2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) | 978-1-7281-7586-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/PIMRC50174.2021.9569615

Deep Learning for Ultra-Wideband Indoor Positioning

Yi-Min Lu, Jang-Ping Sheu, Yung-Ching Kuo
Department of Computer Science, National Tsing Hua University, Taiwan
Emails: s105065524@m105.nthu.edu.tw, sheujp@cs.nthu.edu.tw, cludbuster@gmail.com

***Abstract*—In recent years, the Ultra-wideband (UWB) system has been investigated for indoor localization and navigation by academia and industry. However, the UWB localization accuracy deteriorates when the signal propagates under severe non-line-of-sight (NLoS) conditions. We use two deep learning network models, the long short-term memory (LSTM) network and deep neural network (DNN), to analyze five different UWB signal features. The five features are received signal strength indication (RSSI), time of arrival (ToA), time difference of arrival (TDoA), first path (FP) amplitude from channel impulse response (CIR), and metric Mc (the ratio of the first path amplitude to peak amplitude). Then, we combine the five features into six different datasets for our deep learning models. Based on the prediction accuracy of the deep learning models for each combined feature, we propose a weighted indoor positioning (WIP) algorithm. The experiment results show that the WIP algorithm has better positioning accuracy than baseline works.**

***Index Terms*—Indoor positioning, ultra-wideband, fingerprint, deep learning, time difference of arrival.**

I. Introduction

Location-based service has been used for various scenarios, such as communication, sensing, robotics control [1], localization, behavior, and health analysis for elderly persons [2]. In recent years, UWB has become a promising communication technique. UWB is a technology for transmitting information across a wide bandwidth exceeding 500 MHz. It uses nanosecond (ns) to pico second (ps) level non-sine wave narrow pulses to transmit data and has a more precise positioning ability. Benefit from the high data rate transmission and nanosecond timestamp recording. The UWB indoor localization systems can make precise signal measurements, such as time of arrival (ToA) or time difference of arrival (TDoA), which can achieve the centimeter-level error of indoor positioning even in a critical multipath environment. In addition, due to the large bandwidth, their signals have low transmission power to avoid interference with other wireless signals in the same frequency spectrum. However, the UWB localization accuracy still deteriorates when the signal propagates under a severe non line-of-sight (NLoS) environment [3].

Existing indoor localization systems can be divided into two categories: fingerprint-based and range-based. The fingerprint-based localization uses a matching algorithm to estimate the target position by comparing the online and offline signal characteristics with Wi-Fi [4] or Bluetooth [5]. The signal characteristics include Received Signal Strength Indication (RSSI) [6] and Channel State Information (CSI) [7]. The range-based localization calculates the distances between the anchors and targets [8]. The distance can be estimated based on the time of arrival (ToA) [9] or the time difference of arrival (TDoA) [10].

However, the significant challenge in fingerprint-based localization is how to handle the unstable fluctuation of signals, and the difficulty of range-based localization is the uncertainty of signal propagation in NLoS and multipath environments. The TDoA/ToA measurement will result in significant localization error since the multipath effect deteriorates the accuracy of synchronization errors in sensor positions [11]. Recently, using deep learning to estimate positions becomes increasingly popular. For example, the recurrent neural network (RNN) with long short-term memory (LSTM), deep neural network (DNN), and convolutional neural network (CNN) can increase the robustness of the fingerprint-based and range-based localization methods [12]. Deep learning can extract relevant features embedded in the signals and represent initial data better than the original features. We can train neural networks and obtain critical features to enhance localization performance.

In [13], the authors use RSSI to calculate the position of the tag through a neural network, based on a multilayer perceptron, which is trained and tested with a radio map and learns to compute the position of the tags. In [14], the authors demonstrate a large-scale DNN architecture with a scalable stacked denoising auto-encoder for fingerprint-based indoor localization. In [15], the authors propose an improved localization algorithm called for source localization using LSTM to address a TDoA measurement error or missing data in an asynchronous localization. In [16], the authors propose a beam estimation for applying DNN that derives the angle of arrival by phase differences. The authors in [17] develop a UWB system with arbitrary target orientation and optimal anchor location. The UWB system implements a genetic algorithm (GA) to minimize the average positioning error. Besides, the adaptive NLoS mitigation with deep learning models is introduced to improve the accuracy of wireless positioning and tracking in a dense multipath environment. However, all the above methods extract shallow features that only represent trivial information and result in a rough estimation.

