Self-Attention-Assisted TinyML With Effective Representation for UWB NLOS Identification

Yifeng Wu, Xu He, Lingfei Mo, *Member, IEEE*, and Qing Wang

***Abstract*—Ultra-wideband (UWB) non-line-of-sight (NLOS) identification is a crucial task in wireless localization systems. Various deep learning (DL) solutions have demonstrated promising outcomes in UWB NLOS identification by utilizing channel impulse response (CIR) and channel characteristics. However, effective and robust UWB NLOS identification on resource-constrained edge devices remains a challenge. Hence, this article presents a self-attention-assisted tiny machine learning (TinyML) solution that offers an effective representation for UWB NLOS identification. To overcome computational limitations, a feature selection method is devised for the proposed data-driven DL-based approach. By leveraging feature selection, the self-attention mechanism enhances the representation capability of a pretrained model for UWB NLOS identification. The proposed method is evaluated on both personal computer (PC) and edge platforms and compared against multiple baselines. The evaluation demonstrates its effective representation and optimal performance on both PC and edge platforms, as indicated by various metrics. Thanks to the effective representation, the proposed method also enables the quantized model to achieve state-of-the-art (SOTA) in UWB NLOS identification, while significantly accelerating inference efficiency at the edge.**

***Index Terms*—Channel impulse response (CIR), non-line-of-sight (NLOS) identification, self-attention, tiny machine learning (TinyML), ultra-wideband (UWB).**

I. Introduction

IDENTIFYING non-line-of-sight (NLOS) signals in wireless positioning tasks is a challenge faced by wireless systems [1], [2]. Ultra-wideband (UWB), a wireless radio frequency (RF) technology, stores a substantial amount of channel impulse response (CIR) information in the DW1000 chip, as stated in [3]. This chip also offers various analytical channel characteristics that can be computed and extracted. In the presence of NLOS conditions during UWB signal propagation, there is often a delay in signal arrival and a decrease in energy compared to line-of-sight (LOS) conditions [3], [4], [5]. Consequently, the UWB CIR signals exhibit noticeable differences between NLOS and LOS propagation conditions [4]. Additionally, the DW1000 chip provides several significant channel characteristics that can be utilized for evaluating the quality of UWB signals [3], [19], [20].

Manuscript received 19 October 2023; revised 2 December 2023; accepted 19 December 2023. Date of publication 3 January 2024; date of current version 25 July 2024. This work was supported by the National Key Research and Development Program of China under Grant 2020YFD1100201. *(Xu He and Yifeng Wu are the co-first authors.) (Corresponding author: Lingfei Mo.)*

The authors are with the School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: wyfengever@163.com; 1165443547@qq.com; lfmo@seu.edu.cn; w3398a@263.net).

Digital Object Identifier 10.1109/JIOT.2024.3349462

The rapid advancement of Internet of Things (IoT) technology has brought edge computing and embedded artificial intelligence (AI) to the forefront, holding significant project and research value [13], [21]. Similarly, achieving robust and efficient inference for UWB NLOS identification under resource-constrained devices is crucial for wireless localization systems. Tiny machine learning (TinyML), an intersection of IoT and machine learning (ML), enables the implementation of ML algorithms on low-resource, low-power microcontroller units (MCUs) [13], [14], [15]. Tensorflow lite for microcontrollerss (TFLMs) [14], [16] is currently a reliable solution that effectively supports the deployment and operation of TinyML on microcontrollers.

In UWB-based wireless positioning systems, UWB NLOS propagation is a major cause of abnormal measurement errors [1], [10], [19] and undermines confidence in positioning accuracy [1], [6], [8]. UWB NLOS identification is conducive to the reasonable selection of base station positioning strategy, or the favorable compensation of observation errors in NLOS/LOS mixed conditions [25]. Therefore, UWB NLOS identification is critical to reliable and accurate wireless positioning estimation. UWB NLOS identification is a binary classification task that heavily relies on establishing a strong relationship between input features and labels. In deep learning (DL) models, the shallow networks primarily function as feature extractors, whereas the deep networks play a crucial role in capturing the specific task functionality. For instance, in [9] and [17], convolutional neural network (CNN) is utilized as a feature extractor for CIR signals, where the extracted features are then fed into deep classifier networks.

Transfer learning, a popular technique in the field, often employs the pretraining strategy to enhance the performance of target tasks [22]. This strategy involves gathering a substantial amount of inexpensive training data, learning their shared characteristics through pretraining, transferring these commonalities to the model for the specific task, and subsequently fine-tuning the model using a limited amount of annotated data within the specific domain. Consequently, the model only needs to learn the task-specific aspects starting from the established commonalities [22], [23]. Considering the concept of pretraining, it becomes evident that the pretraining strategy also serves as a valuable means of feature extraction.

Presently, attention mechanisms have gained significant attention as a means of enhancing the performance of DL models [18], [20]. However, it is important to exercise caution when incorporating attention modules, as their blind insertion

can potentially result in model degradation. In [19], a noteworthy attention-assisted algorithm is discussed, showcasing that a thoughtfully designed attention module surpasses the mere addition of additional layers when it comes to optimizing model performance.

To address the challenge of achieving effective and robust UWB NLOS identification on resource-constrained edge devices, a self-attention-assisted TinyML solution is proposed in this article. This solution leverages the self-attention mechanism to enhance the effective representation ability of a pretrained classifier model, and the post-train quantization (PTQ) strategy [33] to accelerate inference efficiency at the edge. Through the developed feature selection technique (detailed in Section III), the computational resource consumption required by the proposed method is sharply reduced compared to using the full CIR data. At the same time, the key features selected that are significantly related to NLOS identification also enable the quantized model to achieve effective and robust representation. Notably, the proposed method surpasses all baseline strategies in performance on the personal computer (PC) side. Moreover, it also exhibits the best inference performance on a resource-constrained embedded MCU, achieving results comparable to those obtained on the PC side. The main contributions are as follows.

1) This article proposes a self-attention-assisted algorithm that significantly enhances the performance of a pretrained classifier model, empowering it with a more effective representation ability for robust UWB NLOS identification.
2) By leveraging the developed feature selection technique, this article reports a TinyML solution using the embedded AI technique with quantization for UWB NLOS identification at the edge. This solution is specifically tailored to efficiently operate within wireless systems with limited resources.
3) In addition to validating the performance of the proposed method on the PC side, this article also presents the testing results after quantization and deployment on an embedded MCU. Furthermore, comprehensive comparisons between the proposed method and multiple baseline strategies are documented in this article.

