

Research Article

SL-BiLSTM: A Signal-Based Bidirectional LSTM Network for Over-the-Horizon Target Localization

Wanting Yu ¹, Hongyi Yu ², Ding Wang ¹, Jianping Du,² and Mengli Zhang³

¹PLA Strategic Support Force Information Engineering University, Zhengzhou, China

²National Digital Switching System Engineering and Technological Research Center, Zhengzhou, China

³State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China

Correspondence should be addressed to Hongyi Yu; xxgcmayu@163.com

Received 11 March 2021; Accepted 1 July 2021; Published 14 July 2021

Academic Editor: Abdul Qadeer Khan

Copyright © 2021 Wanting Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep learning technology provides novel solutions for localization in complex scenarios. Conventional methods generally suffer from performance loss in the long-distance over-the-horizon (OTH) scenario due to uncertain ionospheric conditions. To overcome the adverse effects of the unknown and complex ionosphere on positioning, we propose a deep learning positioning method based on multistation received signals and bidirectional long short-term memory (BiLSTM) network framework (SL-BiLSTM), which refines position information from signal data. Specifically, we first obtain the form of the network input by constructing the received signal model. Second, the proposed method is developed to predict target positions using an SL-BiLSTM network, consisting of three BiLSTM layers, a maxout layer, a fully connected layer, and a regression layer. Then, we discuss two regularization techniques of dropout and randomization which are mainly adopted to prevent network overfitting. Simulations of OTH localization are conducted to examine the performance. The parameters of the network have been trained properly according to the scenario. Finally, the experimental results show that the proposed method can significantly improve the accuracy of OTH positioning at low SNR. When the number of training locations increases to 200, the positioning result of SL-BiLSTM is closest to CRLB at high SNR.

1. Introduction

High-precision location of the over-the-horizon (OTH) target is a critical issue in the fields of space target surveillance and navigation. Although wireless localization theories and methods have made great progress in recent years, there are still many challenging problems existing in OTH scenarios [1]. The existing passive localization algorithm is mainly classified into two modes [2–6], two-step and direct positioning algorithms. Azimuth of arrival (AOA), time difference of arrival (TDOA), frequency difference of arrival (FDOA), and multiparameter joint estimation methods are two-step positioning mode. Two-step positioning methods based on the second-level information fusion inevitably have information loss. The location precision is strictly limited by the accuracy and matching degree of measuring parameters. Another type

of algorithm is direct position determination (DPD) [7–11], which overcomes the shortcomings in two-step position methods and significantly improves positioning performance under low signal-noise ratio conditions. By establishing the received signal model, DPD obtains information directly from the signal data according to the maximum likelihood criterion. DPD can also integrate channel information (such as ionospheric structure) into the position estimation model [12]. However, the actual OTH scene is extremely complicated, and the shortwave communication often relies on ionospheric reflection, resulting in a serious decrease in location precision [7, 12]. Uncertain ionospheric conditions and lower signal-noise ratio make it difficult for the existing position algorithms to model complex channel scenes. Thus, OTH positioning still faces huge challenges even with current advanced DPD algorithms.

Meanwhile, deep learning technology [13] developed rapidly and had achieved effects comparable to humans in image recognition [14], speech, and natural language processing [15, 16]. Deep neural networks, such as convolutional neural networks (CNN) and long short-term memory (LSTM) [17], have a high degree of nonlinearity and information fusion capabilities. Deep learning technology provides new solutions to solve the localization problem in complex scenarios. On the one hand, the complex environment of indoor localization has greatly promoted the application and research of deep learning technology in indoor localization [18–21]. Recently, some fingerprint indoor positioning methods with deep learning [22, 23] are proposed to solve the multipath fading effect. In 2020, Zhou et al. [24] proposed to combine the backpropagation neural network (BPNN) and adaptive genetic algorithm (AGA) with CSI tensor decomposition for indoor Wi-Fi fingerprint localization. On the other hand, scholars also begin to utilize some deep learning methods for outdoor large-scale sensor network positioning [22], passive localization [23, 24], and target tracking [25–27]. Especially when solving the localization problems under many nonideal scenarios, which are difficult for traditional methods to handle, deep learning methods show great advantages [28]. However, there is currently no research on deep learning methods suitable for OTH positioning. Owing to the powerful ability of LSTM in handling sequential data and capturing temporal dependencies, we propose an SL-BiLSTM for long-distance OTH positioning, which is a deep learning structure. In contrast to the DPD algorithm [7–11], our SL-BiLSTM algorithm can directly predict the position without any information of the signal path once it has been well trained.

The rest of this paper is organized in the following manner: in Section 2, we build the received signal model of the OTH target and get the appropriate data form to input into the network. In Section 3, the SL-BiLSTM network is constructed to learn the localization features from signal data based on the bidirectional LSTM structure. We utilize two regularization methods to avoid network overfitting and give a summary of our method. In Section 4, we build the simulation scenario of OTH positioning to get signal data for network learning. Hyperparameters and the number of training locations are discussed to train our model. Then, a comparison between traditional methods, DPD, and the proposed SL-BiLSTM is investigated. Finally, our work is summarized in Section 5.

2. The Received Signal Model of OTH Target

The multiarray passive location of the OTH target is presented in Figure 1. The antenna arrays of multiple receiving stations receive signals and transmit the signal data to the central station. The central station combines the signals received by each station and the prior observation information of the ionosphere to directly estimate the target position. OTH positioning has the characteristics of non-line-of-sight and long target distance, which is mainly realized by shortwave communication and ionosphere reflection. To simplify the model, the positioning model in this

paper only considers the influence of ionospheric virtual heights on the signal propagation path, assuming that the geometric structure of the signal reflected by the ionosphere is symmetrical.