This paper combined five key signal features into six

This work was supported by Qualcomm Technologies, Inc. under Grant SOW NAT-435533.

different datasets for our deep learning models. The five features are RSSI, ToA, TDoA, first path (FP) amplitude from channel impulse response (CIR), and DW1000 metric Mc [18]. We collect the RSSI with the DW1000 UWB transceiver to estimate the receive strength level. The TDoA measurements are calculated by ASync-TDoA [19] model, which can measure the time difference between the target node and anchor nodes directly based on the timestamps. The ToA is modeled by a single side two-way ranging algorithm [20]. DW1000 transceiver measures the first path (FP) peak amplitude from the channel impulse response (CIR) on the DW1000 transceiver. And it uses a DW1000 metric, Mc, representing the ratio FP amplitude to peak amplitude, which can indicate whether the indoor environment is LoS or NLoS. We combine the five features into six different datasets: (i) RSSI, (ii) TDoA, (iii) ToA, (iv) RSSI and TDoA, (v) FP and TDoA, and (vi) Mc, RSSI and TDoA. We use the six datasets to train the LSTM and DNN networks. Based on the prediction accuracy of the six datasets, we propose a weighted indoor positioning (WIP) algorithm. Experiment results show that our deep learning models with the WIP algorithm have better positioning accuracy and robustness than baseline works.

The rest of this paper is organized as follows. Section II describes the background of RSSI, TDoA, ToA, DW1000 metrics, and two neural network models, DNN and LSTM. Section III presents the system architecture and WIP algorithm. The performance of our method is presented in Section IV. Finally, conclusions are drawn in Section V.

II. Background

This section introduces the DW1000 metrics and the schemes we use to estimate RSSI, TDoA, and ToA. Furthermore, we describe the detail of the deep learning models, DNN and LSTM.

A. *Channel Impulse Response* (CIR) *and DW1000 metrics*

Observing the channel impulse response (CIR) at each UWB antenna, we can determine the CIR amplitudes containing the multipath effect [18]. In the LoS situation, the signal propagation from the transmitter to the receiver only in its direct path also called the first path (FP). In the DW1000 accumulator, the FP amplitude is always equal to the peak amplitude and decreasing with time series. In the NLoS situation, the signal propagation from the transmitter to the receiver has an obstacle in its direct path, and the reflection happened. In the DW1000 accumulator, the FP amplitude is deteriorated by an obstacle, but the reflection path amplitude is not. Therefore, the peak amplitude corresponds to one of the reflection path amplitudes, and the peak amplitude is always higher than the FP amplitude.

We can measure the FP amplitude Amp_{fp} and the peak amplitude Amp_{pk} from the DW1000 accumulator. From [18], we can detect the occurrence of the saturation with a DW1000 metric Mc as:

$$Mc = \frac{Amp_{fp}}{Amp_{pk}} \tag{1}$$

If Mc is greater than 0.9, we can conclude that the saturation has occurred. Then it is likely that there is an LoS path between the transmitter and the receiver.

B. *Received Signal Strength Indicator (RSSI)*

The following formula can calculate an estimate of the received signal strength indicator (RSSI) [18]:

$$RSSI = 10 \times log_{10}\left(\frac{C \times 2^{17}}{N^2}\right) - A \text{ (in dBm)}, \tag{2}$$

where C is the CIR power value, N is the DW1000 preamble accumulation count = 128, and A is the constant 113.77. The CIR power value is a 16-bit value reporting the sum of the squares of the accumulator's magnitudes from the estimated highest power portion of the channel related to the receive signal power.

C. *Time Difference of Arrival* (TDoA)

We utilize the ASync-TDoA model to make the TDoA measurements [19]. The steps of ASync-TDoA are as follows. The reference anchor sends the signals to the anchors at regular intervals. It needs to record timestamps from the tag and the reference anchor for an anchor and then send them to the server. For each reference anchor, the server can directly measure the TDoA between the tag and the anchors.

D. *Time of Arrival* (ToA)

Unlike the ASync-TDoA model, the measurement of ToA is highly dependent on the Times-of-Flight between the transmitter and receiver. Although the anchors have accurate timing references, tags use coarse crystals still bring out severe clock offsets, causing a huge-ranging error. Hence, time synchronization is a critical issue for precise ToA measurements. The single side two-way ranging [20] algorithm can reduce the ranging error by using the relative clock offset.

E. *Deep neural network* (DNN)

DNN is a more in-depth version of the artificial neural network (ANN) that consists of an input layer, multiple hidden layers, and an output layer. There are multiple layers of neurons in a directed graph, each fully connected to the next one. Although the DNN is a primary deep learning method, we would like to use DNNs to estimate our target position, and compare the prediction results to LSTM, and utilize the prediction results to our proposed algorithm.

F. *Long short-term memory* (LSTM)

LSTM is a recurrent neural network (RNN) architecture. Unlike the traditional network, the output results depend not only on the current input value but also on the historical data. During the training phase, the LSTM model learns to leverage the latent motion features and trajectories from a sequence of locations and their corresponding datasets of signal features. The model can then predict the tags' current position with raw signal features during the online phase.