This article is structured as follows. Section II presents the related works. Section III is method design, encompassing the designed feature selection, algorithm design, and quantization and deployment solution. Section III provides an evaluation of the proposed method on both PC and embedded MCU platforms, along with comparisons against multiple baselines. In Section V, an in-depth analysis and discussion of the observed phenomena and performance of the proposed method and baselines are presented. Finally, Section VI concludes this article by summarizing the key findings.

II. Related Works

Solutions of UWB NLOS identification can be broadly categorized into two types: 1) nonfeature-based methods and 2) feature-based methods [4]. The former typically relies on prior information or contextual cues, while the latter leverages the distinctions captured by CIR information under different UWB propagation conditions.

Over the years, starting from the early usage of support vector machine (SVM) [6], a range of state-of-the-art (SOTA) ML and DL methods [7], [8], [9], [10] have been extensively employed in feature-based UWB NLOS identification tasks.

Successful UWB NLOS identification has been achieved using CNN and image-based CIR data [7], [11], [12]. CNN models have demonstrated superior accuracy and consistent performance compared to traditional SVM-based classification methods. Long short-term memory (LSTM) networks, known for their excellent representation capability in time-series signals, have been employed to distinguish NLOS propagation in UWB signals [8]. To combine the feature extraction capability of CNN with LSTM, a CNN-LSTM-based method for UWB NLOS identification is reported in [9]. Another similar approach, utilizing CNN and LSTM to extract spatiotemporal features for UWB NLOS identification is reported in [10].

Attention-based models have emerged as a crucial concept in DL and have been extensively investigated across various application domains. Notably, the self-attention mechanism, a variant of the attention mechanism, exhibits reduced reliance on external information and excels in capturing intrinsic correlations within data or features [35]. In [11], a fully convolution network (FCN)-Attention-based method achieves 88.24% accuracy in UWB NLOS identification using full CIR data in an end-to-end manner. Additionally, [11] also presents comparisons of several baselines, including LSTM, CNN, CNN-LSTM, and FCN, to demonstrate the superiority of FCN-Attention. However, it is important to note that previous studies have inevitably increased computational resource costs to improve the performance of UWB NLOS identification.

In [25], a CNN-based UWB NLOS identification strategy using CIR data is proposed to enhance the performance of wireless positioning systems using a resource-constrained device. However, their solution is not suitable for TinyML, which aims to deploy models on embedded MCUs. When using TinyML, the primary consideration is compressing the model size while maintaining performance, enabling deployment on resource-constrained edge devices, particularly MCUs.

Han et al. [38] introduced the "Deep Compression" method, which significantly reduced the scale of the AlexNet and VGG16 models by 35× and 49×, respectively, without sacrificing noticeable accuracy. A quantization scheme is reported in [32] that enables integer-arithmetic-only inference, improving efficiency on commonly available integer-only hardware compared to floating-point operations. Moreover, an 8-bit quantization workflow [31] is provided that maintains accuracy within 1% of the floating-point baselines across various networks, even for models with high quantization difficulty. Additionally, [33] introduces EasyQuant, an efficient post-training method that achieves accuracy levels comparable to training-based methods through scale optimization.

III. Method Design

The proposed method is outlined in Fig. 1, providing an overview of the method design. Serving for resource-constrained conditions, the method design focuses

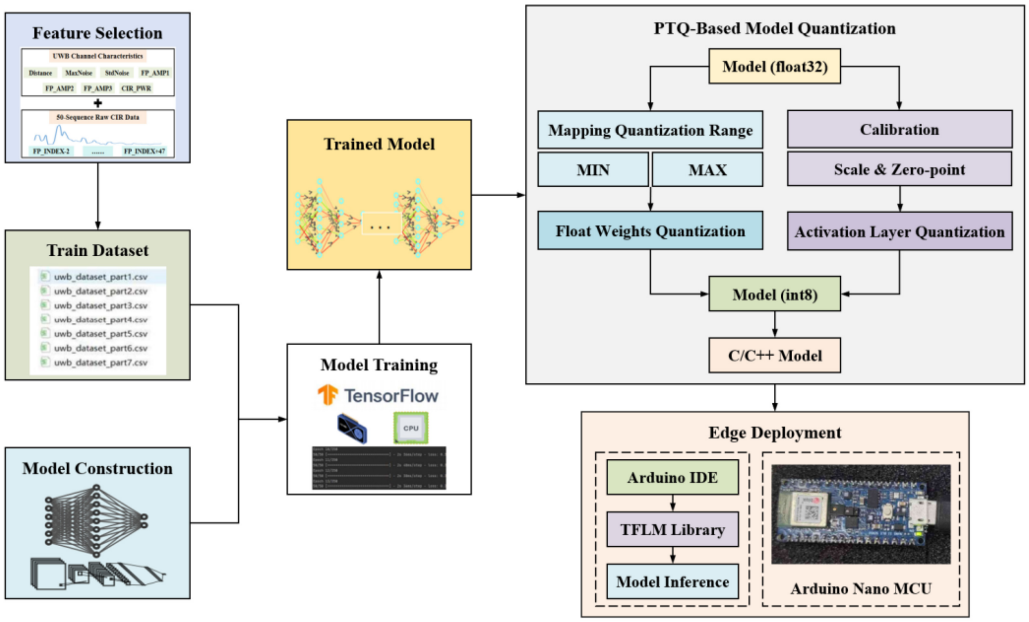


Fig. 1. Overview of the proposed method.

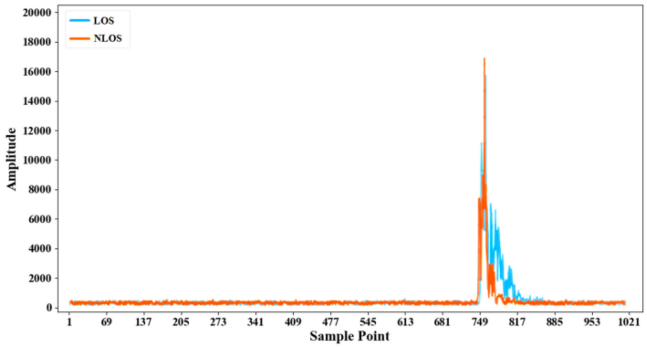


Fig. 2. Example sets of raw CIR signals sampled under NLOS and LOS conditions.

