UWB NLOS/LOS Classification Using Deep Learning Method

Changhui Jiang, Jichun Shen, Shuai Chen, Yuwei Chen, Di Liu, and Yuming Bo

***Abstract*— Ultra-Wide-Band (UWB) was recognized as its great potential in constructing accurate indoor position system (IPS). However, indoor environments were full of complex objects, the signals might be reflected by the obstacles. Compared with the Line-Of-Sight (LOS) signal, the signal transmitting path delay contained in None-Line-Of-Sight (NLOS) signal would induce positive distance errors and position errors. Before employing ranging information from the channels to calculate the position, LOS/NLOS classification or identification was necessary for selecting the "clean" channels. In conventional method, features extracted from the UWB channel impulse response (CIR) or some other signal properties were employed as the input vector of the machine learning methods, e.g. Support Vector Machine (SVM), Multi-layer Perception (MLP). Deep learning methods represented by Convolutional neural network (CNN) and Long Short-Term Memory (LSTM) had performed superior performance in dealing with time series data classification. In this pap er, deep learning method CNN-LSTM was employed in the UWB NLOS/LOS signal classification. UWB CIR data was directly input to the CNN-LSTM. CNN was employed for exploring and extracting the features automatically, and then, the CNN outputs were fed into the LSTM for classification. Open source datasets collected from seven different sites were employed in the experiments. Classification accuracy of CNN-LSTM with different settings was compared for analyzing the performance. The results showed that CNN-LSTM obtained stat e-of-art classification performance.**

***Index Terms*— UWB, NLOS, CNN, LSTM.**

I. Introduction

WITH the rapid development of the Internet-Of-Thing (IoT) technology, Location Based Service (LBS) has gained a boom [1]. In outdoors, Global Navigation Satellite System (GNSS) can provide reliable navigation solutions while sufficient satellites are visible. However, GNSS signal is too weak to pass through wall, and it attenuates heavily indoors. There are still some challenges for building accurate Indoor Positioning System (IPS). Ultra-Wide-Band (UWB) communication technologies have gained wide attention in the community to construct IPS with decimeter accuracy [2]. In UWB, the signal spectrum is consisted of multiple sub-bands, which intends to improve the data rate of wireless communications and the ability to propagate through the wall [3]. In addition, the baseband pulse of the UWB has high time resolution (order of nanoseconds), which can contribute to accurate distance measurement.

However, indoor environments are usually full of various objects, e.g. walls, desks, chairs and some other objects, direct UWB signals might be blocked by them and the reflected signal is received. Compared with the Line-Of-Sight (LOS) signal, the signal reception will induce additional ranging bias in the distance measurements. Without None-Line-Of-Sight (NLOS) signals detection, ultimate location accuracy will be decreased [4], [5]. In the UWB based IPS, NLOS detection is important for selecting "clean" distance information for location estimation, the NLOS contaminated distance should be excluded or corrected before being employed for location information determination.

NLOS detection is the premise of the correction, and there are many approaches presented in the past papers to deal with this problem. These methods can be categorized into three different types:

(1) The first method is built based on the statistical parameters differences of the estimated distance information under LOS/NLOS conditions [6]; distance noise from the LOS condition is subject to Gaussian distribution with zero mean values, and while the NLOS occurs, the noise could be modeled as Gaussian distribution with non-zero mean values, the NLOS induced bias is the mean value. Hypothesis testing can be carried out based on the distance variance and mean values differences [7]. However, it is hard to determine the detection threshold, which might vary at different sites and environments.

(2) The second one is based on the signal propagation path loss model or the study of the channel impulse response (CIR). The main idea behind these solutions is that the energy of the first path is noticeably greater than the energy of the delayed paths. In addition, some non-parameter machine learning methods are utilized to NLOS/LOS classification, e.g. Support Vector Machine (SVM), Multi-layer Perception (MLP), Decision Tree and some other machine learning methods [8]–[11]. Kurtosis, Peak to Lead delay, Mean Excess delay and RMS delay spread and some other features are also employed in these classifiers as features [8], [9]. Effect of the NLOS propagation was quantified with Monto Carlo simulations, and features representative of the LOS/NLOS conditions were extracted for classification [10], [11]. A SVM regressor was employed for mitigating the NLOS induced errors [10], [11]. However, the signal propagation path loss model is affect by lots of factors, and the manually selected features might be not enough for the classification with these classifiers.

