

Application of A Long Short-Term Memory Neural Network Algorithm Fused with ResNet in UWB Indoor Positioning

Chunhua Zhu, Manyi Li, Mingyang Cui

Abstract—Ultra-wideband (UWB) signals are susceptible to multipath effects and non-line-of-sight (NLOS) propagation in indoor environments, necessitating the simultaneous capture of the signal's temporal dynamic changes and spatial structural characteristics. To improve the localization accuracy and real-time tracking capability, a novel UWB indoor localization algorithm is proposed that integrates ResNet and LSTM neural networks, fusing the spatiotemporal characteristics of UWB signals. Initially, the distance data collected by the UWB device is transformed into the time series, then, the ResNet is employed to extract high-dimensional features and suppress the impact of noise and multipath effects, and the LSTM network is integrated with ResNet to capture temporal dependencies, thereby enhancing positioning accuracy. Simulation results indicate that the proposed ResNet-LSTM algorithm outperforms traditional BP neural networks and LSTM networks in terms of localization accuracy, error stability, and noise immunity, effectively improving the performance of UWB indoor positioning.

Index Terms—UWB, Indoor localization, ResNet, LSTM

I. INTRODUCTION

Ultra-wideband (UWB) technology is a carrierless communication method, utilizing nanosecond narrow non-sinusoidal pulses to transmit data. By measuring the distance or angle between the base station and the tag, the tag's position can be determined using various geometric measurement approaches, including Angle-of-Arrival (AOA) [1], Received Signal Strength (RSSI), Time of Arrival (TOA) [2-3], Time Difference of Arrival (TDOA), Time of Flight (TOF) [4], etc. However, in indoor environments, the captured UWB information is disturbed by noise, multipath effect and signal attenuation during transmission, which can degrade the location accuracy [5]. Several algorithms have been proposed to improve the performance of tag localization in UWB systems. Deep learning, with its powerful pattern recognition and learning capabilities, excels in feature

extraction and other related fields, and has obtained approved application in UWB localization by learning key features of the UWB signal [6]. Convolutional neural networks (CNN) are employed for extracting spatial features, and then to estimate the tag's position using a fully connected layer based on these spatial features. These extracted features can improve the localization accuracy and mitigate the noise to some extent during the extraction process. Bregar K et al. [7] adapt CNN to process the raw channel impulse response (CIR) data of the signals and propose the ranging error regression method to reduce NLOS error by weighting the individual distance contributions based on the estimated ranging error. Jiang C et al. [8] propose an invertible transform method to denoise CIR data and classify the signals using CNN. Specifically, the CIR data undergo a leftward cyclic shift, after which the processed data is summed with the initial data to generate the transformed CIR data. This transformation increases the signal-to-noise ratio (SNR) of the new CIR data. The denoised CIR dataset is then obtained by applying an anti-convolution operation using Lucy-Richardson function. Besides spatial features, temporal features of ranging are also incorporated to improve the accuracy of positioning. A variety of recurrent neural networks (RNN), for instance long short-term memory (LSTM), are employed to capture temporal features from the obtained signal datasets. J. Shen et al. [9] input the CIR data of UWB into a Convolutional Neural Network-Long Short-Term Memory Network (CNN-LSTM) model. The CNN extracts features, which are the input into LSTM network. Experimental results demonstrate that the spatial-temporal features are capable of enhancing the performance of UWB positioning in NLOS scenarios. Yalin Tian et al. [10] propose the KF-LSTM algorithm, which introduces the Kalman filter to mitigate the effects of noise in UWB data and employs the LSTM network to learn temporal features, thereby enhancing precision in tag location estimation. However, the KF-LSTM is complex, and the Kalman filter shows limited performance in high-noise environments.

To address these issues, a method that combines a residual neural network (ResNet) with LSTM is proposed, referred as the ResNet-LSTM. ResNet, a deep convolutional neural network, is primarily used to extract features from images and videos. The introduction of residual connectivity can effectively alleviate the problem of gradient vanishing in deep neural networks, allowing the network to be trained with deeper layers to extract deep signal features, providing richer input information for the subsequent LSTM network and improving the overall model performance [11]. This is

Manuscript received November 8, 2024; revised April 29, 2025.

This work was supported in part by the Innovative Funds Plan of Henan University of Technology under Grant 2022ZKCJ13.

