Data-Driven Tuning of Neural Network Aided Least Squares for UWB-TDoA Indoor Positioning

Ryoichi Kawaguchi*, Shinsuke Ibi*, and Hisato Iwai*
* Faculty of Science and Engineering, Doshisha University, 1-3 Tataramiyakodani, Kyotanabe, 610-0394, Japan
E-mail: ctwn0333@mail4.doshisha.ac.jp, {sibi, iwai}@mail.doshisha.ac.jp

***Abstract*—Ultra-wideband (UWB) communications are expected to be used for indoor positioning technology owing to their wide bandwidth and low power consumption. In particular, the time difference of arrival (TDoA) method can provide highly accurate positioning, and the transmitter position is estimated using the least squares (LS) method. However, positioning accuracy is unavoidably affected by non-line of sight (NLOS) and multipath. This study proposes the application of a neural network (NN) to the weighted least squares method for typical TDoA positioning algorithms. We used two NN models: a simple model, the multi-layer perceptron (MLP), and a model specialized for time-series data, the long-short-term memory (LSTM). The validity of the proposed method is demonstrated by tracking a moving transmitter through experimentation.**

I. Introduction

Position estimation using wireless technology has made significant progress. Its applications are diverse, including automotive navigation, which uses satellite radio waves to provide position information [1], asset management, inventory tracking, and assembly process management in industrial fields [2]. The global navigation satellite system (GNSS) is a representative position estimation technology and one of the most accurate position estimation technologies [3]. However, GNSS is designed for use in environments where there are no obstacles between the satellite and the target receiver. On the other hand, exterior walls and other obstacles block satellite signals indoors, causing severe signal attenuation and significantly reducing positioning accuracy [4]. As a result, many studies are being conducted on indoor positioning technologies that can replace GNSS [5]. The main indoor positioning methods include the use of Wi-Fi and Bluetooth signals; however, in this study, we focus on ultra-wideband (UWB) signals.

UWB is a standard for low-power, short-range wireless communications [6]. UWB signals have the advantage of a wide bandwidth, allowing information to spread over a wide range of frequencies. This spread reduces the power spectral density, making it possible to minimize the interference from other wireless systems [7]. The primary positioning method for UWB is time difference of arrival (TDoA). TDoA takes advantage of the differences in the propagation time of a transmitted signal measured by different receivers [8]. TDoA positioning uses the range differences between three or more receivers and a transmitter to estimate the position. Least squares (LS) and Taylor series methods are used to calculate the estimated position. Furthermore, the positioning accuracy of the TDoA depends on the signal bandwidth, the sampling rate at the receiver, and the presence or absence of a direct line of sight (LOS) between the transmitter and the receiver. Other time-based techniques include time of flight (ToF) and time of arrival (ToA). However, TDoA has the advantage that it does not require synchronization between the transmitter and receivers among the receivers themselves. Therefore, this study adopts TDoA for positioning. The positioning of UWB-TDoA uses wideband communication, which improves time resolution and enables highly accurate indoor positioning in LOS environments [9], and has superior resistance to multipath and non-line of sight (NLOS) environments compared to other indoor positioning technologies. However, its influence cannot be completely eliminated, leaving room for improvement in positioning accuracy.

To address this problem, we introduce machine learning (ML) techniques that have been widely used in recent years. In [10], a method was proposed to mitigate the effects of NLOS and correct the measurement error of the distance between the transmitter and receiver by applying a convolutional neural network (CNN). The input data for CNN is the UWB channel impulse response (CIR), which has been preprocessed to focus only on the relevant data of the direct and reflected waves. In [11], this method proposed an unsupervised ML approach to NLOS classification using techniques such as the Gaussian mixture model (GMM) as a classification model. Many studies have proposed methods that use the CIR to correct for measurement errors in the distance between the transmitter and the receiver, but few papers have proposed methods to apply ML to the TDoA positioning algorithm itself.

With the above as background, this study aims to improve positioning accuracy by applying neural networks (NN), a type of ML, to TDoA-based positioning. Specifically, we propose to introduce correction values into the LS method of UWB-TDoA positioning and to adjust the correction values using data-driven NN. As NN models, we propose and evaluate the effectiveness of two types: a simple multi-layer perceptron (MLP) and a time-series-specialized model called long short-term memory (LSTM).