(3) Different from the previous two methods detecting NLOS based on the signal characteristics, the third method is built based on the context awareness of the surrendering

Manuscript received May 20, 2020; accepted June 1, 2020. Date of publication June 4, 2020; date of current version October 9, 2020. The authors acknowledge the support of National Natural Science Foundation of China (Grant No. 61601225). The associate editor coordinating the review of this article and approving it for publication was S. Bartoletti. *(Corresponding authors: Shuai Chen; Yuwei Chen.)*

Changhui Jiang, Shuai Chen, and Yuming Bo are with the School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: chagnhui.jiang1992@gmail.com; chenshuai@njust.edu.cn; byming@njust.edu.cn).

Jichun Shen is with Hesai Technology, Building L2-B, Hongqiao World Centre, Shanghai 201702, China (e-mail: s365445689@hotmail.com).

Yuwei Chen is with the Department of Photogrammetry and Remote Sensing, Finnish Geospatial Research Institute, FI-02430 Masala, Finland (e-mail: yuwei.chen@nls.fi).

Di Liu is with the School of Automation, Nanjing Institute of Technology, Nanjing 210094, China (e-mail: liudinust@163.com).

Digital Object Identifier 10.1109/LCOMM.2020.2999904

environment. In this method, the channel status is identified by observing the previous position of the mobile user or environment data (e.g., geometries and attenuation factors), the signal ray tracing can be carried out with aiding from the 2D or 3D maps, the NLOS signals can be identified and the NLOS induced errors can be estimated [5]. However, 3D map and ray-tracing method need large computation and also a prior position is necessary.

Inspired by the superior performance of the deep learning method in the data classification [12], in this letter, we presented a novel UWB NLOS detection and classification method based on the Convolution Neural Network (CNN) and Long-Short Term Memory Recurrent Neural Networks (LSTM-RNN). CNN was employed to extract the non-temporal features from the raw CIR signals, and the CNN outputs were then feed into the LSTM-RNN for classifying LOS/NLOS signals.

Reminder of the letter is organized as: Section II describes the NLOS problem and the CIR model; Section III presents the details of the proposed CNN and LSTM; and then, the dataset collecting and the experimental results, comparison and the analysis are given in detail, some discussions and future works are added.

II. LOS/NLOS Problem Statement

Assuming the UWB transmitting with white Gaussian noise, the UWB signal transmitting model can be listed as [4]:

$$s(t) = \sum_{i=-\infty}^{\infty} \sum_{j=0}^{N_c-1} \alpha_j q(t - iT_s - jT_c) \tag{1}$$

where, $q(t)$ is the single Gaussian pulse with repetition time T_c, a symbol duration T_s consists of N_c pulses, $\alpha_j \in \{-1, +1\}$ is the polarization sequence for the spectrum shaping.

The UWB CIR of the transmitted signals can be given as:

$$h(t) = \sum_{l=1}^{L} \beta_l \delta(t - \tau_l) \tag{2}$$

where, β_l is the fading coefficient and τ_l is the time delay of the l_{th} path. The received signals is the summation of the multiple attenuated and delayed replicas of $s(t)$.

$$r(t) = \sum_{l=1}^{L} \beta_l s(t - \tau_l) + v(t) \tag{3}$$

where, $v(t)$ is the additive white Gaussian noise.

In UWB based IPS, distance information from different channels is obtained to calculate the location results. Since the Time of Arrival (TOA) method has the advantages of accuracy, it is employed in UWB based IPS for distance calculation [12]. While utilizing the TOA method, the distance information is extracted through the CIR measurements. CIRs of the LOS and NLOS are presented in Figure 1. It can be seen the magnitude of the LOS is much larger than that of NLOS, and the curves are different. The CIR can be regarded as time series, and the NLOS and LOS CIRs are different due to their different transmitting paths. Especially, the NLOS reception affects the CIR curves heavily. Deep learning methods, e.g. CNN, LSTM, can be employed to deal with NLOS/LOS classification directly utilizing the CIR as the input vector.

