This is the Pre-Published Version.

The following publication Y. Sartayeva and H. C. B. Chan, "Towards More Accurate Mobile Direction Finding with UWB," 2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC), Toronto, ON, Canada, 2025, pp. 749-758 is available at https://doi.org/10.1109/COMPSAC65507.2025.00101.

Towards More Accurate Mobile Direction Finding with UWB

Yerkezhan Sartayeva
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
yerkezhan.sartayeva@connect.polyu.hk

Henry C. B. Chan
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
henry.chan.comp@polyu.edu.hk

***Abstract*—UWB is becoming increasingly more available to the general public as part of consumer devices such as smartphones and smartwatches. This opens opportunities for new indoor positioning paradigms because UWB in consumer devices not only supports ranging but also AoA (Angle of Arrival) estimation, meaning dependence on additional positioning infrastructure can be reduced. Since this is a recent development, however, not much research has been conducted on evaluating UWB performance in these consumer devices and how to improve it for better indoor positioning accuracy. To contribute to this research gap, this paper is the first to propose a machine learning solution to AoA accuracy improvement in UWB-equipped iPhones when communicating with DWM3001CDK sensors while in motion. The distinguishing feature of our solution is that, unlike previous works, it uses AoA measurements for training instead of raw CIR (Channel Impulse Response) data, meaning the anchors do not need to be attached to a computer for data collection, which makes the installation of anchors more convenient. In addition, our solution combines machine learning with a collaborative approach based on our positioning vector framework, which further improves AoA error. We compiled a training dataset based on real UWB measurements collected in a large indoor environment. Extensive experiments were conducted to evaluate different machine learning models, and our results show that machine learning can improve the 90th percentile AoA error from about 60° to 11° and thus improve the average direction estimation accuracy to 96.85%.**

***Index Terms*—UWB, indoor positioning, AoA**

I. Introduction

With the rise of indoor location-based services, a growing demand for more accurate indoor positioning has been observed [1]. GPS is known to be widely used for outdoor navigation, but its performance indoors is often poor and does not meet the accuracy requirements of indoor navigation [2]. Despite the wide range of alternative technologies for indoor positioning, e.g., BLE (Bluetooth Low Energy), WiFi, VLC (Visible Light Communication), ultrasound, infrared, RFID (Radio Frequency Identification), it is still an open research area due to the individual limitations of these technologies, such as inability to penetrate walls, high susceptibility to interference, high cost, high energy consumption and short range [3]. Among these technologies, UWB (Ultrawideband) can potentially tackle existing issues in indoor positioning systems for the following reasons:

- Highly accurate ranging: unlike other radiofrequency-based technologies, like BLE, which uses RSSI [3], UWB estimates the distance between two devices using time-based ranging, which has centimetre-level accuracy [4].
- AoA (Angle of Arrival) support: multi-antenna UWB sensors can estimate the angle of signal transmission relative to the device's orientation [4]. This can reduce the number of anchors in range required for positioning.
- Increasing integration in consumer devices: many indoor positioning technologies require positioning targets to carry custom tags, but UWB is being embedded in smartphones and other consumer devices, making them more available to the general public, so dependence on custom equipment can be reduced.
- Multipath fading resistance [5]: UWB has been shown to be resistant to interference, meaning it can produce accurate measurements even in NLOS (Non-Line-of-Sight) conditions.