Consider a transmitter and L base stations intercepting the transmitted signal. Each base station is equipped with an antenna array composed of M elements. In this paper, we mainly consider an X - Y plane coordinate of localization. Thus, the two-dimensional coordinates of target position and l th base station position are defined as $\mathbf{p} = [x_0, y_0]^T$ and $\mathbf{q}_l = [x_l, y_l]^T$. The signal observed by the l th base station array is given by

$$\mathbf{r}_l(t) = \mathbf{a}_l(\mathbf{p})\mathbf{O}_l(\mathbf{p})s(t - t_0) + \mathbf{n}_l(t), \quad 0 \leq t \leq T, \quad (1)$$

where $\mathbf{r}_l(t)$ is a time-dependent $M \times 1$ vector, $\mathbf{a}_l(\mathbf{p})$ is the l th array response to a signal transmitted from position \mathbf{p} , $\mathbf{O}_l(\mathbf{p})$ is a complex matrix representing the channel effect, and $s(t - t_0)$ is the signal waveform, transmitted at time t_0 . The vector $\mathbf{n}_l(t)$ represents noise and interference, including multipath observed by the array. T presents the length of observing time. The sampled version of the signal in (1) is given by

$$\mathbf{r}_l(k) = \mathbf{a}_l(\mathbf{p})\mathbf{O}_l(\mathbf{p})s(k) + \mathbf{n}_l(k), \quad 0 \leq k \leq N_s - 1, \quad (2)$$

$$s_l(k) = s(t - t_0)|_{t=kT}, \quad (3)$$

$$\mathbf{r}_l(k) = \mathbf{r}_l(t)|_{t=kT}, \quad (4)$$

$$\mathbf{n}_l(k) = \mathbf{n}_l(t)|_{t=kT}. \quad (5)$$

The received data of each station can be combined into a matrix:

$$\mathbf{r}(k) = [\mathbf{r}_1^T(k), \mathbf{r}_2^T(k), \dots, \mathbf{r}_L^T(k)]^T. \quad (6)$$

Then, the discrete Fourier transform (DFT) of the signal in (2) is given by

$$\bar{\mathbf{r}}_l(j) = \mathbf{a}_l(\mathbf{p})\bar{\mathbf{O}}_l(j)s(j)e^{-i\omega_j t_0} + \bar{\mathbf{n}}_l(j), \quad j = 1, 2, \dots, J, \quad (7)$$

where $j = 1, 2, \dots, J$ indicates the DFT coefficient of the corresponding time samples. We assume that the propagation path of electromagnetic waves is mainly affected by ionospheric virtual heights at reflection points [12]. Therefore, the channel effect $\mathbf{O}_l(\mathbf{p})$ which is mainly composed of signal transmission delay and attenuation can be calculated as

$$\bar{\mathbf{O}}_l(j) = b_l e^{-i\omega_j \tau_l(\mathbf{p}, \mathbf{h})}, \quad (8)$$

where b_l is an unknown complex scalar representing the attenuation related to the l th base station, $\tau_l(\mathbf{p}, \mathbf{h})$ is the l th station signal delay transmitted from position \mathbf{p} , and $\mathbf{h} = [h_1, h_2, \dots, h_l]^T$ is the unknown ionospheric virtual heights at reflection points. It is well known that the distribution of the ionosphere is related to the spatial position. In this paper, we define $h_l = f(\mathbf{p}_{lr}) = f(x_{lr}, y_{lr})$, where $\mathbf{p}_{lr} = [x_{lr}, y_{lr}]^T$ is the coordinate reflection point, and $f(\cdot)$ is the nonlinear

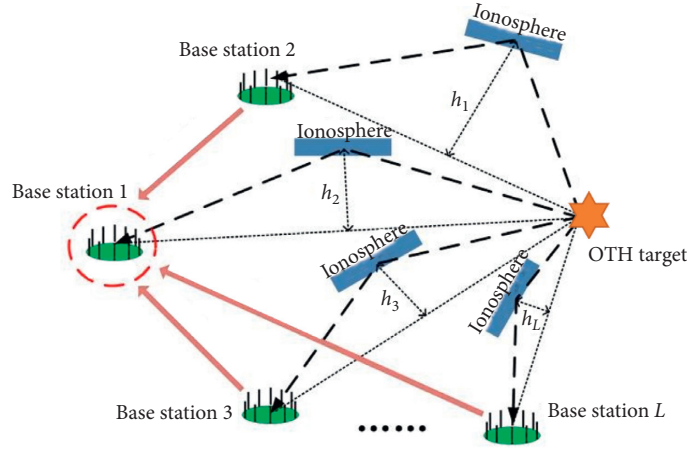


FIGURE 1: The multiarray passive location of the OTH target.

mapping from space position to ionospheric virtual height. We set $T \gg \max\{\tau_l(\mathbf{p}, \mathbf{h})\}$ to ensure a reasonable range of time delay estimation.

Define the vectors:

$$\begin{aligned} \mathbf{A}_l(j) &= b_l \mathbf{a}_l(\mathbf{p}) e^{-i\omega_j \tau_l(\mathbf{p}, \mathbf{h})} = \mathbf{F}(j, \mathbf{p}, \mathbf{h}) \mathbf{Z}, \\ \mathbf{F}(j) &= \text{diag}\{\mathbf{a}_1^T(\mathbf{p}) e^{-i\omega_j \tau_1(\mathbf{p}, \mathbf{h})}, \dots, \mathbf{a}_L^T(\mathbf{p}) e^{-i\omega_j \tau_L(\mathbf{p}, \mathbf{h})}\}, \\ \bar{\mathbf{s}}(j) &= \mathbf{s}(j) e^{-i\omega_j t_0}, \\ \mathbf{b} &= [b_1, b_2, \dots, b_L]^T, \\ \mathbf{Z} &= \mathbf{I}_{L \times L} \otimes \mathbf{1}_{M \times 1}, \end{aligned} \quad (9)$$

where \otimes represents the Kronecker product, $\mathbf{I}_{L \times L}$ represents the unit matrix of size $L \times L$, and $\mathbf{1}_{M \times 1}$ is M column vector whose elements are all one. We can rewrite (7) as

$$\bar{\mathbf{r}}_1(j) = \mathbf{A}(j) \mathbf{b} \bar{\mathbf{s}}(j) + \mathbf{n}(j), \quad j = 1, 2, \dots, J. \quad (10)$$

Then, the received data of each station can be combined into a matrix:

$$\mathbf{r}(j) = [\bar{\mathbf{r}}_1^T(j), \bar{\mathbf{r}}_2^T(j), \dots, \bar{\mathbf{r}}_L^T(j)]^T. \quad (11)$$

The next step is to determine the probable location information of the Q radio sources from $\mathbf{r}(j)$. If we assume that the virtual height \mathbf{h} of the ionosphere is known, the Cramér-Rao lower bound (CRLB) of the conventional DPD algorithm is provided in [7, 12]. However, ionospheric parameters including virtual height distribution are uncertain and changeable, so we need highly nonlinear algorithms to handle the localization of complex scenes.