II. Indoor Positioning Using TDoA

Fig. 1 shows the overview of the TDoA-based positioning system assumed in this study. A transmit (TX) terminal, located at an arbitrary position within the designated positioning area, sends ranging signals to the receivers (RXs) mounted in the area. Using multiple receivers, the arrival time at

979-8-3315-7206-8/25/$31.00

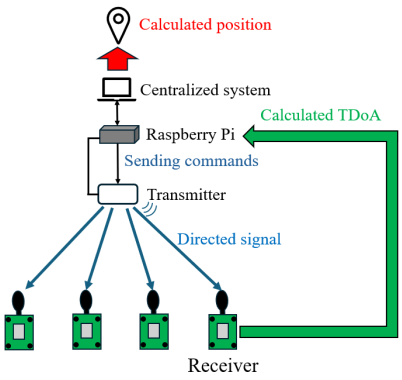


Fig. 1: Overview of TDoA-based positioning system.

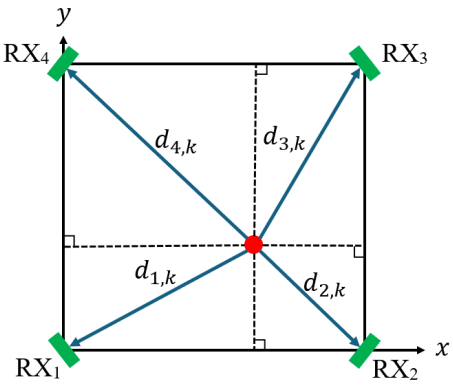


Fig. 2: TDoA-based estimation of transmitter position.

one of them is used as a reference, and then the TDoA is observed with respect to the other receivers. After that, a position estimator aggregates this information and determines the position coordinates of the TX using the LS method.

Fig. 2 illustrates an overview of the TDoA-based indoor positioning environment. The green boxes represent mounted RXs (RX_1 to RX_4), while the red circle represents a TX. TDoA calculates the time differences of arrival of signals from a transmitter to plural fixed receivers, then multiplies these differences by the speed of light $\mathrm{c} = 3.0 \times 10^8$m/s to obtain distance differences. The pseudo-distance $d_{j,k}$ between the transmitter and receiver can be calculated by solving simultaneous equations derived from the observed differences in distance. These calculated distances $d_{j,k}$ [m] between the j-th RX and the TX at the k-th observed data point are used as measurement data, where $k \in [1, K]$. In vector form, we have $\boldsymbol{d}_k = [d_{1,k}, d_{2,k}, d_{3,k}, d_{4,k}]^\mathsf{T}$. Since this study is based on two-dimensional positioning, when the transmitter and receiver are at different heights, the data are converted to horizontal distances using the Pythagorean theorem, ensuring its applicability to the method employed in this study. The coordinate vectors of position of TX and RX are denoted by $\boldsymbol{x}_k = [x_k^{\mathrm{TX}}, y_k^{\mathrm{TX}}]^\mathsf{T}$ and $\boldsymbol{u}_j = [x_j^{\mathrm{RX}}, y_j^{\mathrm{RX}}]^\mathsf{T}$, respectively. Let the coordinate vector of position of the estimated value be denoted by $\hat{\boldsymbol{x}}_k = [\hat{x}_k, \hat{y}_k]$.

In principle, the distance $d_{j,k}$ is formulated by a nonlinear simultaneous equation, which is expressed as

$$d_{j,k}^2 = \left(x_j^{\mathrm{RX}} - \hat{x}_k\right)^2 + \left(y_j^{\mathrm{RX}} - \hat{y}_k\right)^2. \tag{1}$$

However, it is difficult to solve $\hat{\boldsymbol{x}}_k$ directly due to the enormous amount of calculation required. To relax this infeasibility, linearization is applied by taking the difference of $d_{j-1,k}^2 - d_{4,k}^2$, resulting in

$$\boldsymbol{A}\hat{\boldsymbol{x}}_k = \boldsymbol{b} \tag{2}$$

where we have

$$\boldsymbol{A} = 2[\boldsymbol{u}_1 - \boldsymbol{u}_4, \boldsymbol{u}_2 - \boldsymbol{u}_4, \boldsymbol{u}_3 - \boldsymbol{u}_4]^\mathsf{T}, \tag{3}$$

$$\boldsymbol{b} = [b_1, b_2, b_3]^\mathsf{T}, \tag{4}$$

$$b_l = \left(x_l^{\mathrm{RX}}\right)^2 - \left(x_4^{\mathrm{RX}}\right)^2 + \left(y_l^{\mathrm{RX}}\right)^2 - \left(y_4^{\mathrm{RX}}\right)^2 + d_{4,k}^2 - d_{l,k}^2. \tag{5}$$