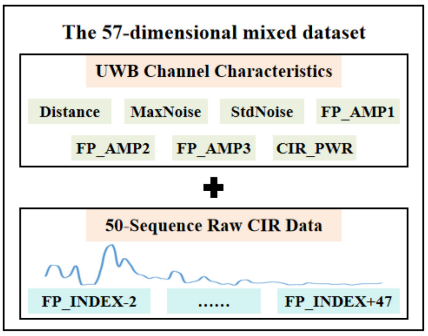


Fig. 3. 57-D mixed data set.

on addressing the following core issues: 1) feature selection; 2) algorithm design; and 3) quantization and deployment.

A. Feature Selection

The required raw CIR signals and channel characteristics can be obtained using commercial UWB modules integrated with the DW1000 chip. According to [3], the raw CIR data is stored in the Accumulator register of the DW1000 chip, with 992 samples at a pulse repetition frequency of 16 MHz, and 1016 samples at 64 MHz. Each CIR data corresponds to a period of approximately 1 μs, which corresponds to a flight distance of about 30 cm for the UWB radio signal in the air.

An example set of raw CIR signals within one complete time period is shown in Fig. 2. Due to slight variations in each pulse response, the initial rise time of the pulse envelope in the Accumulator register also exhibits corresponding differences. The DW1000 uses the built-in leading-edge detection (LDE) algorithm to detect and determine the starting index position of the signal (which can be obtained by reading the value of the FP_INDEX register of the chip) [3]. Due to the different propagation conditions of NLOS and LOS, the impact on the curve of the raw CIR sequence data is significant.

For algorithm and model evaluation convenience, this article utilizes the publicly available elastic Wireless Networking Experimentation (eWINE) UWB data set [25]. The data set consists of samples observed in seven different scenarios, with 3000 groups of LOS samples and 3000 groups of NLOS samples in each scenario. Each sample in these observations has a length of 1031, including 1016 raw CIR sequence data and seven UWB channel characteristics closely associated with the first path records and quality of UWB signals. Details of first path (FP) records and these channel characteristics can be found in [3] and [19].

Considering that DL models are data-driven algorithms, using the complete raw CIR sequence data or image-based CIR data undoubtedly costs more computational resources to ensure the effectiveness of feature extraction and task fitting. Since UWB is an RF technology, the important channel characteristics that are highly correlated with RF signal quality can be collected from its DW1000 chip [3], [19], as well as its complete CIR data [25]. According to [3], [24], [25],

and the raw CIR examples shown in Fig. 2, it is not difficult to find that the crucial features highly related to the UWB NLOS identification are concentrated. Most of the effective data that contains important features for NLOS identification, as mentioned in [24], is concentrated within the interval of 50 groups with FP_INDEX as the index. As reported in [24], to reduce the computational scale, this article truncates and retains 50-sequence key CIR data starting from the index ranging from FP_INDEX-2 to FP_INDEX+47.

In addition to the CIR data, the channel characteristics that are closely related to UWB signal analysis with clear physical meaning have been used in [19] and [37] for UWB ranging error compensation, which can prove their effectiveness for UWB signal quality analysis. More details can be found in [3], [19], and [37] and their references.

The selected 50-sequence data are concatenated with the available seven channel characteristics to form a 57-D mixed data set Fig. 3. The data set dimension (1×57) used in this article is reduced by nearly 94.4% compared to the fully CIR data with the size (1×1016) used in [9] and [11]. In addition, it is also reduced by about 2.67 times compared to [25]. This allows faster inference under resource-constrained conditions while ensuring inference performance at the edge.

Additionally, the data set is randomly split into three parts: 1) training set; 2) validation set; and 3) test set, with proportions of 60%, 20%, and 20%, respectively. In order to avoid the overfitting phenomenon, the data set is shuffled multiple times before the split to ensure a balanced data distribution.

B. Algorithm Design

The attention mechanism encompasses three key elements: Query (Q), Key (K), and Value (V). Soft-attention mechanisms typically require extra source and target information to construct the Q, K, and V vectors [25]. In contrast, self-attention is designed to extract interactions between input features without relying on additional source and target information [11], [20], [35]. Given input data X, the vectors Q, K, and V can be calculated by formula

$$\begin{cases} Q = f_1(X) = \sum W_1 X + B_1 \\ K = f_2(X) = \sum W_2 X + B_2 \\ V = f_3(X) = \sum W_3 X + B_3 \end{cases} \tag{1}$$

where W_1, W_2, and W_3 are linear matrices that transform the input X vectors into new vectors through linear computations. The three linear transformation matrices have the same structure and are independent of each other, with specific parameter values obtained during model training. B_1, B_2, and B_3 are bias vectors that do not change the vector structure of Q, K, and V and can be obtained through model training.

The self-attention output can be updated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V. \tag{2}$$

The attention distribution can be calculated by computing the inner product of Q and K, where d_k is the dimension of X. To normalize the attention score distribution, the softmax function is applied along with a scaled dot-product operation. The final attention output is obtained by multiplying the attention score distribution with V.

The attention mechanism, known for its high-performance feature selection capabilities, is widely recognized as a method for enhancing the performance of DL models [10], [11], [20]. In this study, a self-attention-assisted method is devised to strengthen the representation ability of UWB NLOS identification in the pretrained classifier model. The proposed method leverages a pretraining strategy to enable the shallow network of the pretrained model to learn and capture effective commonalities in feature distributions. Subsequently, the shallow network, responsible for feature extraction, is frozen, and a new self-attention-assisted classifier network is retrained. This approach heavily depends on the feature selection and re-evaluation capabilities of self-attention, which play a crucial role in re-evaluating the captured feature contributions and enhancing the effective representation ability.

In Fig. 4, the pretrained classifier model is represented in (a), while the retrained self-attention-assisted classifier network is depicted in (b). As illustrated in (a), the pretrained model comprises five fully connected (FC) layers and three batch normalization (BN) layers. The activation function of the first FC layer is linear, and the second FC layer is responsible for extracting abstract features from the input data.

