sensors

MDPI

Article

GSOS-ELM: An RFID-Based Indoor Localization System Using GSO Method and Semi-Supervised Online Sequential ELM

Fagui Liu and Dexiang Zhong *

School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China; fgliu@scut.edu.cn

* Correspondence: csdexiangzhong@mail.scut.edu.cn; Tel.: +86-135-7020-5194

Received: 10 May 2018; Accepted: 18 June 2018; Published: 21 June 2018

Abstract: With the rapid development of indoor positioning technology, radio frequency identification (RFID) technology has become the preferred solution due to its advantages of non-line-of-sight, non-contact and rapid identification. However, the accuracy of existing RFID indoor positioning algorithms is easily affected by the tag density and algorithm efficiency, and their environmental robustness is not strong enough. In this paper, we have introduced an RFID positioning algorithm based on the Glowworm Swarm Optimization (GSO) fused with semi-supervised online sequential extreme learning machine (SOS-ELM), which is called the GSOS-ELM algorithm. The GSOS-ELM algorithm automatically adjusts the regularization weights of the SOS-ELM algorithm through the GSO algorithm, so that it can quickly obtain the optimal regularization weights under different initial conditions; at the same time, the semi-supervised characteristics of the GSOS-ELM algorithm can significantly reduce the number of labeled reference tags and reduce the cost of positioning systems. In addition, the online learning phase of the GSOS-ELM algorithm can continuously update the system to perceive changes in the environment and resist the environmental interference. We have carried out experiments to study the influence factors and validate the performance, both the simulation and testbed experiment results show that compared with other algorithms, our proposed GSOS-ELM localization system can achieve more accurate positioning results and has certain adaptability to the changes of the environment.

Keywords: radio frequency identification (RFID); indoor localization; Glowworm Swarm Optimization (GSO); semi-supervised online sequential extreme learning machine (SOS-ELM)

1. Introduction

With the development of Internet of Things technology, people's demand for applications has grown rapidly. Among these technologies, wireless location-aware technologies have shown great activity in both military and civilian applications. Wireless location-aware technologies and services play a more and more important role in people's daily life. In outdoor location awareness technologies, Global Positioning System (GPS) [1] is the most famous and most representative of location sensing technology and is widely used in military and civilian applications. The demand for indoor location-aware applications is increasing, and there is great potential for indoor real-time and dynamic location-awareness needs. Due to the advantages of non-line-of-sight, non-contact and rapid identification, radio frequency identification (RFID) technology has become the preferred solution to indoor location-aware applications. There are many kinds of location-aware algorithms using RFID, such as Received Signal Strength Indication (RSSI) [2], Angle of Arrival (AOA) [3], Time of Arrival (TOA) [4], Time Division of Arrival (TDOA) [5] and other distance-based RFID positioning methods, but these methods are sensitive to the environment nor their environmental robustness is not strong enough.

At the same time, another kind of RFID positioning method based on scene analysis has attracted a lot of research attention due to its higher adaptability to environment and lower cost. For example, LANDMARC [6] is a classic RFID location-aware algorithm. The algorithm introduces the concept of reference tags to assist the localization. Zhao et al. [7] propose the VIRE algorithm to improve the LANDMARC algorithm by inserting virtual reference tags and introducing the concept of fuzzy maps. Xu et al. [8] use Bayesian probability and *k*-nearest neighbor (KNN) to reduce the location fluctuation and error caused by multipath and environmental interference in LANDMARC. In the scene-based analysis method, the use of neural network methods for RFID positioning has also become another research hotspot. Kung et al. [9] propose a passive RFID indoor positioning scheme that combines LANDMARC scheme with a back propagation (BP) neural network. After LANDMARC location perception, the BP neural network is further used to process location-aware results to obtain more accurate location-aware results. Guo et al. [10] propose an algorithm using radial basis function neural network (RBFNN) for RFID indoor location sensing. The RSSI values and RSSI difference (ΔRSSI) are used as the input of the RBFNN, and the positioning result was obtained. Gholoobi et al. [11] use the weighted *k*-nearest neighbor (WKNN) method to process the captured signal and achieve indoor localization. Mazan et al. [12] design a feed-forward artificial neural network (ANN) to process data and produce estimated coordinates that denote the position of the user. Zou et al. [13] propose an RFID positioning method using an extreme learning machine (ELM), which takes signal strength values as input and coordinates as output. Yang et al [14] propose to use the online extreme learning machine (OS-ELM) to locate the indoor manufacturing execution system. This algorithm uses the known labels in the environment to learn and can adapt to changes in the environment over time, but it requires more tags and higher system costs to reach high positioning accuracy.

In recent years, many studies use evolutionary algorithms to optimize artificial neural networks (ANNs). The optimization of artificial neural networks mainly focuses on three aspects: optimizing the initial weight of the network, optimizing and selecting the network structure and learning algorithms for training. For example, Lin et al. [15] suggest five ANNs in parallel and use the genetic algorithm (GA) [16] to optimize weight values of each network for RFID positioning. Wang et al. [17] proposed a method using particle swarm optimization (PSO) [18] optimized BP neural network for RFID indoor positioning. Kuo et al. [19] proposed an algorithm that uses immune-based feed-forward neural network to learn the relationship between RSSI values and actual locations. Krishnanand et al. [20,21] propose a new type of evolutionary algorithms called Glowworm Swarm Optimization (GSO) algorithm to simulate the behavior of natural fireflies for feeding or courtship. Xu et al. [22] employ the GSO algorithm to optimize the initial weights and biases of the BP neural network. Wang et al. [23] improve the BP neural network using the adaptive-step-size glowworm swarm optimization. Li et al. [24] propose a parallel ensemble learning algorithm based on improved binary glowworm swarm optimization algorithm (IBGSO) and BP neural network. However, for the ELM algorithm, since the weights of the input layer and the biases of the hidden layer are randomly selected and will not be corrected in the training process, in the past ELM optimization research, most studies use different forms of optimization algorithm to optimize the input weights and biases of the ELM to obtain better and more stable models. For example, Zhu et al. [25] use the standard differential evolution algorithm to select and optimize the ELM input weights and hidden layer biases, which is called evolutionary extreme learning machine (E-ELM). Cao et al. [26] use a variety of mutation strategies to improve the standard differential evolution algorithm and proposes an improved version of the E-ELM algorithm, which is called self-adaptive evolutionary extreme learning machine (SaE-ELM). The PSO-ELM fusion algorithm proposed by Xu et al. in literature [27] uses the PSO algorithm to optimize the input weight of the ELM and the bias of the hidden layer. Han et al. [28] have considered the complexity of the ELM model in the optimization, improved the PSO-ELM algorithm, and proposed the ELM algorithm combining with an improved PSO method (IPSO-ELM).

In this paper, we propose an RFID positioning algorithm based on the Glowworm Swarm Optimization (GSO) and semi-supervised online sequential extreme learning machine (SOS-ELM), which is called the GSOS-ELM algorithm. First, we use improved Gaussian filter algorithm to preprocess the RFID data. Then, we use GSO algorithm to optimize the regularization coefficients and train the initial model. Finally, we use the continuously arriving data to update the model and process the user localization requests. The proposed algorithm can quickly obtain the optimal regularization weights under different initial conditions; at the same time, the semi-supervised characteristics can significantly reduce the number of labeled reference tags and reduce the cost. In addition, the online learning phase can continuously update the system to resist the environmental interference. The experimental results show that compared with other algorithms, the proposed localization system can achieve more accurate positioning results and has certain adaptability to the changes of the environment.

The rest of this article is organized as follows: section "Algorithms" introduces the principles of the algorithms; section "Simulation Experiment" explains the algorithm simulation and the simulation results; section "Experimental Evaluation" shows the experimental evaluation results in realistic environment; and section "Conclusion" gives a conclusion of this article and explains our future work.