3. Proposed Location Method

3.1. Signal-Based Localization BiLSTM. Since LSTM has the powerful ability in handling sequential data and capturing temporal dependencies in data, it has been successfully applied in many applications [29–31]. In this paper, we utilize a BiLSTM network based on signal data for the OTH target localization. A standard structure of BiLSTM can be

found in Figure 2, and the architecture of LSTM is formulated as follows:

$$\begin{aligned} \mathbf{i}_k &= \sigma(\mathbf{W}_i \mathbf{x}_k + \mathbf{Q}_i \mathbf{h}_{k-1} + \mathbf{b}_i), \\ \mathbf{f}_k &= \sigma(\mathbf{W}_f \mathbf{x}_k + \mathbf{Q}_f \mathbf{h}_{k-1} + \mathbf{b}_f), \\ \mathbf{o}_k &= \sigma(\mathbf{W}_o \mathbf{x}_k + \mathbf{Q}_o \mathbf{h}_{k-1} + \mathbf{b}_o), \\ \theta_k &= \tanh(\mathbf{W}_g \mathbf{x}_k + \mathbf{Q}_g \mathbf{h}_{k-1} + \mathbf{b}_g), \\ \mathbf{c}_k &= \mathbf{f}_k \circ \mathbf{c}_{k-1} + \mathbf{i}_k \circ \theta_k, \\ \mathbf{h}_k &= \mathbf{o}_k \circ \tanh(\mathbf{c}_k), \end{aligned} \quad (12)$$

where \circ denotes Hadamard product, \tanh is the hyperbolic tangent function, and \mathbf{x}_k is the input vector. $\mathbf{W}_m, \mathbf{Q}_m, \mathbf{b}_m$ ($m \in \{i, f, o, g\}$) are learnable parameters. The output of LSTM at time k is \mathbf{h}_k .

Location features of OTH targets are difficult to model and have high complexity. The received signal segments from each base station at their specific location can be treated as data with complex position feature distribution and multiple noises. Our proposed architecture of SL-BiLSTM is shown in Table 1, which is developed with a BiLSTM encoder (including three BiLSTM layers and a maxout layer [32]), a fully connected layer, and a regression layer. Received signal segments are fed into the BiLSTM encoder to obtain the high-level encoding for the localization feature. The fully connected layer and linear regression layer are utilized to estimate final target positions. To learn the target position from the complex data, a large amount of signal data from different positions is required.

Under the same assumption as [7, 12] that the signal waveform is known to the receivers, we consider the problem of OTH localization as learning the mapping from received signal segments $X = \{\mathbf{r}^i(t)\}_{t=1}^n$ to the target location $Y = \{(x_0^i, y_0^i)\}_{i=1}^n$. In our model, X are the T long signal segments of L base stations. As shown in Figure 1, we compute $\mathbf{r}(k)$ as equation (6) of these signal segments and use both their real and imaginary parts as the network input. Through the SL-BiLSTM encoder and regression layer, we can get the predicted coordinates of target locations $\hat{\mathbf{p}}^i = (\hat{x}_0^i, \hat{y}_0^i)$. Then, we calculate the mean square errors of

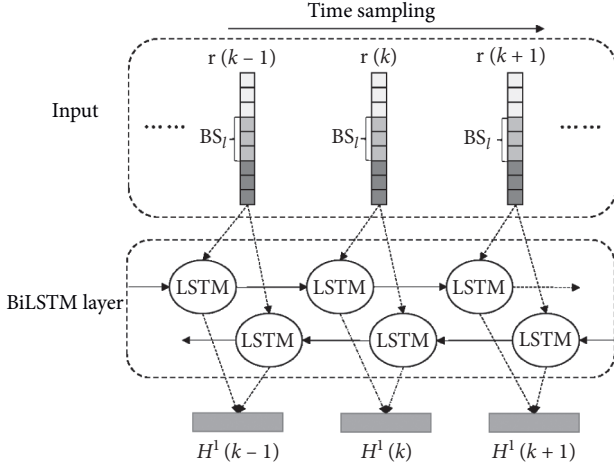


FIGURE 2: Structure of the standard BiLSTM. BS_{*l*}: *l*th base station. $H^1(k)$: the output of the first BiLSTM layer.

the predicted locations and the true locations of training samples and backpropagate the prediction loss to update our SL-BiLSTM parameters. The prediction loss is calculated as

$$L_Y(X, Y) = \|F_Y(X) - Y\|_2^2, \quad (13)$$

where $F_Y(X) = F_Y(\mathbf{r}^i(t), W_b, W_r)$ and W_b, W_r are the weight of BiLSTM and regression layer. We adopt an optimization method named Adam [33] to calculate the adaptive learning rate of network parameters W_b and W_r and then optimize the parameters on the objective function:

$$(\widehat{W}_b, \widehat{W}_r) = \arg \min_{W_b, W_r} L_Y(X, Y). \quad (14)$$

Specifically, assuming that θ_t is the parameter of the network and g_t is the corresponding gradient, Adam algorithm gets the updating θ_{t+1} as

$$\begin{aligned} \alpha_t &= \lambda_1 \alpha_{t-1} + (1 - \lambda_1) g_t, \\ \beta_t &= \lambda_2 \beta_{t-1} + (1 - \lambda_2) g_t^2, \\ \alpha_t &= \frac{\alpha_t}{(1 - \lambda_1)}, \\ \beta_t &= \frac{\beta_t}{(1 - \lambda_2)}, \\ \theta_{t+1} &= \theta_t + \frac{\gamma}{\sqrt{\beta_t} + \varepsilon} \alpha_t, \end{aligned} \quad (15)$$

where α_t and β_t are the first and second moments of the gradient, respectively; γ is the learning rate; and λ_1, λ_2 , and ε are set to be 0.9, 0.999, and 10^{-8} , respectively. In this paper, we select the step declining learning rate (StepLR) as our learning rate reduction strategies with an initial learning rate γ .