In (b), the first three layers are frozen to preserve the pre-extracted feature representations, which are utilized by the self-attention-assisted classifier network for UWB NLOS identification. The introduction of the self-attention mechanism changes the network structure of the pretrained model, and thus the retraining operation should not be consistent with the traditional fine-tuning operation [38], which does not alter the structure of the pretrained model.

During the retraining process, the size of the retrained classifier network was trimmed to an appropriate amount to optimize the overall computational resource consumption compared to the pretrained model. Subsequently, while keeping the first three layers of the pretrained network frozen, the self-attention and new classifier in (b) are considered as a new model to be retrained. The Adam optimizer was used for training the self-attention-assisted classifier network, and the cross-entropy method was used as the loss function. The batch size was set to 256 and the number of epochs was 350. The details of these two models, including model structures, parameter sizes, and activation functions, are presented in Fig. 4.

C. Quantization and Deployment

Currently, model quantization is a prominent area of research in the field of TinyML.

1) Model Compression: Raw DL models trained and saved in TensorFlow are typically large and cannot be directly deployed on embedded devices [26]. TFLM offers a dedicated converter that compresses the model into a specialized TFlite file [16], [26]. This TFlite file is fully compatible with the sequential format of the runtime memory, eliminating the need for additional copying and parsing operations during inference.

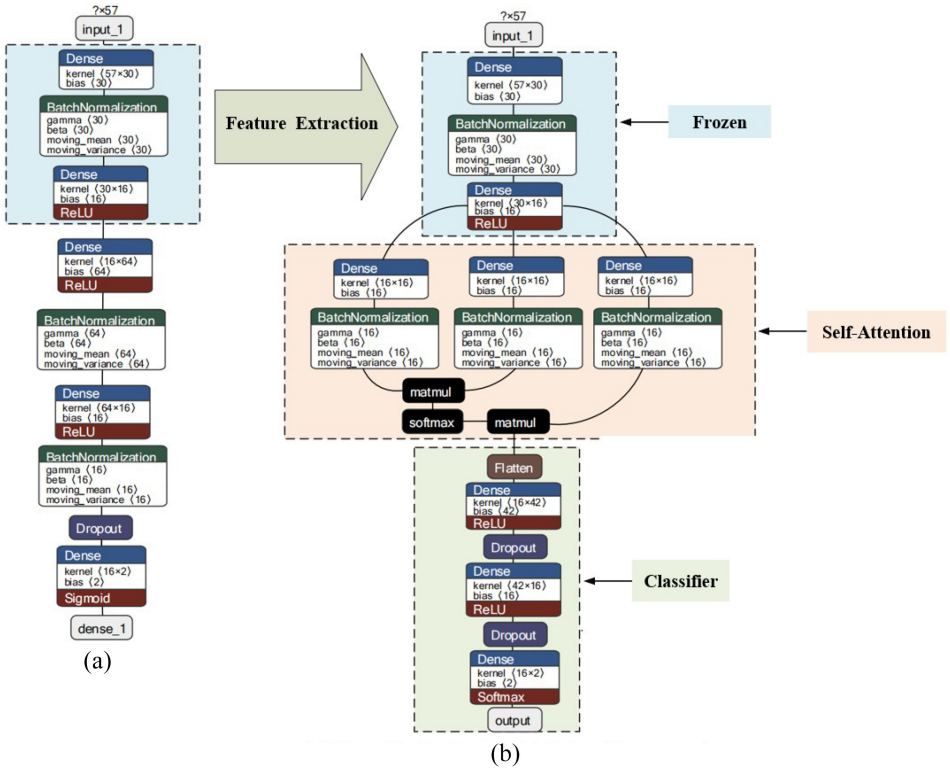


Fig. 4. Algorithm design for UWB NLOS identification. (a) Pretrained classifier model. (b) Self-attention-assisted classifier network.

Additionally, the model can be converted into a hexadecimal C/C++ array file for MCU devices without a file system.

During the model compression process, unnecessary training operators, such as the Adam optimization algorithm, and forward and backward propagation algorithms, are removed from the model since they are not needed during inference. For operators like BN and dropout, which have differences between training and inference but cannot be directly removed, folding techniques are used to merge the layer weight parameters into other layer parameters. These layers are then removed after updating the parameters of other layers [16], [29], [30], [31], [32], [33].

2) Model Quantization: The parameters in the compressed TensorFlow models are stored in the "float32" format by default. Full integer quantization [31] is a quantization method with strong hardware platform compatibility, where the full integer model does not involve any floating-point computations during inference [31], [32], [33], [34]. In practice, it has been observed that using the "int8" format to store weights and activation values can produce inference results almost identical to those of "float32" models [31], [33]. Some researchers have even successfully quantized weight values into binary networks with values of 0 and 1, and ternary networks with values of 1, 0, and −1 [27], [28], [29].

Undoubtedly, model quantization is an effective approach to reduce the model size and computational costs during inference. However, converting the parameters in the compressed models from "float32" format to "int8" format is not a straightforward task. One reliable solution is to employ the PTQ strategy, where weights are quantized to "int8" values while the activation layers still use the "float32" format. This reduces the model file size to 1/4 of its original size and improves inference speed.

The accuracy of model quantization greatly depends on correct scaling factors, zero-point, and activation layer range [29], [30], [33]. To ensure accurate quantization of activation layers, a calibration set consisting of 500 samples is used in this article. Typically, the calibration samples are randomly selected from the training or test set. Through the calibration operation, the input–output range and activation layer range of the model can be determined [33], [34].

3) Model Deployment: The TFlite model files can be stored in the memory of edge devices and accessed using the Interpreter constructor in the program. However, most MCUs do not have a file system, and even if a small number of MCUs do have one, it would still require additional resources for loading and caching files. As an alternative, the TFlite file can be converted into a C/C++ source code file [16]. The final step in deploying the model involves downloading the compiled C/C++ source code file as firmware into the MCU. This allows for immediate access to the model during program execution and inference at the edge.

