper is supported by National Natural Science Foundation of China (No. 61375121), the China Postdoctoral Science Foundation under Grant 2016M601861, the Jiangsu Postdoctoral Research Funds under Grant 1701049A, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 17KJA520001, Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX17_0361, and the Open Project Program of Jiangsu Key Laboratory of Remote Measurement and Control under Grant YCCK201603.

Author Contributions: Wei Zheng and Xiaoyong Yan conceived and designed the experiments; Xiaoyong Yan performed the experiments; Wei Zheng and Xiaoyong Yan analyzed the data; Wei Zheng wrote the paper; Wei Zhao and Chengshan Qian modified the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, X.; Wang, P.; Niyato, D.; Kim, D.I.; Han, Z. Wireless Charging Technologies: Fundamentals, Standards, and Network Applications. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1413–1452. [CrossRef]
2. Naranjo, P.G.V.; Shojafar, M.; Mostafaei, H.; Pooranian, Z.; Baccarelli, E. P-SEP: A prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks. *J. Supercomput.* **2017**, *73*, 733–755. [CrossRef]
3. Cheng, H.; Xiong, N.; Yang, L.T.; Chen, G.; Zhuang, X.; Lee, C. Links organization for channel assignment in multi-radio wireless mesh networks. *Multimedia Tools Appl.* **2013**, *65*, 239–258. [CrossRef]
4. Yang, Z.; Wu, C.; Liu, Y. *Location-based Computing: Localization and Localizability of Wireless Networks*; Tsinghua University Press: Beijing, China, 2014; ISBN 978-7-302-35113-9.
5. Halder, S.; Ghosal, A. A survey on mobile anchor assisted localization techniques in wireless sensor networks. *Wirel. Netw.* **2016**, *22*, 2317–2336. [CrossRef]
6. Safa, H. A novel localization algorithm for large scale wireless sensor networks. *Comput. Commun.* **2014**, *45*, 32–46. [CrossRef]
7. Lee, J.; Choi, B.; Kim, E. Novel Range-Free Localization Based on Multidimensional Support Vector Regression Trained in the Primal Space. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 1099–1113. [CrossRef] [PubMed]
8. Lee, J.; Chung, W.; Kim, E. A new kernelized approach to wireless sensor network localization. *Inf. Sci.* **2013**, *243*, 20–38. [CrossRef]
9. Lim, H.; Hou, J.C. Distributed localization for anisotropic sensor networks. *ACM Trans. Sens. Netw. (TOSN)* **2009**, *5*, 1–26. [CrossRef]
10. Niculescu, D.; Nath, B. DV Based Positioning in Ad Hoc Networks. *Telecommun. Syst.* **2003**, *22*, 267–280. [CrossRef]
11. Alsheikh, M.A.; Lin, S.; Niyato, D.; Tan, H.-P. Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1996–2018. [CrossRef]
12. Wang, J.; Zhang, X.; Gao, Q.; Yue, H.; Wang, H. Device-Free Wireless Localization and Activity Recognition: A Deep Learning Approach. *IEEE Trans. Veh. Technol.* **2017**, *66*, 6258–6267. [CrossRef]
13. Hogben, L. *Handbook of Linear Algebra*; CRC Press: Boca Raton, FL, USA, 2007; ISBN 9781138199897.
14. Gu, B.; Sheng, V.S.; Tay, K.Y.; Romano, W.; Li, S. Incremental Support Vector Learning for Ordinal Regression. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 1403–1416. [CrossRef] [PubMed]
15. Gu, B.; Sheng, V.S. A Robust Regularization Path Algorithm for ν-Support Vector Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 1241–1248. [CrossRef] [PubMed]
16. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*; Wiley-Interscience: Hoboken, NJ, USA, 2000; ISBN 978-0-471-05669-0.

17. Xiao, Q.; Xiao, B.; Cao, J.; Wang, J. Multihop Range-Free Localization in Anisotropic Wireless Sensor Networks: A Pattern-Driven Scheme. *IEEE Trans. Mob. Comput.* **2010**, *9*, 1592–1607. [CrossRef]
18. Xiao, B.; Chen, L.; Xiao, Q.; Li, M. Reliable Anchor-Based Sensor Localization in Irregular Areas. *IEEE Trans. Mob. Comput.* **2010**, *9*, 60–72. [CrossRef]
19. Liu, X.; Zhang, S.; Bu, K. A locality-based range-free localization algorithm for anisotropic wireless sensor networks. *Telecommun. Syst.* **2015**, *62*, 3–13. [CrossRef]
20. Zhang, S.; Liu, X.; Wang, J.; Cao, J.; Min, G. Accurate Range-Free Localization for Anisotropic Wireless Sensor Networks. *ACM Trans. Sens. Netw.* **2015**, *11*, 1–28. [CrossRef]
21. Shahzad, F.; Sheltami, T.R.; Shakshuki, E.M. DV-maxHop: A Fast and Accurate Range-Free Localization Algorithm for Anisotropic Wireless Networks. *IEEE Trans. Mob. Comput.* **2017**, *16*, 2494–2505. [CrossRef]
22. Shahzad, F.; Sheltami, T.R.; Shakshuki, E.M. Multi-objective optimization for a reliable localization scheme in wireless sensor networks. *J. Commun. Netw.* **2016**, *18*, 796–805. [CrossRef]
23. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
24. Zhu, Q.-Y.; Qin, A.K.; Suganthan, P.N.; Huang, G.-B. Evolutionary extreme learning machine. *Pattern Recognit.* **2005**, *38*, 1759–1763. [CrossRef]
25. Sheng, X.; Hu, Y.H. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Trans. Signal Process.* **2005**, *53*, 44–53. [CrossRef]
26. Yan, X.; Song, A.; Yang, Z.; Yang, W. An improved multihop-based localization algorithm for wireless sensor network using learning approach. *Comput. Electr. Eng.* **2015**, *48*, 247–257. [CrossRef]