3.2. Regularization Methods. Deep networks often become overfitting, especially in regression problems where the number of training samples is limited. To obtain a robust

TABLE 1: Network structure of SL-BiLSTM.

Name	Configuration
Input layer	Initial size: $M \times L$, length: Ns
BiLSTM Block 1	Hidden size: 256, dropout: 0.2, length: Ns
BiLSTM Block 2	Hidden size: 256, dropout: 0.2, length: Ns
BiLSTM Block 2	Hidden size: 512, dropout: 0.2, length: Ns
Maxout layer	Output size: 128, sigmoid, dropout: 0.2
FCL layer	Output size: 64, sigmoid, dropout: 0.5
Output layer	Output size: 2, linear

network that avoids overfitting the training data, appropriate regularization methods are essential [34]. In this paper, techniques of dropout and randomization are mainly adopted to prevent overfitting.

3.2.1. Dropout Layers. First, we adopt two dropout layers after the last layer of the LSTM and the fully connected layer, respectively. In the dropout layer, parts of the neuron are randomly masked with a certain probability during each round of parameter update. Assuming that the dropout probability is p_d and \mathbf{w}_i is the node weight during round i , the output z_i at $(l+1)$ th layer during the training process can be expressed as

$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)} (\mathbf{r}_i * \mathbf{y}_i^{(l)}) + b_i^{(l+1)}, \quad (16)$$

where $\mathbf{r}_i \sim \text{Bernoulli}(p_d)$, $\mathbf{y}_i^{(l)}$ is the output of the previous layer, and $b_i^{(l+1)}$ is the network bias parameter. To achieve better generalizability for coordinate regression, the dropout rate is increased from 0.1 to 0.6. The maxout layer is also a dropout method which deletes a part of the network that is not sensitive to the input data to reduce the risk of overfitting.

3.2.2. Sample Randomization. Second, since the number of training samples in actual scenes is limited, the distribution characteristics of training samples that are not related to the localization scene may also cause overfitting. For example, we cannot directly apply the grid points in the area as training samples, because this will cause the neural network to overlearn “on-grid” as one of the location features. Therefore, we uniformly randomize the positions of the training samples in the target area. The specific operation is illustrated in Figure 3. We divide the normalized area evenly (gray line) according to the predetermined sample number and then randomly select the training sample position (blue dots) within the divided subregions. This regularization method not only ensures the adaptability of the network to the entire positioning area but also avoids the overfitting problem caused by the limited training samples.

We take I_t target emission source location as training samples (blue dots) to train our network, and randomly generate I_v validation samples (red circles) to verify the network convergence performance. Finally, 100 Monte Carlo simulation experiments were performed under several SNR conditions at five test positions (red asterisks).

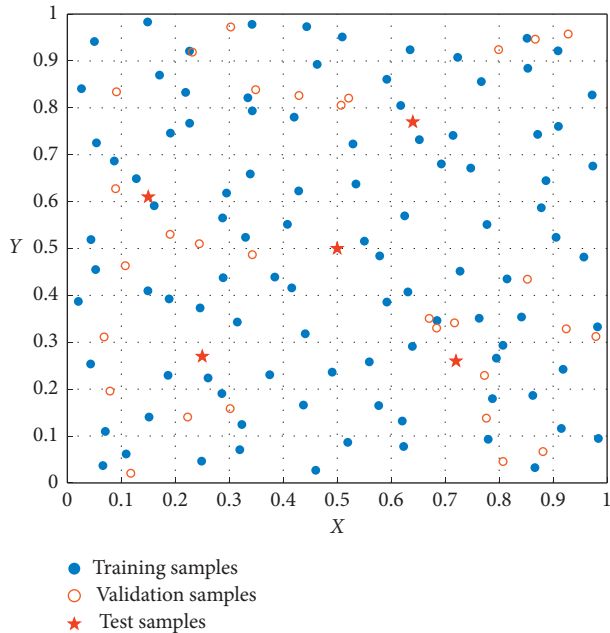


FIGURE 3: Training samples, validation samples, and test samples in the normalized target area. The dotted lines indicate the division of subregions, the blue dots indicate the location of training samples, the red circles indicate the location of validation samples, and the red asterisks indicate the location of test samples.

3.3. *Summary.* Based on the OTH localization model, our SL-BiLSTM is summarized in Algorithm 1. We first create two different datasets to train and update our network which can directly output the predicted location once well-trained.

4. Experiments

In order to examine the method's performance and compare it with existing approaches, we perform extensive Monte Carlo simulations. Since the OTH localization problem is related to the ionosphere, traditional localization systems usually utilize the multistation direction finding and intersection method for positioning. Thus, AOA and DPD methods based on the MUSIC algorithm [2, 11] and the OTH DPD (ODPD) [7, 12] method with fixed virtual height hypothesis are selected for performance comparison. We use machines equipped with Intel Core i7CPU, 32 GB RAM, and NVIDIA GeForce RTX 2080Ti GPUs to train our model and complete simulation experiments.