Chunhua Zhu is a professor and the deputy dean of the School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, 450001, China (e-mail: zhuchunhua@haut.edu.cn)

Manyi Li is a postgraduate student in the College of Information Science and Engineering, Henan University of Technology, Zhengzhou, 450001, China (e-mail: 522102000@qq.com)

Mingyang Cui is a lecturer in College of Information Science and Engineering, Henan University of Technology, Zhengzhou, 450001, China (corresponding author to provide phone: (+86)18623715327, e-mail: cmy_1205@haut.edu.cn)

because residual blocks of ResNet can capture complex features of the input data and stably extract effective features under varying noise levels, ensuring that the ResNet-LSTM model maintains high localization accuracy and robustness in different noise conditions and complex environments. The spatial features extracted by ResNet are then input to the LSTM network for time series prediction and localization computation.

By integrating ResNet and LSTM networks and taking advantage of their advantages, the UWB indoor positioning performance can be effectively improved. In summary, the contributions of this paper include:

1) The residual module of ResNet is commonly employed for extracting the spatial distribution features of the signal, while the deep network enhances the expressive power of complex nonlinear relationships. Design the integration of multiple convolutional layers and residual blocks that allow the network to efficiently learn deeper features, and the residual module is able to capture subtle changes in the signal by adding input data directly to the transformed output. The extracted features are then fed to an LSTM network, which processes the time series features to further improve positioning accuracy and stability.

2) LSTM networks play a key role in extracting temporal features from time series data, using their gating mechanism to extract relevant information from signals while suppressing irrelevant or noisy data, leading to improved accuracy and prediction stability in location prediction. Additionally, the LSTM layer captures the temporal dependencies of UWB signals in the time series, addressing the continuity issue of the movement trajectory of mobile terminals.

3) The double-sided two-way ranging (DS-TWR) technique is used to effectively mitigate the clock synchronization error, provide more accurate ranging data for the algorithm, and improve the accuracy of the whole positioning system.

II. DOUBLE-SIDED TWO-WAY RANGING

To obtain the data for localization, the Double-sided Two-way Ranging (DS-TWR) [12] method is employed to measure the distance between the tag and the base station (BS). In UWB positioning technology, the DS-TWR method is extensively utilized to attain high-accuracy ranging by calculating the round-trip time of signal propagation. The ranging process is illustrated in Fig. 1. The distance between the tag and the base station is calculated by capturing the time difference between the two timestamps before and after the

UWB signal travels from one transceiver to the other, respectively. Specifically, the anchor transmits the initial round of messages to initiate the ranging request. When the tag receives the response message, it generates and sends back the response after a certain period, which includes the time interval from the reception of the signal to the generation of the new signal. The time for the tag to communicate with BS can be determined by

$$\hat{T}_{prop} = \frac{1}{2}(T_{round} - T_{reply}) \quad (1)$$

where T_{round} is the time taken from tag initiating the transmission of a signal until it receives a response; T_{reply} is the reply time, which refers to the duration for the BS to send a response after receiving the signal.

In practical applications, the issue of clock synchronization is unavoidable. Consequently, DS-TWR can effectively mitigate clock drift errors by enhancing the number of signal propagation paths [13]. Subsequently, by the DS-TWR method, the time for the tag to communicate with BS can be computed by

$$T_{prop} = \frac{T_{round1} \times T_{round2} - T_{reply1} \times T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}} \quad (2)$$

where T_{round1} represents the time interval between device Tag transmitting data and BS receiving the response. T_{round2} represents the duration from when the BS transmits data to when the Tag acknowledges its reception. T_{reply1} also denotes the corresponds to time of device BS in processing information from device Tag. T_{reply2} represents the response time of device Tag in processing information from device BS.

According to Equation (2), the fundamental ranging data can be obtained by measuring the round-trip propagation time of the signal. The data will subsequently be input into the ResNet-LSTM network for deep feature extraction and time series prediction, thereby further enhancing positioning accuracy.

III. UWB LOCALIZATION BASED ON RESNET-LSTM NETWORK

The network structure of ResNet-LSTM is illustrated in Fig. 2. LSTM is a deep learning model specifically designed for processing time series data, enabling it to capture dependencies within the sequences. The data utilized in UWB ranging and positioning are inherently time-dependent.