Then, by solving (2) based on the LS method [12], the estimated position $\hat{\boldsymbol{x}}_k$ of TX can be obtained as

$$\hat{\boldsymbol{x}}_k = \left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\mathsf{T}\boldsymbol{b}. \tag{6}$$

The LS shown above is severely affected by the estimation errors of $d_{j,k}$, which are enhanced in NLOS environments. To mitigate the negative impacts of errors, this study uses a weighted LS (WLS) method by introducing weights $\boldsymbol{Z}_k$. Furthermore, the method proposed in this study introduces the correction bias $\hat{\boldsymbol{x}}'_k = [x'_k, y'_k]^\mathsf{T}$ to correct the WLS as

$$\hat{\boldsymbol{x}}_k = \left(\boldsymbol{A}^\mathsf{T}\boldsymbol{Z}_k\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\mathsf{T}\boldsymbol{b} + \hat{\boldsymbol{x}}'_k. \tag{7}$$

where $\boldsymbol{Z}_k$ is a diagonal matrix with diagonal elements $[z_{1,k}, z_{2,k}, z_{3,k}]$. The method of introducing weights into the TDoA positioning itself has not been sufficiently studied, and the way to adjust the weights has not been explored. This is because it is difficult to analytically determine the optimal weight matrix and bias vector. Therefore, this paper proposes a novel positioning algorithm that adjusts the weight $\boldsymbol{Z}_k$ and the correction bias $\hat{\boldsymbol{x}}'_k$ in a data-driven manner using ML models.

III. Machine Learning-Aided Positioning

A. Weighted least squares

Fig. 3 shows a schematic diagram of the positioning algorithm using WLS. The weights $\boldsymbol{Z}_k = \mathrm{diag}[z_{1,k}, z_{2,k}, z_{3,k}]$ and the correction biases $\hat{\boldsymbol{x}}'_k = [x'_k, y'_k]^\mathsf{T}$ are training parameters. The positioning accuracy is improved by using the optimal vectors obtained by the ML-based search. ML involves a process that requires two phases: training and testing. During the training phase, the model is trained using training and validation data. During the testing phase, the model is evaluated using test data. Yielding the transmitter-receiver distance vector $\boldsymbol{d}_k$ into WLS Layer (WLSL), the estimated position vector $\hat{\boldsymbol{x}}_k$ is calculated by (7) with the assistance of weights $\boldsymbol{Z}_k$ and correction biases $\hat{\boldsymbol{x}}'_k$. To adjust $\boldsymbol{Z}_k$ and $\hat{\boldsymbol{x}}'_k$, supervised learning is performed during the training phase by minimizing

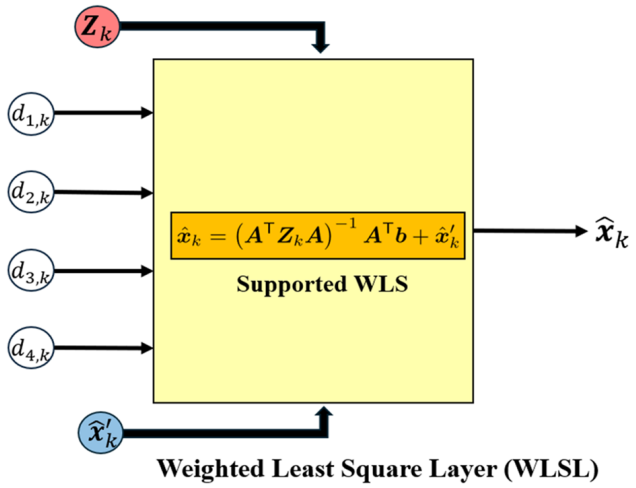


Fig. 3: Schematic of WLSL.