4.1. *Simulation Scenarios.* Consider three base stations placed at three corners of $2000 \text{ km} \times 2000 \text{ km}$ square as shown in Figure 4, and their coordinates are shown in Table 2. Each base station is equipped with a circular array of nine antenna elements. The radius of the array is set at one wavelength. The OTH target area is $1000 \text{ km} \times 1000 \text{ km}$ square. As shown in Table 2, these 5 test locations are not the same as the 100 locations (denoted as training locations with labels 1, 2, . . . , 100 in Table 2 and Figure 3) in the training dataset and the coordinates are accurate to two decimal places to ensure high-precision positioning capability.

The actual distribution of the ionosphere changes with time and geographic location. In this scenario, we ignore time-varying effects and the influence of the Earth sphere. Conventional methods often assume that the ionospheric virtual height is a known constant, which increases the positioning error. Without loss of generality, we use the DCT low-dimensional coefficients of a random matrix (uniformly distributed in $[160 \text{ km}, 240 \text{ km}]$) to simulate the spatial distribution of the ionospheric virtual height, as shown in Figure 5. Since the reflection point is difficult to measure, we assume that the reflection is geometrically symmetric to simplify the model. Then, we consider the situation that the propagation path of the signal through the ionosphere is only affected by ionospheric virtual heights at reflection points. Thus, ionospheric virtual height can be written as $h_l = f_c(\mathbf{p}_{lr}) = f_c[(\mathbf{p} + \mathbf{q}_l)/2]$. The emission signal is an 8PSK signal with a known waveform, the sampling frequency is 1 MHz with ten times of sampling per symbol. The number of signal samples per position is 100. Each location prediction is based on 64 times of samples of the signal. The path-loss attenuation magnitude is set to be normal distribution (mean = 1, STD = 0.1) with phase uniformly distributed in $[-\pi, \pi]$. Then, we can calculate the received signal waveform matrix of the corresponding target position according to the model in Section 2 as the network input data. We combine all the position labels and the corresponding received waveform data to finally form the dataset required for network training. We utilize the root mean square error (RMSE) of the target positioning obtained by the Monte Carlo experiment to measure the localization performances:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{p}}_i - \mathbf{p}\|_2^2}. \quad (17)$$

4.2. *Discussion on Training Hyperparameters.* In this subsection, we discuss the training hyperparameters, including the learning rate, batch size, dropout rate, and training loss threshold. Appropriate hyperparameters will improve network convergence performance to a certain extent. The method of selecting network hyperparameters is usually based on data characteristics and empirical. In this paper, considering the data dimension and network scale, we first adjust the batch sizes and the corresponding learning rate (initial rate and delay step) according to the training loss. Then, we adjust the dropout rate from 0 to 0.5 according to the validation loss to avoid the network from falling into overfitting.

In the entire training procedure, different hyperparameters are used to train the network. Both 64 and 128 are suitable batch sizes. Considering the GPU parallel efficiency and learning rate decay strategy, $N_b = 128$ is finally selected. In addition, if the initial learning rate is too large or too small, it may make it difficult for the network to converge to the best result. After multiple sets of parameters debugging, we finally select 0.001 as the initial learning rate.

Input: The received signals $X_t = \{\mathbf{r}^i(t)\}_{i=1}^{I_t}$, $X_v = \{\mathbf{r}^j(t)\}_{i=1}^{I_v}$, $Y_t = \{(x_0^i, y_0^i)\}_{i=1}^{I_t}$, $Y_v = \{(x_0^j, y_0^j)\}_{i=1}^{I_v}$.

Initialize: Learning rate γ , decline step d , training batch size N_b , maximum train epoch 500; the initial state (h_1^0, c_1^0) of LSTM is $(0, 0)$; and network weight W^0 is zeros;

- (1) **Dataset creation:** Training dataset D_t (data, label) = $\{X_t, Y_t\}$, validation dataset D_v (data, label) = $\{X_v, Y_v\}$.
- (2) **for** each epoch **do**
- (3) **for** batch index $k(X_t^k, Y_t^k)$ **in** D_t
- (4) $\hat{\mathbf{p}}^k = \text{SL-BiLSTM}(X_t^k, W^{k-1})$.
- (5) $L_{Y_t}(X_t^k, Y_t^k) = \|\hat{\mathbf{p}}^k - Y_t^k\|_2^2$.
- (6) Update $W^k \leftarrow \text{Adma}(W^{k-1} + (\gamma/\sqrt{\beta} + \varepsilon)\alpha)$.
- (7) **end for**
- (8) Evaluate network performance on D_v (data, label) = $\{X_v, Y_v\}$
- (9) **for** X_v, Y_v **in** D_v
- (10) $L_{Y_v}(X_v, Y_v) = \|\text{SLBiLSTM}(X_v; W) - Y_v\|_2^2$
- (11) **end for**
- (12) **end for**

Output: Training and validation loss and network weights W .

Test: New received signals $X = \{\mathbf{r}^i(j)\}_{i=1}^N$ from positions $Y = \{(x_0^i, y_0^i)\}_{i=1}^N$. The predicted locations are $\{\hat{\mathbf{p}}\}_{i=1}^N = \text{SLBiLSTM}(X; W)$.

ALGORITHM 1: SL-BiLSTM for OTH localization.

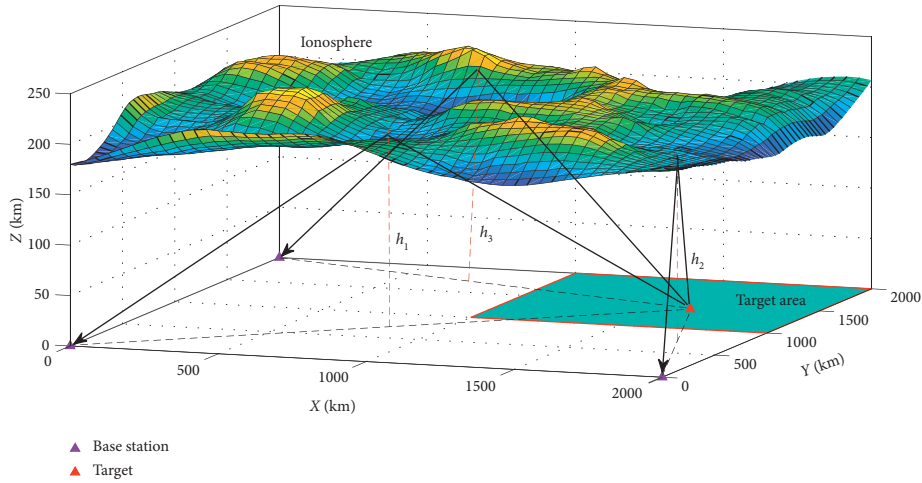


FIGURE 4: Geometry diagram of the OTH localization scenario and the target area.