IV. Performance Evaluation

The UWB NLOS identification method proposed in this article primarily relies on feature selection, pretraining strategy, and classifier pruning and retraining based on self-attention. The evaluation of the proposed method is conducted using comprehensive metrics, such as accuracy, precision, recall, and

TABLE I
PERFORMANCE OF STRATEGIES ON THE PC SIDE

	Strategy	Accuracy	Precision	Recall	F1 score
1	CNN [25]	87.38%	85.92%	89.41%	87.63%
2	CNN-LSTM [9]	82.37%	79.44%	87.36%	83.21%
3	FCN [11]	86.26%	83.82%	89.87%	86.74%
	LSTM-FCN [11]	86.35%	83.87%	90.01%	86.83%
	FCN-Attention [11]	88.24%	85.85%	91.56%	88.62%
4	Depthwise CNN (Unquantized)	91.33%	89.30%	93.22%	91.22%
	Depthwise CNN (Quantized)	90.05%	86.83%	90.88%	88.81%
5	MLP (Unquantized)	92.30%	91.48%	93.13%	92.30%
	MLP (Quantized)	90.53%	92.23%	89.34%	90.76%
6	***Ours (Unquantized)***	***93.10%***	***93.21%***	***93.13%***	***93.17%***
	Ours (Quantized)	***92.99%***	***92.65%***	***93.42%***	***93.03%***

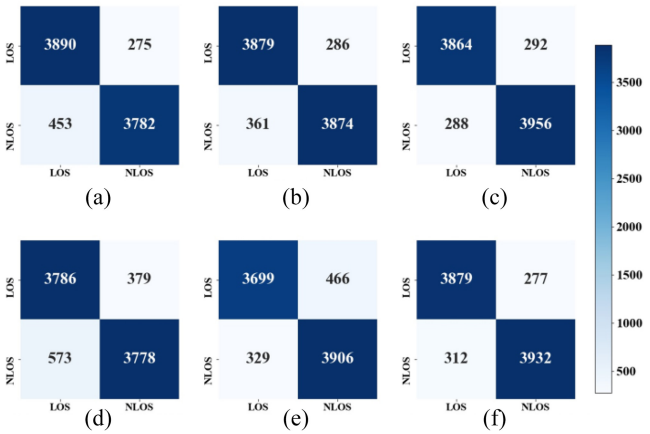


Fig. 5. Confusion matrices of the three groups of selected strategies before and after quantization on the PC side. (a) Depthwise CNN. (b) MLP. (c) Proposed method. (d) Quantized depthwise CNN. (e) Quantized MLP. (f) Quantized proposed method.

F1 score. To facilitate the evaluation process, Table I provides a summary of the reported results from recent SOTA baseline strategies. The following section describes these baseline strategies in detail.

Strategy 1 (CNN [25]): CNN has been a widely favored UWB NLOS classification method [7], [11], [12]. In [25], the performance of the CNN-based strategy for UWB NLOS identification on the eWINE data set is reported and tested under resource-constrained conditions. Compared to classical SVM-based classification algorithms, the reported CNN-based strategy achieves more significant success.

It is worth mentioning that the eWINE data set, which is publicly available and widely cited, has supported a large amount of research on UWB NLOS identification.

Strategy 2 (CNN-LSTM [9]): The performance of CNN-LSTM-based strategy for UWB NLOS identification on the eWINE data set is reported in [9]. Similarly, a better-performing strategy based on CNN-stacked-LSTM can also be found in [9].

Strategy 3 (FCN-Attention [11]): The performance of the FCN-Attention-based strategy on the eWINE data set can be found in [11]. Furthermore, [11] also compares the performance of strategies, including LSTM, FCN, and LSTM-FCN.

Strategy 4 (Depthwise CNN [36]): Depthwise CNN is a variant of CNN aimed at reducing the parameters and computational complexity while maintaining good performance. Traditional convolution operations involve a large number of parameters because they have a separate convolution kernel for each input channel. In contrast, Depthwise CNN utilizes depthwise convolution operations, applying a separate convolution kernel to each input channel. This means that each input channel has its corresponding convolution kernel for feature extraction, reducing the number of parameters required.

To our knowledge, there are currently no specific reports on the use of a Depthwise CNN-based strategy for UWB NLOS identification. In this article, Depthwise CNN, an improved strategy of conventional CNN methods, is compared and validated against the proposed method under the same conditions.

Strategy 5 [Multilayer Perceptron (MLP)]: This strategy is the pretrained classifier model used in this article, as shown in Fig. 4(a). Thanks to the designed feature selection method, even with the use of a low-complexity MLP, excellent UWB NLOS identification performance can be achieved on the PC side. This phenomenon highlights the significant contribution of feature selection to data-driven DL algorithms.

Note: Strategy 4 and Strategy 5 are additional baselines designed in this article, aiming to accelerate edge inference under resource-constrained conditions with minimal parameter count. With the developed feature selection method, strategy 5, namely, MLP, can even outperform all the performance of summarized baseline strategies on the PC side. Among the numerous baselines, strategy 4 (Depthwise CNN) and strategy 5 (MLP) described above are the best-performing with the fewest parameters.

UWB NLOS identification is a binary classification issue. The confusion matrix is an effective tool to evaluate the performance of classification algorithms [11], [25]. The results are usually divided into four categories. In this article, correctly predicted NLOS samples are referred to as true positives (TP), and incorrectly predicted ones are referred to as false positives (FP). Similarly, correctly predicted LOS samples are set as true negatives (TN), and incorrectly predicted ones are false negatives (FN). In Figs. 5 and 6, the *rows* represent the true labels, while the *columns* represent the predicted labels.

This article not only reports the performance of the proposed method on the PC side but also tests the proposed method at the edge based on the quantization and deployment scheme described in Section II. Furthermore, evaluations of the three groups of selected strategies, that have achieved the best results on the PC side, are recorded before and after quantization both on the PC side and at the edge. It is worth mentioning that the three groups of selected strategies also

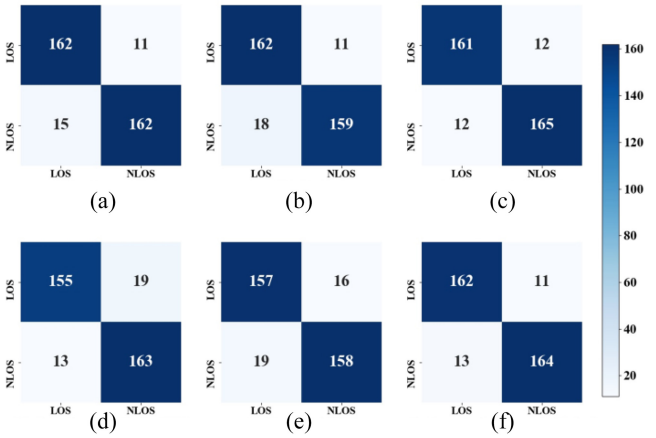


Fig. 6. Confusion matrices of the three groups of selected strategies before and after quantization at the edge. (a) Depthwise CNN. (b) MLP. (c) Proposed method. (d) Quantized depthwise CNN. (e) Quantized MLP. (f) Quantized proposed method.