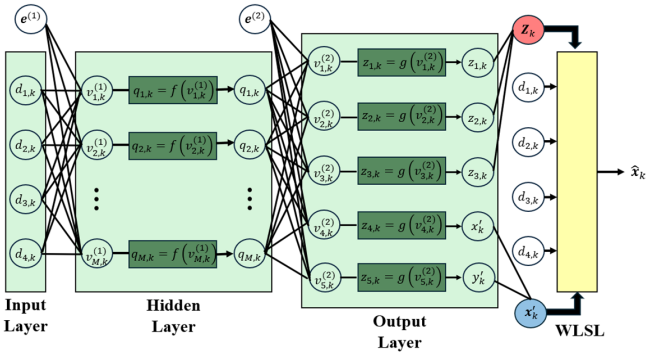


Fig. 4: Schematic of MLP-aided positioning algorithm.

the mean squared error (MSE) loss function, which is defined as

$$E = \frac{1}{K}\sum_{k=1}^{K}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|^2. \tag{8}$$

The gradients of $\boldsymbol{Z}_k$ and $\hat{\boldsymbol{x}}'_k$ are then calculated using the backpropagation method based on the error function. The weights $\boldsymbol{Z}_k$ and $\hat{\boldsymbol{x}}'_k$ are updated by an optimization algorithm using the obtained gradients. During the testing phase, the weights are fixed to the optimized values obtained during the training phase.

B. Multi-layer perceptron

Fig. 4 is a schematic diagram of the positioning algorithm using MLP. The considered MLP consists of an input layer, a hidden layer, and an output layer. Here, we define a weight matrix $\boldsymbol{W}^{(1)}$ of size $M \times 4$. The (j,m)-th element $w^{(1)}_{j,m}$ of $\boldsymbol{W}^{(1)}$ is the weight between the m-th node of the hidden layer and the j-th node of the input layer. Furthermore, $\boldsymbol{e}^{(1)} = [e^{(1)}_1, \ldots, e^{(1)}_m, \ldots, e^{(1)}_M]^\mathsf{T}$ is a bias vector of size $M \times 1$. When $\boldsymbol{d}_k$ is yielded into the input layer, the input

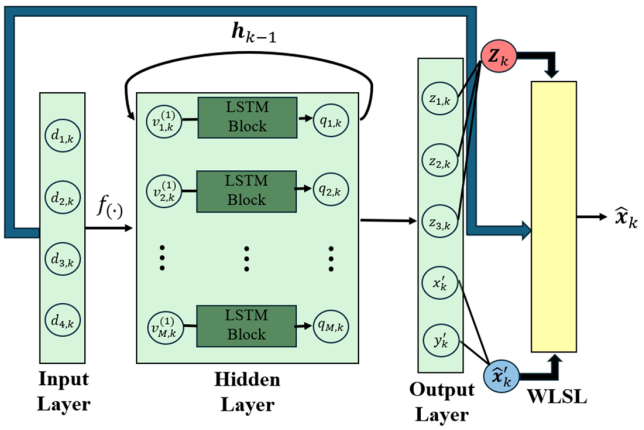


Fig. 5: Measurement system configuration.

$\boldsymbol{v}^{(1)}_k = [v^{(1)}_{1,k}, \ldots, v^{(1)}_{m,k}, \ldots, v^{(1)}_{M,k}]^\mathsf{T}$ to the hidden layer is expressed as

$$\boldsymbol{v}^{(1)}_k = \boldsymbol{W}^{(1)}\boldsymbol{d}_k + \boldsymbol{e}^{(1)}. \tag{9}$$

The resultant $\boldsymbol{v}^{(1)}_k$ is then input into the activation function as

$$\boldsymbol{q}_k = f\left(\boldsymbol{v}^{(1)}_k\right) = \max\left(0, \boldsymbol{v}^{(1)}_k\right), \tag{10}$$

where the activation function $f(\cdot)$ of the hidden layer is the ReLU function.

In MLP, linear processing with weights and biases and nonlinear processing with activation functions are performed alternately, as described above. The input to the output layer is $\boldsymbol{v}^{(2)}_k = \boldsymbol{W}^{(2)}\boldsymbol{q}_k + \boldsymbol{e}^{(2)}$, where $\boldsymbol{W}^{(2)}$ and $\boldsymbol{e}^{(2)}$ are the weight matrix and bias vector between the hidden and output layers, respectively. In this study, we use a sigmoid function scaled by a trainable parameter $\alpha \in [0, 2]$ as the activation function $g(\cdot)$ for the output layer.

$$\left[\mathrm{diag}[\boldsymbol{Z}_k]^\mathsf{T}, (\hat{\boldsymbol{x}}'_k)^\mathsf{T}\right]^\mathsf{T} = g\left(\boldsymbol{v}^{(2)}_k\right) = \frac{\alpha}{1 + \exp\left(-\boldsymbol{v}^{(2)}_k\right)}. \tag{11}$$

