Scientific Reports

https://doi.org/10.1038/s41598-026-35802-0

Article in Press

Application of a temporal convolutional network algorithm fused with channel attention module for UWB indoor positioning

Received: 27 April 2025

Accepted: 8 January 2026

Published online: 27 January 2026

Cite this article as: He L., Lian Z., Núñez-Andrés M.A. *et al.* Application of a temporal convolutional network algorithm fused with channel attention module for UWB indoor positioning. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-35802-0

Liuhui He, Zengzeng Lian, M. Amparo Núñez-Andrés, Hao Chen, Chenrui Zhao & Yalin Tian

Scientific
reports

Application of a temporal convolutional network algorithm fused with channel attention module for UWB indoor positioning

Liuhui He[1,#], Zengzeng Lian[1,#], M. Amparo Núñez-Andrés[2], Hao Chen[1], Chenrui Zhao[1], Yalin Tian[1]

[1]School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo 454003, China*
[2]Department of Civil and Environmental Engineering, Universitat Politècnica de Catalunya, Barcelona, 08028, Spain.
Correspondence: zengzenglian@hpu.edu.cn
# These authors contributed equally.

ABSTRACT

Ultra-wideband (UWB) technology offers considerable advantages for indoor positioning. However, its accuracy significantly decreases in non-line-of-sight environments, particularly in dynamic scenarios with frequent human movements. To address this challenge, this study proposed a temporal convolutional network with a channel attention module (TCN-CAM) to enhance positioning performance. The TCN architecture, employing causal and dilated convolutions, effectively mitigates the vanishing gradient problem commonly encountered in neural networks and improves the model's capacity to capture long-range dependencies in time-series data. Concurrently, CAM enhances model adaptability by emphasizing salient features under complex conditions. Simulation and field experiments demonstrated that the TCN-CAM algorithm achieved high positioning accuracy and stability with a mean error of only 3.32 cm. Compared with LSTM-AM, CNN-CAM, and conventional TCN algorithms, the proposed method improved positioning accuracy by 76.12%, 25.06%, and 19.42%, respectively, thereby significantly enhancing the robustness and performance of UWB-based positioning systems.

Introduction

The rapid advancement of Internet of Things (IoT) technology, coupled with the widespread adoption of smart devices, has led to a burgeoning demand for precise location tracking. Global Navigation Satellite Systems (GNSSs) have been widely applied in outdoor environments and can achieve high positioning accuracy. However, satellite positioning becomes inaccurate or unattainable within urban canyons or enclosed indoor spaces, primarily owing to obstruction and multipath effects affecting GNSS signals [1]. Consequently, there is a pressing need to address the challenges of achieving high-precision positioning in indoor environments [2]. Currently, various indoor positioning technologies have emerged, such as Wi-Fi, Bluetooth, ultra-wideband (UWB), audio, 5G positioning, and vision-based positioning technologies [3-7]. Among them, UWB is extensively employed in indoor positioning owing to its notable advantages, such as wide bandwidth, short pulse duration, robust anti-jamming capability, and immunity to interference from other communication devices [8].

UWB technology, which originated in the 1960s for military and radar applications, has gradually transitioned into the commercial sector, where it plays an increasingly important role in wireless communication and positioning systems. Advancements in UWB technology have led to the development of several key ranging and positioning techniques. These techniques are generally categorized into four principal types: Received Signal Strength (RSS), Angle of Arrival (AOA), Time of Arrival (TOA), and Time Difference of Arrival (TDOA) [9-12]. The RSS-based positioning algorithm is simple and cost-effective to implement; however, its accuracy is highly susceptible to environmental interference. AOA-based methods provide goniometric precision but are vulnerable to errors caused by multipath propagation and signal reflections. TDOA algorithms eliminate the need for precise time synchronization required by TOA; however, their accuracy is limited by the complexity of calculating time differences across multiple receivers. Among these, TOA algorithms, relying on signal arrival time measurements, are currently the most widely adopted. The performance of TOA-based positioning depends on the precision of UWB

ranging and synchronization accuracy between base stations and tags [13]. In complex indoor environments, obstacles such as doors, windows, and glass, coupled with human movement, contribute to non-line-of-sight (NLOS) errors in UWB signal propagation, significantly affecting positioning accuracy. Under such conditions, traditional algorithms based on RSS, AOA, TOA, and TDOA are inadequate for achieving high-precision positioning. Therefore, novel algorithms capable of mitigating NLOS errors in UWB ranging are urgently required to enhance positioning accuracy.

Extensive research has been conducted on NLOS errors, leading to notable advancements in this field. To mitigate NLOS errors and inaccuracies in the kinematic model during UWB positioning, literature [14] proposed an improved robust adaptive Kalman filter (IRACKF). The filter adopted polynomial-fitting sliding windows to classify errors and selected an appropriate filter for error correction. The proposed IRACKF algorithm significantly improved the accuracy and stability of UWB positioning. However, the experiments did not consider the mean and variance values of NLOS errors across different channel environments. Literature [15] employed Bayesian filtering to mitigate UWB ranging errors and improve positioning accuracy through probabilistic modeling. Despite its effectiveness, this method is highly dependent on the accuracy of prior probabilities. In dynamic three-dimensional (3D) environments, delays in updating the prior model may lead to error accumulation, reducing its applicability in complex scenarios. Additionally, an improved Residual Improved Chan-Taylor (RICT) algorithm was proposed to suppress NLOS errors in indoor UWB positioning [16]. The RICT algorithm demonstrated superior performance in scenarios with a limited number of anchor nodes. However, its advantages diminished when the anchor node density was sufficient. Traditional methods for handling NLOS errors rely on complex mathematical models, offering significant potential for optimization in accurately modeling and predicting these errors. To address this challenge, machine learning has emerged as an effective solution, leveraging vast datasets and pattern recognition techniques to enhance ranging accuracy and play an important role across various application domains [17-19]. In the literature [20], NLOS errors were effectively mitigated using the Relevant Vector Machine (RVM) technique, and their locations were determined using a two-step iterative (TSI) algorithm. In the literature [21], three machine learning techniques (SVM, RF, and MLP) were employed to classify multi-labels (LOS, NLOS, and MP) in UWB ranging. Furthermore, the literature [22] utilized machine learning algorithms to detect and reduce UWB NLOS errors, leveraging measured data from one scene for training, validation, and testing across various scenes. However, machine learning methods often require manual feature extraction, which oversimplifies the modeling of non-line-of-sight errors. Recently, deep learning has gained attention owing to its robust feature recognition and learning capabilities, enabling direct error correction and improved positioning performance [23]. Literature [24] offers a comprehensive overview of the current state and future trends of UWB systems, highlighting LOS/NLOS classification for indoor tag localization as a key research focus. In the literature [25], a deep-learning-based UWB-NLOS/LOS classification algorithm, FCN-Attention, was introduced. This algorithm employed a Fully Convolutional Network (FCN) for improved feature extraction and incorporated a Self-Attention Mechanism to enhance feature description, thereby increasing classification accuracy. Another study [26] employed a CNN-LSTM (Convolutional Neural Networks- Long Short-Term Memory) method for UWB-NLOS/LOS signal classification, where the CNN automatically extracted features, and the LSTM network performed the classification. The literature [27] proposed a TinyML method incorporating self-attention mechanisms to enhance feature selection and model representation for UWB NLOS identification. In another work [28], a CNN-based method integrated with a channel attention module (CNN-CAM) was developed for NLOS signal classification. This model extracted features from CIR data and replaced the fully connected layer with a pooling layer to enhance classification efficiency. Finally, a novel multi-input learning neural network model [29] was introduced to process both CIR and its time–frequency domain representation (TFDOCIR) for NLOS/LOS identification tasks, achieving higher accuracy and robustness than traditional single GRU or CNN networks.

Although the aforementioned studies have laid a solid foundation for the development of indoor positioning technologies, several limitations remain. First, much of the existing research has focused on signal recognition and classification rather than directly enhancing the accuracy of UWB positioning systems. Second, most current approaches are limited to two-dimensional localization, which constrains their applicability in scenarios requiring high-precision 3D positioning, particularly when accurate height information is essential. Furthermore, the impact of errors induced by dynamic obstacles, particularly human movements, has been largely overlooked. In real-world applications, NLOS errors caused by human motion are not only more frequent but also more challenging to mitigate owing to their inherent unpredictability. These shortcomings underscore the need for further research and highlight the difficulties faced by current methods in addressing NLOS-related positioning errors. To address these challenges, this study proposed a novel TCN-CAM-based localization algorithm aimed at improving the accuracy of 3D indoor positioning by directly mitigating NLOS errors. TCN is capable of processing time-series data and effectively capturing the dynamic temporal characteristics of UWB signals. The integration of a CAM further enhances the model's ability to extract and prioritize salient features, thereby improving positioning performance in complex environments. The main contributions of this study are summarized as follows.

(1) A UWB indoor positioning model based on TCN-CAM was proposed to specifically address NLOS errors caused by human movement.

(2) Unlike conventional methods, the proposed TCN-CAM model eliminated the need for prior signal classification. It leverages deep learning to directly extract features from raw UWB data and suppress NLOS errors, thereby achieving end-to-end optimization of positioning accuracy.

(3) Both simulation and field experiments demonstrated that the TCN-CAM algorithm effectively alleviated dynamic NLOS errors and achieved superior positioning accuracy and robustness compared with existing methods.

The remainder of this paper is organized as follows. Section 2 outlines the principles of UWB positioning. Section 3 describes the proposed TCN-CAM model. Section 4 presents the results of simulations and field experiments. Sections 5 and 6 provide the discussion and conclusion, respectively.

Related work

UWB ranging principle

UWB technology is a wireless communication method that transmits data using nanosecond-scale non-sinusoidal narrow pulses. Owing to its high temporal resolution and transmission speed, UWB is extensively utilized in distance measurements, with Two-Way Ranging (TWR) being a widely adopted technique. TWR is primarily divided into Double-Sided Two-Way Ranging (DS-TWR) and High-Dynamic Scenario Two-Way Ranging (HDS-TWR). By recording the timestamps of signal transmission and reception, DS-TWR calculates the signal's round-trip time from the time difference to determine the distance between devices. In contrast, HDS-TWR builds on DS-TWR to improve efficiency by enabling simultaneous ranging with multiple base stations. This approach significantly reduces the measurement time and is tailored for high-dynamic scenarios. The operational principle of HDS-TWR is detailed in Figure 1.