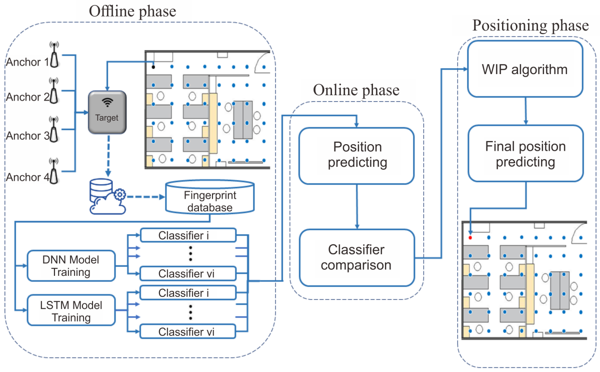


Fig. 1: The Architecture of localization system.

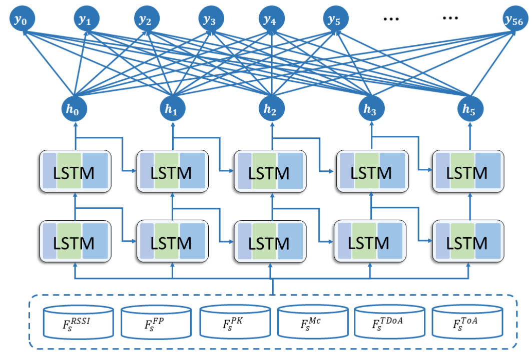


Fig. 2: The proposed LSTM model.

III. System Model

A. System overview

Fig. 1 shows the architecture of the proposed system. The indoor localization system is generally categorized into three phases: offline phase (training phase), online phase (predicting phase), and positioning phase. In the offline phase, we predefine the tags for different locations and collect many signal characteristics at each location, such as RSSI, TDoA, and ToA, and we stored them in a database. We then train six classifiers using the DNN (LSTM) network model corresponding to the six datasets (i) RSSI, (ii) TDoA, (iii) ToA, (iv) RSSI and TDoA, (v) FP and TDoA, (vi) Mc, RSSI, and TDoA. Thus, we have twelve classifiers, six for the DNN model, and six for the LSTM model. In the online phase, we predict the location with different classifiers for the random location of the tags from testing data. In the positioning phase, we can accurately predict the unknown location with the proposed weighted indoor positioning (WIP) algorithm to determine the final location of the tags.

B. Offline phase

In the offline phase, we construct the classifier and make a rough estimation of the tag's location at each data point, which will be utilized to determine the tag's final position in the positioning phase. The offline training includes data collection and the classifier trained by the DNN and the LSTM models. We use the methods mentioned in section II to measure the signals of RSSI, TDoA, ToA, FP, and Mc. Then, build our database via rescaling all the min-max normalization measures, which is the most common way to normalize data. Each feature is transformed into the interval [0, 1].

The database is separated into two partitions: the training set and the testing set. The former is used for model learning and to fit the parameters of a neural network. The latter is applied to validate the model error and evaluate the neural network model. We utilize a fingerprint approach for indoor positioning. Let N_{AN} be the number of the anchor nodes, N_T be the number of the tags, and $R^{s,i}$ be the RSSI on the s-th tag received signal from the i-th anchor. The FP on the s-th tag received signal propagation from the i-th anchor is $fp^{s,i}$. The $m^{s,i}$ is the Mc on the s-th tag received from the i-th anchor. The $T^{s,i}$ and $\tau^{s,i}$ are TDoA and ToA between the s-th tag and the i-th anchor, respectively. Each fingerprint Fs collected at the s-th tag has the signatures as shown in Table I.

TABLE I: The Offline Database of Fingerprints

Feature	Corresponding Fingerprint
RSSI	$F_s^{RSSI} = \{R^{s,1}, R^{s,2}, ..., R^{s,N_{AN}}\}$, for s = 1, ..., N_T
FP	$F_s^{FP} = \{fp^{s,1}, fp^{s,2}, ..., fp^{s,N_{AN}}\}$, for s = 1, ..., N_T
PK	$F_s^{PK} = \{pk^{s,1}, pk^{s,2}, ..., pk^{s,N_{AN}}\}$, for s = 1, ..., N_T
Mc	$F_s^{Mc} = \{m^{s,1}, m^{s,2}, ..., m^{s,N_{AN}}\}$, for s = 1, ..., N_T
TDoA	$F_s^{TDoA} = \{T^{s,1}, T^{s,2}, ..., T^{s,N_{AN}}\}$, for s = 1, ..., N_T
ToA	$F_s^{ToA} = \{\tau^{s,1}, \tau^{s,2}, ..., \tau^{s,N_{AN}}\}$, for s = 1, ..., N_T