Because the integration of UWB into smartphones started a few years ago, research on UWB performance in smartphones for indoor positioning has been limited. Some studies [4], [6], [7] have evaluated the ranging accuracy of UWB-equipped smartphones, but studies on AoA accuracy are lacking. One of the major limitations of UWB for indoor positioning has been the high cost of UWB sensors. However, in recent years, more affordable versions have been released. Another disadvantage was that they were not compatible with UWB-equipped smartphones, but UWB manufacturers have adapted to this change. For example, Qorvo recently released the DW3XXX UWB chips, some of which can support AoA estimation and ranging with iPhones' U1 chips, but UWB performance in these devices has not been studied extensively. In our previous paper [8], we studied distance and AoA error between DWM3001CDK sensors by Qorvo and UWB-equipped iPhones. We found that while distance measurements had decimetre-level accuracy, AoA measurements between these devices while in motion were highly noisy and inaccurate. This is an issue because AoA measurements can help reduce the number of anchors required to estimate the position of a target and enable relative positioning, e.g., determining whether an anchor is to the right, left or in the front. To the best of our knowledge, solutions to this issue have not been investigated. To contribute to this research gap, this paper proposes a novel collaborative method to increase positioning coverage based on our positioning vector framework in [8] (estimating directions to anchors out of range) and explores the efficacy of different machine learning models for improving AoA accuracy, i.e., random forest, XGBoost and a neural network. Previous works [9]–[11] have proposed solutions to improving AoA error in UWB devices, but most of them require an external computer to collect raw UWB data such as CIR (Channel Impulse Response) [12], which is inconvenient for users carrying smartphones. Therefore, our solution denoises AoA measurements directly rather than estimating them using raw UWB data. Overall, the contributions of this paper can be summarised as follows:

- *A novel collaborative approach to estimate directions to anchors out of range*. Many positioning methods such as trilateration require the presence of at least three anchors in range for positioning, which may be hard to achieve if the indoor environment is large or has poor line-of-sight conditions. Our new method based on the positioning vector framework allows the use of an AoA measurement from just one anchor to estimate the relative directions of all other anchors.
- *A dataset of distance and relative angle measurements between UWB-equipped iPhones in motion and the DWM3001CDK module in a real indoor environment*. To the best of our knowledge, this paper is the first to compile a dataset of UWB measurements between these devices. Unlike many previous papers, whose datasets were collected in individual small rooms, the measurements in this paper were collected in a $621.5\,\mathrm{m}^2$ area.
- *Machine-learning-based solutions for AoA estimation improvement in UWB-equipped smartphones*. The dataset collected during the experiments was used to train three machine learning models to improve AoA estimation error in iPhones: random forest, XGBoost and a neural network. Extensive experimental results on the performance of these models are presented.

The rest of the paper is organised as follows. Section II examines previous works related to this paper. Section III describes the dataset design and the proposed methods for AoA error improvement. Section IV presents the experimental results of the direction-finding approaches in section III. Finally, section V provides concluding remarks and directions for future work.

II. Related Work

This section will discuss recent papers related to our work. It is divided into two parts: machine-learning-based (ML-based) methods and non-machine-learning-based (non-ML-based) methods.

A. UWB Background

According to [3], UWB is a radiofrequency-based wireless communication technology that operates in the wide 3.1–10.6 GHz range of the electromagnetic spectrum, hence the name.

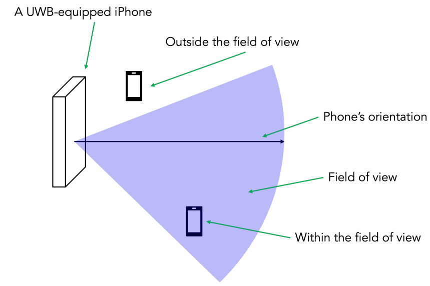

Fig. 1. An illustration of the field of view in UWB devices.

Similar to other radiofrequency-based communication technologies, such as BLE, it supports ranging (estimating distances between two devices with a peer-to-peer connection) and AoA estimation. However, unlike other technologies, UWB's ranging is based on time-based ranging methods, such as TWR (Two-Way Ranging), which has centimetre-level accuracy, making UWB a promising technology for indoor positioning. UWB can be employed to estimate the distance between devices regardless of their orientation, i.e., they do not have to face each other, but AoA estimation is strongly dependent on the devices' orientations. This is due to the fact that UWB devices have a limited field of view, meaning AoA estimation is either not possible or very noisy if one of the devices is more than or close to the maximum AoA degrees away from the other device's orientation [9], as is illustrated in Fig. 1. For example, the iPhone's field of view is [$-90°$, $90°$] when estimating AoA to a DW3110 chip [13]. Another major problem with AoA estimation using UWB is phase ambiguity [10], meaning that AoA estimates can wrap around, making it hard to determine if the other device is in front or behind. Therefore, researchers proposed several solutions to these issues, which are discussed next.