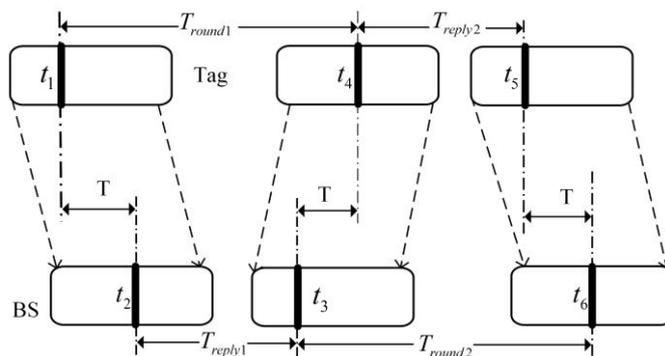


Fig. 1. UWB ranging process

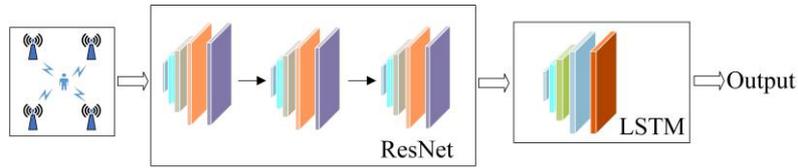


Fig. 2. ResNet-LSTM Network Structure

ResNet efficiently extracts complex and high-level features from the input UWB signals, enabling the model to capture intricate spatial patterns and representations. While LSTM captures the temporal dependencies in these features, thus leveraging the strengths of both models. This integration provides more comprehensive input information, ultimately improving the precision of predictions and robustness of the positioning model.

A. ResNet Network

ResNet is a widely utilized deep neural network architecture in computer vision, including image classification, object detection, and semantic segmentation. UWB signal transmission can be adversely affected by multipath effects and complex environments, leading to diminished positioning accuracy.

To enhance feature extraction from input data, it is often necessary to increase the number of layers in the deep neural network, thereby improving its capacity to fit the data. However, with an increasing number of layers, network performance may degrade as the network depth increases. To address this challenge, ResNet is introduced in the literature [14], incorporating residual connections within neural networks. This innovation facilitates more effective gradient propagation back to the earlier layers, mitigating performance degradation associated with increased network depth. In this paper, we design a deep neural network built upon the core principles of ResNet. By leveraging its feature extraction and representation capabilities, the network is capable of extracting useful features from UWB signals for indoor positioning. Furthermore, to mitigate the vanishing and exploding gradient problems in the process of training deep networks, we introduce the residual network [15] to extract features from UWB signals.

In this residual unit, X and $H(X)$ represent the input and output of the residual unit, respectively. The input X undergoes a series of transformations to produce $H(X)$. In ResNet, the residual block introduces an identity mapping, yielding the output $H(X)=F(X)+X$, where $F(X)$ denotes the residual function.

ResNet exhibits superior performance in handling noise in UWB signals due to its distinctive residual connections. The integration of multiple convolutional layers and residual blocks facilitates the extraction of high-dimensional spatial features from complex signal transmission paths. These residual connections enable the network to effectively learn deeper features without encountering the vanishing gradient problem. Additionally, by directly adding the input data to the transformed output, residual blocks are able to capture subtle variations in the signal. These deeply extracted features are subsequently fed into the LSTM network, which processes the time series features to further enhance positioning accuracy and stability. A common residual unit is illustrated in Fig. 3.

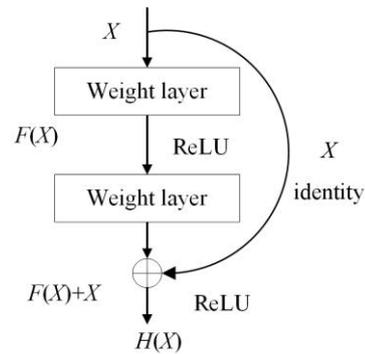


Fig. 3. Residual Unit

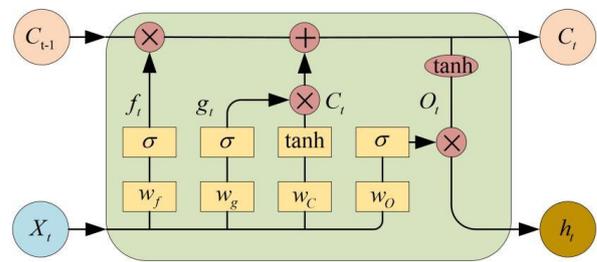


Fig. 4. LSTM Network Structure

B. Indoor Positioning Based on LSTM

LSTM, developed from recurrent neural networks (RNNs), is primarily designed to mitigate the gradient vanishing problem inherent in RNNs [16]. Compared to traditional backpropagation (BP) neural networks and other general neural networks, LSTM introduces an additional temporal dimension at the input and incorporates feedback loops, thereby enabling more effective connections when processing time series data.

The core concept of LSTM is to control the flow of information by the introduction of "gate" structures [17-18]. Each gate functions as a neural network layer that determines whether the information can be transmitted [19]. The LSTM network structure, illustrated in Fig. 4, primarily comprises three gates and a cell state: Input Gate: Regulates the impact of new input information on the current cell state; Forget Gate: Decides whether the information in the current cell state should be forgotten; Output Gate: Regulates how the cell state influences the output result.