TABLE II
PERFORMANCE EVALUATION AT THE EDGE

Strategy	Accuracy	Precision	Recall	F1 score	Model Size (kb)	Inference Duration (ms)
Depthwise CNN (Unquantized)	92.57%	91.53%	93.64%	92.57%	30.1	5.72±0.1
Depthwise CNN (Quantized)	90.86%	92.61%	89.56%	91.06%	11.1	2.89±0.07
MLP (Unquantized)	91.71%	89.83%	93.53%	91.64%	20.6	1.42±0.06
MLP (Quantized)	90.00%	89.27%	90.80%	90.03%	8.7	0.84±0.02
Ours (Unquantized)	93.14%	93.22%	93.22%	93.22%	21.9	1.69±0.06
Ours (Quantized)	***93.14%***	***92.66%***	***93.71%***	***93.18%***	***10.3***	***0.97±0.02***

have the lowest parameters and complexity among all the strategies mentioned above. Thus, they are most suitable for application in TinyML, and their model sizes are listed in Table II.

Due to limited storage resources on Arduino Nano, 350 data samples were randomly selected from the test set for inference on Arduino Nano. We recorded the inference results outputted by Arduino Nano and computed the confusion matrixes, as shown in Fig. 6. The evaluation metrics and performance, using the selected 350 data samples, at the edge are summarized in Table II. To evaluate the inference speed of different strategies, the clock information of the MCU was carefully recorded to obtain the precise inference duration for a single inference.

V. ANALYSIS AND DISCUSSION

Feature selection is crucial for data-driven algorithms such as DL models, as it allows for fitting and representing large amounts of raw data without increasing computational scale. Based on the summarized results in Table I, it can be analyzed that, thanks to the designed feature selection method, even using a simple MLP model can outperform SOTA DL-based strategies [7], [8], [9], [10], [11]. Furthermore, this shapely reduces the computational cost and model size, making the developed DL-based strategies more suitable for TinyML under resource-constrained conditions.

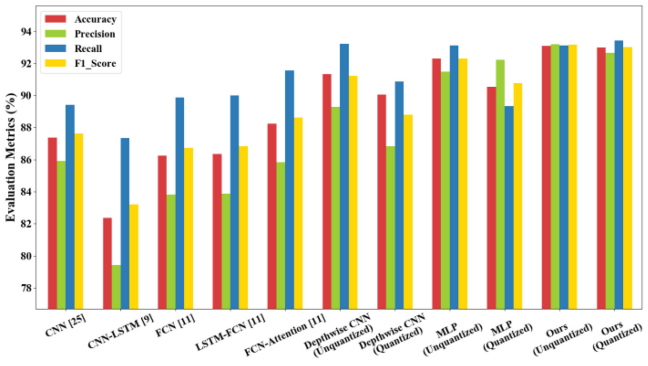


Fig. 7. Evaluation metrics of different strategies on the PC side.

Strategy 4 (Depthwise CNN) designed in this article aims to reduce model size and parameter quantity while ensuring the inference performance. However, as shown in the model size records in Table II, compared to traditional CNNs [25], even though the Depthwise CNN reduces the overall parameters through parameter sharing, it still does not have an advantage in terms of model size and inference speed, despite performing similarly to MLP. Therefore, other baselines that are more complex than CNN were not included in the statistics in Table II, considering the inference efficiency on resource-constrained edge devices and the performance degradation caused by quantization and deployment.

According to Table I, after considering all factors, it is evident that our method consistently outperforms all mentioned baselines in terms of accuracy, precision, and F1 score on the PC side, for UWB NLOS identification, regardless of whether it is quantized or not. Before quantization, our method attained a slightly lower sensitivity (recall) in comparison to the unquantized Depthwise CNN. However, when taking into account the other three crucial evaluation metrics, the proposed model undoubtedly outperforms all baseline strategies, as shown in Fig. 7. The most noteworthy experimental finding is that, in contrast to the specifically designed Depthwise CNN and MLP strategies with an evident performance degradation after quantization, our method merely exhibits a limited decline in performance after quantization. This discrepancy can be attributed to the fact that the proposed model successfully captures effective representations that are minimally affected by the changes in model weights before and after quantization.

Based on Table II, it can be observed that the quantized MLP model achieves the best results in terms of model size and single inference duration on the MCU, followed by our proposed method. At the edge, after quantization, our method has a model size that is 1.6 kB larger than the quantized MLP model and an inference duration approximately 0.13 ms longer than MLP for a single inference. However, our method

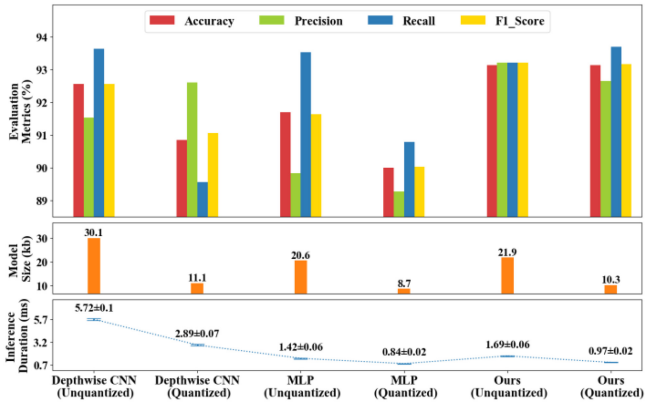


Fig. 8. Evaluation metrics of different strategies at the edge.

surpasses MLP by approximately 3% points in all evaluation metrics, as shown in Fig. 8. Despite incurring a slight increase in computational resource cost and inference duration compared to MLP, our method outperforms at the edge by demonstrating superior performance. Moreover, its robust and exceptional performance on the MCU is comparable to that on the PC side.