After collecting the above signal features, the database is reorganized into six datasets: (i) F_s^{RSSI}; (ii) F_s^{TDoA}; (iii) F_s^{ToA}; (iv) F_s^{RSSI} and F_s^{TDoA}; (v) F_s^{FP} and F_s^{TDoA}; (vi) F_s^{Mc}, F_s^{RSSI}, and F_s^{TDoA}, and each dataset is used as classifiers' training input. For the LSTM and DNN models, we train each classifier corresponding to each dataset to observe the indoor multipath effect. The prediction results will apply to the positioning phase. We design our LSTM model, which captures the target location correlations at each moment by referring to the previous five locations. Then it can critically select the relevant previous information for predictions of the target location.

The proposed LSTM model is shown in Fig. 2. The LSTM network consists of one input layer, two hidden layers, and one output layer. The input layer's neurons depend on the number of the anchor nodes N_{AN} and the type of fingerprints. It consists of N_{AN} neurons for datasets (i), (ii), and (iii), $2N_{AN}$ neurons for datasets (iv) and (v), and $3N_{AN}$ neurons for the dataset (vi). Each hidden layer consists of five LSTM units. The output layer neurons depend on the number of the target node N_T. We can obtain the probability that every target node is predicted to the correct location through the $softmax$ function. Here, we treat mean square error (MSE) as the loss function, which is denoted as:

$$Loss_{MSE}(y, \hat{y}) = \frac{1}{N} \sum_{i}^{N} (y_i - \hat{y}_i)^2 \quad (3)$$

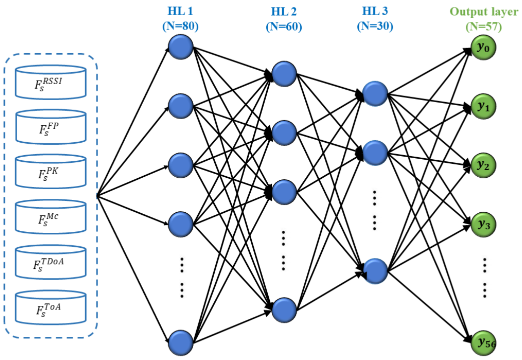


Fig. 3: The proposed DNN model.

The proposed DNN model consists of one input layer, three hidden layers, and one output layer, as shown in Fig. 3 The input layer and output layer are the same as the LSTM network. The hidden layers are composed of three fully connected layers 80, 60, and 30 neurons. To avoid the vanishing gradient problem of the sigmoid function, the ReLU function activates the neurons after the linear operation. Regarding the indoor localization as multiclass classification problems, we consider the cross-entropy function as a loss function, which is denoted as:

$$Loss_{CE}(y, \hat{y}) = -\frac{1}{N}\sum_{i}^{N}\left[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right] \tag{4}$$

Therefore, the location can be predicted through the twelve classifiers. Different classifiers with corresponding data can reveal different features in neural networks. By doing so, we can increase the accuracy of our localization system and achieve better learning.

C. Online phase

In the online phase, we predict the tag's location by feeding the testing data to all classifiers. Some tags are placed in LoS, and some of them are placed in NLoS. We predict the tags' location from the twelve classifiers in the LoS and NloS, respectively. Then, the accuracy of each classifier with LoS and NLoS will be used as the weights to determine the tags' final location. The set of weights are defined as $W_{LoS}=\{w_1, w_2, ..., w_{12}\}$ and $W_{NLoS}=\{\acute{w}_1, \acute{w}_2, ..., \acute{w}_{12}\}$ for LoS and NLoS environment, where w_1 to w_6 (or $\acute{w}_1$ to $\acute{w}_6$) are the weights calculated by the LSTM classifiers with the corresponding dataset (i) to (vi), and w_7 to w_{12} (or $\acute{w}_7$ to $\acute{w}_{12}$) are the weights calculated by the DNN classifier with the corresponding dataset (i) to (vi). The weights can be calculated by

$$w_i = \frac{Accuracy\ of\ classifier_i}{\sum_{i=1}^{N} Accuracy\ of\ classifier_i} \tag{5}$$

D. Positioning phase

After online testing, we find out some classifiers have good performance in particular situations. For instance, the classifier with the RSSI data has better accuracy than the one with the TDoA data in the NLoS indoor environment. We will show the experiment results in the next section. Therefore, we design a weighted indoor positioning algorithm called WIP. We separated the testing dataset into LoS dataset and NLoS dataset in the online phase, then calculated all classifiers' accuracy with the prediction results. Furthermore, we computed the weight W_{NLoS} and W_{LoS} by the accuracy of all classifiers in the NLoS and LoS environments. We use the Mc metric to distinguish whether a tag is in the NLoS environment during the positioning phase. If Mc is bigger than 0.9, we use the weight W_{LoS} to determine the final location of the tag, on the opposite, we use the weight W_{NLoS} to determine the final position of the tag. The pseudo-code of our algorithm is given in Algorithm 1. In severe multipath interference, using the WIP algorithm can avoid overfitting in the training phase and have good robustness in our system.