2. Algorithms

In this part, we will introduce the algorithms in detail. First of all, we will present the semi-supervised online sequential extreme learning machine (SOS-ELM); then we will introduce the glowworm swarm optimization (GSO) method; and finally we demonstrate our proposed RFID-based indoor localization system using GSOS-ELM.

2.1. Semi-Supervised Online Sequential Extreme Learning Machine

To overcome the problems of traditional BP algorithm such as slow learning rate and local minimum in the process of training single-hidden layer feedforward neural networks (SLFNs), Huang et al. [29,30] proposed a simple SLFNs, which is called Extreme Learning Machine (ELM). Its characteristic is that only the number of hidden layer nodes of a SLFNs needs to be set. Without adjusting the input weight and the bias of the hidden element, the input weights of the SLFNs and the offsets of the hidden layer neurons are randomly given in the ELM. The weight of the output layer can be calculated by the Moore–Penrose generalized inverse of the hidden layer output matrix. The results from ELM algorithm have better generalization performance and its learning speed has been greatly improved compared to traditional neural networks. The basic principle of the ELM algorithm is as follows:

Given a training set with N samples $\{(x_i, y_i)|i = 1, 2, \ldots, N\}$, where $x_i \in R^n$ is the input space and $y_i \in R^m$ is the output space. The output of the ELM model with L hidden layer nodes can be expressed as:

$$f(x_i) = h(x_i)^T \beta = \sum_{i=1}^{L} \beta_i G(w_i, b_i, x_i) = y_i, i = 1, 2, \ldots, N \tag{1}$$

where $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{in}]^T$ is the set of weights between the i-th hidden layer neuron and output layer nodes, $w_i = [w_{i1}, w_{i2}, \ldots, w_{im}]^T$ is the set of weights connecting the input layer node and the i-th hidden layer, b_i is the bias term used, $G(\cdot)$ is the activation function.

The above model can be represented as a matrix:

$$H\beta = Y \tag{2}$$

where $H = \begin{pmatrix} G(w_1, b_1, x_1) & \cdots & G(w_L, b_N, x_1) \\ \vdots & \ddots & \vdots \\ G(w_1, b_1, x_N) & \cdots & G(w_L, b_N, x_N) \end{pmatrix}_{N \times L}, \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, Y = \begin{bmatrix} Y_1^T \\ \vdots \\ Y_N^T \end{bmatrix}_{N \times m}.$

Weights w_i and offsets b_i are randomly generated in the model, without training and correction. According to the Moore–Penrose generalized inverse theory [31], solving equations can yield $\beta = H^+$, where H^+ is the Moore–Penrose generalized inverse of matrix H.

The most common method of solving the generalized inverse matrix H^+ is the singular value decomposition method [32]. This method can be used whether $H^T H$ is a singular matrix or a nonsingular matrix, and its execution speed is also better than orthogonal iterations method. The solution formula is as follows:

$$\beta = \begin{cases} (H^T H)^{-1} H^T Y), N > L \\ H^T (H H^T)^{-1} Y, N < L \end{cases} \tag{3}$$

where N is the number of rows of matrix H, L is the number of columns of matrix H, that is, the number of hidden layer nodes.

At the same time, to improve the stability of the ELM solution, Huang et al. [33] introduce the structural risk as a regularization term, added the $L2$ regularization term to the objective function of the ELM, give a ridge regularization version of the ELM algorithm. They convert the original optimization problem to optimization problem:

$$\underset{\beta}{min}(\frac{1}{2}\|H\beta - Y\|^2 + \frac{C_1}{2}\|\beta\|^2) \tag{4}$$

where C_1 is the $L2$ regularization parameter used to adjust the ratio of structural risk and empirical risk to prevent overfitting. Solving Equation (4) results in Equation (5):

$$\beta^* = \begin{cases} (H^T H + \frac{I_{L\times L}}{C_1})^{-1} H^T Y, N > L \\ H^T (H H^T + \frac{I_{N\times N}}{C_1})^{-1} Y, N < L \end{cases} \tag{5}$$

where N is the number of rows of matrix H, L is the number of columns of matrix H, and $I_{L\times L}$ and $I_{N\times N}$ are identity matrices.

To solve the problems of excessive training capacity and long training time when training data is large, Liang et al. [34] improved the batch learning ELM algorithm and proposed an online sequential learning ELM algorithm (OS-ELM). The incremental approach learns new knowledge from data that arrives one by one or from chunk by chunk, and after the current data training is completed, the original data can be discarded. Furthermore, to solve the problem of labeled samples shortage, Huang et al. [35] propose a semi-supervised extreme learning machine (SS-ELM) for semi-supervised learning, which reduces the number of labeled samples and uses unlabeled samples to avoid the high cost of sample labels.

Jia et al. [36] comprehensively consider the merits of empirical risk and structural risk in SS-ELM algorithm and the advantage of incremental learning in OS-ELM algorithm, proposed a semi-supervised online sequential extreme learning machine (SOS-ELM), which maintains generalization ability to take online learning at the same time. The description of the SOS-ELM algorithm is as follows:

Given a data set containing the labeled samples $\{(x_i, y_i)|x_i \in R^n, y_i \in R^m, i = 1, 2, \ldots, N_l\}$ and the unlabeled sample $\{x'_i|x'_i \in R^n, i = 1, 2, \ldots, N_u\}$, the number of hidden neurons is L, the activation function is $g(x)$, the hyperparameters C_1, C_2 correspond to the $L2$ regularization term and the regularization term of the manifold respectively, J is a diagonal matrix, $[J_{ii}] = E_i, i = 1, 2, \ldots, l$, the remaining elements are 0, E_i is the penalty factor, and is set to 1 when used to solve the regression problem. The specific steps of the SOS-ELM algorithm are described in Procedure 1.

Procedure 1 Semi-Supervised Online Sequential Extreme Learning Machine

Initialization phase: Let the training initial data set size be N_0, then the initialized data set is $\{(x_i, y_i)\ or\ x_i' | i = 1, 2, \ldots, N_0\}$.

Step 1.1: Randomly generate the hidden layer weights w_i and offsets b_i.

Step 1.2: Record the labeled sample n_l and unlabeled sample n_u, calculate the similarity matrix of the sample:

$$W = [w_{ij}]_{u \times u} \tag{6}$$

where w_{ij} is the similarity between sample i and sample j, w_{ij} = w_{ji}.

Step 1.3: Calculate the initial diagonal matrix J_0 and the Laplacian matrix $L_{\aleph_0}$:

$$L_{\aleph_0} = D_{\aleph_0} - W_{\aleph_0} \tag{7}$$

where $D_{\aleph_0}$ is a diagonal matrix, the i-th diagonal element is $\sum_{j=1}^{u} w_{ij}$.

Step 1.4: Calculate the initial output weight:

$$\beta^0 = P_0 H_0^T J_0 Y_0 \tag{8}$$

where $P_0 = (C_1 I + H_0^T J_0 H_0 + C_2 H_0^T L_{\aleph_0} H_0)^{-1}$

Step 1.5: Let $k = 0$.

Online learning phase: The $(k+1)$-th new data block arrives:

Step 2.1: Record the labeled samples n_l and unlabeled samples n_u, calculate the diagonal matrix J_{k+1} and Laplacian matrix $L_{\aleph_{k+1}}$.