In Fig. 4, X_t represents the input to the LSTM network, which encompasses tag coordinates, base station location information, and the tag coordinates predicted in the previous time step. h_t represents the output of the network, which consists of the optimized predicted tag coordinates.

A. ResNet-LSTM Algorithm

The data utilized for UWB ranging and positioning exhibits temporal dependencies. LSTM networks excel at capturing long-term dependencies in sequential data,

rendering them highly effective for processing indoor positioning signals. In UWB indoor positioning, the LSTM network is pivotal in extracting temporal features from time series data by leveraging its gating mechanisms to extract relevant information from the signal while suppressing irrelevant or noisy data, thereby enhancing the accuracy and robustness of position predictions.

The ResNet network introduces residual connections, enabling each layer not only to learn new features but also to retain and propagate information from preceding layers. The integration of ResNet and LSTM networks harnesses the strengths of both, improving the accuracy of tag positioning. The algorithm involves three steps: establishing a time series dataset, constructing the ResNet network, and building the LSTM neural network.

In an ideal scenario, when there are three or more base stations, the tag position can be uniquely determined. In the experiment, four UWB base stations are utilized. Consider a tag within the UWB positioning system, and the distance from the tag to each base station at time t is recorded as $[D_1^t \ D_2^t \ D_3^t \ D_4^t]$, where $t=1, 2, 3, 4, \dots, N$. Assuming the required length of the time series in the dataset is N , a segment of data from the dataset can be recorded as

$$\begin{bmatrix} D_1^1 & D_2^1 & D_3^1 & D_4^1 \\ D_1^2 & D_2^2 & D_3^2 & D_4^2 \\ D_1^3 & D_2^3 & D_3^3 & D_4^3 \\ \vdots & \vdots & \vdots & \vdots \\ D_1^N & D_2^N & D_3^N & D_4^N \end{bmatrix}$$

This represents the distances from the tag to each of the four base stations over a time series of length T . A residual block is composed of two convolutional layers and two normalization layers, with an activation function applied between each pair of layers, along with a shortcut connection that adds the input to the output. This structure is repeated three times, with each subsequent input being the output from the preceding residual block. The specific steps of the algorithm are as follows:

Input: Store the input as part of the shortcut connection, which will be directly added to the transformed output later.

First convolutional layer: Perform convolution operations through Conv1D to extract features. This layer utilizes a specified number of filters and kernel size; Subsequently, apply Batch Normalization to the output of the convolutional layer to stabilize the training process; Finally utilize the Rectified Linear Unit (ReLU) [20] activation function for non-linear transformation.

Second convolutional layer: Perform convolution, batch normalization, and activation operations again. The results of these two convolutions constitute the residual component.

Residual connection: Add the input from the residual connection to the output of the second convolutional layer. This establishes the residual connection, enabling the network to learn the residual function.

Activation function: Finally, apply a ReLU activation function to the result of the residual connection prior to generating the output.

In the entire residual module, residual blocks are stacked multiple times. The initial feature extraction is performed with a single convolution, followed by deepening the network through the stacking of multiple residual blocks.

This structure enables the network to capture more complex features while preserving the original input features.

The output of ResNet, the initially predicted tag coordinates h_{t-1} , and the cell state C_t value (determined by the forget gate and input gate functions in the LSTM network) are input to the LSTM network. The output of the LSTM network is expressed as:

$$h_t = [\tilde{x}_t, \tilde{y}_t] \quad (3)$$

here, \tilde{x}_t and \tilde{y}_t are the predicted tag coordinates at time t after passing through the LSTM network. The computation process of the forget gate is expressed as:

$$f_t = \sigma(W_f \times [h_{t-1}, D_{x,y}] + b_f) \quad (4)$$

here, σ represents the ReLU of the LSTM network, W_f is the weight matrix, and b_f is the bias term. The computation process of the input gate function is expressed as:

$$g_t = \sigma(W_g \times [h_{t-1}, D_{x,y}] + b_g) \quad (5)$$

here, W_g denotes the weight coefficient of the input gate, and b_g is the bias term. The input \tilde{C}_t and output C_t of the cell state at time t are given as:

$$\tilde{C}_t = \tan h(W_c \times [h_{t-1}, D_{x,y}] + b_c) \quad (6)$$

$$C_t = g_t \times \tilde{C}_t + f_t \times C_{t-1} \quad (7)$$

The long-term memory in the LSTM network is updated iteratively, and the output function O_t is expressed as:

$$O_t = \sigma(W_o \times [h_{t-1}, D_{x,y}] + b_o) \quad (8)$$

The final output of the LSTM, which represents the tag's position coordinates x and y , is determined by the output gate and the cell state, which can be expressed as:

$$h_t = O_t \times \tanh(C_t) \quad (9)$$

The LSTM network constructs a regression model through an iterative process and the initialization of weights.