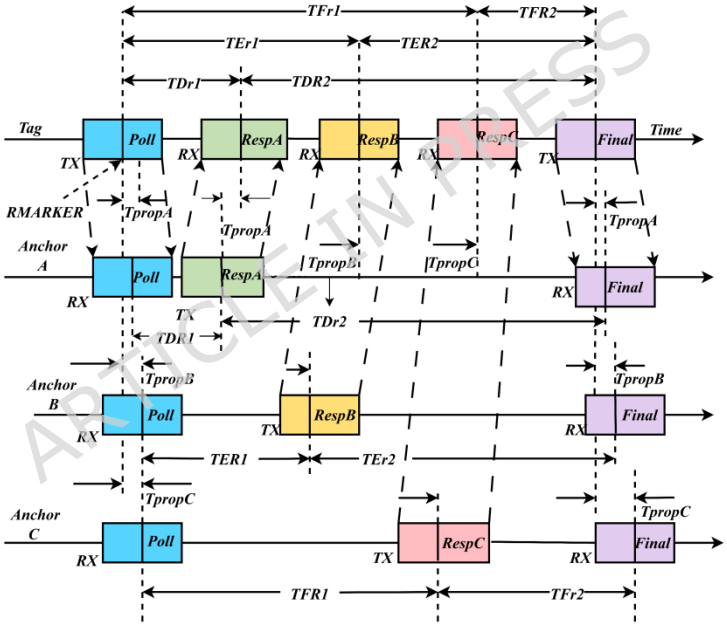


Figure 1. HDS-TWR ranging principle.

In Figure 1, T_{propA}, T_{propB}, and T_{propC} represent the UWB signal flight times to base stations A, B, and C, respectively. TX and RX represent the transmitting and receiving ends, respectively. The time required for the UWB signal to be transmitted from the tag to base stations A, B, and C is defined by Equation (1).

$$T_{prop(i)} = \frac{T_{(j)r1} \times T_{(j)r2} - T_{(j)R1} \times T_{(j)R2}}{T_{(j)r1} + T_{(j)r2} + T_{(j)R1} + T_{(j)R2}}, (i, j) \in \{(A, D), (B, E), (C, F)\} \quad (1)$$

where T_{Dr1}, T_{Er1}, and T_{Fr1} denote the time intervals between the tag transmitting signals to base stations A, B, and C, respectively. T_{Dr2}, T_{Er2}, and T_{Fr2} denote the time intervals between base stations A, B, and C transmitting signals to the tag and receiving its response, respectively. T_{DR1}, T_{ER1}, and T_{FR1} denote the response times of base stations A, B, and C when processing signals from the tag, respectively. T_{DR2}, T_{ER2}, and T_{FR2} denote the response times of the tag when processing signals from base stations A, B, and C, respectively. Equation (2) provides the calculated distances D_{propA}, D_{propB}, and D_{propC} from the tag to base stations A, B, and C.

$$D_{prop(i)} = C \times T_{prop(i)} \quad (2)$$

where C represents the speed of light.

Temporal Convolutional Network-Channel Attention Module Algorithm

Temporal Convolutional Networks (TCN)

TCN is an extension of CNN [30] for time-series data using one-dimensional convolutions to capture temporal dependencies. Unlike traditional recurrent neural networks (RNNs) and their variants (including LSTM and GRU), TCN uses a unique network architecture that enables parallel processing of time-series data while effectively capturing long-range temporal dependencies [31-33]. Furthermore, in contrast to LSTM and GRU, TCN processes temporal data non-sequentially, which improves computational efficiency and simplifies the architecture by eliminating the need for complex gating mechanisms.

The TCN architecture comprises five fundamental components: an input layer, convolutional layers, residual connections, a fully connected layer, and an output layer.

1) Input layer: The input data consists of raw time series observations, giving the network the essential information required for subsequent convolutional processes. The expression for the input layer is as follows:

$$X=[x_1, x_2, x_t] \tag{3}$$

where X represents the input data for computations in time-series convolutional network models; T represents a time step; and x_t represents an observation or signal at a time t.

2) Convolutional layers: Feature extraction is performed sequentially through convolutional layers 1 to 4. This hierarchical structure progressively captures temporal features at varying scales from the UWB signal data. Among these, the dilated convolution (Figure 2) employs an expansion factor d=[1,2,4,8] to achieve multi-scale modeling. Specifically, d=1 captures local features, d=2 covers mid-range dependencies, and d=4 and 8 are used to extract long-range temporal relationships through progressive sense field expansion. In parallel, causal convolution (Figure 3) ensures strict temporal causality by restricting the convolution kernel to operate only on the current and historical inputs. This design, combined with the efficient 1D processing capability of a 1×1 convolution kernel, is particularly suitable for temporal analysis of UWB signals. The computational expression for the convolutional layer is as follows:

$$y_t = f\left(\sum_{i=1}^{k} w_i x_{t+(i-1)*d}\right) \tag{4}$$

where y_t is the output of the convolution operation; f is the activation function; w_i is the weight of the convolution kernel; d is the expansion factor of the convolution kernel; k is the size of the convolution kernel; x_t is the input data; and i is the index of the current position of the convolution kernel concerning the input sequence.

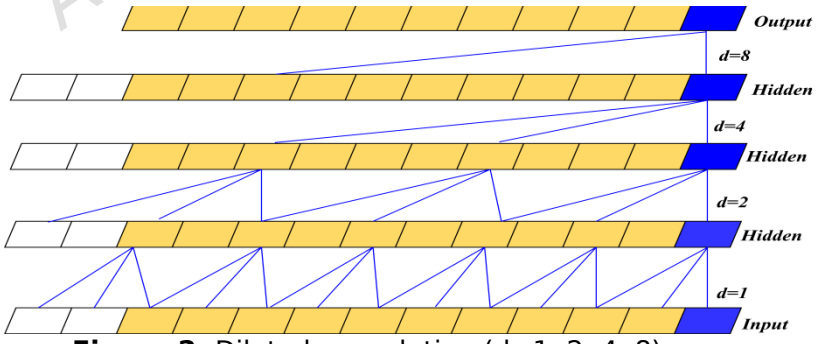


Figure 2. Dilated convolution (d=1, 2, 4, 8).

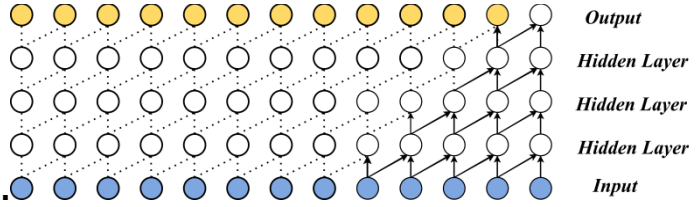


Figure 3. Causal convolution.

3) Residual connections: The data processed by the convolutional layer were combined with the input data and subjected to the ReLU activation function to establish the residual connection, as shown in Figure 4. Residual connections can address gradient vanishing issues, accelerate model training, and enhance model performance and expressiveness. The expression for calculating the residual connection

is as follows:

$$Z_t = y_t + x_t \tag{5}$$

where Z_t is the output of the residual join; y_t is the data processed by the convolutional layer; and x_t is the input data.

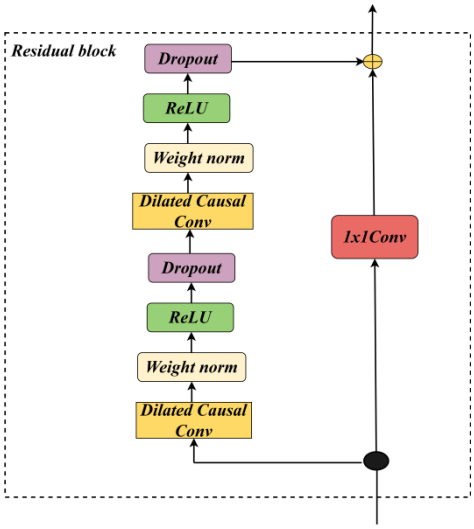


Figure 4. Residual connection.

4) Fully connected layer: After being linked through residual connections, the data were fed into the fully connected layer. This layer facilitates feature transformation and output prediction and completes the sequence modeling process. The expressions for the fully connected layer are as follows:

$$Z_i = \sum_{j=1}^{N} w_{ij} \cdot x_j + b \tag{6}$$

$$a_i = f(z_i) \tag{7}$$

where N is the number of neurons in the previous layer; w_{ij} is the weight that connects the j input neuron to the i output neuron; x_j is the output of the j input neuron; b is the bias of the i output neuron; Z_i is the weighted input, denoting the input weighted sum of the i neuron; $f(\cdot)$ is an activation function, usually nonlinear, such as ReLu; and a_i is the output of the i neuron, processed by the activation function.

5) Output layer: This layer outputs the final prediction results.

The specific network structure is shown in Figure 5.

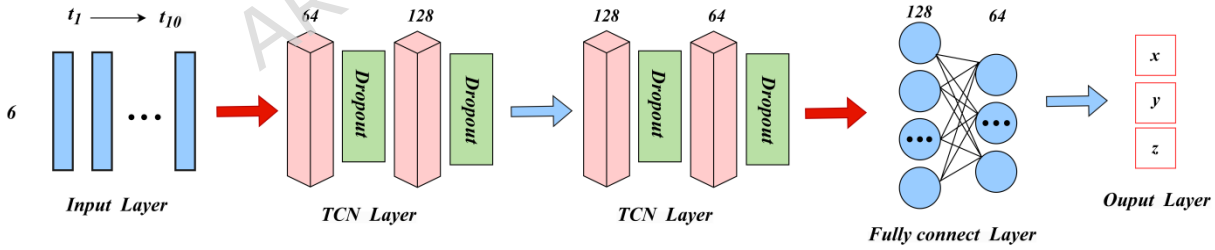


Figure 5. TCN network structure.

Channel Attention Module

The channel attention mechanism, widely used in computer vision tasks, dynamically adjusts the importance of each channel, enabling the model to automatically learn and enhance feature representations of the input data [34]. The processing flow is as follows in Figure 6.

1) Squeeze Operation: First, an input feature map F of size H×W×C undergoes global maximum pooling (GMP) and global average pooling (GAP) along the spatial dimensions, producing two 1×1×C feature maps. GAP captures the global average features, whereas GMP emphasizes the most prominent features. This squeeze operation reduces the spatial dimensions of the feature map to a 1×1 representation while preserving the essential global characteristics of each channel. The expressions for the squeeze operation are as follows:

$$S_{avg} = GlobalAveragePool(F) \in R^C \tag{8}$$

$$S_{max} = GlobalMaxPool(F) \in R^C \tag{9}$$

where S_{avg} and S_{max} represent the channel feature vectors from GAP and GMP, respectively, both with

dimensionality C; and F is the input feature map.

2) Excitation Operation: GMP and GAP results are processed through a shared multi-layer perceptron (MLP). This transformation facilitates the modeling of cross-channel dependencies and generates a refined intermediate feature representation. The expressions for the excitation operation are as follows:

$$S_{avg} = \mathrm{ReLu}(W_1 S_{avg} + b_1) \tag{10}$$

$$S_{\max} = \mathrm{ReLu}(W_1 S_{\max} + b_1) \tag{11}$$

where S_{avg} and $S_{\max}$ are intermediate representations obtained by feeding the global average pooling results; and W_1 and b_1 are the learnable weights and biases of the first fully connected layer.

3) Channel Attention Weights: The output of the MLP passes through a sigmoid activation function that normalizes the channel attention weights to a range of [0, 1]. The channel attention weight is expressed as follows:

$$A = s(W_2(S_{avg} + S_{\max}) + b_2) \tag{12}$$

where A represents the channel attention weights; W_2 and b_2 are the learnable weights and biases of the second fully connected layer; and s denotes the sigmoid function.