Step 2.2: Calculate P_{k+1} and output weight β^{k+1}:

$$P_{k+1} = P_k - P_k H_{k+1}^T (C_1 I + (J_{k+1} + C_2 L_{\aleph_{k+1}}) H_{k+1}^T P_k H_{k+1}^T)^{-1} (J_{k+1} + C_2 L_{\aleph_{k+1}}) H_{k+1} P_k \tag{9}$$

$$\beta^{k+1} = \beta^k + P_{k+1} H_{k+1}^T (J_{k+1} Y_{k+1} - (J_{k+1} + C_2 L_{\aleph_{k+1}}) H_{k+1} \beta^k) \tag{10}$$

Step 2.3: Let $k = k + 1$, return to **Step 2.1**.

2.2. Glowworm Swarm Optimization Method

The Glowworm Swarm Optimization (GSO) algorithm simulates the glow behavior of the firefly in nature, uses its luminescence properties to find partners based on its search area, and moves to a firefly with a superior position in the neighborhood structure to achieve evolution.

The relative brightness of the firefly is $I = I_0 \times e^{-\gamma r_{ij}}$, where I_0 is the maximum fluorescent intensity of the firefly, i.e., the fluorescence intensity of itself $(r = 0)$, which is related to the objective function value; γ is the light intensity absorption coefficient, to reflect the fact that fluorescence gradually decreases with distance increases and absorption of the media; r_{ij} is the spatial distance between fireflies i and j.

The degree of attraction of fireflies is $\beta = \beta_0 \times e^{-\gamma r_{ij}^2}$, where β_0 is the maximum degree of attraction, that is, the degree of attraction at the light source $(r = 0)$.

The position of the firefly i being attracted to the firefly j is updated by $x_i = x_i + \beta \times (x_j - x_i) + \alpha \times (rand - 1/2)$, where x_i, x_j are the spatial positions of the fireflies i and j; α is the constant step length factor on [0, 1]; *rand* is a random factor that follows uniform distribution on [0, 1].

The specific steps of the GSO algorithm are described as Procedure 2.

Procedure 2 Glowworm Swarm Optimization Method

Step 1: Initialize the basic parameters: set the number of fireflies m, the maximum attraction β_0, the light absorption coefficient γ, the step length factor α, the maximum number of iterations $maxT$ or the search accuracy ϵ.
Step 2: Randomly initialize the position of the firefly and calculate the target value of the firefly as its maximum fluorescent intensity I_0.
Step 3: Calculate the relative brightness I and the attractiveness β of the firefly in the population and determine the direction of movement of the firefly based on the relative brightness:

$$I = I_0 \times e^{-\gamma r_{ij}} \tag{11}$$

$$\beta = \beta_0 \times e^{-\gamma r_{ij}^2} \tag{12}$$

where γ is the light intensity absorption coefficient and r_{ij} is the spatial distance between fireflies i and j.
Step 4: Update the spatial position of the firefly:

$$x_i = x_i + \beta \times (x_j - x_i) + \alpha \times (rand - 1/2) \tag{13}$$

where x_i, x_j are the spatial positions of the fireflies i and j; α is the constant step length factor on [0, 1]; $rand$ is a random factor that follows uniform distribution on [0, 1].
Step 5: Randomly disturb the firefly at the best position.
Step 6: Recalculate the firefly's brightness I_{new} based on the location of the updated firefly.
Step 7: When the search accuracy is satisfied, or the maximum number of searches is reached, skip to **Step 8**; otherwise, increase the number of searches by 1 and skip to **Step 3** for the next search.
Step 8: Output global extreme points and optimal individual values.

2.3. RFID-Based Indoor Localization System

Based on the semi-supervised online sequential extreme learning machine (SOS-ELM) and Glowworm Swarm Optimization (GSO) method, this paper proposes an RFID indoor localization algorithm that combines GSO algorithm and SOS-ELM algorithm, which is called the GSOS-ELM algorithm.

The framework of the proposed system is shown in Figure 1. The system includes n readers, l reference tag with known position, u reference tags with unknown positions, the signal strength of the i-th reader reading the tag is $RSSI_i$, the first l elements of diagonal matrix J are set to 1 and the following u elements are 0.

The specific steps of the GSOS-ELM system are described as Procedure 3, it is divided into the offline phase and the online phase.

Procedure 3 Semi-Supervised Online Sequential Extreme Learning Machine

Offline phase:

Step 1.1: Preprocessing the data using an improved Gaussian filter algorithm:

(i): The i-th reader repeatedly reads the signal strength of the same tag for a total of N times, and records the signal strength of the k-th read signal as $RSSI_{ik}, k = 1, 2, \ldots, N$;

(ii): Calculate the variance δ^2 of the RSSI value:

$$\delta^2 = \frac{1}{N-1} \sum_{k=1}^{N} (RSSI_{ik} - \overline{A})^2 \tag{14}$$

where $\overline{A} = \frac{1}{N} \sum_{k=1}^{N} RSSI_{ik}$

(iii): For the k-th signal strength $RSSI_{ik}$, perform culling if $|RSSI_{ik} - \overline{A}| > 3\delta$, finally an RSSI set of size m is obtained, and the average value of the RSSI set is calculated as the average signal strength:

$$\overline{RSSI_l} = \frac{1}{m} \sum_{k=1}^{m} RSSI_{ik} \tag{15}$$

Step 1.2: Determine the number of hidden neurons of the GSOS-ELM algorithm, the activation function $g(x)$, the regularization coefficients C_1 and C_2, and generate hidden layer weights w_i and offsets b_i;

Step 1.3: The initial data set size is N_0, record the labeled sample n_l and the unlabeled sample n_u, calculate the similarity matrix $W_{\aleph_0}$ of the sample:

$$W_{\aleph_0} = [w_{ij}]_{u \times u} \tag{16}$$

where w_{ij} is the similarity between sample i and sample j, the measure formula is:

$$w_{ij} = e^{-\left\| x_i - x_j \right\|^2 / (2\delta^2)} \tag{17}$$

Step 1.4: Calculate the initial Laplacian matrix $L_{\aleph_0}$:

$$L_{\aleph_0} = D_{\aleph_0} - W_{\aleph_0} \tag{18}$$

where $D_{\aleph_0}$ is a diagonal matrix, the i-th diagonal element is $\sum_{j=1}^{u} w_{ij}$.

Step 1.5: Calculate the initial output weight $\beta^{0_t} = P_0 H_0^T J_0 Y_0$, where $P_0 = (C_1 I + H_0^T J_0 H_0 + C_2 H_0^T L_{\aleph_0} H_0)$, J_0 is a diagonal matrix, $[J_{ii}] = 1, i = 1, 2, \ldots, l$, the remaining elements are 0, get the output matrix $\widehat{Y}$ of the labeled sample and the real value Y;

Step 1.6: Optimize the regularization coefficients C_1 and C_2 according to the specific steps in Procedure 2, the fitness function is:

$$fitness = \frac{1}{\sum_{i=1}^{l} \left\| \widehat{Y_l} - Y_i \right\|} \tag{19}$$

Step 1.7: The optimized initial output weight β^0 is output to online learning phase and online working phase.

Online phase: The online phase includes online learning phase and online working phase, they can be performed parallel.

Online learning phase: The $k+1$ new data block arrives:

Step 2.1.1: Record the labeled samples n_l and unlabeled samples n_u, use improved Gaussian filter to process the data and calculate the Laplacian matrix $L_{\aleph_{k+1}}$.

Step 2.1.2: Calculate P_{k+1} and output weight β^{k+1}:

$$P_{k+1} = P_k - P_k H_{k+1}^T (C_1 I + (J_{k+1} + C_2 L_{\aleph_{k+1}}) H_{k+1}^T P_k H_{k+1}^T)^{-1} (J_{k+1} + C_2 L_{\aleph_{k+1}}) H_{k+1} P_k \tag{20}$$

$$\beta^{k+1} = \beta^k + P_{k+1} H_{k+1}^T (J_{k+1} Y_{k+1} - (J_{k+1} + C_2 L_{\aleph_{k+1}}) H_{k+1} \beta^k) \tag{21}$$

Step 2.1.3: Let $k = k+1$, return to **Step 2.1.1**.