The obtained weights $\boldsymbol{Z}_k$, the correction biases $\hat{\boldsymbol{x}}'_k$ and the transmitter-receiver distance vector $\boldsymbol{d}_k$ are input into the WLSL, and the estimated position coordinates $\hat{\boldsymbol{x}}_k$ are obtained. The weights $\boldsymbol{W}^{(1)}$, $\boldsymbol{W}^{(2)}$, and the variables α, biases $\boldsymbol{e}^{(1)}$, $\boldsymbol{e}^{(2)}$ between each layer of the MLP are updated iteratively to minimize the error function. In the testing phase, positioning is performed by yielding $\boldsymbol{d}_k$ into the MLP, which has fixed weights, the variables and biases optimized in the training phase, and the resultant weights, biases, and the distance vector are delivered to WLSL. Finally, WLSL determines the position of TX $\hat{\boldsymbol{x}}_k$.

C. Long short-term memory

Fig. 5 shows a model of LSTM. In this study, LSTM, which can learn both long-term and short-term time dependence, was employed because it is capable of handling time-sequence data actually measured by moving-point positioning. The model consists of three layers: an input layer, a hidden layer (LSTM),

and an output layer. In this study, to stabilize the model training and enable the learning of complex features, we apply an activation function $f(\cdot)$ before the LSTM block. In time-series data of length K, in addition to the input at time step k, $\boldsymbol{d}_k$, the hidden layer output at the previous time step, $\boldsymbol{h}_{k-1}$ is retained as the state at time $h-1$ and fed back to the current state at time k. As a result, the hidden state $\boldsymbol{h}_{k-1}$ recursively reflects all previous states. In the hidden layer, time series information is maintained and propagated through a two-layer LSTM block, and this part of the system controls the internal state through input gates, forget gates, and output gates. After the output layer, the structure is the same as the MLP structure described in Sect. III.B.

IV. Experimental Results

A. Collection of dataset

The system configuration used to collect the dataset is shown in Fig. 6. It consists of UWB transmitter and receiver, a Raspberry Pi, a mobile battery, an autonomous robot, a fisheye lens camera, and a PC. The UWB module used in this study is the Swarm Bee ER manufactured by Nanotron, and its operating frequency band is 3.5 GHz to 6.5 GHz. The Raspberry Pi with UWB transmitter and the mobile battery is attached to an autonomous robot, which autonomously moves within the designated positioning area. To enable remote data acquisition, the Raspberry Pi is configured to be accessible and controllable from another PC. The autonomous robot is assembled using the LEGO Education SPIKE Prime Set. A custom program is installed on the robot to move randomly within the designated positioning area surrounded by black tape. When a color sensor on the robot detects the black tape on the floor, the robot changes its direction randomly inside the designated area.

Fig. 7 shows the experimental environment of the designated positioning area. The size of the room is 7.824 m $\times$ 6.216 m, the receiver height is 1.5 m, and the transmitter height is 0.78 m. The positioning area was defined as the lower half of the entire room. The measurement area was marked with black tape, and measurements were conducted by allowing a robot equipped with a transmitter to move randomly within this area. The dataset was collected over a 6-hour period, with four hours used for training data and the remaining two hours used for test data.

Based on the TDoA obtained from the UWB ranging signal, the distance $\boldsymbol{d}_k = [d_{1,k}, d_{2,k}, d_{3,k}, d_{4,k}]$ between the transmitter and the four receivers at the discrete time k is calculated. The resulting data $\boldsymbol{d}_k$ are saved on the Raspberry Pi and then transferred to the PC. In addition, the exact position of the robot, denoted by $\boldsymbol{x}_k = \left[x_k^{\mathrm{TX}}, y_k^{\mathrm{TX}}\right]^{\mathsf{T}}$ was calculated using video data recorded with a fisheye lens. The computation was performed on a PC connected to the fisheye lens, utilizing the Computer Vision Toolbox in MATLAB. This tool extracts one frame per second from the captured video and uses an object detector trained to recognize the robot's features in order to detect the position of the robot in each frame, which corresponds to the position of the transmitter. Furthermore, detection accuracy is improved by applying a Kalman filter to the detected positions. Through the above processes, $\boldsymbol{x}_k$ of the exact position coordinates of the transmitter are calculated to create the dataset.