TABLE 2: Configuration of base stations, training locations, and test locations.

BS	1	2	3			
X (km)	0	0	2000			
Y (km)	0	2000	0			
Training	1	2	3	...	99	100
X (km)	1065.87	1070.35	1043.12	...	1972.30	1909.17
Y (km)	1036.99	1109.90	1253.71	...	1827.53	1921.63
Test	A	B	C	D	E	
X (km)	1500	1150.25	1640.75	1250.25	1720.50	
Y (km)	1500	1610.50	1770.25	1270.75	1260.50	

Finally, increasing the dropout rate to 0.5 can effectively avoid network overfitting and the loss of validation data gradually converges. The five sets of parameters and corresponding network losses are given in Table 3, where parameter (d) is set as the final parameter. We use machines equipped with NVIDIA GeForce RTX 2080 Ti GPUs to train our model.

4.3. Experimental Results and Discussions. In this subsection, we evaluate the performance of our method through the experimental results of $N_M = 100$ Monte Carlo simulations at 5 testing target locations. First, Figure 5 shows the predicted locations of 5 testing targets for 100 experiments and their RMSE at SNR 10 dB, and the corresponding CDF curves of localization error are shown in Figure 6. Then, we

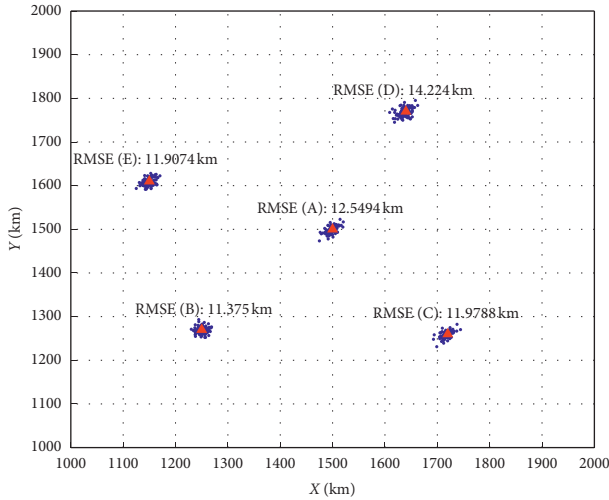


FIGURE 5: Predicted locations of five testing targets for 100 experiments and their RMSE at SNR 10 dB.

TABLE 3: Configuration of base stations, training locations, and test locations.

Parameters	(a)	(b)	(c)	(d)	(e)
N_b	64	64	128	128	200
γ	0.0001	0.001	0.001	0.001	0.001
d	50	50	20	20	20
p_d	0	0.2	0.2	0.5	0.5
Training epoch	500	500	500	400	400
Training loss	0.05379	0.03940	0.03581	0.03581	0.03547
Validation loss	0.12884	0.06163	0.07201	0.03616	0.03624

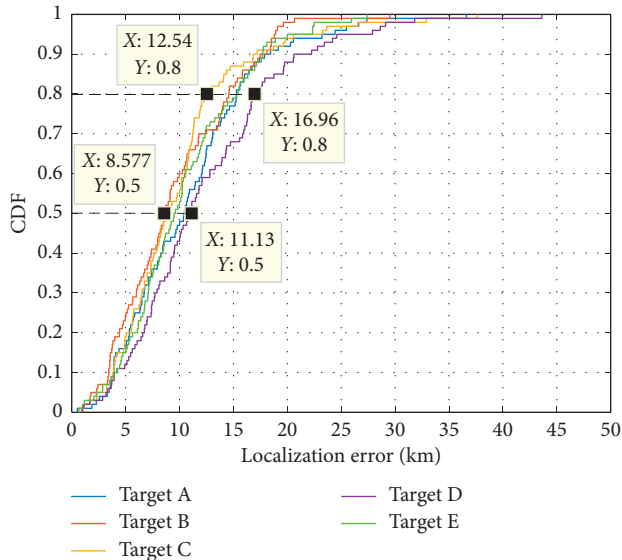


FIGURE 6: CDF curves of localization error using the proposed SL-BiLSTM network for five testing locations (targets A, B, C, D, and E).

select several SNR values between -10 dB and 10 dB. At each SNR value, we conducted 100 experiments on target A to obtain the performance statistics of our proposed method. In

traditional positioning systems, the OTH problem usually adopts the method of multistation direction finding and intersection. Therefore, we select the following different approaches for performance comparison: AOA estimation based on MUSIC algorithm, MUSIC-DPD, ML-DPD, and ODPD [11, 12]. The comparison curve of the localization RMSE performance of the above-mentioned methods is shown in Figure 7.

Figures 5 and 6 indicate that although the localization accuracy of different geometric positions is different, the results of 100 location experiments on five test targets are still stable. Target B has the highest accuracy, with 50% and 80% localization errors within 8.57 km and 12.54 km, respectively. Location accuracy of target D is relatively low, and 50% and 80% localization errors are within 11.13 km and 16.96 km. The plots in Figure 7 indicate that SL-BiLSTM is superior to AOA, MUSIC-DPD, ML-DPD, and ODPD, especially at low SNR. Owing to the uncertainty of ionospheric distribution, all methods fail to reach the theoretical level even at high SNR.