4) Feature Rescaling: In the final stage, the computed channel attention weights are applied to the original feature maps via element-wise multiplication. This operation enhances the most informative channels while suppressing less significant channels and effectively recalibrating feature maps. The expression for the feature scaling is as follows:

$$F_{CAM} = F \mathbin{e} A \tag{13}$$

where F_{CAM} represents the feature map adjusted by the channel attention module; and e denotes element-wise multiplication.

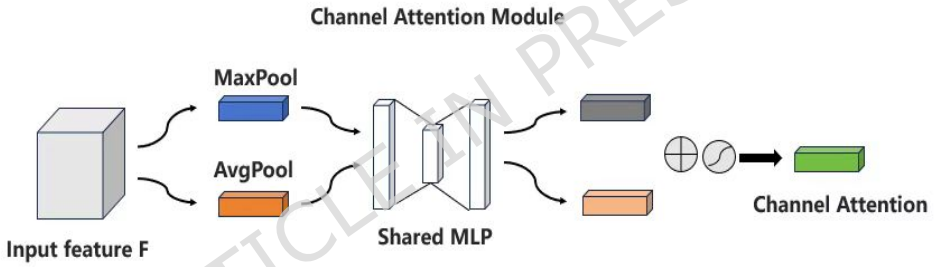


Figure 6. CAM network structure.

Temporal Convolutional Network-Channel Attention Module

Traditional temporal convolutional networks (TCNs) often struggle to effectively capture the dynamic importance of channel features, whereas channel attention mechanisms (CAMs) may lose critical local temporal information owing to the use of global pooling in time-series tasks. To address these limitations, this study proposed a TCN-CAM network that enhances hierarchical feature selection through multilevel temporal feature refinement and a dynamic bi-dimensional attention mechanism spanning both channel and temporal dimensions.

As illustrated in Figure 7, the network adopted a four-level one-dimensional causal convolutional architecture. The input layer processed a 6 × 10 dimensional matrix derived from the UWB localization dataset. The first convolutional layer applied 64 filters of size 1 × 3 to initialize feature extraction. The second layer increased the filter count to 128 (1 × 3) to enhance representational capacity, and the third and fourth layers maintained 128 filters of the same size to stabilize deep feature learning. For temporal modeling, multi-scale dependencies were captured using dilated convolutions with expansion rates d=[1,2,4,8]. To dynamically recalibrate channel-wise feature importance, CAM1 with a compression ratio r=8 was embedded after the second layer (d=2) to refine mid-level features. CAM2 (r=4) was introduced after the fourth layer (d=8) to optimize high-level semantic information. This forms an alternating pattern of convolutional extraction–attention calibration. The final output feature map passes through two fully connected layers (128 and 64 nodes), and the output layer then produces the final predicted 3D coordinates.

During model training, the Adam optimizer was used for parameter updates, with a fixed learning rate of 1×10^{-3} to balance convergence speed and training stability. The batch size was set as 64. A dual regularization strategy was applied to enhance generalization. A dropout (rate = 0.2) was used to randomly deactivate redundant neuron connections, and L2 regularization ($\lambda = 1 \times 10^{-4}$) was implemented to penalize large weight magnitudes and prevent overfitting. This combined strategy significantly

improves the model's generalization ability while ensuring training efficiency. To quantify the model performance, a weighted mean square error loss function was designed as follows:

$$L = 2.0 \times MSE_1 + 2.0 \times MSE_2 + 1.0 \times MSE_3 + \lambda \|w\|_2^2 \quad (14)$$

where the weighting coefficients of $[2.0, 2.0, 1.0]$ are used to balance the contributions of different sample categories; and $\lambda = 1 \times 10^{-4}$ is consistent with the L2 regularization coefficient used during training, ensuring consistency between the evaluation objective and the optimization objective.

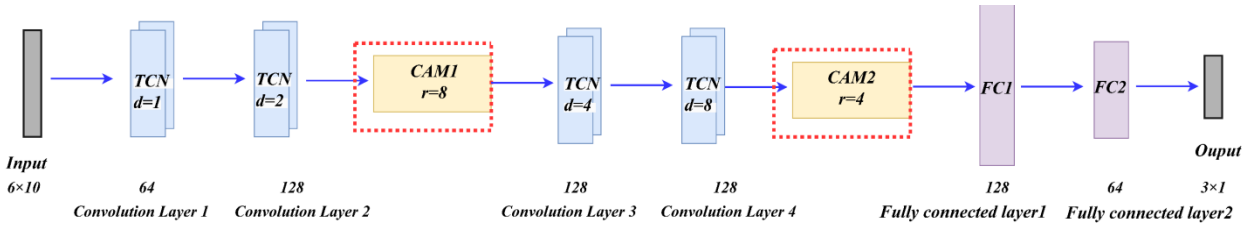


Figure 7. TCN-CAM network structure.

Experiments

Both simulations and field experiments were conducted to verify the effectiveness of the proposed algorithm.

Simulation experiments

Simulation experiments were conducted to evaluate the positioning performance of the proposed TCN-CAM algorithm under NLOS conditions. To simulate the NLOS environment, a synthetic dataset was generated using a real-distance-plus-noise model implemented in Python 3.6. Simulations were performed on a Lenovo Savior Y7000 laptop equipped with 16 GB of RAM and an NVIDIA GeForce GTX 1050 Ti GPU.

During the simulation, six UWB base stations and one UWB tag were deployed. The base stations were labeled A, B, C, D, E, and F, with coordinates denoted as (X_A, Y_A, Z_A), (X_B, Y_B, Z_B), (X_C, Y_C, Z_C), (X_D, Y_D, Z_D), (X_E, Y_E, Z_E), and (X_F, Y_F, Z_F), respectively. Assuming that the coordinates of the tag G are (X_G, Y_G, Z_G), the true distance between base station A and tag G is calculated using Equation (15):

$$D_{AG} = \sqrt{(X_A - X_G)^2 + (Y_A - Y_G)^2 + (Z_A - Z_G)^2} \quad (15)$$

In this simulation, the coordinates of base stations A, B, C, D, E, and F were set to (0, 0, 200), (400, 0, 0), (800, 0, 200), (800, 400, 100), (400, 400, 200), and (0, 400, 100), respectively, with all units in centimeters. The simulation was conducted within an 8 m × 4 m × 2 m (length × width × height) 3D field, where 1,000 coordinate points were randomly generated. Among these, 950 points were utilized as training data, and the remaining 50 were selected as test data for model validation. The actual distance between each base station and the tag was calculated using Equation (15). To simulate real-world conditions, random noise with a standard deviation of 10 cm was added to the actual distance between each base station and the tag. Additionally, NLOS noise in the ranges of 0–10, 0–20, 0–30, 0–40, and 0–50 cm was introduced to further replicate the NLOS conditions. The final input feature for the TCN-CAM network was constructed using the sum of the true distance, random noise, and NLOS error, and the corresponding true coordinates were used as the output target.

Statistical Validation of Simulation Data

To ensure the realism and diversity of the generated NLOS scenarios — which are crucial for robustly evaluating the proposed algorithm — a comprehensive statistical analysis was performed on the simulated ranging errors across all six base stations (A–F) under four noise conditions combining 10 cm of random noise with NLOS errors of 0–10 cm, 0–20 cm, 0–30 cm, and 0–50 cm.

The frequency distribution of the errors was analyzed and fitted with a Gaussian model. The results, detailed in Supplementary Figures 8–11, show high consistency in the Gaussian fitting parameters (e.g., y_0, xc, w) across different base stations under the same condition ($R^2 \geq 0.98$), indicating a uniform error generation mechanism and excellent experimental controllability. As the NLOS error range increased, the parameters evolved systematically: the center (xc) shifted positively, indicating accumulated systematic bias, and the width (w) expanded, reflecting increased dispersion due to more random signal paths.

This progression convincingly demonstrates that our simulation successfully reproduces a wide and realistic spectrum of challenging scenarios, from weak to strong interference, thereby providing a solid foundation for the subsequent algorithm evaluation.

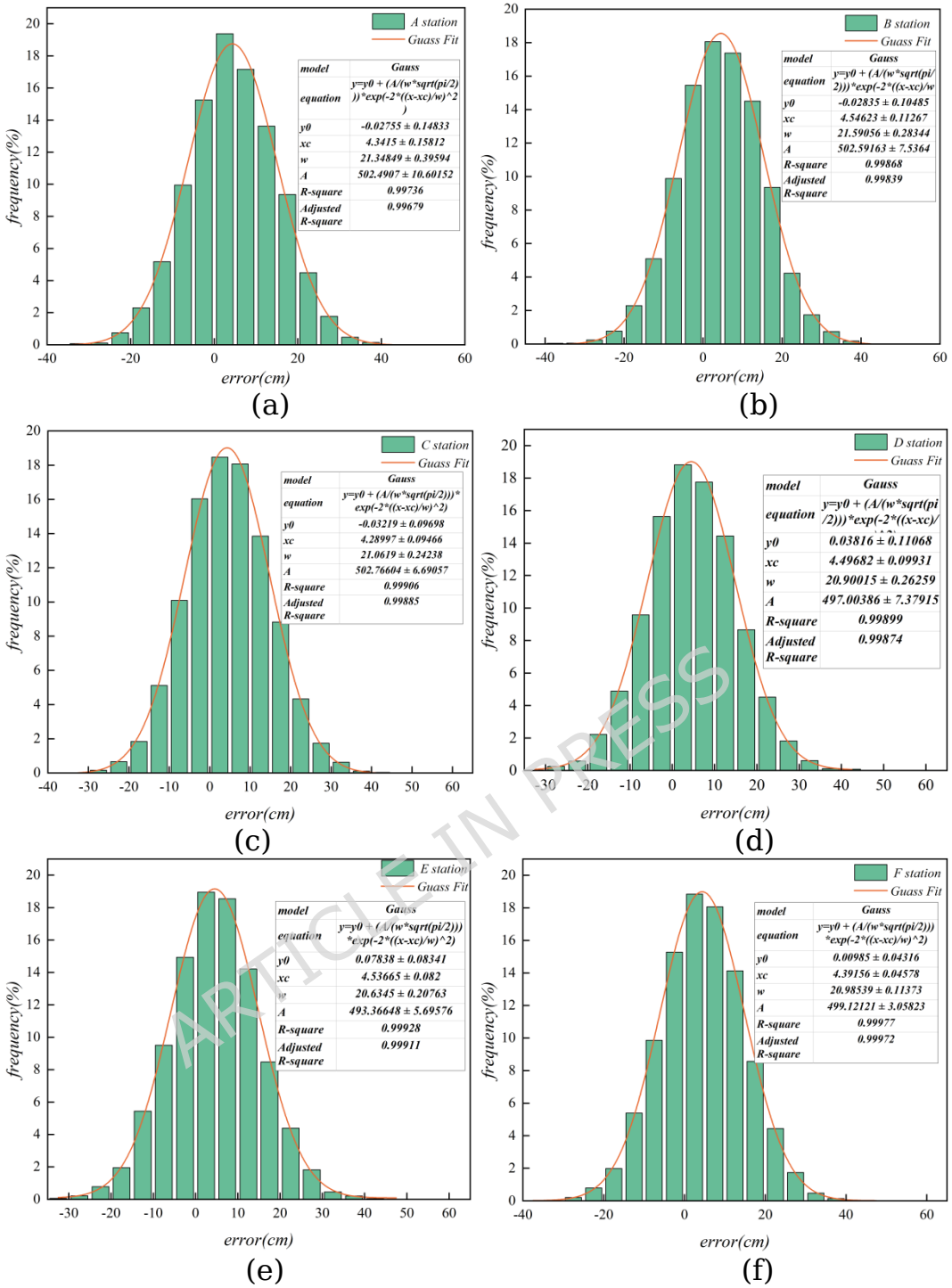


Figure 8. The error frequency histograms and normal fit of (a) A station, (b) B station, (c) C station, (d) D station, (e) E station, and (f) F station. (10 cm noise standard deviation + 0-10 cm NLOS error)