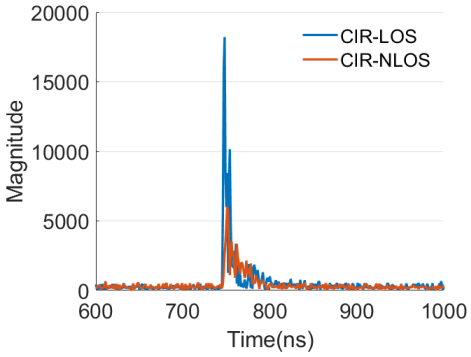

Fig. 1. Channel impulse response (LOS/NLOS).

Fig. 2. Overview of the CNN-LSTM Fig. 3. CNN structure.

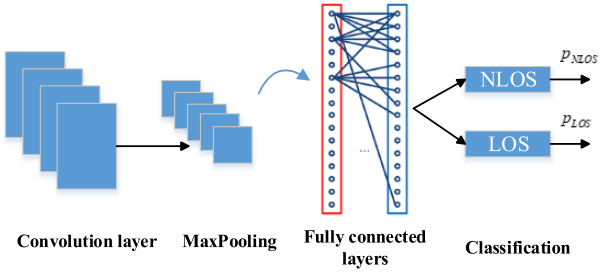

Fig. 3. CNN structure.

III. Proposed CNN-LSTM

Brief overview of the CNN-LSTM is presented in Figure 2; two basic blocks CNN and LSTM are included in the framework. In the CNN-LSTM, the CIR is employed as the input vector for the CNN. CNN exploits the non-temporal structure from the input signals by learning the frequency characteristics of the CIR. The CNN outputs are feed into the LSTM for learning long-range dependencies of the time series data and classifying the NLOS and LOS signals.

A. Convolutional Neural Network (CNN)

There are usually multiple hidden layers between the input and output layer in CNN. Among the hidden layers, there are usually convolutional layers, pooling layers and fully connected layers. A CNN structure is presented in Figure 3.

A convolutional layer is designed to extract features. Multiple kernel filters are employed in the convolutional layer to extract the features and characteristics from the input data. Assuming the l_{th} conventional layer has N_F^l kernel filters, and the filters are employed to convolve the input data. Each kernel filter uses the same kernel function to extract the features, and the operation is modelled as [12]:

$$y_i^{(l)} = a\left(\sum_{r=1}^{N_k^{(l)}} w_r^{(l)} x_{r+i\times N_S^l} + b^{(l)}\right) \tag{4}$$

$$0 \le i \le \frac{N - N_K^{(l)}}{N_S^{(l)}}, \quad l = 1, 2, \ldots, L. \tag{5}$$

where, the $N_K^{(l)}$ is the filters' kernel size, the $w_r^{(l)}$ and $b^{(l)}$ are the weight and bias of the kernel.

After the convolution operation using the kernel filters, the outputs are fed into an activation function. Here, the Rectified Linear Units (ReLU) activation function is selected and the equation is given by:

$$f_{ReLU}(x) = \begin{cases} x & if\ x > 0 \\ 0 & otherwise' \end{cases} \tag{6}$$

where, x is the argument of the ReLU function.

Pooling layer is usually added after the convolution layer for spatial reduction through down-sampling the outputs from the convolutional layer. Pooling layer reduces the computation load and time complexity through reducing the tensors' dimension of the outputs from previous convolutional layer. Here, MaxPooling function is employed to select the maximum values from current pooling window.

Convolutional and MaxPooling layers work together to extract the features, and then a fully-connected layer is responsible for selecting the classes' probability using SoftMax function. The class with highest probability is selected as the output of the classifier. In the fully-connected layer, the neurons are all connected to that in previous layer.