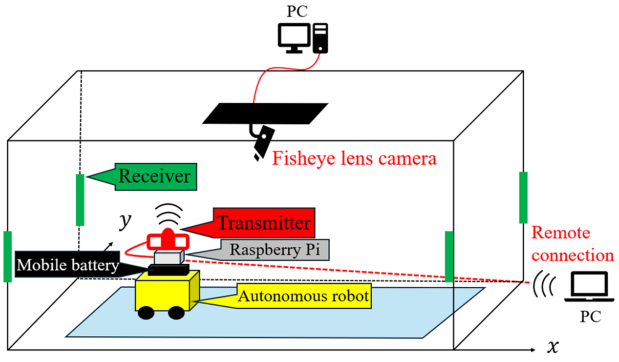


Fig. 6: Configuration of measurement system.

TABLE I: Measurement data [m].

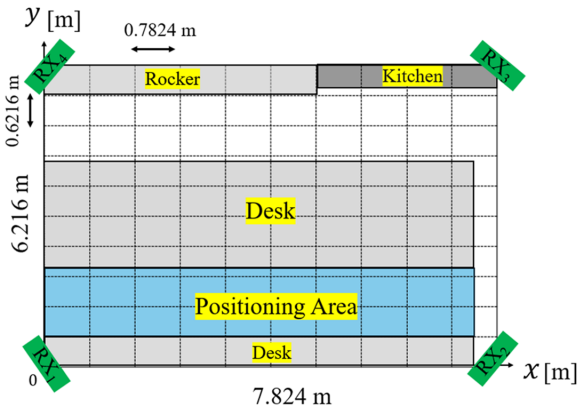
Time [s]	$d_{1,k}$	$d_{2,k}$	$d_{3,k}$	$d_{4,k}$	x_k^{TX}	y_k^{TX}
8	7.05	2.44	5.35	8.61	6.27	1.16
12	6.98	2.39	5.41	8.61	6.26	1.08
16	6.90	2.33	5.52	8.73	6.21	0.99
20	6.77	2.26	5.73	8.70	6.20	0.92
⋮	⋮	⋮	⋮	⋮	⋮	⋮

The distance $d_{j,k}$ between the transmitter and receivers was measured simultaneously while capturing the transmitter using a camera equipped with a fisheye lens. The exact position coordinates of the transmitter were calculated from the camera footage, and this data was transferred from the PC connected to the fisheye lens camera to a remote control PC where preprocessing was performed to create a dataset of accurate coordinates. On the other hand, a dataset was also created by preprocessing the distance data between the transmitter and receivers. These two datasets were time-synchronized using a common timestamp and then merged to create the final dataset shown in Table I

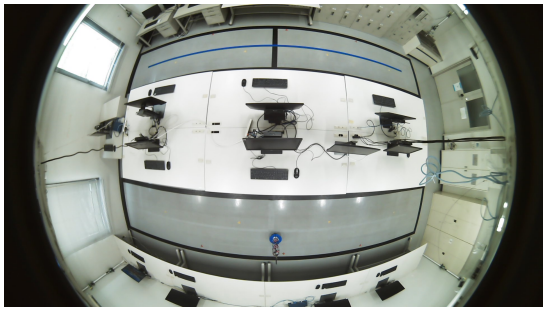
The above dataset was divided into training, validation, and test sets, resulting in 3,438 training samples, 644 validation samples, and 645 test samples. In this study, NN model was trained using the training data, and its positioning accuracy was evaluated using the test data.

B. Evaluation using test data

In this study, we compare the proposed method using MLP with the conventional LS algorithm in terms of the distance error (DE) between the estimated and true coordinates of the transmitter, which is defined by $\sqrt{\left(\hat{x}_k - x_k^{\mathrm{TX}}\right)^2 + \left(\hat{y}_k - y_k^{\mathrm{TX}}\right)^2}$. Fig. 8 shows a bar graph of the mean DE (MDE). The MDEs of LS, MLP, and LSTM are 0.416 m, 0.233 m, and 0.177 m, respectively. The results of MLP and LSTM indicate that the positioning accuracy



(a) Size of designated positioning area.



(b) Snapshot from fisheye camera.

Fig. 7: Experimental environment of positioning area.