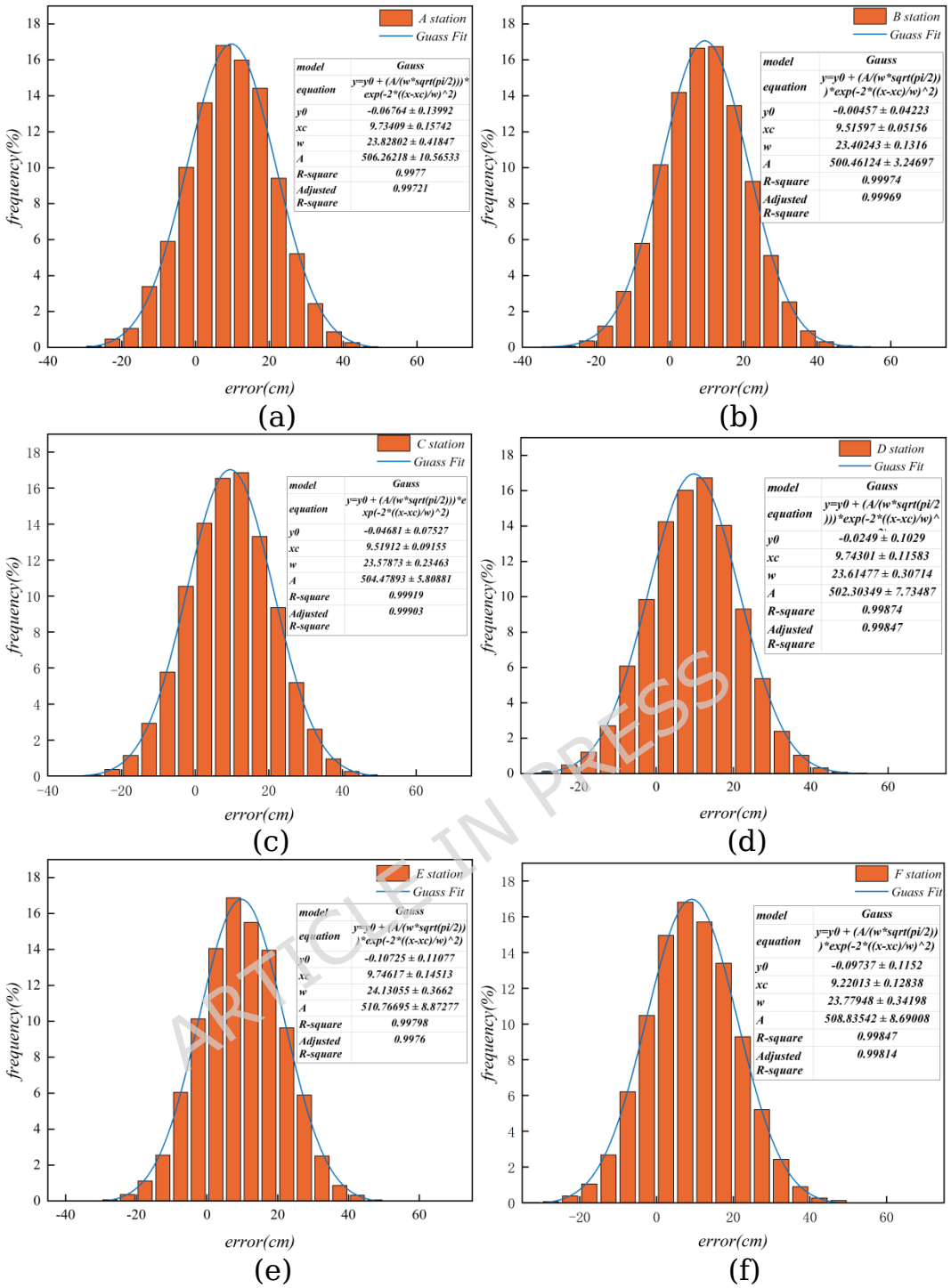


Figure 9. The error frequency histograms and normal fit of (a) A station, (b) B station, (c) C station, (d) D station, (e) E station, and (f) F station. (10 cm noise standard deviation + 0-20 cm NLOS error)

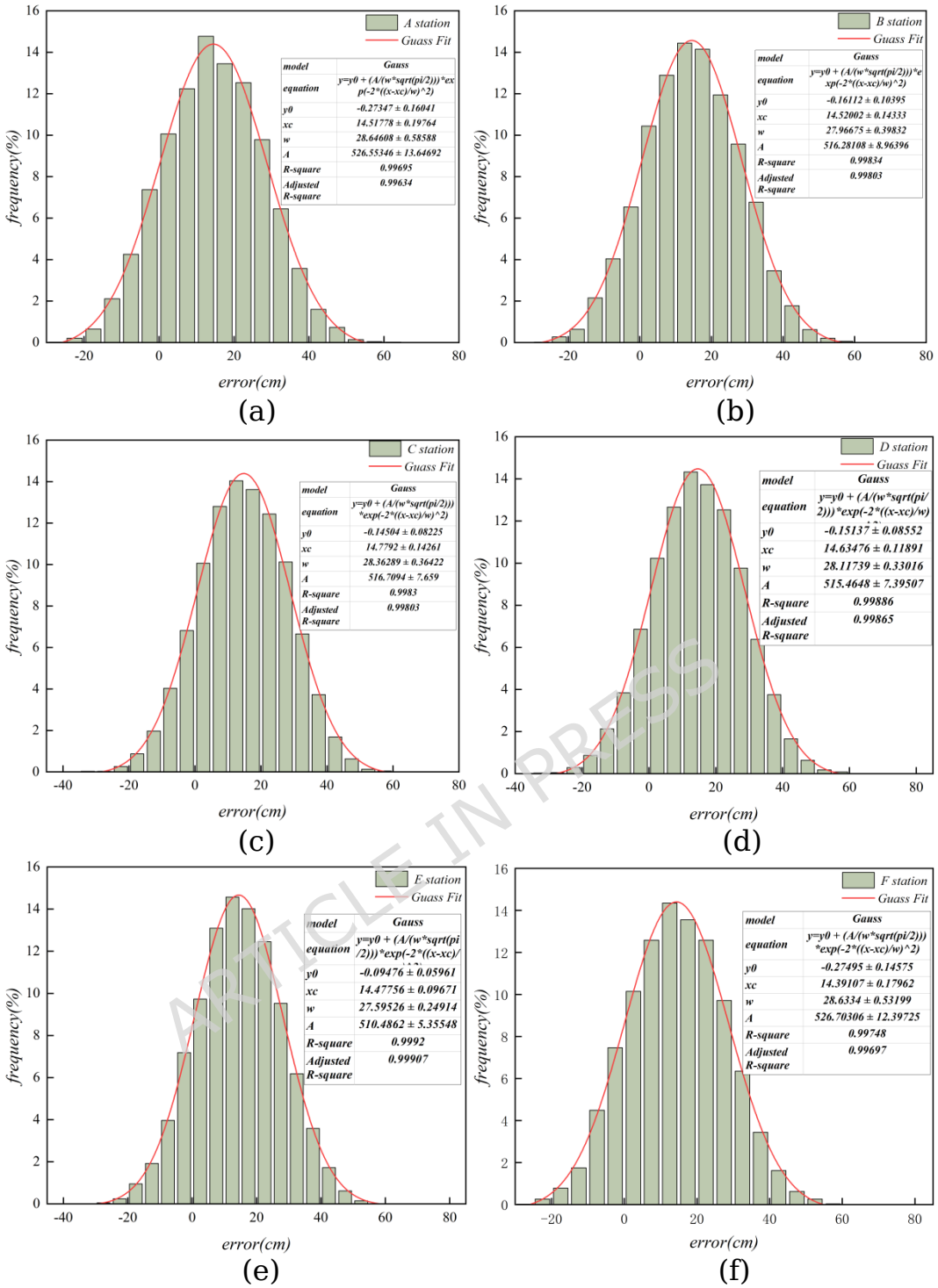


Figure 10. The error frequency histograms and normal fit of (a) A station, (b) B station, (c) C station, (d) D station, (e) E station, and (f) F station. (10 cm noise standard deviation + 0-30 cm NLOS error)