To adapt to complex situations that are difficult to model by conventional mathematical methods, deep learning models often need a large number of parameters and gradient calculations, which makes the training process take more time. However, the training process of deep learning models is generally completed offline, and the trained model can be directly used for online positioning tasks. The comparison results of the online positioning time complexity of 100 Monte Carlo experiments are given in Table 4. The results show that the well-trained SL-BiLSTM model is significantly faster than other methods when performing online positioning.

4.4. Impact of the Number of Training Locations. The network performance depends on the training dataset to a certain extent. In this paper, the impact of the dataset is mainly reflected in the number and distribution of training locations. Therefore, we further study the influence of the number of training locations on the positioning accuracy in the target area in Figure 5. The localization RMSE varies with SNR, corresponding to the number of training locations 50, 100, and 200, shown in Figure 8.

The experimental results show that, as training positions increase, the positioning accuracy increases accordingly. The performance advantage of the proposed method is more obvious at low SNR. When training locations are insufficient such as the number of 50, the network performance does not reach the desired accuracy even at a high SNR. In addition, training locations represent the global reference of the entire area, while the accuracy of different locations is different. Using global reference data to estimate a single target may lose part of the local accuracy. Choosing appropriate reference locations according to actual scenarios is also one of the issues that worth further researching. In summary, SL-BiLSTM can learn complex channel features from signal data after proper training and finally give more accurate location predictions.

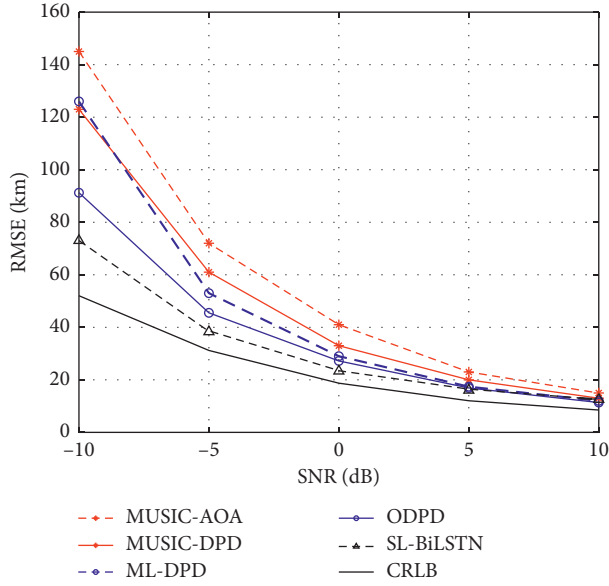


FIGURE 7: RMSE for five different methods and the CRLB.

TABLE 4: Time complexity of MUSIC-AOA, MUSIC-DPD, ML-DPD, ODPD, and SL-BiLSTM methods with 100 Monte Carlo experiments, respectively.

Method	MUSIC-AOA	MUSIC-DPD	ML-DPD	ODPD	SL-BiLSTM
Time	0.82 s	51.96 s	51.35 s	56.02 s	0.16 s

5. Conclusions

In this paper, a deep learning method for OTH positioning called SL-BiLSTM was proposed. The proposed SL-BiLSTM encodes location features in signal data based on BiLSTM structure and obtains a position estimation model by training on reference locations. The number of reference positions is limited in practical applications. We utilize regularization methods to solve the network overfitting caused by these limitations. Simulation experiments verify that SL-BiLSTM has higher positioning accuracy in OTH scenarios, while conventional localization methods generally suffer performance losses.

Our work provides a novel method and experimental basis for long-distance OTH positioning. There are still many points worthy of further study, such as how to choose as few reference locations as possible to achieve high accuracy. The experimental results further also illustrate that deep learning methods have obvious advantages especially in complex scenarios that cannot be modelled.

The deep learning method is affected by the scene data, so in the application with the proposed method, we need to consider different OTH positioning problems with different distributions of the ionosphere. When the ionosphere distribution is significantly different, the model will need more data to be further adjusted. In addition, since the ionosphere distribution will change with time, to improve the generalization performance of our SL-BiLSTM network, data with sufficient observation time and improved models are

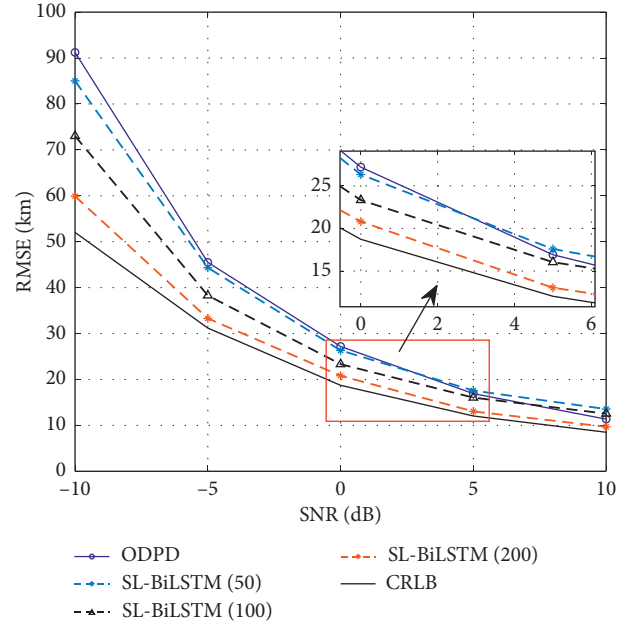


FIGURE 8: The localization RMSE corresponding to the number of training locations 50, 100, and 200.

required for sufficient pretraining of the network, which is a promising direction for future work.

Data Availability

The method of generating simulation data has been described in the experiments of the paper. All data included in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors acknowledge support from National Natural Science Foundation of China (Grant nos. 62071029, 61772548 and 61201381), China Postdoctoral Science Foundation (Grant no. 2016M592989), and Key Scientific and Technological Research Project in Henan Province (Grant nos. 192102210092 and 192102210117).