The CNN structures, layer numbers and the parameters are explored and adjusted by experiments in Section IV. Training and parameters optimization are important for the CNN, and the overfitting problem should be considered for improving the generality and prediction accuracy. Two methods are included for overcoming this problem.

(1) A dropout rate is introduced after the fully-connected layer, some subsets of the neurons are ignored and the nodes are dropped at the training stage. The dropout regularization rate values of different models are mentioned in Section IV. Also, the dropout is helpful to prevent building site specific model.

(2)Abundant training datasets collected from different sites are added to the training stage, which guarantees the neural networks are fully trained without redundancy.

B. Long Short Term Memory (LSTM)

Different from the conventional RNN, LSTM-RNN is composed of three "gates" with different functions. The basic structure of the LSTM unit is presented in the Figure 4. Specifically, the "gates" are "input gate", "forget gate" and the "output gate" [12], [13].

As shown in Figure 4, the first "gate" is the "forget gate" from the left to right. The function of the "forget gate" is to decide the "updating" degree and regulate the cell state from previous epoch. The input vector is fed into a sigmoid function, and then the output vector $\boldsymbol{f}_t$ is multiplied with cell state vector from the last epoch ($\boldsymbol{C}_{t-1}$). The output vector $\boldsymbol{f}_t$ equation is written as:

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f\right) \tag{7}$$

where, $\sigma(\cdot)$ is the sigmoid function, $\boldsymbol{W}_f$ is the updating weights vector, $\boldsymbol{b}_f$ is the bias vector. The vector $\boldsymbol{f}_t$ contains the values between 0 and 1, which decides the keeping degree of the $\boldsymbol{C}_{t-1}$ through a multiplication operation.

After the "forget gate" is the "input gate", which regulates the input data $\boldsymbol{x}_t$ and the processed state vector from "forget gate". The"input gate" function is described as:

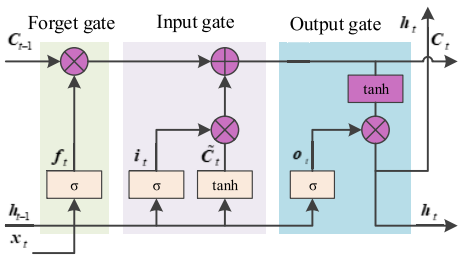

Fig. 4. Basic Structure of the LSTM unit.

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i\right) \tag{8}$$

$$\tilde{\boldsymbol{C}}_t = \tanh\left(\boldsymbol{W}_C \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_C\right) \tag{9}$$

where, $\boldsymbol{W}_i, \boldsymbol{W}_C$, $\boldsymbol{b}_i$ and $\boldsymbol{b}_C$ are parameters, and they are determined through the training. $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ are the weights and the bias for calculating the vector $\boldsymbol{i}_t$, $\boldsymbol{W}_C$ and $\boldsymbol{b}_C$ are the weights and bias for calculating the candidate cell state $\tilde{\boldsymbol{C}}_t$ with the *tanh* function, then, the candidate cell state vector $\tilde{\boldsymbol{C}}_t$ is multiplied with the vector $\boldsymbol{i}_t$, and the results are fed into updating the cell state at current epoch.

The last one is the "output gate", which decides the outputs of the LSTM. There are two different functions (*sigmoid* and *tanh*) employed in this gate. The Equations are written as:

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o\right) \tag{10}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \cdot tanh\left(\boldsymbol{C}_t\right) \tag{11}$$

In addition, the cell state vector is as:

$$\boldsymbol{C}_t = \boldsymbol{f}_t \cdot \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \cdot \tilde{\boldsymbol{C}}_t \tag{12}$$

IV. Dataset and Experimental Results

A. Experimental Setting Up and Dataset Description

For convenient assessing the proposed method, open source datasets were employed in the experiments. NLOS and LOS measurements were collected from seven different indoor locations: Office1, Office2, Small apartment, Small workshop, Kitchen with a living room, Bedroom and Boiler room [9]. At each location, 3000 NLOS and 3000 LOS measurements were collected. Collecting the NLOS/LOS samples from different sites aimed to avoid the occurrence of the location-specific model. While building the classification model using the dataset, 35000 samples (5000 samples from each site) were randomly selected from the dataset. The selected samples were random shuffled, 25000 samples were randomly employed as the training dataset and the rest 10000 samples were treated as the testing dataset. Randomly selecting these samples also helped to prevent the possible overfitting of the model in particular site.