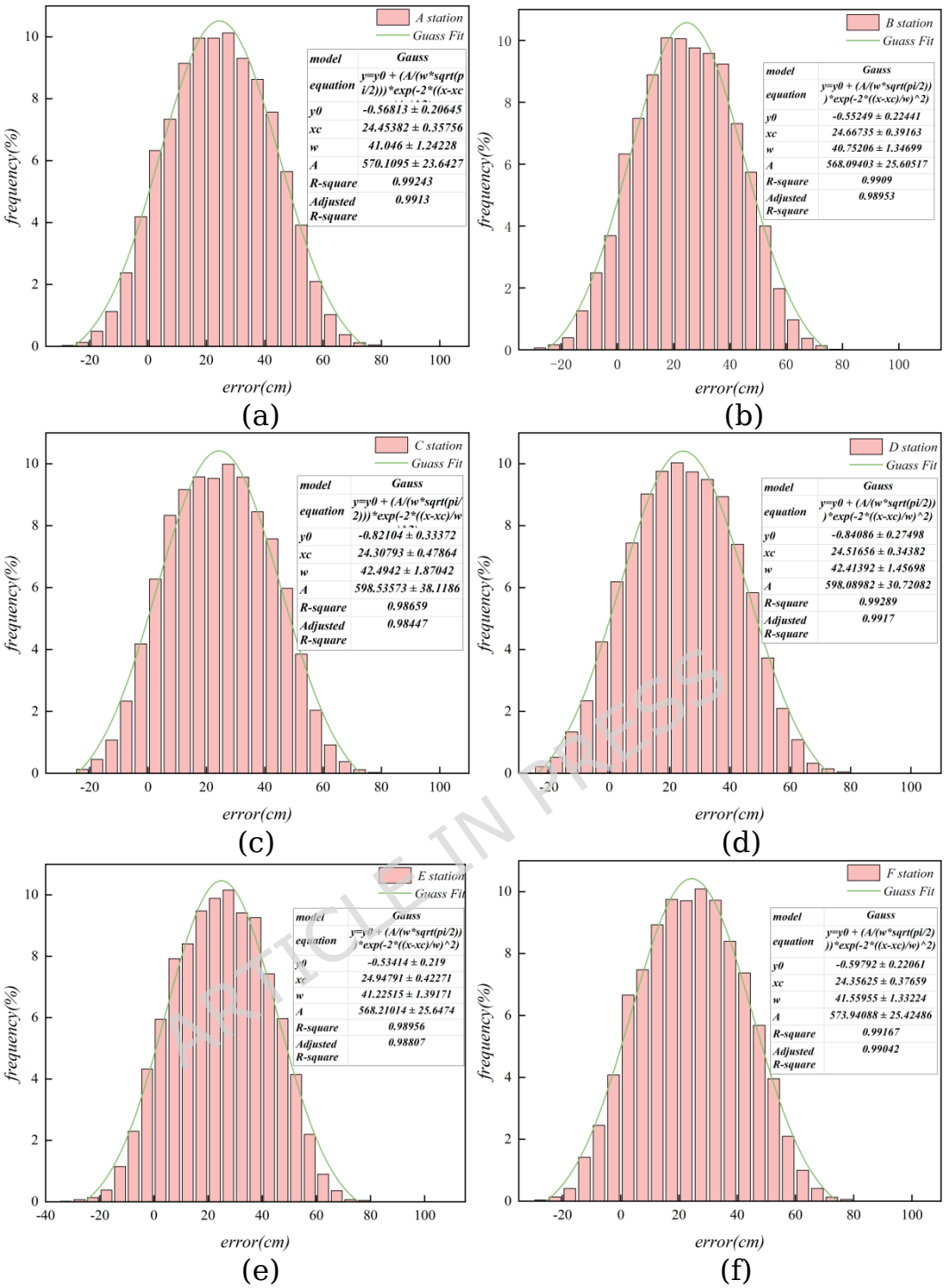


Figure 11. The error frequency histograms and normal fit of (a) A station, (b) B station, (c) C station, (d) D station, (e) E station, and (f) F station. (10 cm noise standard deviation + 0-50 cm NLOS error)

Simulation results

The 3D coordinate data derived from the simulation experiments were processed using multiple algorithms such as TCN-CAM, TCN, CNN-CAM, and LSTM-AM. These algorithms analyze the 3D tag coordinate data to validate the accuracy and stability of their positioning results.

Figure 12 shows the root mean square error (RMSE) curves of the four algorithms. The positioning error increased with the noise level. Among these algorithms, the LSTM-AM algorithm demonstrated the poorest positioning performance, whereas the TCN-CAM algorithm exhibited the lowest positioning error. Initially, with noise of 0–10 cm, the CNN-CAM algorithm had positioning errors similar to those of the TCN algorithm. However, as the noise increased, the positioning errors of the TCN and CNN-CAM algorithms also increased. Moreover, the differences in positioning errors among the algorithms became more pronounced at higher noise levels.

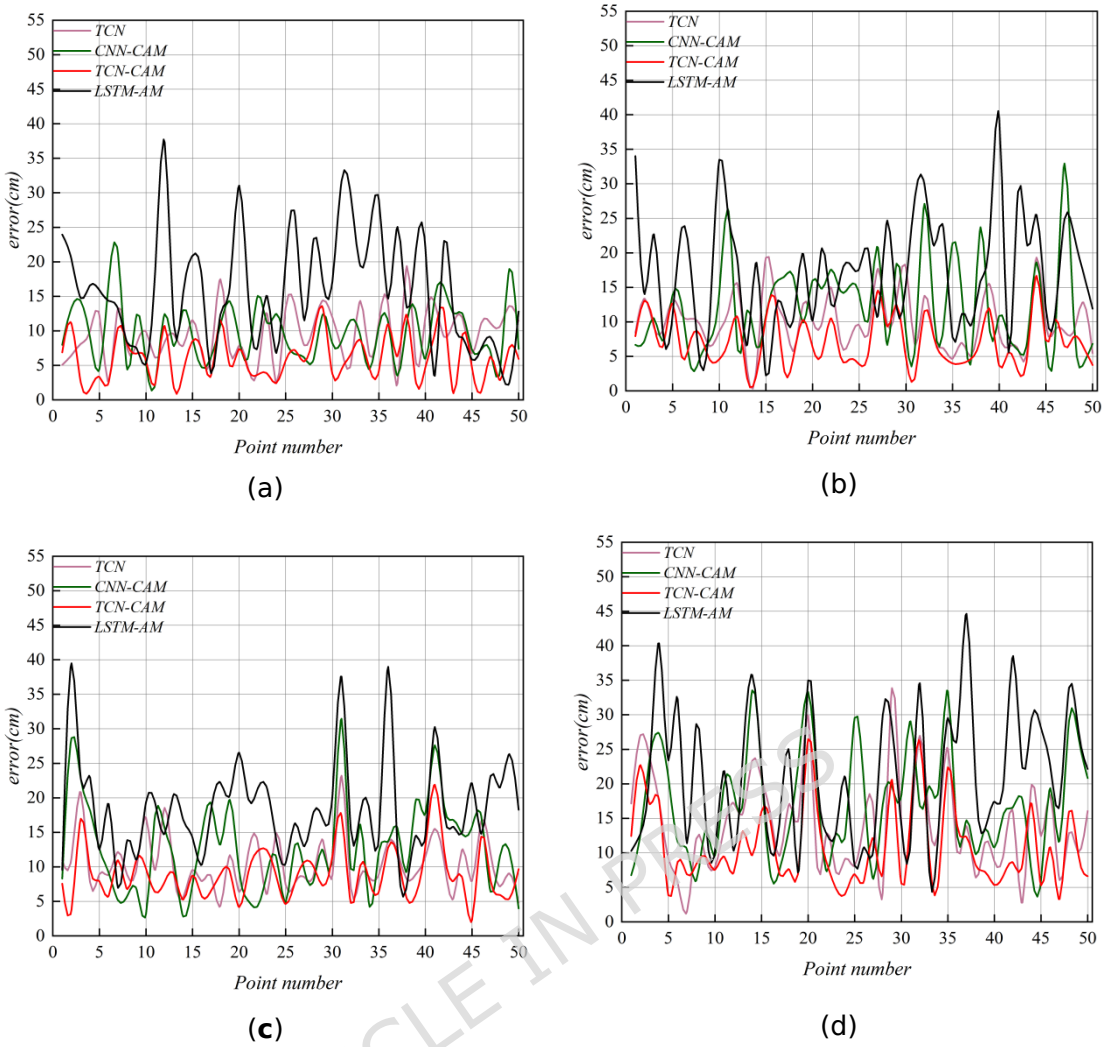
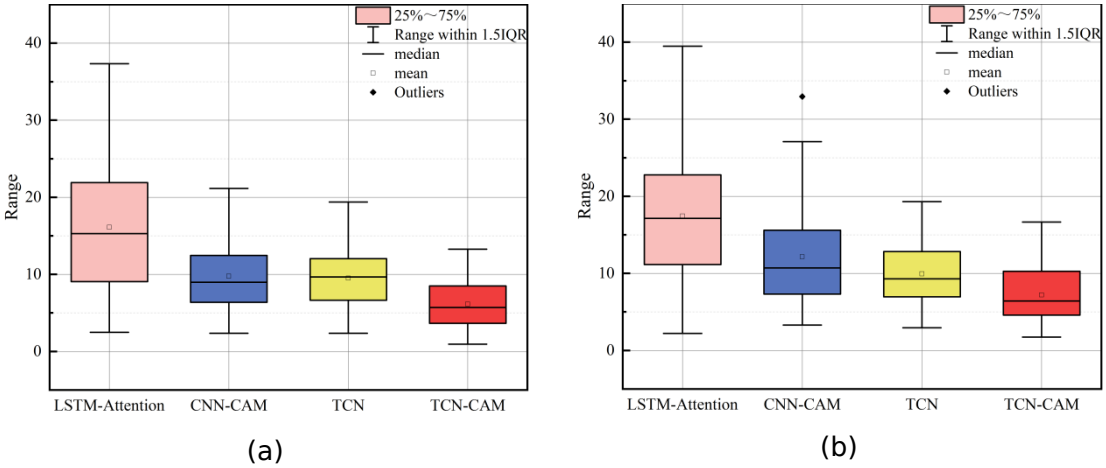
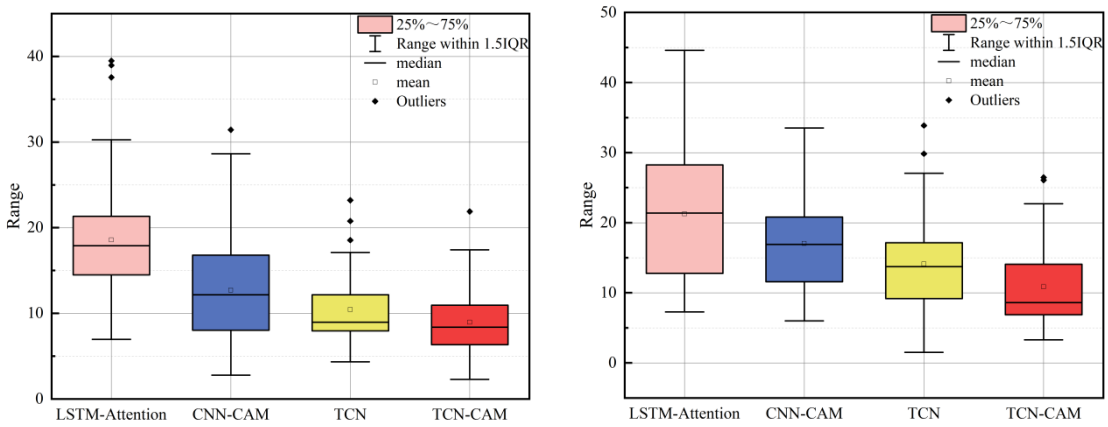


Figure 12. Error curves of the four algorithms under different noise conditions: (a), (b), (c), and (d) correspond to noise ranges of 0–10, 0–20, 0–30, and 0–50 cm, respectively.

Figure 13 presents the box plots of the four algorithms under NLOS noise levels of 0–10, 0–20, 0–30, and 0–50 cm. The results showed a clear trend. The average positioning error of each algorithm increased progressively with increasing noise levels. Among all noise conditions, the LSTM-AM algorithm consistently exhibited the largest box size, widest error distribution, and highest average error. In contrast, the TCN-CAM algorithm achieved the smallest box size, most compact error distribution, and lowest average error.





(c) (d)

Figure 13. Box plots of the four algorithms under different noise conditions: (a), (b), (c), and (d) correspond to noise ranges of 0–10, 0–20, 0–30, and 0–50 cm, respectively.