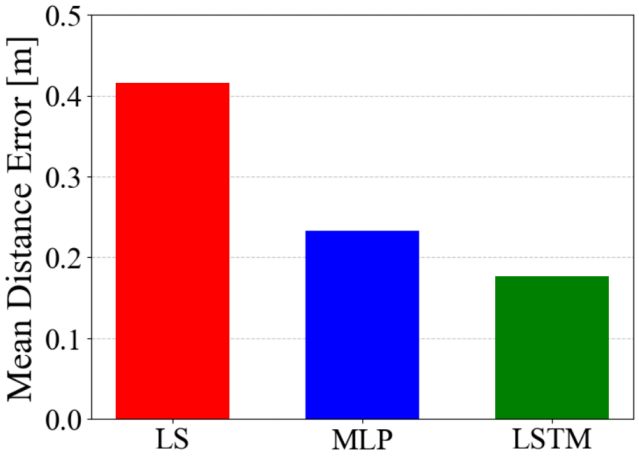


Fig. 8: MDE of estimated position coordinates.

is improved compared to the conventional LS method. More specifically, in terms of MDE, MLP and LSTM achieve 44% and 57.5% improvements over the LS method, respectively. From this, we can see that the MDE accuracy is highest for LSTM, followed by MLP, and lowest for LS. This study uses a dataset of moving point positioning, and it is expected that LSTM has higher accuracy than MLP because the dataset is a time series.

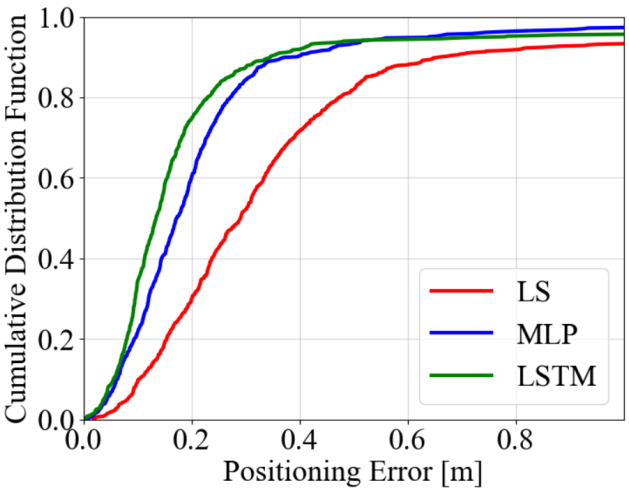


Fig. 9: CDF of MDE in estimated position coordinates.

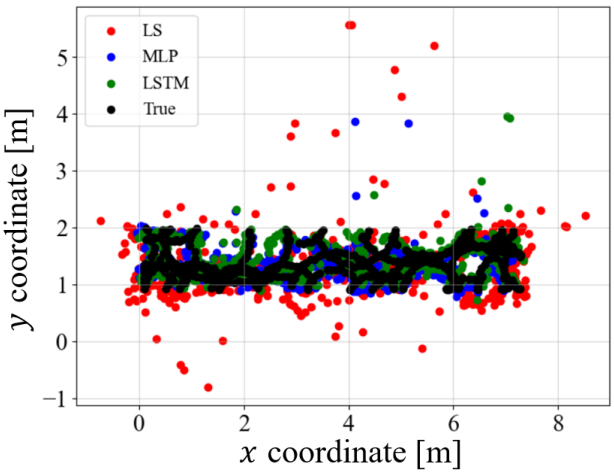


Fig. 10: Scatter plot of estimated position coordinates.

Fig. 9 shows the cumulative distribution function (CDF) of the DE for each method. This study evaluates the CDF 90% values of DE. For LS, 90% of the data has an error below 0.662 m, whereas the values for MLP and LSTM are 0.396 m and 0.346 m, respectively. These results indicate that all methods achieve centimeter-level positioning accuracy, as 90% of the data fell within an error margin of less than 1 m in the given experimental environment. This study focuses on the positioning of moving points. The probability that DE is less than 0.1 m, evaluates based on CDF, is 0.093 m for LS, 0.216 m for MLP, and 0.353 m for LSTM. These results indicate that LSTM, which can utilize temporal information, is particularly effective for moving point positioning where high-accuracy estimation is required.