Algorithm 1: Weighted Indoor Positioning Algorithm

Input: Data signal features of a tag: RSSI $\boldsymbol{R}$, TDoA $\boldsymbol{T}$, ToA $\boldsymbol{\tau}$, FP $\boldsymbol{fp}$ and Mc $\boldsymbol{m}$
Output: The final location $\boldsymbol{L_E}$

In the online phase, we compute the weights $W_{LoS}=\{w_1,w_2,\dots,w_{12}\}$ and $W_{NLoS}=\{\acute{w}_1,\acute{w}_2,...,\acute{w}_{12}\}$ through the LSTM and DNN models.
Predict the tag's possible locations $L=\{l_1,l_2,\dots,l_{12}\}$ according to all classifiers.
Compute the final location $\boldsymbol{L_E}$
if $m > 0.9$ **then** /* LoS environment */
  $L_E = w_1 l_1 + w_2 l_2 + ... + w_{12} l_{12}$
else /* NLoS environment*/
  $L_E = \acute{w}_1 l_1 + \acute{w}_2 l_2 + ... + \acute{w}_{12} l_{12}$
end if
return L_E;

IV. Experiment Results

This section uses a desktop computer with Intel Core i7-6700K and GeForce GTX 750 Ti to train and evaluate the accuracy of our network models. The version of the TensorFlow backend is 1.9. In addition, we adopt the DecaWave DWM1001 modules as transceivers, which is compliant with the IEEE802.15.4 standard. Then, we use the proposed models with WIP algorithm for UWB indoor positioning to demonstrate our experiments. We do the experiments to evaluate our method's accuracy for localization in both the NLoS and LoS environments. To test the proposed method's performance, we consider a two-dimensional localization model in a 10×6 m^2 office room containing the NLoS and LoS environments. Finally, we predict the localization error in this environment and compare it with other models.

A. Experimental Environment

In the office room, there are one reference node, four anchor nodes, and 57 tags. Besides, a gateway and a server are deployed in the room as shown in Fig. 4. Every tag has an

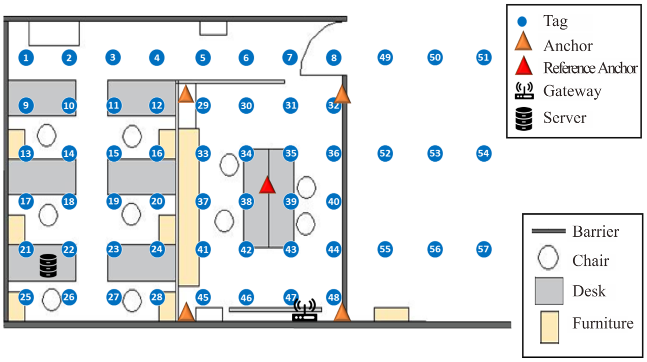


Fig. 4: The experimental environment.

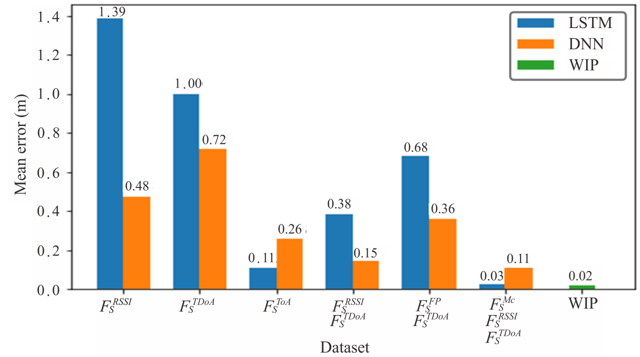


Fig. 5: The mean error of different classifiers' predictions.

identification number from 1 to 57, where no. 1 to no. 8 are placed on the floor of the corridor, no. 9 to no. 28 placed on the floor of the office room, no. 29 to no. 48 placed in the ceiling of the conference room, no. 49 to no. 57 placed in the ceiling of the outer corridor, i.e., no. 1 to no. 28 are in the NLoS environment, and no. 29 to no. 57 are in the LoS environment. The reference anchor transmits the UWB signal periodically to the anchor nodes, recording timestamps and sending them to the server through the gateway.