Figure 14 shows the average error and standard deviation variation across the four algorithms for a more intuitive assessment of their positioning accuracy and stability. The histograms illustrate the average error of the four algorithms across noise levels of 0–10, 0–20, 0–30, and 0–50 cm, offering insights into their localization accuracy. The line graph shows the standard deviation of these algorithms under various noise levels, serving as an indicator of their stability. Under the 0–10 cm noise condition, the positioning errors of the CNN-CAM and TCN algorithms were 9.77 cm and 9.51 cm, respectively, with corresponding standard deviations of 4.16 cm and 3.76 cm, indicating comparable performance. In contrast, the TCN-CAM algorithm achieved a lower positioning error of 6.13 cm and a standard deviation of 3.26 cm, demonstrating improved accuracy and stability. The LSTM-AM algorithm exhibited the highest positioning error and variability, with a mean error of 16.12 cm and a standard deviation of 8.06 cm. Compared with LSTM-AM, CNN-CAM, and TCN, the TCN-CAM algorithm improved positioning accuracy by 61.97%, 37.26%, and 35.54%, respectively. In terms of stability (i.e., lower standard deviation), improvements were 59.18%, 21.63%, and 13.30%, respectively. As the noise increased from 0–30 to 0–50 cm, the standard deviations of CNN-CAM and TCN became increasingly similar. Nonetheless, TCN consistently outperformed CNN-CAM in both accuracy and stability across all noise levels. At the 0–50 cm noise level, the positioning errors of LSTM-AM, CNN-CAM, TCN, and TCN-CAM were 21.26 cm, 17.06 cm, 14.11 cm, and 10.87 cm, respectively. The corresponding standard deviations were 9.47 cm, 7.58 cm, 6.86 cm, and 5.70 cm. These results confirm that the TCN-CAM algorithm outperforms all comparison algorithms in terms of both positioning accuracy and stability. Specifically, it improved stability by 39.81%, 24.80%, and 16.91%, and enhanced positioning accuracy by 48.87%, 36.28%, and 22.96% compared with LSTM-AM, CNN-CAM, and TCN, respectively.

The results clearly indicated that under low noise conditions (e.g., 0–10 cm), the positioning accuracy and stability of the CNN-CAM and TCN algorithms were comparable. However, as the noise level increased, the positioning accuracy of all algorithms decreased. Among them, the TCN-CAM algorithm consistently achieved the lowest positioning error and highest accuracy across all noise levels. Notably, the accuracy gap between CNN-CAM and TCN widened with increasing noise, whereas the stability differences gradually diminished. Although stability declined for all algorithms as noise increased, the TCN-CAM algorithm consistently demonstrated superior stability relative to the others. In contrast, the LSTM-AM algorithm exhibited the poorest performance in terms of both positioning accuracy and stability. The average positioning errors and standard deviations of all four algorithms across different noise ranges are summarized in Table 1.

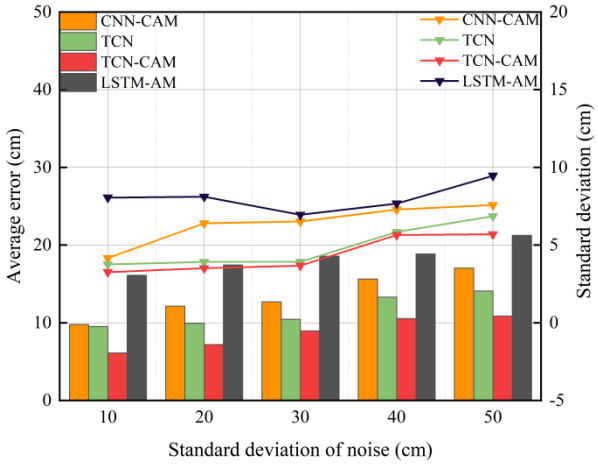


Figure 14. Average error (bars) and standard deviation (lines) for the four algorithms under different noise conditions.

Table 1. Average positioning error and standard deviation for the four algorithms across varying noise ranges.

Noise standard deviation (cm)	Average positioning error (cm)				Standard deviation(cm)			
	LSTM-AM	CNN-CAM	TCN	TCN-CAM	LSTM-AM	CNN-CAM	TCN	TCN-CAM
0–10	16.12	9.77	9.51	6.13	8.06	4.16	3.76	3.26
0–20	17.45	12.16	9.94	7.20	8.12	6.41	3.92	3.51
0–30	18.57	12.71	10.44	8.97	6.95	6.52	3.92	3.67
0–40	18.86	15.63	13.31	10.54	7.68	7.29	5.85	5.65
0–50	21.26	17.06	14.11	10.87	9.47	7.58	6.86	5.70

Field experiments

To further validate the reliability of the TCN-CAM algorithm, field experiments were conducted using the DWM1000 UWB positioning module. The experiments were performed at the School of Surveying and Mapping, Henan Polytechnic University, and involved a setup comprising a computer, a total station, six UWB base stations, and a UWB tag. Figure 15 shows the complete experimental setup and corresponding schematic diagram.

Data acquisition was performed using the LinkPG positioning system developed by the Guangzhou Network Technology Company, which was used to collect UWB ranging data. The true coordinates of the UWB tag at each time point were recorded using a Topcon total station with millimeter-level accuracy, serving as the ground-truth reference. The coordinates of the six UWB base stations were as follows (unit: m): Base Station A (0.4709, 1.6176, 1.0259), Base Station B (0.5042, 5.0750, 1.2905), Base Station C (0.6250, 9.6745, 1.0331), Base Station D (4.4659, 9.5698, 2.0746), Base Station E (4.3152, 5.5233, 1.1126), and Base Station F (4.3744, 1.5687, 2.0728). The total station was positioned at the origin of the coordinate system (0, 0). As the primary aim of this study was to mitigate NLOS errors caused by human movement during UWB positioning, individuals were allowed to move randomly within the experimental area during data collection, thereby intentionally introducing NLOS errors into the UWB-ranging data.

Figure 15. Field experimental setup and scene layout.

The collected experimental data were processed to evaluate the practical effectiveness of the proposed algorithm. Figure 16 presents the coordinate point errors, and Figure 17 illustrates the average errors and standard deviations. Table 2 summarizes the localization error values of the four algorithms.

Figure 16 shows that the LSTM-AM algorithm exhibited the lowest positioning accuracy with the largest positioning errors among all algorithms assessed. The CNN-CAM and TCN algorithms demonstrated smaller localization errors than the LSTM-AM algorithm. However, the TCN algorithm outperformed the CNN-CAM algorithm concerning localization effectiveness. Remarkably, the TCN-CAM algorithm proposed in this study achieved the lowest localization error and the highest localization accuracy.

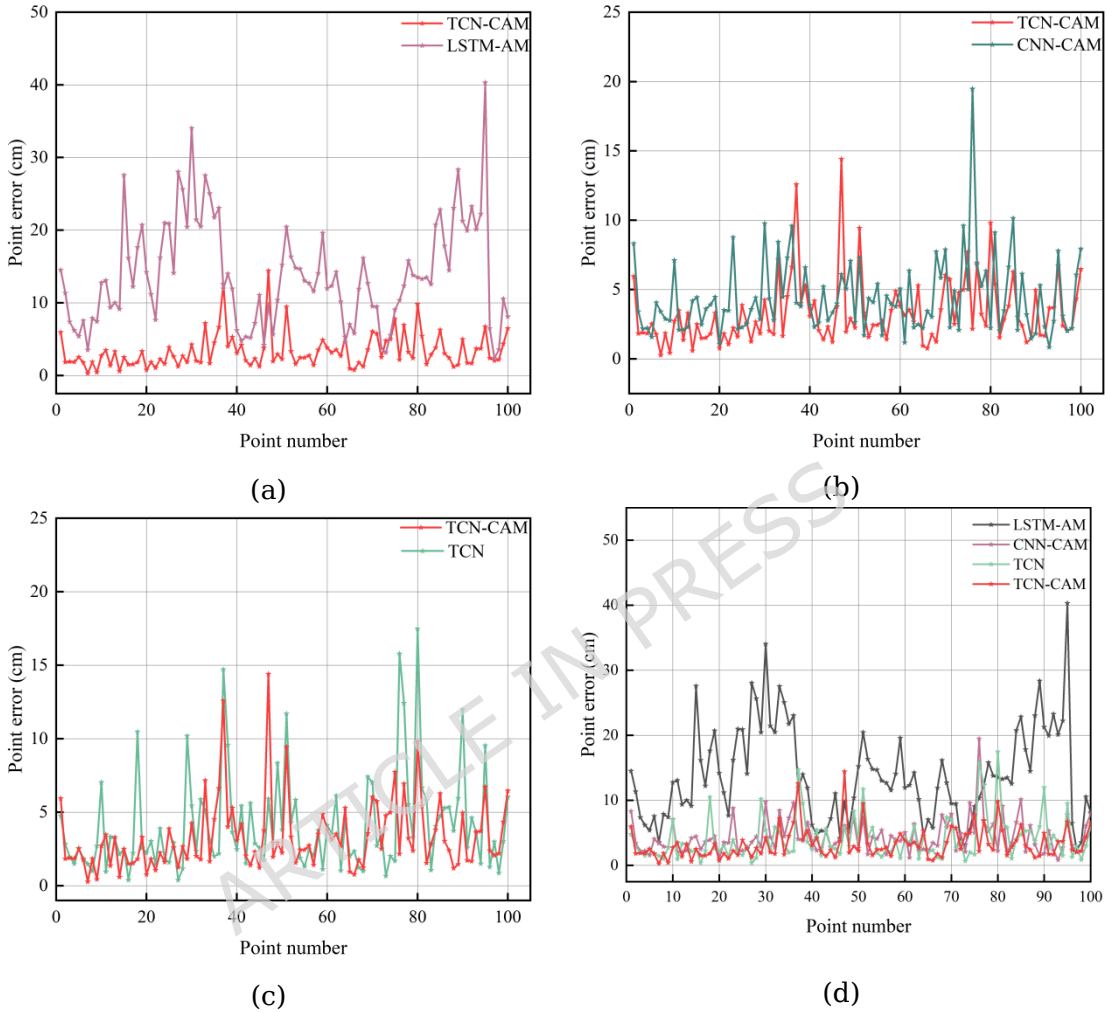


Figure 16. Point-wise localization errors: (a) TCN-CAM vs. LSTM-AM; (b) TCN-CAM vs. CNN-CAM; (c) TCN-CAM vs. TCN; (d) comparison among TCN-CAM, TCN, CNN-CAM, and LSTM-AM.

Figure 17 shows the average error and standard deviation for the four algorithms. The average point error was 3.32 cm for TCN-CAM, 4.12 cm for TCN, 4.43 cm for CNN-CAM, and 13.9 cm for LSTM-AM. Compared with LSTM-AM, CNN-CAM, and TCN algorithms, the TCN-CAM algorithm demonstrated substantial improvements in 3D tag localization accuracy, achieving 76.12%, 25.06%, and 19.42% higher accuracies, respectively. In terms of stability, the standard deviations for TCN-CAM, TCN, CNN-CAM, and LSTM-AM were 2.4 cm, 3.36 cm, 2.72 cm, and 7.2 cm, respectively. Among all algorithms, the TCN-CAM algorithm exhibited the highest stability.