Compared to traditional BP models, it demonstrates faster convergence. The positioning algorithm of the ResNet-LSTM network is summarized in Table I.

TABLE I
POSITIONING ALGORITHM BASED ON RESNET-LSTM NETWORK

Algorithm 1: Positioning Algorithm Based on ResNet-LSTM Network

Input: $D_{(x,y)}$, the true coordinates of the tag at time t .

Output: predicted position coordinates (\tilde{x}, \tilde{y})

1. Obtain the distance data from the anchors to the target position and the true coordinates.
 2. Construct the ResNet module to extract spatial features from the distance data through multiple convolutional layers and residual blocks.
 3. Adjust the output shape of the ResNet module to conform to the input format required by the LSTM.
 4. Utilize the LSTM layer to process the features extracted by ResNet, capturing long-term dependencies in the time series.
 5. Convert the LSTM output into two-dimensional positional coordinates using a fully connected layer.
 6. Compile and train the ResNet-LSTM model by utilizing an optimizer and the mean squared error loss function.
 7. Employ the trained model to make predictions on the test set and compute the error for evaluation.
-

II. EXPERIMENTS AND RESULTS

To evaluate the prediction accuracy and stability of the proposed ResNet-LSTM model, the python 3.9.0 simulation

environment is employed along with its built-in tools. The simulation encompasses the UWB base stations and tag positioning environment. The simulated TOF distance model estimates the distances of all tag positions from each base station within the experimental area, which are used as the training data. Four UWB base stations are employed, labeled A, B, C, and D, with their coordinates denoted as (x_A, y_A) , (x_B, y_B) , (x_C, y_C) and (x_D, y_D) , respectively. The tag coordinates is $O(x_o, y_o)$. The true distance from A to O is calculated as

$$d_{AO} = \sqrt{(x_A - x_o)^2 + (y_A - y_o)^2} \quad (10)$$

The predicted result is $(\tilde{x}_t, \tilde{y}_t)$, leading to a positioning error calculated as

$$E = \sqrt{(\tilde{x}_t - x_o)^2 + (\tilde{y}_t - y_o)^2} \quad (11)$$

To thoroughly assess the positioning performance of the proposed ResNet-LSTM, four commonly used evaluation metrics are employed: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics are well-established tools for evaluating model performance, as they quantify the divergence between model predictions and truth data from different perspectives, providing insights into the model's accuracy, stability, and sensitivity to outliers.

1) MSE: MSE is a frequently employed metric for assessing regression models. It represents the mean of the squared differences between the predicted and observed values. The formula is expressed as follows:

$$MSE = \frac{1}{N} \sum_{t=1}^N [(\tilde{x}_t - x_o)^2 + (\tilde{y}_t - y_o)^2] \quad (12)$$

2) RMSE: RMSE is derived by taking the square root of MSE. As a crucial and intuitive metric in model evaluation, it provides valuable reference that aid in optimizing and improving the model during both training and testing phases. The specific formula is as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N [(\tilde{x}_t - x_o)^2 + (\tilde{y}_t - y_o)^2]} \quad (13)$$

3) MAE: Unlike MSE and RMSE, MAE uses the absolute value of the error rather than the squared value, providing a straightforward measure of the model's overall accuracy. The specific formula is as follows:

$$MAE = \frac{1}{N} \sum_{t=1}^N [|\tilde{x}_t - x_o| + |\tilde{y}_t - y_o|] \quad (14)$$

4) MAPE: MAPE measures the percentage of the prediction error relative to the actual values, providing an effective assessment of relative accuracy. The specific formula is as follows:

$$MAPE = 100 \times \frac{1}{N} \sum_{t=1}^N \left[\frac{|\tilde{x}_t - x_o|}{\tilde{x}_t} + \frac{|\tilde{y}_t - y_o|}{\tilde{y}_t} \right] \quad (15)$$

The experimental setup configured the locator coordinates as (0, 0), (0, 400), (400, 400), and (400, 0), with the coordinate unit expressed in centimeters. The data samples consist of 1000 distances, each including four distance measurements along with the user's x and y position coordinates. Ninety percent of the data samples are utilized for training the model, while the remaining ten percent are reserved for testing. Noise is introduced to the generated true

distances, which are time series required for the LSTM. Table II and Table III separately present the positioning performance comparisons of different models (BP Neural Network, LSTM, and ResNet-LSTM) on the training set and test set, including their MSE, RMSE, MAE, and MAPE values.