References

- [1] J. M. Kelso, "Ray tracing in the ionosphere," *Radio Science*, vol. 3, no. 1, pp. 1–12, 2016.
- [2] Q. S. Ren and A. J. Willis, "Extending MUSIC to single snapshot and on line direction finding applications," in *Proceedings of the Radar 97 (Conference Published No 449)*, pp. 783–787, Edinburgh, UK, October 1997.
- [3] K. Yang, J. An, X. Bu, and G. Sun, "Constrained total least-squares location algorithm using time-difference-of-arrival measurements," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1558–1562, 2010.

- [4] M. R. Gholami, S. Gezici, and E. G. Strom, "A concave-convex procedure for TDOA based positioning," *IEEE Communications Letters*, vol. 17, no. 4, pp. 765–768, 2013.
- [5] Y. L. Wang and Y. Wu, "An efficient semidefinite relaxation algorithm for moving source localization using TDOA and FDOA measurements," *IEEE Communications Letters*, vol. 21, no. 1, pp. 80–83, 2017.
- [6] G. Wang, Y. Li, and N. Ansari, "A semidefinite relaxation method for source localization using TDOA and FDOA measurements," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 853–862, 2013.
- [7] A. Amar and A. J. Weiss, "Direct position determination in the presence of model errors-known waveforms," *Digital Signal Processing*, vol. 16, no. 1, pp. 52–83, 2006.
- [8] T. Tirer and A. J. Weiss, "High resolution direct position determination of radio frequency sources," *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 192–196, 2016.
- [9] E. Tzoreff and A. J. Weiss, "Expectation-maximization algorithm for direct position determination," *Signal Processing*, vol. 133, pp. 32–39, 2017.
- [10] T. Tirer and A. J. Weiss, "Performance analysis of a high-resolution direct position determination method," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 544–554, 2017.
- [11] J. Du, D. Wang, W. Yu, and Y. Hong-Yi, "Direct position determination of unknown signals in the presence of multipath propagation," *Sensors*, vol. 18, no. 3, 2018.
- [12] W. Yu, H. Yu, J. Du, and D. Wang, "A direct position determination method for over-the-horizon target on known radiation source waveforms," *Acta Electronica Sinica*, vol. 441, no. 11, pp. 130–139, 2019.
- [13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 770–778, IEEE Computer Society, Las Vegas, NV, USA, June 2016.
- [15] H. Li, J. Zhu, J. Zhang, C. Zong, and X. He, "Keywords-guided abstractive sentence summarization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, pp. 8196–8203, 2020.
- [16] G. Hinton, L. Deng, Y. Dong, and G. E. Dahl, "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.
- [17] L. Houegnigan, P. Safari, C. Nadeu, M. André, and M. van der Schaar, "Machine and deep learning approaches to localization and range estimation of underwater acoustic sources," in *Proceedings of the 2017 IEEE/OES Acoustics in Underwater Geosciences Symposium (RIO Acoustics)*, pp. 1–6, Rio de Janeiro, Brazil, July 2017.
- [18] D. V. Le, N. Meratnia, and P. J. M. Havinga, "Unsupervised deep feature learning to reduce the collection of fingerprints for indoor localization using deep belief networks," in *Proceedings of the 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–7, Nantes, France, September 2018.
- [19] H. Jiang, C. Peng, and J. Sun, "Deep belief network for fingerprinting-based RFID indoor localization," in *Proceedings of the ICC 2019–2019 IEEE International Conference on Communications (ICC)*, pp. 1–5, Shanghai, China, May 2019.
- [20] W. He, P. Motlicek, and J. Odobez, "Adaptation of multiple sound source localization neural networks with weak supervision and domain-adversarial training," in *Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 770–774, Brighton, UK, May 2019.
- [21] Z. Chen, H. Zou, J. Yang, H. Jiang, and L. Xie, "Wi-Fi fingerprinting indoor localization using local feature-based deep LSTM," *IEEE Systems Journal*, vol. 14, no. 2, pp. 3001–3010, 2019.
- [22] J. Li, W. Hsu, and C. Peng, "Fingerprint indoor positioning method with deep learning," in *Proceedings of the 8th International Conference on Innovation, Communication and Engineering (ICICE)*, pp. 117–120, Zhengzhou, China, October 2019.
- [23] Z. Zhang, M. Lee, and S. Choi, "Deep learning-based indoor positioning system using multiple fingerprints," in *Proceedings of the International Conference on Information and Communication Technology Convergence*, pp. 491–493, Jeju, Korea (South), October 2020.
- [24] M. Zhou, Y. Long, and W. Zhang, "Adaptive genetic algorithm-aided neural network with channel state information tensor decomposition for indoor localization," *IEEE Transactions on Evolutionary Computation*, vol. 99, p. 1, 2021.
- [25] B. G. Maciel-Pearson, S. Akçay, A. Atapour-Abarghouei, C. Holder, and T. P. Breckon, "Multi-task regression-based learning for autonomous unmanned aerial vehicle flight control within unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4116–4123, 2019.
- [26] Y. Xue, W. Su, H. Wang, D. Yang, and Y. Jiang, "DeepTAL: deep learning for TDOA-based asynchronous localization security with measurement error and missing data," *IEEE Access*, vol. 7, pp. 122492–122502, 2019.
- [27] L. Shen, Q. Zhang, J. Pang, H. Xu, and P. Li, "PRDL: relative localization method of RFID tags via phase and RSSI based on deep learning," *IEEE Access*, vol. 7, pp. 20249–20261, 2019.
- [28] Y. Yang and Z. Chen, "Pedestrian tracking algorithm for dense crowd based on deep learning," in *Proceedings of the 2019 6th International Conference on Systems and Informatics (ICSIAI)*, pp. 568–572, Shanghai, China, November 2019.
- [29] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.
- [30] X. Wang, Z. Yu, and S. Mao, "DeepML: deep LSTM for indoor localization with smartphone magnetic and light sensors," in *Proceedings of the IEEE International Conference Communications (ICC)*, pp. 1–6, Kansas, MO, USA, May 2018.
- [31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [32] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1319–1327, Atlanta, GA, USA, June 2013.
- [33] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.
- [34] G. E. Hinton, N. Srivastava, and A. Krizhevsky, "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.