B. Classification Accuracy of CNN With Different Number of Layers

For selecting the proper number of the convolutional layers in CNN, the CNN-LSTM models were carried out with different numbers of the convolutional layers. The batch size was set to 64, the neurons of the fully-connected layer was set to 128. In the convolution layer, we selected "valid" padding function. A dropout regularization was added after the feature

TABLE I
ACCURACY COMPARISON FOR DIFFERENT NUMBER OF LAYERS

Number of the convolutional layers	Accuracy (%)	Feature Extraction Module Architecture
1	57.3%	Cov1D(10,4) MaxPooling(2,2)
2	81.56%	Cov1D(10,4) Cov1D(20,5) MaxPooling(2,2)
3	81.14%	Cov1D(10,4) Cov1D(20,4) MaxPooling(2,2) Cov1D(20,4)
4	75.48%	Cov1D(10,4) Cov1D(20,4) MaxPooling(2,2) Cov1D(20,4) Cov1D(40,4)

extraction module and the dropout regularization rate was set to 0.5. These CNN-LSTM models were trained and tested for searching the best number of the layers. The input vector dataset was the CIR data with the size (1×1016). These CNNs were trained with 10 epochs, the training and testing dataset was described in Section IV. A.

The classification results of the CNNs with different amount of convolutional layers were listed in Table I. For the filters, we mean Conv(p, q) a convolution layer with p filters and q kernel size, and we mean the MaxPooling(a, b) with a pooling size and b strides. Among these CNN structures, the CNN with two convolutional layers was superior to the others. While more convolutional layers were included in the model, the classification accuracy decreased on the contrary. We thought two reasons might account for this phenomenon. The first reason is that CNN with more than two convolutional layers might cause overfitting here; the second reason is that deeper CNN structure might lead to the learning efficiency decreasing; residual CNN might be helpful for improving the classification accuracy. Therefore, CNN with two convolutional layers were deployed here for the NLOS/LOS classification.

C. Classification Accuracy Comparison for Different Methods

After selecting approximate CNN, this Sub-section aimed to search for suitable LSTM architectures. Bi-direction LSTM and stacked LSTM were deployed to classify the NLOS/LOS. For comparing these methods fairly, they were incorporated with the same CNN structure described in Section IV.C. Dataset for training, testing and validating were the same as that described in Section IV. B. The classification accuracy comparison results were listed in the Table II. LSTM performed the lowest classification accuracy. However, with the CNN included, the CNN-LSTM obtained significant improvement compared with single CNN and LSTM. Without the CNN, the redundant information contained in the input data might influence the LSTM training and its performance.

The LSTM hidden layers was set with 32, and the learning rate was 0.001. As the classification results listed in Table II, the CNN-stacked-LSTM obtained minor improvement of the classification accuracy. However, the CNN-bidirectional-LSTM performed a slightly worse in the classification accuracy, which might be initialed by the fact that the bidirectional LSTM had more parameters needed to be determined during the training phase.

TABLE II
ACCURACY COMPARISON FOR DIFFERENT NUMBER OF LAYERS

Method	Accuracy (%)
LSTM	51.11%
CNN+LSTM	81.56%
CNN+bidirectional-LSTM	78.93%
CNN+stacked-LSTM	82.14%

D. Limitations and Future Work

Although above methods obtained satisfying performance in the UWB NLOS/LOS classification, there were still following limitations.

(1) Deep learning methods were all implemented in Python with Keras and Tensorflow library, and the Alienware PC with I7 CPU (3.3GHz) and 16GB RAM were employed to run these programs. Each method was trained with 10 epochs, and the procedure always cost over one hour. Methods should be considered for reducing the computation load and accelerate the training.