Online working phase:

Step 2.2.1: The user requests positioning from the server through the client and sends the signal strength RSSI information at the unknown location to the server.

Step 2.2.2: The server uses the RSSI information sent from the client as input to the current GSOS-ELM model to estimate the positioning result and send it back to the client.

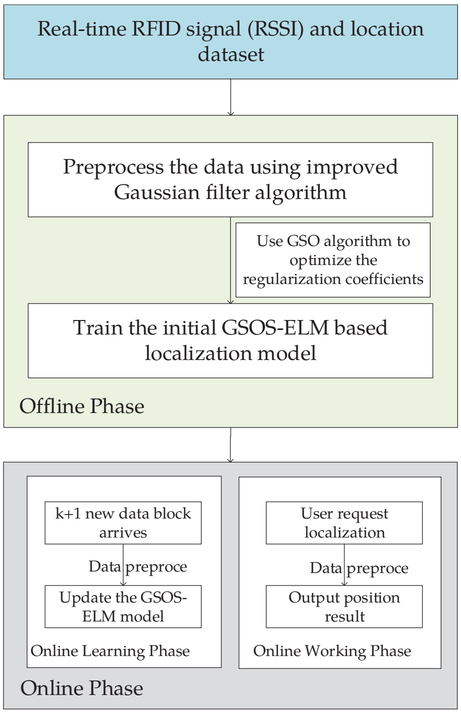
Real-time RFID signal (RSSI) and location dataset
Preprocess the data using improved Gaussian filter algorithm
Use GSO algorithm to optimize the regularization coefficients
Train the initial GSOS-ELM based localization model
Offline Phase
k+1 new data block arrives
Data preproce
Update the GSOS-ELM model
Online Learning Phase
User request localization
Data preproce
Output position result
Online Working Phase
Online Phase

Figure 1. The Framework of RFID Localization System Using GSOS-ELM.

3. Simulation Experiment

As shown in Figure 2, we have performed a simulation experiment on the MATLAB platform and simulated a 6 m × 7.2 m area, the reference readers are placed in the corners and the edges, and the RFID tags are distributed in the region. The objective of our simulation experiment is to analyze the impact factors and compare the localization effect in different environments. The experimental configurations are as follows: (1) operating systems: Windows 10 x64 v1803; (2) CPU: Intel(R) Core(TM) i3-4160 @ 3.60 GHz; (3) memory: 4 GB; and (4) software: MATLAB R2015b.

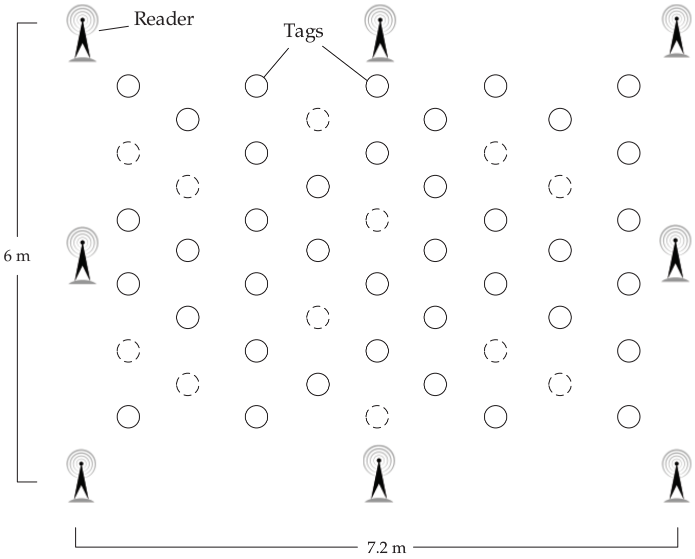

Figure 2. Simulation Experiment Layout.

In the simulation experiment, we use the log-normal path loss model to model the attenuation of the signal in the indoor environment with distance:

$$PL(d) = PL(d_0) + 10n \log_{10} \frac{d}{d_0} + X_\delta \tag{22}$$

where d_0 is the reference distance, n is the path loss factor, d is the distance between the sender and the receiver of the signal, and X_δ is zero mean Gaussian random variable with standard deviation of δ.

The RSSI value of RFID can be expressed by Equation (23):

$$RSSI(d) = P_t + G_t - PL(d) \tag{23}$$

where P_t is the transmit power, and G_t denotes the antenna gain of transmit node.

Since the G_t is fixed, combine Equations (22) and (23), we can get Equation (24):

$$RSSI(d) = RSSI(d_0) - 10n \log_{10} \frac{d}{d_0} + X_\delta \tag{24}$$

where d_0 is the reference distance and n is the path loss factor. According to the analysis in [37,38], we set $RSSI(d_0) = -45$ dBm, $n = 2$, $\delta = 2$ in the simulation experiment.

The experimental error is defined as the Euclidean distance between the estimated target tag position (x_e, y_e) and the actual target tag position (x_t, y_t), the i-th error result is denoted as ρ_i:

$$\rho_i = \sqrt{((x_e - x_t)^2 + (y_e - y_t)^2)} \tag{25}$$

The average error of the system is $\overline{\rho_i} = \sum_{i=1}^{n} \rho_i / n$, where n is the number of target tags in the positioning process.

We set the number of hidden layer nodes of the GSOS-ELM model to $L = 100$ and the activation function $g(x)$ to sigmoid. We use the k-fold cross-validation to evaluate the performances. The samples are randomly split into k subsets equally; then, the subsets are divided into two sets, the testing set with only one subset and the training set with the reset $(k - 1)$ subsets; and here we set $k = 10$. After the division of training set and testing set, we further divide the samples into two groups, one group with labels, the rest are unlabeled.

3.1. Influence Factors

The main factors influencing the positioning result include the density of reference tags (Δ), the number of reference readers (N), the proportion of labeled samples (%), the data preprocessing, and the placement of the reference tags. In the following sections, we will analyze the impact of each factor respectively.

3.1.1. Density of Reference Tags

In this experiment, we set the number of readers to $N = 8$ and the proportion of labeled samples to 40%. We divide the reference tag density into 1.0 m, 0.8 m, 0.5 m and 0.3 m.

As shown in Figure 3, when the reference tag density changes from 1.0 m to 0.8 m and 0.5 m, the positioning error decreases significantly, but when the density of reference tags is changed from 0.5 m to 0.3 m, the positioning error does not change much. Because the number of reference tags in the unit area increases as the density of the reference tag increases, thereby reducing the positioning error and obtaining a more accurate positioning result; but when the reference tags reach to a relative high density (here is 0.3 m), it may contain redundant information compared to relatively low density situation (0.5 m) and do not increase the accuracy significantly.

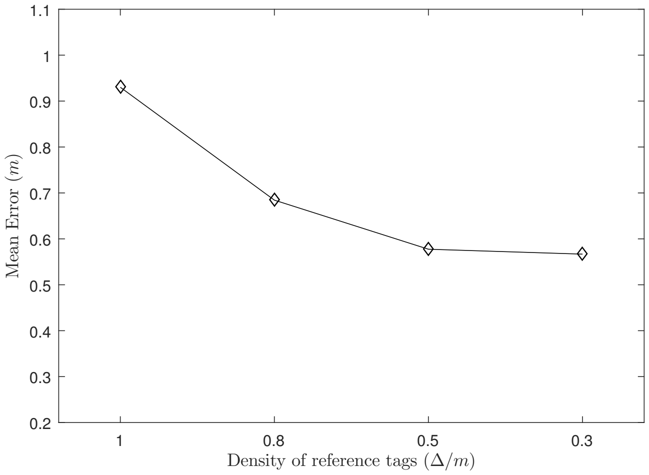

Figure 3. The Influence of Density of Reference Tags.