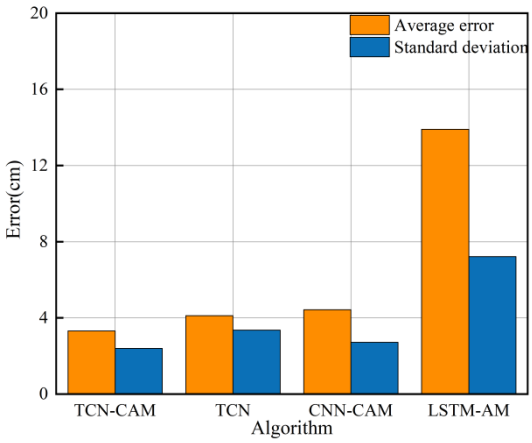


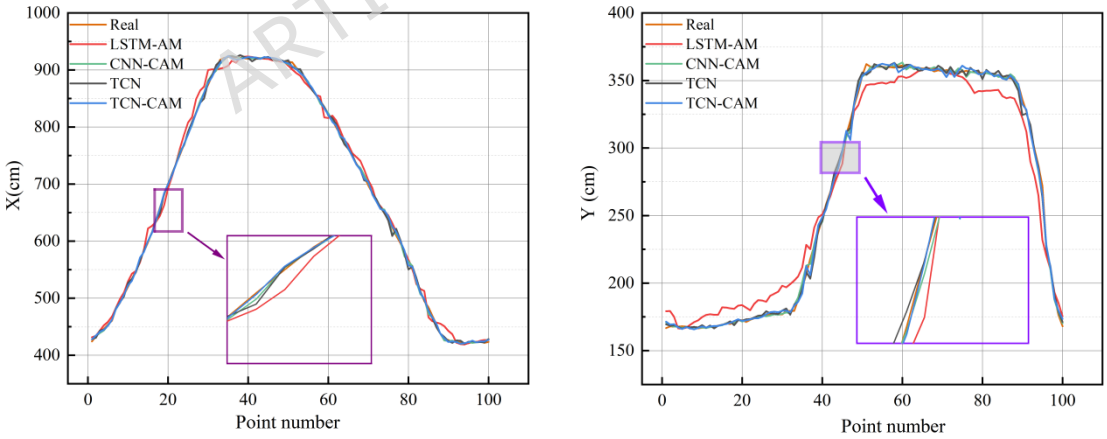
Figure 17. Average error and standard deviation for the four algorithms.

Table 2 summarizes the localization errors of the four algorithms and compares their maximum, minimum, and average errors relative to those of the LSTM-AM algorithm. When using the CNN-CAM algorithm, the maximum localization error was reduced by 51.74%, the minimum error by 64.56%, and the average point accuracy improved by 68.13%. With the TCN algorithm, the maximum and minimum errors were reduced by 56.65% and 83.54%, respectively, whereas the average point accuracy increased by 70.36%. The proposed TCN-CAM algorithm achieved the most significant improvements, reducing the maximum error by 64.26% and the minimum error by 88.61% and enhancing average point accuracy by 76.12%.

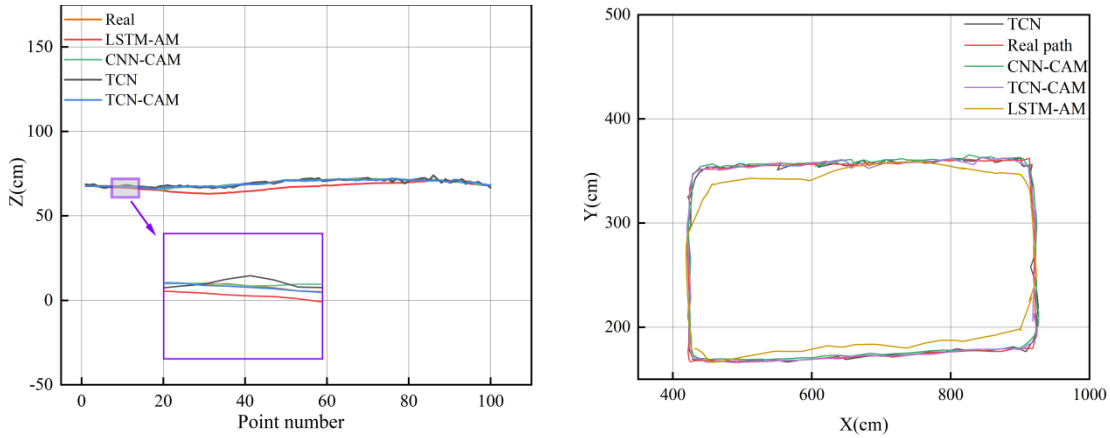
Table 2. Localization accuracy based on measured data for each algorithm.

Algorithms	Maximum error (cm)	minimum error (cm)	Average error (cm)	Standard deviation (cm)
LSTM-AM	40.32	2.37	13.9	7.22
CNN-CAM	19.46	0.84	4.43	2.72
TCN	17.48	0.39	4.12	3.36
TCN-CAM	14.41	0.27	3.32	2.40

Figure 18 illustrates the trajectories of the four algorithms in the X, Y, and Z directions along with the actual motion routes obtained from the field test. Among the algorithms, the trajectory produced by the TCN-CAM algorithm exhibited the closest alignment with the actual path across all three spatial dimensions.



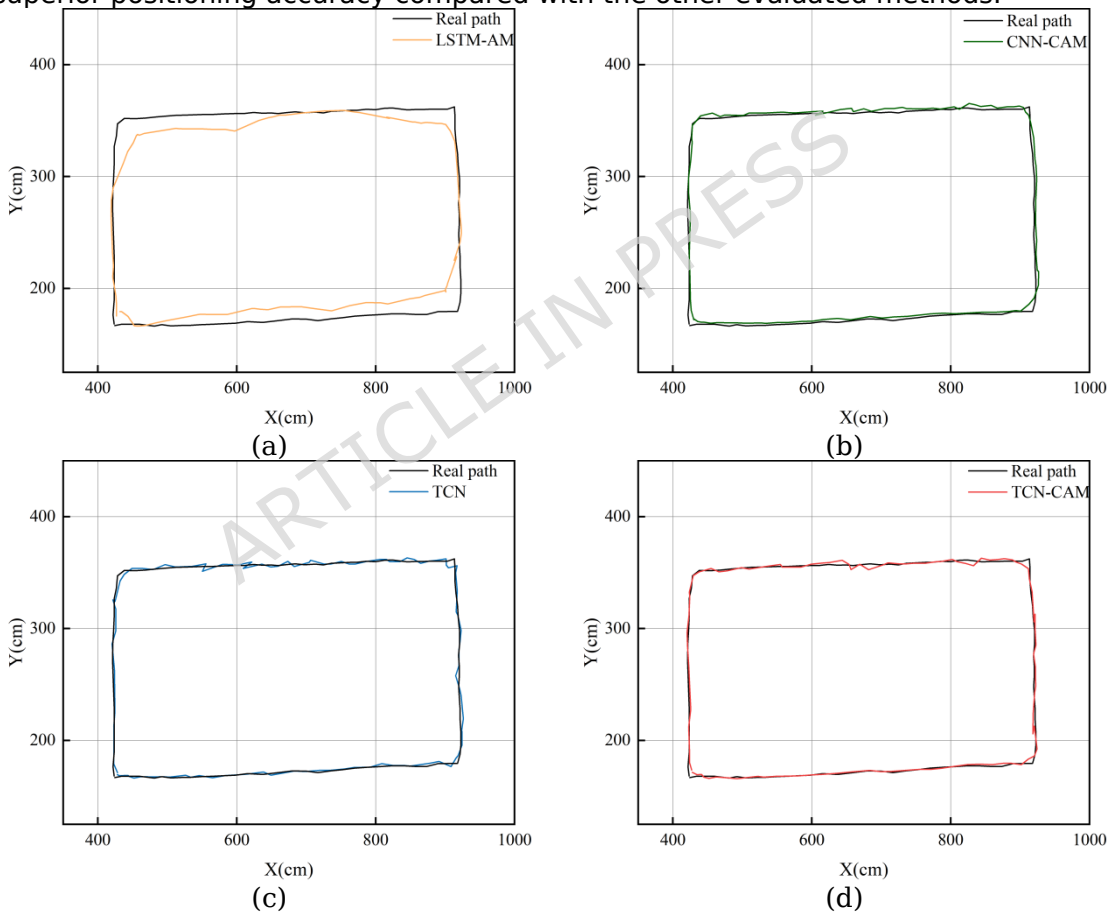
(a) (b)



(c) (d)

Figure 18. Comparison of algorithm trajectories with the actual measured path: (a) X-direction, (b) Y-direction, (c) Z-direction, and (d) actual measurement trajectory.

Figure 19 compares the motion trajectories generated by the different algorithms with the actual paths measured during field tests. The results demonstrate that the proposed TCN-CAM algorithm achieves superior positioning accuracy compared with the other evaluated methods.



(a) (b)

(c) (d)

Figure 19. Comparison of measured trajectories for each algorithm: (a) LSTM-AM, (b) CNN-CAM, (c) TCN, and (d) TCN-CAM.

Discussion

In the field of indoor positioning, UWB technology has attracted significant industrial attention owing to its centimetre-level ranging accuracy, low power consumption, and strong resistance to multipath interference. However, in practical applications, NLOS propagation often leads to a substantial reduction in positioning accuracy. Among the various sources of interference, dynamic obstruction caused by human movement is particularly prominent. This type of interference is characterized by time-varying randomness, presenting considerable challenges for maintaining accurate and reliable positioning.

This study proposed a TCN-CAM algorithm and evaluated its effectiveness in 3D indoor positioning applications. TCN effectively extracted temporal features while mitigating the gradient vanishing problem

commonly observed in recurrent neural networks, particularly LSTM-based models. CAM enhanced the model's adaptability to complex environments by improving feature selection. To evaluate the algorithm's performance, simulation experiments were conducted by introducing NLOS noise with varying ranges, followed by validation through field experiments. Compared with LSTM-AM, CNN-CAM, and TCN, the proposed TCN-CAM algorithm demonstrated superior performance in both positioning accuracy and stability.

Despite its strong performance in indoor positioning, the TCN-CAM algorithm has certain limitations. Its effectiveness is highly dependent on the structure of input data. Specifically, the model relies on time-series data collected from fixed base station positions, requiring that the input data originate from the same base stations used during training. Although the model can directly output target 3D coordinates, significant changes in base station locations render the pre-trained model parameters invalid. In such cases, new data must be collected, and the model must be retrained. Nevertheless, the TCN-CAM algorithm remains highly promising for applications in environments in which base station locations remain stable over extended periods.