Additionally, the proposed method also surpasses the Depthwise CNN, which is specifically designed to reduce the computational cost and model size of traditional CNNs. While other SOTA strategies may demonstrate strong performance on high-performance workstations or the PC side, they are not suitable for efficient and reliable inference on resource-constrained edge devices. This is primarily due to their high complexity and the performance degradation after quantization. Hence, considering the balanced factors, the proposed method emerges as the optimal choice under resource-constrained conditions.

Above, the performance of the proposed self-attention-assisted TinyML has been discussed and analyzed. Next, we will discuss something about the deployment and application of the TinyML solution proposed in this article.

The proposed self-attention-assisted TinyML algorithm requires at least 10.3 kB of storage memory for model deployment in practical usage, as shown in Table II. Meanwhile, the actual inference speed for each input data on the Arduino nano MCU used in this article is faster than 1 ms. With the detailed specifications of Arduino nano (the edge device used in this article) listed in Table III, the practical computational resource requirements of the proposed self-attention-assisted TinyML solution can also be used as a reference for other MCU devices.

TinyML is a relatively new concept in the AI and ML industry, and although progress has been made, it still faces many challenges in practical deployment and application. The data frame rate and throughput during the actual operation of the TinyML devices are important considerations for the normal operation of the device. Sufficient power supply, memory constraints of MCUs and operating frequency may become potential challenges in the practical application and deployment of the proposed TinyML solution. In addition, it is worth mentioning that, the heterogeneity of hardware devices makes deploying TinyML difficult. Therefore, currently, it is necessary to adjust the developed TinyML algorithm for different devices for their different hardware specifications and requirements, which also becomes a problem.

TABLE III
SPECIFICATIONS OF ARDUINO NANO

MCU	nRF52840
Working Voltage	3.3 V
Clock Frequency	64 MHz
Flash Memory	1MB (nRF52840)
SRAM	256KB (nRF52840)
Power Consumption	25mA@64MHZ / 0.5uA@System OFF mode

VI. CONCLUSION

This article presents a self-attention-assisted TinyML solution to solve the challenge of effective and robust UWB NLOS identification in resource-constrained edge devices. This article utilizes a developed feature selection method to reduce resource consumption, ensuring optimal performance while facilitating the implementation and deployment of the TinyML solution.

The designed feature selection method proves to be highly effective in reducing computational costs and model size, surpassing several SOTA strategies, including the low-complexity MLP strategy. The key aspect of the proposed method lies in leveraging the self-attention mechanism to enhance the effective representation ability of the pretrained model for UWB NLOS identification.

On the PC side, validations for the proposed self-attention-assisted algorithm showcase its superior performance by comparing it to several SOTA strategies reported in recent years. Through experimental observations, it can be inferred that the proposed algorithm enables the model to learn effective representations for UWB NLOS identification, resulting in only limited performance degradation after quantization. Then, the validations conducted at the edge also demonstrate its effective representation ability of UWB NLOS identification, leading to advanced edge inference performance, which is on par with the performance observed on the PC side.

Overall, the proposed method significantly enhances the pretrained model with effective representations in UWB NLOS identification, delivering robust and exceptional performance both on the PC side and at the edge. Consequently, it emerges as an advanced choice for UWB NLOS identification in resource-constrained edge devices.

ACKNOWLEDGMENT

The authors would like to thank Bregar and Mohorčič [25] for their open-source eWINE data set, which can be available online: https://github.com/ewine-project/UWB-LOS-NLOS-Data-Set.

REFERENCES

[1] J. Khodjaev, Y. Park, and A. S. Malik, "Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments," *Ann. Telecommun. Ann. Des Télécommun.*, vol. 65, no. 5, pp. 301–311, Jun. 2010.

[2] V. B. Vales, T. Domínguez-Bolaño, C. J. Escudero, and J. A. García-Naya, "Using the power delay profile to accelerate the training of neural network-based classifiers for the identification of LOS and NLOS UWB propagation conditions," *IEEE Access*, vol. 8, pp. 220205–220214, 2020.

[3] *DW1000 User Manual*, 2nd ed., DecaWave Limited Co., Dublin, Ireland, 2017.

[4] F. Che et al., "Feature-based generalized gaussian distribution method for NLoS detection in ultra-wideband (UWB) indoor positioning system," *IEEE Sensors J.*, vol. 22, no. 19, pp. 18726–18739, Oct. 2022.

[5] S. Angarano, V. Mazzia, F. Salvetti, G. Fantin, and M. Chiaberge, "Robust ultra-wideband range error mitigation with deep learning at the edge," *Proj. Appl. Artif. Intell.*, vol. 102, Jun. 2021, Art. no. 104278, doi: 10.1016/j.engappai.2021.104278.

[6] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, "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.

[7] T. Zeng, Y. Chang, Q. Zhang, M. Hu, and J. Li, "CNN-based LOS/NLOS identification in 3-D massive MIMO systems," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2491–2494, Dec. 2018.

[8] D.-H. Kim, A. Farhad, and J.-Y. Pyun, "UWB positioning system based on LSTM classification with mitigated NLOS effects," *IEEE Internet Things J.*, vol. 10, no. 2, pp. 1822–1835, Jan. 2023, doi: 10.1109/JIOT.2022.3209735C.

[9] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo, "UWB NLOS/LOS classification using deep learning method," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2226–2230, Oct. 2020.

[10] B. Yang, J. Li, Z. Shao, and H. Zhang, "Robust UWB indoor localization for NLOS scenes via learning spatial-temporal features," *IEEE Sensors J.*, vol. 22, no. 8, pp. 7990–8000, Apr. 2022, doi: 10.1109/JSEN.2022.3156971.

[11] Y. Pei, R. Chen, D. Li, X. Xiao, and X. Zheng, "FCN-Attention: A deep learning UWB NLOS classification algorithm using fully convolution neural network with self-attention mechanism," *Geo-Spat. Inf. Sci.*, vol. 23, no. 22, pp. 1–20, Nov. 2023, doi: 10.1080/10095020.2023.2178334.

[12] Z. Cui, Y. Gao, J. Hu, S. Tian, and J. Cheng, "LOS/NLOS identification for indoor UWB positioning based on Morlet wavelet transform and convolutional neural networks," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 879–882, Mar. 2021.

[13] N. Schizas, A. Karras, C. Karras, and S. Sioutas, "TinyML for ultra-low power AI and large scale IoT deployments: A systematic review," *Future Internet*, vol. 14, no. 12, p. 363, 2022.