3.1.2. The Number of Readers

In this experiment, we set the density of reference tags to $\Delta = 0.5$ m and the proportion of labeled samples to 40%. As shown in Figure 2, we place reference readers in the corners and the edges of the system; the number of reference readers is 4, 6, 8 and 12 respectively.

As shown in Figure 4, the positioning error of the system decreases as the number of reference readers increases. When the number of reader increases from 4 to 6 and from 6 to 8, the localization accuracy improves noticeably, but when the number of reference readers increases to a certain degree and provides duplicate data, it does not change much on the positioning error, and the increasing in the number of readers will obviously increase the cost of the entire positioning system.

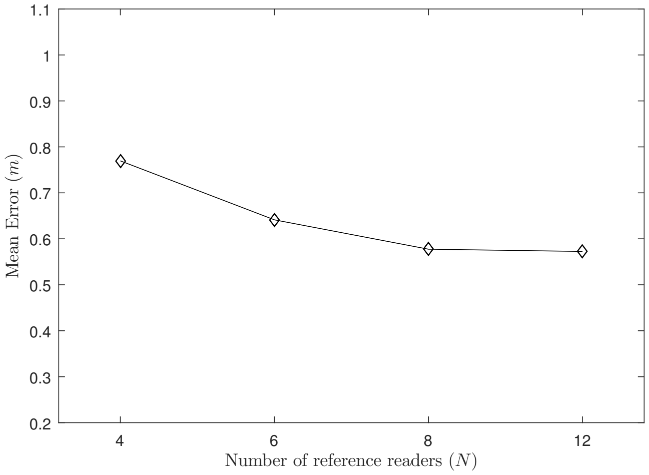

Figure 4. The Influence of the Number of Readers.

3.1.3. The Proportion of Labeled Samples

In this experiment, we analyze the influence of the labeled samples proportion to the GSOS-ELM algorithm. We set the number of readers to $N = 8$ and the density of reference tags to $\Delta = 0.5$ m. The proportion of labeled samples is 20%, 40%, 60%, and 80%, respectively.

As shown in Figure 5, the performance of the positioning system increases most significantly when the proportion of the labeled samples increases from 20% to 40%, but the positioning error does not change much as 40% is increased to 60% and 80%. Because when the labeled samples at a low proportion, the increasing of labeled information will significantly improve the non-regularization term of PSOS-ELM algorithm and reduce positioning error; but when the proportion of labeled samples reaches a high degree, the more labeled samples and less unlabeled samples may balance the non-regularization term and regularization term, and the localization accuracy does not change much.

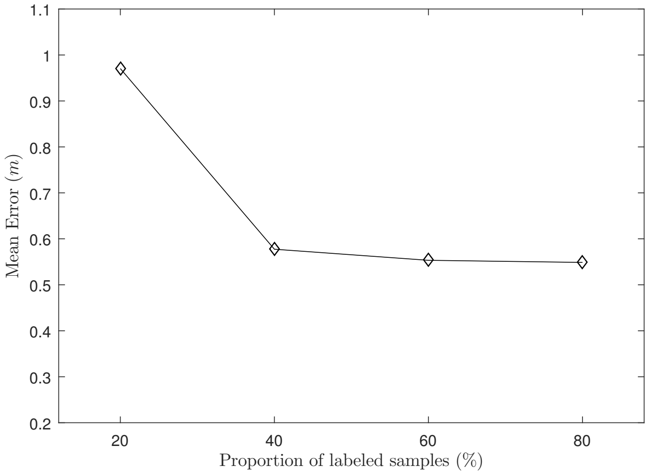

Figure 5. The Influence of Proportion of Labeled Samples.

3.1.4. Preprocessing and Reference Tags Placement

In this experiment, we study the influence of the improved Gaussian preprocessing and the reference tags placement method on the positioning effect of the algorithm. We set the number of readers to $N = 8$, the density of reference tags to $\Delta = 0.5$ m and the proportion of labeled samples to 40%. The reference tags placement method changes without changing the total number of reference tags, which are squares, rectangles, and equilateral triangles.

As shown in Table 1, the performance of the GSOS-ELM algorithm in positioning has been improved by the pre-processing process. The average error has been improved from 0.6439 m to 0.5774 m, and the standard deviation has decreased from 0.7395 m to 0.6496 m. As shown in Table 2, the placement of the reference tag also has a certain impact on the GSOS-ELM algorithm, when the tag is placed in a rectangular manner, the performance of the algorithm is reduced significantly compared to square and equilateral triangle manner. At the same time, the equilateral triangle placement method has a certain improvement to the square placement method, and the average error decreases from 0.6568 m to 0.5774 m.

Table 1. The Influence of Preprocessing.

Preprocessing	Min Error (m)	Max Error (m)	Average Error (m)	Standard Deviation (m)
Without preprocessing	0.1076	2.1157	0.6439	0.7395
With preprocessing	0.0973	1.8447	0.5774	0.6496

Table 2. The Influence of Label Placement.

Reference Tags Placement	Min Error (m)	Max Error (m)	Average Error (m)	Standard Deviation (m)
Square	0.1064	2.2497	0.6568	0.6832
Rectangle	0.1634	2.7769	0.9795	0.8645
Equilateral triangle	0.0973	1.8447	0.5774	0.6496

3.2. Comparison with Other Methods

According to the results from Section 3.1, we set the default condition to: the density of the reference tag is $\Delta = 0.5$ m, the number of readers is $N = 8$, the proportion of labeled samples is 40% and the placement of the reference tags is equilateral triangle.

In this section, we compare the proposed GSOS-ELM algorithm with the NN-Based algorithm proposed in [9], the FA-OSELM algorithm proposed in [39], and the NMDS algorithm proposed in [40]. The NN-Based algorithm uses a BP neural network to enhance the LANDMARC algorithm [6]; the FA-OSELM algorithm uses incremental data to update the original model to a transferred model; and the NMDS algorithm combines nonmetric multidimensional scaling algorithm and fingerprinting algorithm to archive localization. We implement these algorithms and perform experiments in the same simulated space. The parameters setting for these methods is described in Table 3.

Table 3. Parameters Setting for Proposed Method and Compared Methods.

Method	Parameters Setting
GSOS-ELM	Activation function: sigmoid, $L = 100$
NN-Based	Activation function: sigmoid, $L = 100$, number of neighbors $k = 4$
FA-OSELM	Activation function: RBF, $L = 350$, regularization factor $C = 2^{-6}$
NMDS	Goodness fit threshold $\epsilon = 10^{-4}$, maximum number of iterations $k_{max} = 200$

The comparison results are shown in Table 4. The proposed GSOS-ELM algorithm has certain advantages over other algorithms in terms of average error and stability. Compared with the NN-Based algorithm, FA-OSELM algorithm, and NMDS algorithm, the average error of our proposed GSOS-ELM algorithm improves by 13.46%, 16.56% and 11.94%, respectively. Also, we can learn that the max error is higher at 1.8447 m, this is because when the target tag falls into the edges and corners, the decrease in the reference tags leads to the increase in the error. For the average execution time of algorithms, as shown in Table 5, both GSOS-ELM and FA-OSELM reach a better efficiency than NN-Based method and NMDS method. GSOS-ELM takes 17.6613 s and FA-OSELM requires 15.0384 s in average execution time, hence GSOS-ELM is slightly slower than FA-OSELM. GSOS-ELM reduces the average execution time to 21.03% of NN-Based and 40.41% of NMDS.

Table 4. Comparison Result under Simulation Environment.