(2) The NLOS/LOS was just clustered in this letter, and influence of the NLOS/LOS classification accuracy on the location estimation was not investigated.

In future, we thought following works might promote the development of this method.

(1) In the CNN-LSTM, Dropout was employed for preventing the overfitting. Another method Batch Normalization (BN) was able to actuate the training procedure; it was of significance to explore BN in this application.

(2) Training dataset was possibly insufficient for the classifier training and modeling, NLOS/LOS dataset collecting was also labor intensive, other machine learning methods, e.g. few-shot learning, which could provide satisfying performance with limited training dataset. It was of great significance for exploring these methods in this application.

(3) We did not discuss the diffraction effects in UWB limited by the dataset and hardware. It was of great significance to explore the diffraction effects in UWB based IPS.

V. CONCLUSION

In this letter, CNN-LSTM was investigated in the UWB NLOS/LOS classification. Through the experimental results, we could conclude that CNN-LSTM was effective for the UWB NLOS/LOS classification, and CNN effectively reduced the redundant information. With the LSTM aiding, the classification accuracy obtained further increase compared with CNN. The CNN-LSTM method could be extended to other wireless signals based indoor system, e.g. WLAN, Bluetooth.

ACKNOWLEDGMENT

The authors would like to thank Doctor Klemen Bregar providing the UWB NLOS/LOS open source data, and the UWB dataset downloading link is as: https://github.com/ewine-project/ UWB-LOS-NLOS-Data-Set.

REFERENCES

[1] W. Wang *et al.*, "Robust weighted least squares method for TOA-based localization under mixed LOS/NLOS conditions," *IEEE Commun. Lett.*, vol. 21, no. 10, pp. 2226–2229, Oct. 2017.

[2] S. Mazuelas *et al.*, "Soft range information for network localization," *IEEE Trans. Signal Process.*, vol. 66, no. 12, pp. 3155–3168, Jun. 2018.
[3] D. Minoli and B. Occhiogrosso, "Ultrawideband (UWB) technology for smart cities IoT applications," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Sep. 2018, pp. 1–8.
[4] X. Yang, "NLOS mitigation for UWB localization based on sparse pseudo-input Gaussian process," *IEEE Sensors J.*, vol. 18, no. 10, pp. 4311–4316, May 2018.
[5] S. Wu *et al.*, "NLOS error mitigation for UWB ranging in dense multipath environments," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Oct. 2007, pp. 1565–1570.
[6] J. Khodjaev *et al.*, "Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments," *Ann. Telecommun. Ann. Commun.*, vol. 65, nos. 5–6, pp. 301–311, Jun. 2010.
[7] A. H. Muqaibel *et al.*, "Practical evaluation of NLOS/LOS parametric classification in UWB channels," in *Proc. 1st Int. Conf. Commun., Signal Process., Appl. (ICCSPA)*, Feb. 2013, pp. 1–6.
[8] K. Bregar *et al.*, "NLOS channel detection with multilayer perceptron in low-rate personal area networks for indoor localization accuracy improvement," in *Proc. 8th Jožef Stefan Int. Postgraduate School Students Conf.*, Ljubljana, Slovenia, Vol. 31, May 2016, pp. 1–8.
[9] V. Barral *et al.*, "NLOS identification and mitigation using low-cost UWB devices," *Sensors*, vol. 19, no. 16, p. 3464, Aug. 2019.
[10] H. Wymeersch *et al.*, "A machine learning approach to ranging error mitigation for UWB localization," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1719–1728, Jun. 2012.
[11] S. Marano *et al.*, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.
[12] C. Lu *et al.*, "MIMO channel information feedback using deep recurrent network," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 188–191, Jan. 2019.
[13] C. Jiang *et al.*, "A MEMS IMU de-noising method using long short term memory recurrent neural networks (LSTM-RNN)," *Sensors*, vol. 18, no. 10, p. 3470, Oct. 2018.