From Table II and Table III, it can be observed that the ResNet-LSTM demonstrates superior performance on both the training set and the test set. On the training set, the MSE, RMSE, MAE, and MAPE values of ResNet-LSTM are all the lowest, indicating that it can better fit the data during the training process. On the test set, the MSE and RMSE of ResNet-LSTM are significantly lower than those of other models, suggesting that it has stronger generalization capability. Furthermore, the lower MAE and MAPE values further prove the advantages of the ResNet-LSTM model in terms of relative accuracy and stability. The comparison of the four metrics reveals that ResNet-LSTM achieves a lower prediction error. Compared to BP, the MSE, RMSE, MAE, and MAPE on the test set are reduced by 96.89%, 82.76%, 85.93%, and 61.30%, respectively. When compared to LSTM, these metrics reduced by 89.26%, 67.22%, 72.58%, and 43.97%. These results suggest that the ResNet-LSTM model is well-suited for UWB indoor positioning applications that require higher precision.

TABLE II
COMPARISON OF POSITIONING PERFORMANCE ON THE TRAINING SET

Evaluation metrics	BP	LSTM	ResNet-LSTM
MSE	4726.56	1499.42	192.38
RMSE	68.75	38.72	13.87
MAE	52.84	36.58	11.21
MAPE (%)	30.62	19.82	10.62

TABLE III
COMPARISON OF POSITIONING PERFORMANCE ON THE TEST SET

Evaluation metrics	BP	LSTM	ResNet-LSTM
MSE	6483.17	1876.62	201.53
RMSE	82.36	43.32	14.20
MAE	80.58	41.36	11.34
MAPE (%)	38.42	26.54	14.87

The Fig. 5 illustrates the error curves for the proposed positioning algorithm, the LSTM positioning algorithm, and the BP positioning algorithm. From Fig. 5, the proposed positioning algorithm outperforms BP algorithms LSTM algorithms, exhibiting the lowest positioning error, minimal error fluctuation, and superior stability. In contrast, the BP exhibits the highest error and greatest fluctuation, resulting in the poorest performance. The LSTM performs at an intermediate level between the two. For the characterization of error distributions, the scatter plots in Fig. 6 depict positioning errors associated with the three algorithms, highlighting their differential performance.

As Fig. 6, the proximity of the error distribution points to the origin is indicative of the accuracy of the predicted tag coordinates relative to the true coordinates. A tighter clustering near the origin suggests smaller positioning errors.

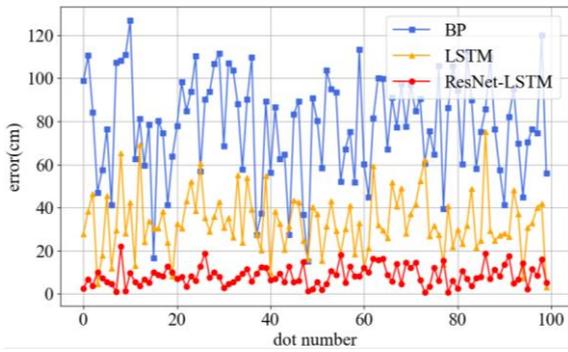


Fig. 5. The error curves of the proposed positioning algorithm

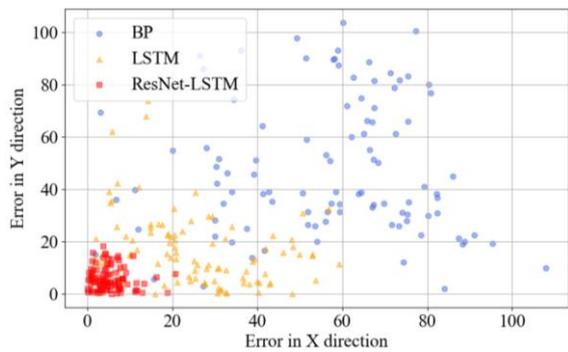


Fig. 6. The error scatter plots of the proposed positioning algorithm

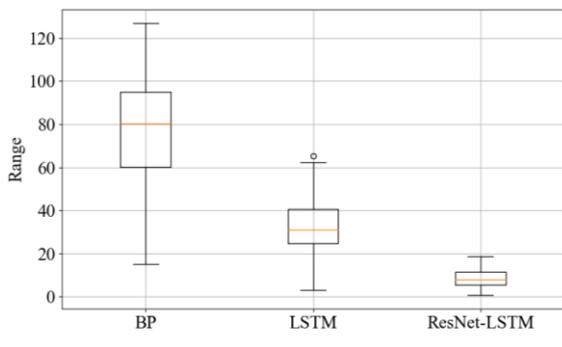


Fig. 7. The box plot of the positioning errors for the different positioning algorithms