The experiment steps are as follows: (a) Collect and compute the signal characteristic data, like RSSI, TDoA, ToA, and CIR amplitude. We combine the five features into 12 datasets: (i) RSSI, (ii) TDoA, (iii) ToA, (iv) RSSI and TDoA, (v) FP and TDoA, and (vi) Mc, RSSI, and TDoA in the case of LoS and NloS, respectively. (b) The 12 datasets is used to train the LSTM and DNN networks independently to obtain the final 24 classifiers. (c) According to the classifiers' prediction results, we compute each classifier's positioning weight based on its accuracy. (d) Using the WIP algorithm to predict the final position of the tags. To obtain enough training data, we set 57 tags located one meter apart to get 140 samples for each tag. We collect 7981 samples with their actual coordinates. Then, we use 7183 samples for the training model and 798 samples for evaluation.

B. Prediction performance

To determine the classifiers' weights, we compute the classifiers' prediction accuracy for the tags' location. Table II shows each classifier's prediction accuracy with the corresponding datasets from 798 samples, and Fig. 5 shows the mean error of each classifier's position prediction. We analyze the accuracy of predictions to evaluate the performance of different classifiers. After feeding 130 data points, the performance of the LSTM model with F_s^{Mc}, F_s^{RSSI}, and F_s^{TDoA} datasets outperform the others in the NLoS environment, and the DNN model with Amp_{fp} and F_s^{TDoA} datasets outperforms the others in LoS environment. We find that DNN can extract more relevant NLoS features from the RSSI data as shown in Table II. The performance of the classifier with F_s^{RSSI} is at least 13% better than the classifier with F_s^{TDoA}. The RSSI data plays an important role in our experiments, and even it is unreliable by its unstable signal characteristic. From the whole area perspective, the LSTM model with F_s^{Mc}, F_s^{RSSI}, and F_s^{TDoA} datasets have the highest accuracy, but its performance in the LoS environment is less than the DNN model with F_s^{FP} and F_s^{TDoA} datasets. All the results in Table II can be used to compute the positioning weights W_{LoS} = {0.051, 0.081, 0.090, 0.081, 0.08, 0.090, 0.072, 0.09, 0.092, 0.092, 0.092, 0.089} and W_{NLoS} = {0.069, 0.056, 0.09, 0.088, 0.081, 0.1, 0.089, 0.064, 0.093, 0.09, 0.081, 0.1}.

In Fig. 5, we compare the mean error of different classifiers with their corresponding datasets. The LSTM model with F_s^{Mc}, F_s^{RSSI}, and F_s^{TDoA} datasets can achieve the lowest error of 3 cm in our experiments. Furthermore, the error can be reduced to 2 cm with the WIP algorithm in the final position estimation. However, all the datasets are collected under the fixed positions (i.e., no.1 to no.57) as shown in Fig. 4. In the following experiments, we will evaluate the accuracy in randomly selected positions.

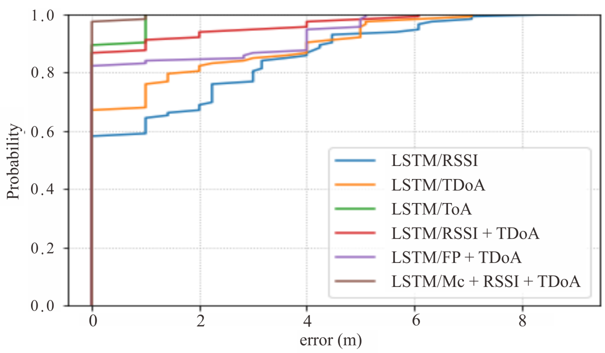
C. Robustness evaluation

Fig. 6 shows the cumulative distribution function (CDF) of positioning errors, evaluating the classifier's robustness. We find that the robustness of the LSTM model is always better than the DNN model. For instance, the range error of the LSTM model with F_s^{Mc}, F_s^{RSSI}, and F_s^{TDoA} datasets are within 0-1 m, but the DNN model results in 0-5 m. As it turned out, the results indicate that the proposed LSTM model can make good predictions by interpreting the localization from the current features and the information of previous tags. Thus, we can prove that the LSTM model makes a more stable prediction than the DNN model attributed to the internal hidden memory of the LSTM units.

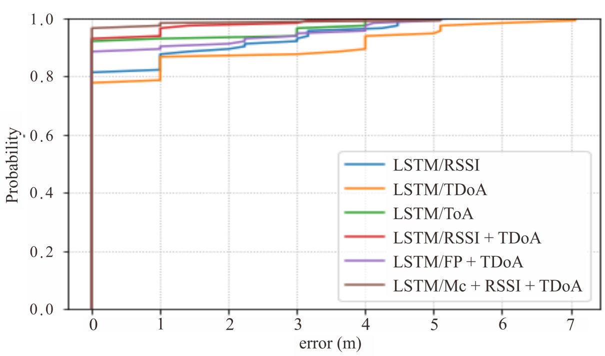
We compare our proposed method with the support vector machine (SVM) classifier, k-nearest neighbor (KNN) classifier, and multi-lateration (MLAT) classifier. The former two classifiers and proposed models use the same dataset, F_s^{Mc}, F_s^{RSSI}, and F_s^{TDoA}, while MLAT is a classical positioning mechanism for determining a position based on measurement of the times of arrival (ToAs) of energy waves, uses F_s^{ToA} to estimate the locations of the tags.