Future research will focus on reducing the algorithm's sensitivity to changes in base station locations. This includes the development of a hybrid architecture that integrates geometric constraint models with deep learning frameworks, aiming to decouple location-dependent features from channel characteristics, thereby reducing reliance on fixed infrastructure. Additionally, multi-sensor fusion has been identified as a promising direction to further enhance positioning robustness and adaptability.

In summary, the TCN-CAM algorithm offers an effective solution for mitigating NLOS errors in indoor positioning systems and demonstrates significant practical value in environments in which base station locations remain stable over time. With continued technological advancements and optimization, the TCN-CAM algorithm is expected to play an increasingly important role across a broader range of real-world applications.

Conclusions

This study proposed an algorithm that integrates a temporal convolutional network with a channel attention module to reduce the impact of NLOS errors caused by human movement on indoor positioning accuracy. The positioning accuracy and stability of the TCN-CAM, LSTM-AM, CNN-CAM, and TCN algorithms were evaluated through both simulations and field experiments.

(1) Simulation results demonstrated that TCN-CAM significantly outperformed TCN, CNN-CAM, and LSTM-AM. Its unique architecture enabled the TCN algorithm to process signal features in noisy environments compared to CNN and LSTM. Additionally, the introduction of the channel attention mechanism further enhanced both stability and positioning accuracy. Specifically, TCN-CAM achieved approximately 20–50% higher positioning accuracy than the other three algorithms.

(2) Field experiment results confirmed the superior performance of the TCN-CAM algorithm. It consistently outperformed LSTM-AM, CNN-CAM, and TCN in terms of positioning accuracy and exhibited the lowest standard deviation, indicating the highest stability. Specifically, the positioning accuracy of TCN-CAM was 76.12%, 25.06%, and 19.42% higher than that of LSTM-AM, CNN-CAM, and TCN, respectively.

References

1 Yao, L. *et al.* GNSS/UWB/INS indoor and outdoor seamless positioning algorithm based on federal filtering. *Measurement Science and Technology* **35**, doi:10.1088/1361-6501/ad03ba (2024).

2 Tian, Y. *et al.* Application of a long short-term memory neural network algorithm fused with Kalman filter in UWB indoor positioning. *Scientific Reports* **14**, doi:10.1038/s41598-024-52464-y (2024).

3 Jin, R. *et al.* Toward Practical Lightweight Passive Human Tracking Using WiFi Sensing. *Ieee Internet of Things Journal* **10**, 13769-13783, doi:10.1109/jiot.2023.3262960 (2023).

4 Song, K. & Paik, J.-H. Bluetooth AoA based Positioning Scheme using Angle and Distance Validation Test. *Journal of Broadcast Engineering* **26**, 790-798, doi:10.5909/jbe.2021.26.6.790 (2021).

5 Qi, M., Xue, B. & Wang, W. Calibration and Compensation of Anchor Positions for UWB Indoor Localization. *IEEE Sensors Journal* **24**, 689-699, doi:10.1109/jsen.2023.3329535 (2024).

6 Yang, S. *et al.* 5G Indoor Positioning Error Correction Based on 5G-PECNN. *Sensors* **24**, doi:10.3390/s24061949 (2024).

7 Liu, Z. *et al.* Precise, Low-Cost, and Large-Scale Indoor Positioning System Based on Audio Dual-Chirp Signals. *IEEE Transactions on Vehicular Technology* **72**, 1159-1168, doi:10.1109/tvt.2022.3205960 (2023).

8 Kim, D.-H., Farhad, A. & Pyun, J.-Y. UWB Positioning System Based on LSTM Classification With Mitigated NLOS Effects. *IEEE Internet of Things Journal* **10**, 1822-1835, doi:10.1109/jiot.2022.3209735 (2023).

9 Efremova, E. V., Kuzmin, L. V. & Itskov, V. V. Measuring Received Signal Strength of UWB Chaotic Radio Pulses for Ranging and Positioning. *Electronics* **12**, doi:10.3390/electronics12214425 (2023).
10 Margiani, T. *et al.* Angle of Arrival and Centimeter Distance Estimation on a Smart UWB Sensor Node. *IEEE Transactions on Instrumentation and Measurement* **72**, doi:10.1109/tim.2023.3282289 (2023).
11 Deng, W., Li, J., Tang, Y. & Zhang, X. Low-Complexity Joint Angle of Arrival and Time of Arrival Estimation of Multipath Signal in UWB System. *Sensors* **23**, doi:10.3390/s23146363 (2023).
12 Zhao, W., Goudar, A., Qiao, X. & Schoellig, A. P. UTIL: An ultra-wideband time-difference-of-arrival indoor localization dataset. *International Journal of Robotics Research*, doi:10.1177/02783649241230640 (2024).
13 Wang, P. *et al.* Application of the Least Squares-Adaptive Vector Projection Iteration Algorithm to Ultra-Wideband Positioning. *IEEE Sensors Journal* **24**, doi:10.1109/JSEN.2024.3461155 (2024).
14 Dong, J., Lian, Z., Xu, J. & Yue, Z. UWB Localization Based on Improved Robust Adaptive Cubature Kalman Filter. *Sensors* **23**, doi:10.3390/s23052669 (2023).
15 Xin, J., Gao, K., Shan, M., Yan, B. & Liu, D. A Bayesian Filtering Approach for Error Mitigation in Ultra-Wideband Ranging. *Sensors* **19**, doi:10.3390/s19030440 (2019).
16 Li, S. & Wu, J. Research on NLOS error suppression in UWB based on RICT algorithm. *Measurement* **244**, doi:10.1016/j.measurement.2024.116463 (2025).
17 Lu, J., Ma, G. & Zhang, G. Fuzzy Machine Learning: A Comprehensive Framework and Systematic Review. *IEEE Transactions on Fuzzy Systems* **32**, 3861-3878, doi:10.1109/tfuzz.2024.3387429 (2024).
18 Cho, H. *et al.* Machine Learning and Health Science Research: Tutorial. *Journal of Medical Internet Research* **26**, doi:10.2196/50890 (2024).
19 Alzoubi, Y. I., Mishra, A. & Topcu, A. E. Research trends in deep learning and machine learning for cloud computing security. *Artificial Intelligence Review* **57**, doi:10.1007/s10462-024-10776-5 (2024).
20 Thang Van, N., Jeong, Y., Shin, H. & Win, M. Z. Machine Learning for Wideband Localization. *IEEE Journal on Selected Areas in Communications* **33**, 1357-1380, doi:10.1109/jsac.2015.2430191 (2015).
21 Sang, C. L. *et al.* Identification of NLOS and Multi-Path Conditions in UWB Localization Using Machine Learning Methods. *Applied Sciences-Basel* **10**, doi:10.3390/app10113980 (2020).
22 Barral, V., Escudero, C. J., Garcia-Naya, J. A. & Suarez-Casal, P. Environmental Cross-Validation of NLOS Machine Learning Classification/Mitigation with Low-Cost UWB Positioning Systems. *Sensors* **19**, doi:10.3390/s19245438 (2019).
23 Tian, Y. *et al.* The application of gated recurrent unit algorithm with fused attention mechanism in UWB indoor localization. *Measurement* **234**, doi:10.1016/j.measurement.2024.114835 (2024).
24 Hapsari, G. I., Munadi, R., Erfianto, B. & Irawati, I. D. Future Research and Trends in Ultra-Wideband Indoor Tag Localization. *IEEE Access*, 1-1, doi:10.1109/access.2024.3399476 (2024).
25 Pei, Y., Chen, R., Li, D., Xiao, X. & Zheng, X. FCN-Attention: A deep learning UWB NLOS/LOS classification algorithm using fully convolution neural network with self-attention mechanism. *Geo-Spatial Information Science* **27**, 1162-1181, doi:10.1080/10095020.2023.2178334 (2024).
26 Jiang, C. *et al.* UWB NLOS/LOS Classification Using Deep Learning Method. *Ieee Communications Letters* **24**, 2226-2230, doi:10.1109/lcomm.2020.2999904 (2020).
27 Wu, Y., He, X., Mo, L. & Wang, Q. Self-Attention-Assisted TinyML With Effective Representation for UWB NLOS Identification. *Ieee Internet of Things Journal* **11**, 25471-25480, doi:10.1109/jiot.2024.3349462 (2024).
28 Zhang, J. *et al.* Research on None-Line-of-Sight/Line-of-Sight Identification Method Based on Convolutional Neural Network-Channel Attention Module. *Sensors* **23**, doi:10.3390/s23208552 (2023).
29 Wei, J. *et al.* NLOS identification using parallel deep learning model and time-frequency information in UWB-based positioning system. *Measurement* **195**, doi:10.1016/j.measurement.2022.111191 (2022).
30 Joung, J., Jung, S., Chung, S. & Jeong, E. R. CNN-based Tx-Rx distance estimation for UWB system localisation. *Electronics Letters* **55**, 938-+, doi:10.1049/el.2019.1084 (2019).
31 Wu, X. *et al.* RNNtcs: A test case selection method for Recurrent Neural Networks. *Knowledge-Based Systems* **279**, doi:10.1016/j.knosys.2023.110955 (2023).
32 Zarzycki, K. & Lawrynczuk, M. Advanced predictive control for GRU and LSTM networks. *Information Sciences* **616**, 229-254, doi:10.1016/j.ins.2022.10.078 (2022).

33 Kharakhashyan, A. & Maltseva, O. Comparison of the Forecast Accuracy of Total Electron Content for Bidirectional and Temporal Convolutional Neural Networks in European Region. *Remote Sensing* **15**, doi:10.3390/rs15123069 (2023).
34 Wang, Y. *et al.* An attention mechanism module with spatial perception and channel information interaction. *Complex & Intelligent Systems* **10**, 5427-5444, doi:10.1007/s40747-024-01445-9 (2024).

Author Contributions:

Conceptualization, L.H. and Z.L.; methodology, L.H.; software, C.Z.; validation, L.H., Y.T., and H.C.; formal analysis, L.H.; investigation, H.C.; resources, Z.L.; data curation, Z.L.; writing—original draft preparation, L.H.; writing—review and editing, Y.T. and M.A.; visualization, Y.T.; supervision, Z.L.; project administration, H.C.; funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the Fundamental Research Funds for the Universities of Henan Province (Grant Number NSFRF230405), Doctoral Scientific Fund Project of Henan Polytechnic University (Grant Number B2017-10), Henan Polytechnic University Funding Plan for Young Backbone Teachers (Grant Number 2022XQG-08), Henan Province Science and Technology Research Projects (Grant Number: 242102320070), National Natural Science Foundation of China (Grant Number 42374029), and Henan Polytechnic University Surveying and Mapping Science and Technology "Double First-Class" Discipline Creation Project (Grant Number: CHXKYXBS05).

Competing interests

The authors declare no conflicts of interest.

Additional information

Data availability

These were computer-generated and gathered in the experimental area; they are not yet accessible to the general public or the Internet. If necessary, they can be acquired from the corresponding author.

Informed consent statement

All subjects and/or their legal guardian(s) for both study participation and publication of identifying information or images in an online open-access publication (when applicable).