Method	Min Error (m)	Max Error (m)	Average Error (m)	Standard Deviation (m)
GSOS-ELM	0.0973	1.8447	0.5774	0.6496
NN-Based	0.1103	2.4829	0.6672	0.8574
FA-OSELM	0.0935	2.3341	0.6920	0.7933
NMDS	0.1104	2.0745	0.6557	0.7202

Table 5. Comparison of Average Execution Time under Simulation Environment.

Method	GSOS-ELM	NN-Based	FA-OSELM	NMDS
Average Execution Time (s)	17.6613	83.9932	15.0384	37.2103

At the same time, as shown in Figure 6, to verify the adaptability of the GSOS-ELM positioning algorithm in the environment, we move some labeled reference tags during the experiment to observe the average error of the positioning system. The specific steps to process the dynamic changes are shown in Figure 7, when the localization environment changes and the new data blocks arrive, we use improved Gaussian filter algorithm mentioned in Procedure 3 to preprocess the data and update the initial PSOS-ELM.

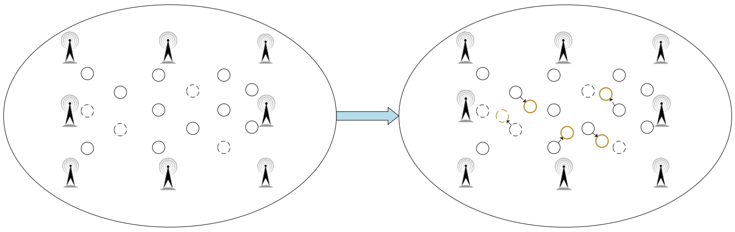

Figure 6. The Dynamic Changes to the Simulation Environment.

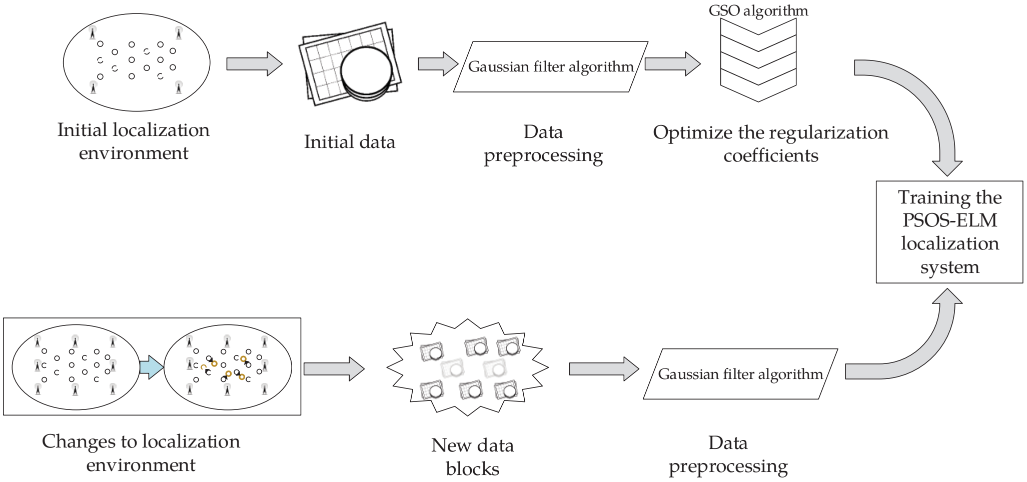
Initial localization environment
Initial data
Gaussian filter algorithm
Data preprocessing
GSO algorithm
Optimize the regularization coefficients
Training the PSOS-ELM localization system
Changes to localization environment
New data blocks
Gaussian filter algorithm
Data preprocessing

Figure 7. The Steps to Process the Dynamic Changes.

As shown in Table 6, after moving the reference tags, the average error of the GSOS-ELM algorithm increases from 0.5774 m to 0.6428 m, and the increasing rate is 11.33%. At the same time, the average error increasing rate of the FA-OSELM algorithm using the online learning method is 12.69%. However, the average error of the NN-Based and NMDS algorithms without online learning phase raise significantly, with the increasing rates of 28.25% and 25.55%, respectively.

Table 6. Comparison Result under Dynamic Simulation Environment.

Method	Min Error (m)	Max Error (m)	Average Error (m)	Standard Deviation (m)
GSOS-ELM	0.0997	2.0408	0.6428	0.7155
NN-Based	0.1246	2.6368	0.8557	0.9149
FA-OSELM	0.1041	2.4076	0.7798	0.8581
NMDS	0.1227	2.3718	0.8232	0.8061

4. Experimental Evaluation

4.1. Experiment Setting

To further demonstrate the appropriateness of the proposed method under static and dynamics environments compared to the other methods, we have conducted several realistic experiments in Guangzhou Research Institute of O-M-E Technology.

The reader model we used is Alien ALR-9900+ [41], the main working frequency is 920 MHz, the maximum power strength is 30.7 dBm. We use two kinds of antennas, their model are Alien ALR-8696-C [42] (8.5 dBic gain) and ALR-9611-CR [43] (6 dBic gain). Our experiment setting is shown in Figures 8 and 9, we finish the experiments under a realistic environment with 4.00 m long and 3.63 m wide, we place 8 antennas in the corners and edges and 20 passive RFID tags inside the area. The computer communicates with the ALR-9900+ reader via TCP/IP protocol. Same as previous, the parameters setting is in Table 3, the proportion of labeled samples is 40%, and the 10-fold cross-validation is adopted to evaluate the performances.

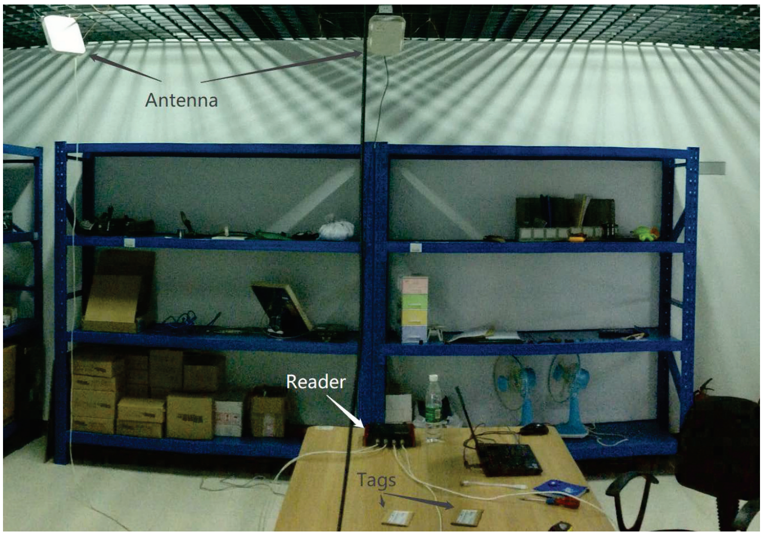

Figure 8. Experimental Setup in Realistic Environment.

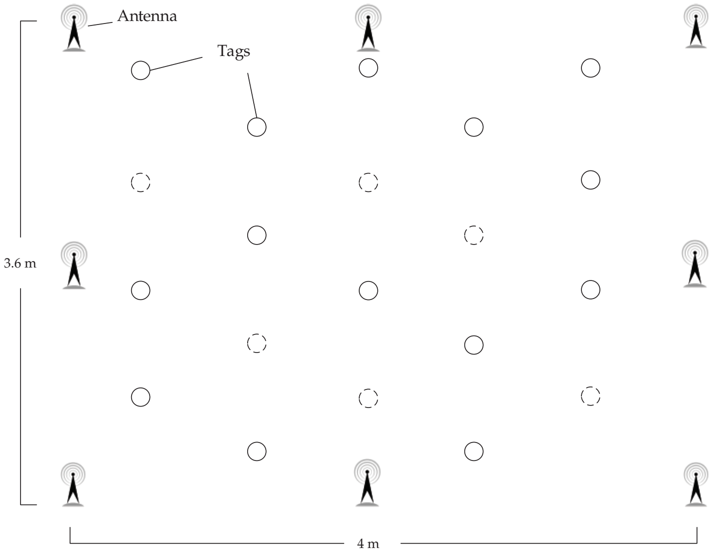

Figure 9. Realistic Experiment Layout.