TABLE II: The Performance of Each Classifier with the Corresponding Dataset

Model	LSTM						DNN					
Dataset	F_s^{RSSI}	F_s^{TDoA}	F_s^{ToA}	F_s^{RSSI} F_s^{TDoA}	F_s^{FP} F_s^{TDoA}	F_s^{Mc} F_s^{RSSI} F_s^{TDoA}	F_s^{RSSI}	F_s^{TDoA}	F_s^{ToA}	F_s^{RSSI} F_s^{TDoA}	F_s^{FP} F_s^{TDoA}	F_s^{Mc} F_s^{RSSI} F_s^{TDoA}
NLoS Acc.	67.24%	54.38%	87.09%	85.48%	79.03%	96.77%	86.88%	62.29%	90.38%	87.09%	79.03%	96.77%
LoS Acc.	55.10%	88.23%	97.89%	88.23%	86.27%	98.03%	78.01%	98.03%	99.24%	99.47%	99.67%	96.07%
Overall Acc.	58.40%	67.25%	89.38%	86.72%	82.30%	97.34%	81.41%	77.87%	92.03%	92.92%	88.49%	96.46%



(a) The LSTM model



(b) The DNN model

Fig. 6: The CDF of positioning errors with (a) the LSTM model and (b) the DNN model.

Fig. 7: The trajectory of positioning between the proposed method and the others.

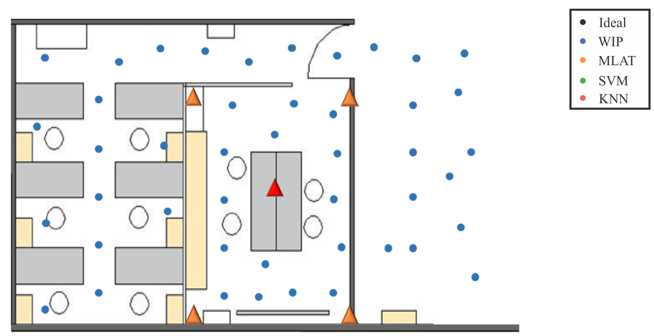


Fig. 8: The randomly select positions in experimental environment.

Fig. 7 shows the indoor tracking experiment by a human moving with 1 m/s. A human walks along the trajectory, which is shown in the dotted line in Fig. 7. We collect 82 samples at each meter of the walk, 31 positions in total. The tracking mean errors of the proposed models with the WIP algorithm, SVM, KNN, and MLAT are 7.6 cm, 76.5 cm, 72.5 cm, and 37 cm, respectively. We can observe that when the tags in the NLoS environment, the other models perform a poor accuracy. To compare the performance of our WIP algorithm with other baseline works, we collect 178 samples for each of 45 randomly selected positions, as shown in Fig. 8. Based on 8010 testing data, Fig. 9 shows the CDF of the mean errors between the WIP algorithm and other baselines. The mean errors of the proposed WIP algorithm, SVM, KNN, and MLAT, are 8.1 cm, 81.7 cm, 76.6 cm, and 38 cm. The results show that our WIP algorithm outperforms the other methods.

The experiments conclude that the proposed method can accurately predict the location of the tags. The proposed network models can effectively extract abstract location features by our network design to enhance localization performance. In addition, we use the classifiers' accuracy to implement our WIP algorithm can improve the prediction accuracy. These results demonstrate that the proposed method has better accuracy for indoor positioning in NLoS and LoS environments.

V. Conclusion

This paper combines five signal features into six different datasets for our deep learning models and proposed a WIP algorithm to predict the indoor positions in the NLoS and LoS environments. In the proposed algorithm, every dataset plays a

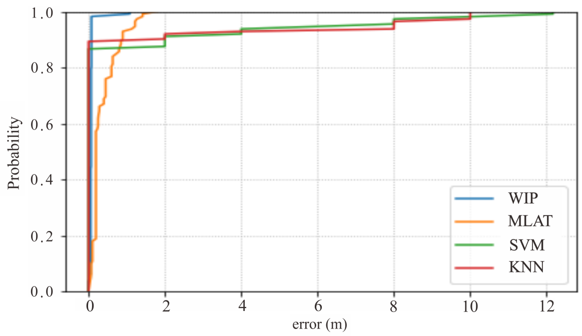


Fig. 9: The CDF of mean error for the proposed method and other methods in random positions.

significant role in our neural network models. Besides, we use DNN to express features more abstractly at a higher level and take advantage of LSTM units to enhance the accuracy of our method. With random positions, we achieve the positioning error within 8.1 cm in the experiments of indoor tracking. The results show that the proposed models with the WIP algorithm have better accuracy than baselines.