[14] M. Giordano, N. Baumann, M. Crabolu, R. Fischer, G. Bellusci, and M. Magno, "Design and performance evaluation of an ultralow-power smart IoT device with embedded TinyML for asset activity monitoring," *IEEE Trans. Instrumen. Meas.*, vol. 71, Apr. 2022, Art. no. 2510711, doi: 10.1109/TIM.2022.3165816.

[15] S. S. Saha, S. S. Sandha, and M. Srivastava, "Machine learning for microcontroller-class hardware: A review," *IEEE Sensors J.*, vol. 22, no. 22, pp. 21362–21390, Nov. 2022, doi: 10.1109/JSEN.2022.3210773.

[16] "TensorFlow Lite for microcontrollers." Mar. 2023. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers?hl=en

[17] M. Si, Y. Wang, H. Siljak, C. Seow, and H. Yang, "A lightweight CIR-based CNN with MLP for NLOS identification in a UWB positioning system," *IEEE Commun. Lett.*, vol. 27, no. 5, pp. 1332–1336, May 2023, doi: 10.1109/LCOMM.2023.3260953.

[18] H. Lim, C. Park, and H. Myung, "RONet: Real-time range-only indoor localization via stacked bidirectional LSTM with residual attention," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Macau, China, 2019, pp. 3241–3247, doi: 10.1109/IROS40897.2019.8968551.

[19] X. He, L. Mo, and Q. Wang, "An attention-assisted UWB ranging error compensation algorithm," *IEEE Wireless Commun. Lett.*, vol. 12, no. 3, pp. 421–425, Mar. 2023, doi: 10.1109/LWC.2022.3229104.

[20] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An attentive survey of attention models," *ACM Trans. Intell. Syst. Technol*, vol. 12, no. 5, pp. 1–32, Oct. 2021, doi: 10.1145/3465055.

[21] G. M. Iodice and R. Naughton, *TinyML Cookbook: Combine Artificial Intelligence and Ultra-Low-Power Embedded Devices to Make the World Smarter*. Birmingham, U.K.: Packt Publ., 2022.

[22] M. Sugiyama, J. Quiñonero-Candela, N. Lawrence, and A. Schwaighofer, "When training and test sets are different: Characterizing learning transfer," in *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009, pp. 3–28.

[23] Z. Sun, K. Wang, R. Sun, and Z. Chen, "Channel state identification in complex indoor environments with ST-CNN and transfer learning," *IEEE Commun. Lett.*, vol. 27, no. 2, pp. 546–550, Feb. 2023, doi: 10.1109/LCOMM.2022.3220506.

[24] S. Kram, M. Stahlke, T. Feigl, J. Seitz, and J. Thielecke, "UWB channel impulse responses for positioning in complex environments: A detailed feature analysis," *Sensors*, vol. 19, no. 24, p. 5547, 2019, doi: 10.3390/s19245547.

[25] K. Bregar and M. Mohorčič, "Improving indoor localization using convolutional neural networks on computationally restricted devices," *IEEE Access*, vol. 6, pp. 17429–17441, 2018, doi: 10.1109/ACCESS.2018.2817800.

[26] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[27] M. Courbariaux, Y. Bengio, and J. P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 1–9.

[28] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "BinaryNet: Training deep neural networks with weights and activations constrained to $+1$ or -1," 2016, *arXiv:1602.02830*.

[29] W. Tang, G. Hua, and L. Wang, "How to train a compact binary neural network with high accuracy?" in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2625–2631, doi: 10.1609/aaai.v31i1.10862.

[30] A. Gholami et al., "A survey of quantization methods for efficient neural network inference," 2021, *arXiv:2103.13630*.

[31] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, "Integer quantization for deep learning inference: Principles and empirical evaluation," 2020, *arXiv:2004.09602*.

[32] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2704–2713, doi: 10.1109/CVPR.2018.00286.

[33] D. Wu, Q. Tang, Y. Zhao, M. Zhang, Y. Fu, and D. Zhang, "EasyQuant: Post-training quantization via scale optimization," 2020, *arXiv:2006.16669*.

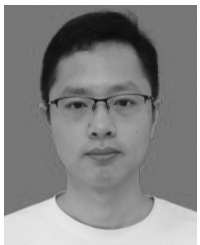
[34] M. Nagel, M. Fournarakis, R. Ali Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A white paper on neural network quantization," 2021, *arXiv:2106.08295*.

[35] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 6000–6010.

[36] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[37] C. Jiang, L. Mo, and Q. Wang, "Transfer learning based ranging error mitigation with mobile anchor in village cadastral survey," *IEEE Trans. Instrum. Meas.*, vol. 71, Sep. 2022, Art. no. 5502710, doi: 10.1109/TIM.2022.3205690.

[38] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.



Yifeng Wu received the master's degree from Southeast University, Nanjing, China, in 2023.

He was with the School of Instrument Science and Engineering, Southeast University. His research interests focus on IoT, wireless positioning, and TinyML.

Xu He is currently pursuing the Ph.D. degree in instrumentation science and technology with the School of Instrument Science and Engineering, Southeast University, Nanjing, China.

His research interests focus on IoT, intelligent perception, general AI, intelligent PNT, and brain-like navigation.

Lingfei Mo (Member, IEEE) received the B.S. degree in automation engineering from Beijing Jiaotong University, Beijing, China, in 2004, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2009.

He is currently an Associate Professor with the School of Instrument Science and Engineering, Southeast University, Nanjing, China. He worked as a Postdoctoral Fellow with the Department of Mechanical Engineering, University of Connecticut, Storrs, CT, USA, from 2011 to 2012, before joining the School of Instrument Science and Engineering, Southeast University in Fall 2012. He has authored over 100 technical publications in standard journals and conferences. He also has more than 30 China patents in the area of IoT and AI. His research interests focus on IoT, AI, and brain-like intelligence.

Dr. Mo is an Associate Editor of the IEEE JOURNAL OF RADIO FREQUENCY IDENTIFICATION.

Qing Wang received the Ph.D. degree from the School of Instrument Science and Engineering, Southeast University, Nanjing, China, in 1996.

He is currently a Chair Professor with the School of Instrument Science and Engineering, Southeast University. His main research interests include outdoor and indoor positioning and multisensor fusion.