B. ML-based Methods

Although UWB is characterised by resistance to NLOS conditions, its signal can still be affected by severe interference. To address this issue, many papers on UWB-based positioning focused on the use of CIR for NLOS identification. According to [9], CIR contains detailed information on the UWB signal quality and environmental interference, so it can be used for noise detection. For example, [14] collected CIR from four DW1000 anchors and trained a deep CNN (Convolutional Neural Network) to generate AoA estimates. The study compared the machine learning approach to two rule-based AoA estimation methods, i.e., PDoA and MUSIC, and found that the ML approach outperformed the traditional algorithms and achieved an error of $2.8°$ in noisy conditions. In another study [9], the authors trained a neural network and a XGBoost model to accurately estimate distances and AoA values based on CIR from the DW3220 IC (Integrated Circuit) by Qorvo, which supports TWR, TDoA (Time Difference of Arrival) and PDoA (Phase Difference of Arrival) because of the UWB chip's dual-antenna design. They compiled a dataset by collecting about 20,000 measurements in a $8.6 \times 7.7\,\mathrm{m}^2$

area. Their empirical results showed that the neural network and the XGBoost model achieved a $19°$ and a $43.7°$ error on the test set, respectively. In addition, a $6.25\,\mathrm{cm}$ and $2.25\,\mathrm{cm}$ distance error was obtained by the respective models. While these results are promising, the test area where measurements were collected was rather small, and for this approach to work in other areas, new custom models would have to be trained based on datasets collected in those areas, which takes significant time and effort. One of the solutions to this issue is transfer learning [11], [15], i.e., fine-tuning a large model that was already trained on a wide variety of samples. For example, the authors of [11] developed such a model by training it on samples collected in both LOS (Line-of-Sight) and NLOS conditions in different environments, including a complex industrial setting. Their generalisation study demonstrated that the transfer learning solution could reduce AoA error to under $5°$ in unseen environments.

It is evident that CIR is an effective solution to improving distance and AoA estimation using UWB. However, access to CIR data usually requires the devices to be connected to a computer, which may not be possible or convenient. The authors of [16] proposed a solution to this issue by employing UWB distance estimates instead of CIR for training, which did not require the anchors to be attached to a computer as the distance measurements were directly exchanged between anchors and targets. Specifically, the authors combined a Kalman filter and a LSTM (Long-Short-Term Memory) on UWB distance measurements to denoise them. The authors conducted a simulation and real experiments in a $64\,\mathrm{m}^2$ indoor area to compare their approach to a backpropagation network with and without a Kalman filter. They found that the combination of a Kalman filter and a LSTM yielded the best performance and achieved an average error of $16.8\,\mathrm{cm}$. However, the experimental area was quite small, so it is unclear how their approach would work in a larger area, e.g., a mall or an airport.

C. Non-ML-Based Methods

While ML-based methods can considerably improve UWB measurements, they require substantial effort in the form of dataset collection and model maintenance due to changes in indoor environments. For applications where these requirements cannot be met, other solutions are needed. One such system is PnPLoc [17], which utilises a custom TDoA protocol where only anchors engage in TWR, while targets determine their own locations by eavesdropping on packets exchanged between the anchors. The system achieved an error of $28.9\,\mathrm{cm}$ in a $695\,\mathrm{m}^2$ area. One of the major strengths of this system is that no AoA measurements are required for positioning. That said, both anchors and targets are required to be attached to microcontrollers, which may be costly for large-scale applications with many users, e.g., in malls. In addition, at least three anchors are required for 2-D TDoA-based positioning [18]. AoA measurements can be used to address this problem as they can help reduce the number of anchors required, but they can be highly noisy [19].

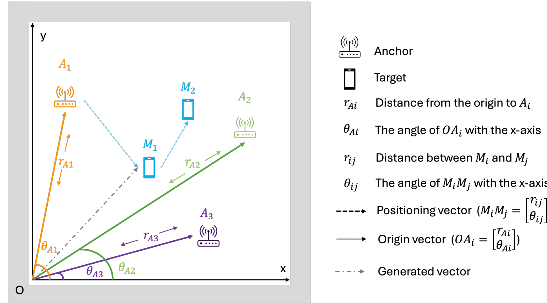

Fig. 2. An illustration of the problem setup.