4.2. Experiment Results

As shown in Table 7, the proposed GSOS-ELM algorithm has certain improvement in average error and standard deviation. The average error of our proposed GSOS-ELM method overcomes by a rate of 15.18%, 18.07% and 12.45% over NN-Based method, FA-OSELM method and NMDS method, respectively. We can also find that the maximum localization error is significantly higher at 1.2489 m, it is due to the reduced reference tags when target tag is in the edges and corners, leading to the higher localization error. The comparison of average execution time is shown in Table 8, we can learn that GSOS-ELM and FA-OSELM take less average execution time than NN-Based method and NMDS method, while GSOS-ELM requires 9.7748 s and FA-OSELM is 1.3017 s faster. The average execution time of GSOS-ELM method is 5.73 times less than NN-Based method and 2.58 times less than NMDS method.

Table 7. Comparison Results under Realistic Environment.

Method	Min Error (m)	Max Error (m)	Average Error (m)	Standard Deviation (m)
GSOS-ELM	0.0872	1.2489	0.4302	0.4837
NN-Based	0.0974	1.6307	0.5072	0.5704
FA-OSELM	0.0892	1.5513	0.5251	0.5481
NMDS	0.0992	1.4877	0.4914	0.5113

Table 8. Comparison of Average Execution Time under Realistic Environment.

Method	GSOS-ELM	NN-Based	FA-OSELM	NMDS
Average Execution Time (s)	9.7748	56.0564	8.4731	21.8764

At the same time, as shown in Figure 10, we move some labeled reference tags to checkout the adaptability of the algorithms in realistic environment.

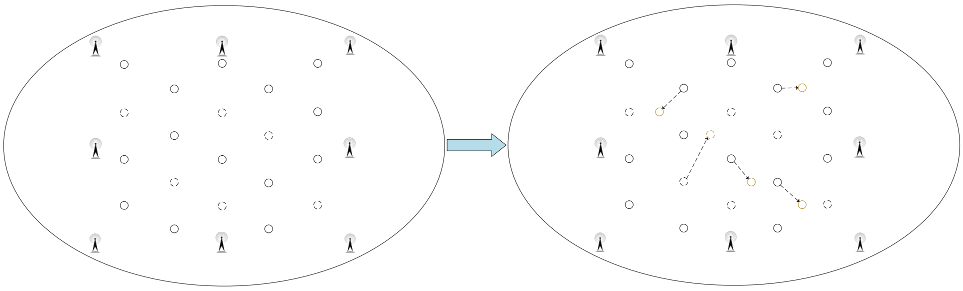

Figure 10. The Dynamic Changes to the Realistic Environment.

The comparison results are in Table 9, after moving the reference tag, the average error of proposed GSOS-ELM algorithm raises from 0.4302 m up to 0.4851 m, with an increasing rate of 12.76%. Meanwhile, the average error raising rate of the FA-OSELM algorithm with the online learning phase is 13.54%, with not much differences. However, the average error of the NN-Based and NMDS algorithms raise significantly, with the rates of 30.78% and 28.14%, respectively. The experimental results show that adopting online learning method can improve the adaptability of the algorithm in the environment.

Table 9. Comparison Result under Dynamic Realistic Environment.

Method	Min Error (m)	Max Error (m)	Average Error (m)	Standard Deviation (m)
GSOS-ELM	0.0882	1.3258	0.4851	0.5204
NN-Based	0.0993	1.7434	0.6633	0.6620
FA-OSELM	0.0931	1.6041	0.5962	0.5685
NMDS	0.1064	1.5381	0.6297	0.5828

5. Conclusions

This paper proposes an RFID positioning algorithm, which is called the GSOS-ELM algorithm. It is a fusion semi-supervised online sequential extreme learning machine (SOS-ELM) based on the Glowworm Swarm Optimization (GSO), aiming at improving the disadvantages of the existing RFID indoor positioning algorithms, which are susceptible to the tag density and algorithm efficiency, and lack of environmental adaptability. The GSOS-ELM algorithm uses the semi-supervised method to reduce the number of labeled reference tags and the cost of positioning systems; and we use the GSO method to adjusts the regularization weights of the SOS-ELM algorithm, so that it can quickly obtain the optimal regularization weights under different initial conditions. In addition, the online learning phase of the GSOS-ELM algorithm can continuously update the system to perceive changes in the environment and resist the environmental interference.

In the simulation section, we have studied the influence factors of the GSOS-ELM algorithm, including the reference tag density, the number of reference readers, the proportion of labeled samples, data preprocessing and the placement of reference tags. The results show that increasing the proportion of labeled samples, the density of readers and reference tags can increase the localization accuracy, but the rate of increase gradually decreases and the cost of the system raises. Besides, we have given simulation experiment to compare GSOS-ELM method with the other methods and carried out testbed experiment to evaluate our proposed method. Both the simulation and the testbed experiment results have shown that, compared with other algorithms, our proposed algorithm has certain advantages in average error; and the smaller increase in localization errors shows that it has certain adaptability to the changes of the environment.

However, the proposed method does not solve the problem that the localization error is higher when the target tag falls into the edges and corners. Our further work will focus on solving this problem and analyzing how the environment and the location of the antennas affect the level of electromagnetic radiation.

Author Contributions: F.L. and D.Z. defined problem and developed the idea. D.Z. carried out the experiments and data analysis, and wrote the relevant sections.

Funding: This paper is partially supported by the Engineering and Technology Research Center of Guangdong Province for Logistics Supply Chain and Internet of Things (Grant No. GDDST[2016]176); the Provincial Science and Technology Project in Guangdong Province (Grant No. 2013B090200055); the Key Laboratory of Cloud Computing for Super—integration Cloud Computing in Guangdong Province (Grant No. 610245048129); and the Engineering and Technology Research Center of Guangdong Province for Big Data Intelligent Processing (Grant No. GDDST[2013]1513-1-11).

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

References

1. Hofmann-Wellenhof, B.; Lichtenegger, H.; Collins, J. *Global Positioning System: Theory and Practice*; Springer Science & Business Media: Berlin, Germany, 2012.
2. Song, X.; Li, X.; Tang, W.; Zhang, W.; Li, B. A hybrid positioning strategy for vehicles in a tunnel based on RFID and in-vehicle sensors. *Sensors* **2014**, *14*, 23095–23118. [CrossRef] [PubMed]
3. Azzouzi, S.; Cremer, M.; Dettmar, U.; Kronberger, R.; Knie, T. New Measurement Results for the Localization of UHF Rfid Transponders Using an Angle of Arrival (AOA) Approach. In Proceedings of the 2011 IEEE International Conference on RFID, Orlando, FL, USA, 12–14 April 2011; pp. 91–97.
4. Cheung, K.W.; So, H.C.; Ma, W.K.; Chan, Y.T. Least squares algorithms for time-of-arrival-based mobile location. *IEEE Trans. Signal Process.* **2004**, *52*, 1121–1130. [CrossRef]
5. Ai, Z.; Liu, Y. Research on the TDOA measurement of active RFID real time location system. In Proceedings of the 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Chengdu, China, 9–11 July 2010; Volume 2, pp. 410–412.
6. Ni, L.M.; Liu, Y.; Lau, Y.C.; Patil, A.P. LANDMARC: Indoor location sensing using active RFID. *Wirel. Netw.* **2004**, *10*, 701–710. [CrossRef]
7. Zhao, Y.; Liu, Y.; Ni, L.M. VIRE: Active RFID-based localization using virtual reference elimination. In Proceedings of the 2007 International Conference on Parallel Processing (ICPP 2007), Xi'an, China, 10–14 September 2007; p. 56.
8. Xu, H.; Ding, Y.; Li, P.; Wang, R.; Li, Y. An RFID Indoor Positioning Algorithm Based on Bayesian Probability and K-Nearest Neighbor. *Sensors* **2017**, *17*, 1806. [CrossRef] [PubMed]
9. Kung, H.Y.; Chaisit, S.; Phuong, N.T.M. Optimization of an RFID location identification scheme based on the neural network. *Int. J. Commun. Syst.* **2015**, *28*, 625–644. [CrossRef]