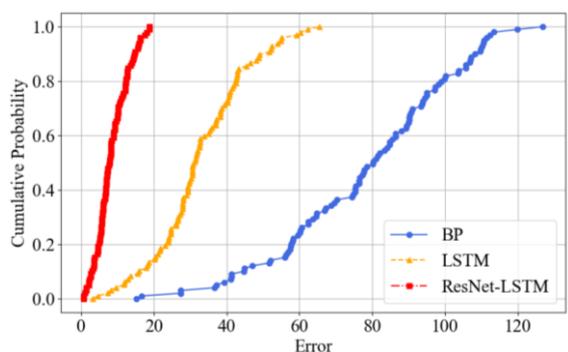


Fig. 8. The cumulative distribution function (CDF) plots for the proposed positioning algorithm

The error distribution of tag coordinates generated by the proposed algorithm is concentrated within a specific region, indicating superior clustering performance. In contrast, the LSTM and the BP display more dispersed error distributions, with certain positioning coordinates exhibiting significantly larger errors. Following an analysis of the relationship between error fluctuations and directional errors across different algorithms (as depicted in Fig. 6), box plots are used to analyze positioning performance.

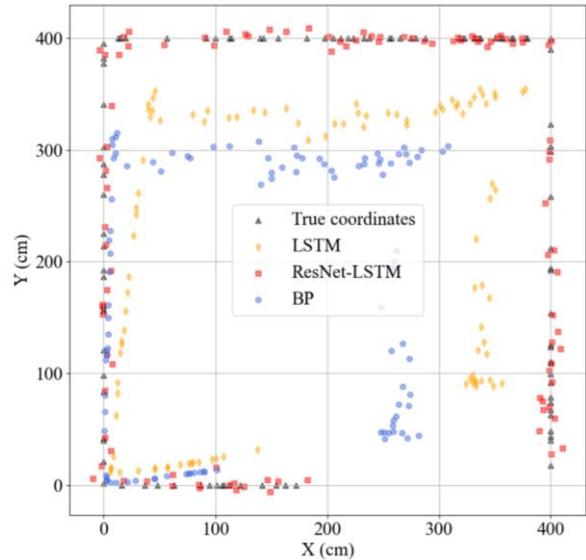


Fig. 9. Trajectory comparison of actual and predicted coordinates

The box plot is adapted to provide a more detailed examination of the overall error distribution characteristics of the positioning algorithms, and a more comprehensive understanding of positioning performance in terms of prediction accuracy and stability.

The box plot provides five key statistical metrics regarding the data distribution: the minimum value (lower bound), the first quartile (bottom of the box), the median (line within the box), the third quartile (top of the box), and the maximum value (upper bound). This visualization facilitates a clear assessment of the error distribution for each algorithm, highlighting both the central tendency (median) and the dispersion of the data (interquartile range). In a box plot, shorter boxes and narrower interquartile ranges typically signify that the algorithm’s errors are more concentrated, indicating greater stability in predictions. As illustrated in Fig. 7, the proposed positioning algorithm demonstrates a concentrated error distribution with minimal fluctuation and almost no outliers compared to other algorithms, indicating that the model’s predictive performance is stable across all test samples.

As shown in Fig. 8, the proposed ResNet-LSTM algorithm reaches a cumulative probability close to 1 when the error is below 20 cm, whereas the LSTM network model and BP neural network reach similar cumulative probabilities only when the error is around 40 cm and 100 cm, respectively.

This demonstrates the outstanding performance of the proposed algorithm within the range of small errors, with most samples having small errors, thereby reflecting its high accuracy and stability.

To evaluate the performance of the proposed ResNet-LSTM algorithm more intuitively, this experiment compares the actual coordinate trajectory with the coordinates predicted by the different positioning model, as illustrated in Fig. 9. From Fig. 9, the coordinates predicted by ResNet-LSTM are closer to the real coordinates. The average errors between the predicted trajectory and the actual trajectory for BP, LSTM, and ResNet-LSTM are 78.01 cm, 35.48 cm, and 13.59 cm, respectively, indicating the superior positioning accuracy of the proposed ResNet-LSTM algorithm.