Fig. 10 shows a scatter plot of DE for each method. From the scatter, it can be seen that the estimation results obtained by LS generally tend to underestimate the $\hat{y}_k$-coordinates

TABLE II: Statistical summary of errors for each method.

statistic	LS [m]	MLP [m]	LSTM [m]
first quartile	0.177	0.11	0.0888
median	0.290	0.173	0.131
third quartile	0.427	0.247	0.190
minimum	0.0184	0.00705	0.00111
maximum	5.04	3.77	2.35
dispersion	0.297	0.0825	0.0454

compared to the ground truth, with some estimates falling below zero, which lies outside the valid area range. On the other hand, in the method using the ML model, all estimated $\hat{y}_k$-coordinates are greater than or equal to 0, indicating that the significant errors observed in LS are effectively suppressed. Furthermore, a comparison of the number of data points with a measurement error of 1 m or more in the overall data shows that LS, MLP, and LSTM had 44, 18, and 10 data points, respectively. This confirms that LSTM can significantly reduce large measurement errors, with time-series models showing particularly strong performance in this study.

In order to understand the variability in the data, statistical measures of the measurement error for each method are compared in Table II.

From Table II, it can be seen that the median distance error decreases in the order of LS, MLP, and LSTM. Furthermore, LSTM has the smallest variance, indicating less variation in the errors. This suggests that LSTM not only has a better central tendency of error but also exhibits higher positioning stability. From the first and third quartilc in Table II, the interquartile ranges are 0.25 [m] for LS, 0.137 [m] for MLP, and 0.101 [m] for LSTM. This indicates that LSTM, which has the smallest interquartile range, exhibits less variation in error and demonstrates stable positioning performance. Furthermore, because the maximum error decreases in the order of LS, MLP, and LSTM, it can be inferred that LSTM demonstrates more stable accuracy.

V. Conclusions

In this study, we focused on the deterioration of positioning accuracy in conventional UWB-TDoA caused by the influence of fixtures. To address this accuracy degradation, we introduce a correction value into the LS method and propose a positioning algorithm that improves UWB-TDoA accuracy by adjusting this correction value in a data-driven manner using ML. Consequently, we employed MLP and LSTM as the ML models, achieving 44% and 57.5% improvements in positioning accuracy, respectively, compared to conventional methods, thereby significantly enhancing the overall positioning performance. We confirmed that all major statistical indicators—including the median, variance, and quartiles—are smallest for LSTM, followed by MLP, and then LS. These results demonstrate the effectiveness of integrating ML into the positioning algorithm. In addition, LSTM of the time-series data model was found to be the most effective means of moving point positioning.

Acknowledgment

A part of this work was financially supported by JSPS KAKENHI Grant Number JP23K20935, Japan.

References

[1] A. Koutris, T. Siozos, Y. Kopsinis, *et al.*, "Deep learning-based indoor localization using multi-view BLE signal," *Sensors*, vol. 22, no. 7, p. 2759, 2022.

[2] A. Prorok, P. Tomé, and A. Martinoli, "Accommodation of NLOS for Ultra-WideBand TDoA localization in single- and multi-robot systems," pp. 1–9, 2011.

[3] D. Dardari, P. Closas, and P. M. Djurić, "Indoor tracking: theory, methods, and technologies," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015.

[4] T. Ma, D. Zhang, J. Zhuang, C. You, G. Yin, and S. Tang, "TDoA-based UWB indoor 1D localization via weighted sliding window filtering," *Digital Signal Processing*, vol. 151, p. 104 544, 2024, ISSN: 1051-2004.

[5] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.

[6] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, 2007, pp. 46–51.

[7] S. Gezici, Z. Tian, G. Giannakis, *et al.*, "Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, 2005.

[8] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," 3, vol. 21, 2019, pp. 2568–2599.

[9] Z. Hajiakhondi-Meybodi, A. Mohammadi, M. Hou, and K. N. Plataniotis, "DQLEL: Deep Q-Learning for energy-optimized LOS/NLOS UWB node selection," *IEEE Transactions on Signal Processing*, vol. 70, 2022, pp. 2532–2547.

[10] B. Van Herbruggen, J. Fontaine, and E. D. Poorter, "Anchor pair selection for error correction in time difference of arrival (TDoA) ultra wideband (UWB) positioning systems," 2021, pp. 1–8.

[11] P. B. Duong, B. Van Herbruggen, A. Bröring, A. Shahid, and E. De Poorter, "Error mitigation for TDoA UWB indoor localization using unsupervised machine learning," *IEEE Sensors Journal*, vol. 25, no. 1, pp. 1959–1968, 2025.

[12] Y. Cheng and T. Zhou, "UWB indoor positioning algorithm based on TDoA technology," pp. 777–782, 2019.