REFERENCES

[1] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[2] V. Djaja-Josko and J. Kolakowski, "UWB positioning system for elderly persons monitoring," in *2015 23rd Telecommunications Forum Telfor (TELFOR)*, 2015, pp. 169–172.

[3] W. Li, T. Zhang, and Q. Zhang, "Experimental researches on an UWB NLoS identification method based on machine learning," in *2013 15th IEEE International Conference on Communication Technology*, 2013, pp. 473–477.

[4] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, Jun. 2005, p. 205–218.

[5] J. Lovón-Melgarejo, M. Castillo-Cara, O. Huarcaya-Canal, L. Orozco-Barbosa, and I. García-Varea, "Comparative study of supervised learning and metaheuristic algorithms for the development of Bluetooth-based indoor localization mechanisms," *IEEE Access*, vol. 7, pp. 26 123–26 135, Feb. 2019.

[6] W. Xue, Q. Li, X. Hua, K. Yu, W. Qiu, and B. Zhou, "A new algorithm for indoor RSSI radio map reconstruction," *IEEE Access*, vol. 6, pp. 76 118–76 125, 2018.

[7] X. Wang, L. Gao, and S. Mao, "CSI phase fingerprinting for indoor localization with a deep learning approach," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1113–1123, Dec. 2016.

[8] Z. Yin, X. Jiang, Z. Yang, N. Zhao, and Y. Chen, "WUB-IP: A high-precision UWB positioning scheme for indoor multiuser applications," *IEEE Systems Journal*, vol. 13, no. 1, pp. 279–288, Mar. 2019.

[9] I. Sharp and K. Yu, "Indoor ToA error measurement, modeling, and analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 9, pp. 2129–2144, Sept. 2014.

[10] P. Du, S. Zhang, C. Chen, A. Alphones, and W.-D. Zhong, "Demonstration of a low-complexity indoor visible light positioning system using an enhanced TDoA scheme," *IEEE Photonics Journal*, vol. 10, no. 4, pp. 1–10, Aug. 2018.

[11] S. Leugner, M. Pelka, and H. Hellbrück, "Comparison of wired and wireless synchronization with clock drift compensation suited for U-TDoA localization," in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, 2016, pp. 1–4.

[12] X. Wang, X. Wang, and S. Mao, "ResLoc: Deep residual sharing learning for indoor localization with CSI tensors," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct. 2017, pp. 1–6.

[13] A. S. Martinez Sala, R. Guzman Quiros, and E. Egea Lopez, "Using neural networks and active RFID for indoor location services," in *European Workshop on Smart Objects: Systems, Technologies and Applications*, June 2010, pp. 1–9.

[14] C. Liu, C. Wang, and J. Luo, "Large-scale deep learning framework on FPGA for fingerprint-based indoor localization," *IEEE Access*, vol. 8, pp. 65 609–65 617, Apr. 2020.

[15] Y. Xue, W. Su, H. Wang, D. Yang, and Y. Jiang, "DeepTAL: Deep learning for TDoA-based asynchronous localization security with measurement error and missing data," *IEEE Access*, vol. 7, pp. 122 492–122 502, Aug. 2019.

[16] A. Niitsoo, T. Edelhäußer, and C. Mutschler, "Convolutional neural networks for position estimation in TDoA-based locating systems," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–8.

[17] Y.-Y. Chen, S.-P. Huang, T.-W. Wu, W.-T. Tsai, C.-Y. Liou, and S.-G. Mao, "UWB system for indoor positioning and tracking with arbitrary target orientation, optimal anchor location, and adaptive NLoS mitigation," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9304–9314, Feb. 2020.

[18] DW1000 metrics for estimation of non line of sight operating conditions. [Online]. Available: https://www.decawave.com

[19] Y. Xue, W. Su, H. Wang, D. Yang, and J. Ma, "A model on indoor localization system based on the time difference without synchronization," *IEEE Access*, vol. 6, pp. 34 179–34 189, 2018.

[20] I. Dotlic, A. Connell, and M. McLaughlin, "Ranging methods utilizing carrier frequency offset estimation," in *2018 15th Workshop on Positioning, Navigation and Communications (WPNC)*, Oct. 2018, pp. 1–6.