III. CONCLUSIONS

This paper proposes a UWB indoor positioning algorithm that integrates ResNet with LSTM neural networks. This algorithm converts UWB-collected distance measurements into time series representations, employs ResNet to mitigate noise and uncertainties within the time series, and subsequently trains the processed data using the LSTM network. Simulation experiments demonstrate that the algorithm outperforms both BP and LSTM algorithms with respect to positioning accuracy and stability, producing more accurate positioning coordinates.

REFERENCES

- [1] Z. Yuan, Y. Jun, L. You, et al, "Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons," *Sensors*, vol. 16, no. 5, pp. 596, 2016.
- [2] A. Poulouse, J. Kim, D. S. Han, "A Sensor Fusion Framework for Indoor Localization Using Smartphone Sensors and Wi-Fi RSSI Measurements," *Applied Sciences*, vol. 9, no. 20, pp. 4379, 2015.
- [3] W. Wang, Q. Zhu, Z. Wang, et al, "Research on Indoor Positioning Algorithm Based on SAGA-BP Neural Network," *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3736-3744, 2021.
- [4] B. J. Jang, "Principles and Trends of UWB Positioning Technology," *The Journal of Korean Institute of Electromagnetic Engineering and Science*, vol. 33, no. 1, pp. 1-11, 2022.
- [5] X. Li, Y. Wang, K. Khoshelham, "Comparative Analysis of Robust Extended Kalman Filter and Incremental Smoothing for UWB/PDR Fusion Positioning in NLOS Environments," *Acta Geodaetica et Geophysica*, vol. 54, pp. 157-179, 2019.
- [6] Y. Tian, Z. Lian, P. Wang, et al, "Application of A Long Short-Term Memory Neural Network Algorithm Fused with Kalman Filter in UWB Indoor Positioning," *Scientific Reports*, vol. 14, no. 1, pp. 1-14, 2024.
- [7] K. Bregar, M. Mohorčič, "Improving Indoor Localization Using Convolutional Neural Networks on Computationally Restricted Devices," *IEEE Access*, vol. 6, pp. 17429-17441, 2018.
- [8] C. Jiang, S. Chen, Y. Chen, et al, "An UWB Channel Impulse Response De-Noiseing Method for NLOS/LOS Classification Boosting," *IEEE Communications Letters*, vol. 24, no. 11, pp. 2513-2517, 2022.
- [9] C. Jiang, J. Shen, S. Chen, et al, "UWB NLOS/LOS Classification Using Deep Learning Method," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2226-2230, 2020.
- [10] C. He, Y. Yuan, B. Tan, "Constrained L1-Norm Minimization Method for Range-Based Source Localization under Mixed Sparse LOS/NLOS Environments," *Sensors*, vol. 21, no. 4, pp. 1321, 2021.
- [11] F. X. Zhu, X. Yang, J. S. Yang, et al, "Design of Digital FIR Filter Based on Long Short-Term Memory Neural Network," *Engineering Letters*, vol. 32, no. 10, pp. 1993-2001, 2024.
- [12] H. Tong, N. Xin, X. Su, et al, "A Robust PDR/UWB Integrated Indoor Localization Approach for Pedestrians in Harsh Environments," *Sensors*, vol. 20, 193.no. 1, pp. 193.
- [13] D. G. Angelis, A. Moschitta, P. Carbone, "Positioning Techniques in Indoor Environments Based on Stochastic Modeling of UWB Round-Trip-Time. Measurements," *IEEE Transactions on Intelligent Transportation Systems*, vol.17, no. 8, pp. 2272, 2016.
- [14] W. Q. Liu, G. Wei, C. F. Gao, et al, "Research on UWB NLOS Propagation Impact Suppression Technology Based on Deep Learning," *Infrared and Laser Engineering*, vol. 52, no. 12, pp. 293-301, 2023.
- [15] K. He, X. Zhang, S. Ren, et al, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.
- [16] Y. Yu, X. Si, C. Hu, et al, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural computation*, vol. 1, no. 7, pp. 1235-1270, 2019.
- [17] Y. Huang, J. Li, et al. "An Improved Hybrid CNN-LSTM-Attention Model with Kepler Optimization Algorithm for Wind Speed Prediction," *Engineering Letters*, vol.32, no. 10, pp. 1957-1965, 2024.
- [18] D. Guang, "Research on Record Named Entity Recognition of Chinese Electronic Medical based on LSTM-CRF," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists*, pp. 88-93, 2023.
- [19] A. Poulouse, D. S. Han, "UWB Indoor Localization Using Deep Learning LSTM Networks," *Applied Science*, vol. 10, pp. 6290, 2020.
- [20] R. Katende, H. Kasumba, G. Kakuba, et al, "On the Error Bounds for ReLU Neural Networks," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 12, pp. 2602-2611, 2024.