10. Guo, D.; Zhang, Y.; Xiang, Q.; Li, Z. Improved radio frequency identification indoor localization method via radial basis function neural network. *Math. Probl. Eng.* **2014**, *2014*, 420482. [CrossRef]
11. Gholoobi, A.; Stavrou, S. RSS based localization using a new WKNN approach. In Proceedings of the 2015 7th International Conference on Computational Intelligence, Communication Systems and Networks, Riga, Latvia, 3–5 June 2015; pp. 27–30.
12. Mazan, F.; Kovarova, A. Optimizing Artificial Neural Network for Beacon Based Indoor Localization. In Proceedings of the 17th International Conference on Computer Systems and Technologies 2016, Palermo, Italy, 23–24 June 2016; pp. 261–268.
13. Zou, H.; Wang, H.; Xie, L.; Jia, Q.S. An RFID indoor positioning system by using weighted path loss and extreme learning machine. In Proceedings of the 2013 IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), Taipei, Taiwan, 19–20 August 2013; pp. 66–71.
14. Yang, Z.; Zhang, P.; Chen, L. RFID-enabled indoor positioning method for a real-time manufacturing execution system using OS-ELM. *Neurocomputing* **2016**, *174*, 121–133. [CrossRef]
15. Lin, Y.S.; Chen, R.C.; Lin, Y.C. An indoor location identification system based on neural network and genetic algorithm. In Proceedings of the 2011 3rd International Conference onAwareness Science and Technology (iCAST), Dalian, China, 27–30 September 2011; pp. 193–198.
16. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
17. Wang, C.; Wu, F.; Shi, Z.; Zhang, D. Indoor positioning technique by combining RFID and particle swarm optimization-based back propagation neural network. *Opt. Int. J. Light Electron Opt.* **2016**, *127*, 6839–6849. [CrossRef]
18. Kennedy, J. Particle swarm optimization. In *Encyclopedia of Machine Learning*; Springer: Berlin, Germany, 2011; pp. 760–766.
19. Kuo, R.; Chang, J. Intelligent RFID positioning system through immunebased feedforward neural network. *J. Intell. Manuf.* **2015**, *26*, 755–767. [CrossRef]
20. Krishnanand, K.; Ghose, D. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In Proceedings of the 2005 IEEE Swarm Intelligence Symposium SIS 2005, Pasadena, CA, USA, 8–10 June 2005; pp. 84–91.
21. Krishnanand, K.; Ghose, D. Theoretical foundations for multiple rendezvous of glowworm-inspired mobile agents with variable local-decision domains. In Proceedings of the 2006 American Control Conference, Minneapolis, MN, USA, 14–16 June 2006; p. 6.
22. Xu, Q.; Liu, Y.Q.; Tian, D.; Long, Q. Rolling bearing fault diagnosis method using glowworm swarm optimization and artificial neural network. *Adv. Mater. Res.* **2014**, *860*, 1812–1815. [CrossRef]
23. Wang, X.W.; Zhu, Q.S.; Tan, A.S.; Zhang, Y.S. Threat Assessment of Target Group Based on Improved Glowworm Swarm Optimization and BP Neural Network. *Command Control Simul.* **2014**. [CrossRef]
24. Li, J.M.; Ni, Z.W.; Zhu, X.H.; Xu, Y. Parallel Ensemble Learning Algorithm Based on Improved Binary Glowworm Swarm Optimization Algorithm and BP Neural Network. *Pattern Recognit. Artif. Intell.* **2017**, *30*, 171–182.
25. Zhu, Q.Y.; Qin, A.K.; Suganthan, P.N.; Huang, G.B. Evolutionary extreme learning machine. *Pattern Recognit.* **2005**, *38*, 1759–1763. [CrossRef]
26. Cao, J.; Lin, Z.; Huang, G.B. Self-adaptive evolutionary extreme learning machine. *Neural Process. Lett.* **2012**, *36*, 285–305. [CrossRef]
27. Xu, Y.; Shu, Y. Evolutionary extreme learning machine–based on particle swarm optimization. In Proceedings of the Third International Symposium on Neural Networks, Chengdu, China, 28 May–1 June 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 644–652.
28. Han, F.; Yao, H.F.; Ling, Q.H. An improved evolutionary extreme learning machine based on particle swarm optimization. *Neurocomputing* **2013**, *116*, 87–93. [CrossRef]
29. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990.
30. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
31. Serre, D. Graduate Texts in Mathematics. In *Matrices*; Springer: New York, NY, USA, 2002; Volume 216.

32. Golub, G.H.; Reinsch, C. Singular value decomposition and least squares solutions. *Numer. Math.* **1970**, *14*, 403–420. [CrossRef]
33. Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. B Cybern.* **2012**, *42*, 513–529. [CrossRef] [PubMed]
34. Liang, N.Y.; Huang, G.B.; Saratchandran, P.; Sundararajan, N. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Trans. Neural Netw.* **2006**, *17*, 1411–1423. [CrossRef] [PubMed]
35. Huang, G.; Song, S.; Gupta, J.N.; Wu, C. Semi-supervised and unsupervised extreme learning machines. *IEEE Trans. Cybern.* **2014**, *44*, 2405–2417. [CrossRef] [PubMed]
36. Jia, X.; Wang, R.; Liu, J.; Powers, D.M. A semi-supervised online sequential extreme learning machine method. *Neurocomputing* **2016**, *174*, 168–178. [CrossRef]
37. Rappaport, T.S. *Wireless Communications: Principles and Practice*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1996; Volume 2.
38. Torres-Sospedra, J.; Moreira, A. Analysis of Sources of Large Positioning Errors in Deterministic Fingerprinting. *Sensors* **2017**, *17*, 2736. [CrossRef] [PubMed]
39. Jiang, X.; Liu, J.; Chen, Y.; Liu, D.; Gu, Y.; Chen, Z. Feature adaptive online sequential extreme learning machine for lifelong indoor localization. *Neural Comput. Appl.* **2016**, *27*, 215–225. [CrossRef]
40. Gao, Z.; Ma, Y.; Liu, K.; Miao, X.; Zhao, Y. An Indoor Multi-Tag Cooperative Localization Algorithm Based on NMDS for RFID. *IEEE Sens. J.* **2017**, *17*, 2120–2128. [CrossRef]
41. Alien Technology Corporation. Alien Technology ALR-9900 Enterprise RFID Reader. 2013. Available online: http://www.alientechnology.com/wp-content/uploads/Alien-Technology-ALR-9900-Enterprise-RFID-Reader.pdf (accessed on 27 November 2013).
42. Alien Technology Corporation. Alien Technology ALR-8696-C Antenna. 2014. Available online: http://www.alientechnology.com/wp-content/uploads/Alien-Technology-ALR-8696-C-Antenna.pdf (accessed on 10 June 2014).
43. Alien Technology Corporation. Alien Technology ALR-9611-CR Antenna. 2014. Available online: http://www.alientechnology.com/wp-content/uploads/Alien-Technology-ALR-9611-CR-Antenna.pdf (accessed on 21 December 2014).