[20] proposed a solution to this issue by combining a least-squares-based method with an extended Kalman filter for AoA estimation using UWB, i.e., the authors leveraged past AoA measurements to denoise current estimates. Their experimental results showed that their solution was able to accurately track the AoA measurements. However, their system also required access to a microcontroller.

III. Methodology

This section will briefly review the positioning framework proposed in our previous papers [8], [19]. The use of the framework as well as machine learning for collaborative direction finding to improve positioning coverage and AoA error will also be discussed.

A. Problem Setup

Suppose there is an indoor space S with a global coordinate system defined in advance. Let the origin of this system be denoted as O. N_A anchors are installed in the indoor space at pre-defined fixed positions, which are recorded in a database. Let an anchor i be denoted as A_i. Each anchor uniquely identifies a place of interest, e.g., a specific shop at a mall, a restaurant, etc. There are also N targets, each target i denoted as M_i, which can move and needs to calculate how to navigate to the places of interest. Fig. 2 illustrates an example of the problem setup. Each target carries a UWB-equipped smartphone that can estimate the AoA and distance to anchors in its range. Instead of using Cartesian coordinates, the position of each device, i.e., an anchor or a target, is defined as a vector with respect to the pre-defined global coordinate system. For example, the position of anchor A_1 is a positioning vector OA_1, which is formed by the distance between the origin and A_1 (r_{A1}) and the angle OA_1 forms with the x-axis (θ_{A1}). The vectors connecting the origin to a device are referred to as origin vectors. There are also positioning vectors between anchors and targets, e.g., M_1M_2 in Fig. 2, which is defined by the distance between M_1 and M_2 (r_{12}) and the angle M_1M_2 forms with the x-axis (θ_{12}). The figure shows that not all devices are connected with vectors. The absence of vectors represents a device being out of the field of view of another device, which was discussed in section II-A. These positioning vectors can be employed to generate the missing vectors. For example, as shown in the figure, if OA_1 and A_1M_1 are added, OM_1 can be generated.

B. Positioning Vectors

As discussed in the previous section, a positioning vector is defined by the distance and angle between two devices. Our framework outlines six types of positioning vectors:

1) Measured vectors – positioning vectors defined by distances and angles measured with UWB. A measured vector M_iM_j is denoted as $M_iM_j^{(m)}$.
2) Actual vectors – positioning vectors defined by ground truth distances and angles. An actual vector M_iM_j is denoted as $M_iM_j^{(a)}$.
3) Intermediate generated vectors of degree K – positioning vectors that are calculated through K intermediate terminals. For example, if M_1M_2 and M_2M_3 are available but M_1M_3 is not, M_1M_3 can be generated through M_2 by adding M_1M_2 and M_2M_3. This is an example of a degree 1 intermediate generated vector because only one intermediate terminal – M_2 – was used. In this case, the intermediate generated vector M_1M_3 is denoted as $M_1M_3^{(g_{1,2,3})}$.
4) Generated vectors of degree K – positioning vectors that are calculated as the average of all possible intermediate generated vectors of degree K for a given vector. A generated vector M_iM_j is denoted as $M_iM_j^{(g)}$. For example, for M_1M_3, if there are only four mobile terminals and $M_1M_3^{(g_{1,2,3})}$ and $M_1M_3^{(g_{1,4,3})}$ can be calculated, then $M_1M_3^{(g)} = \frac{1}{2}(M_1M_3^{(g_{1,2,3})} + M_1M_3^{(g_{1,4,3})})$.
5) Static positioning vectors – positioning vectors that are not parameterised by time and are defined for fixed (static) targets.
6) Dynamic positioning vectors - positioning vectors that are parameterised by time and are defined for mobile (dynamic) targets. A dynamic positioning vector M_iM_j of type l (e.g., (g) for generated) recorded at time t is denoted as $M_iM_j^{(l,t)}$.

[19] focused on static vectors, while [8] served as an introduction to dynamic positioning vectors. Specifically, the latter evaluated the distance and relative angle error in mobile UWB-equipped iPhones and DWM30001CDK sensors. We found that while the distance estimation accuracy between iPhones and DWM3001CDK sensors was on decimetre-level, the AoA accuracy was much lower compared to that between iPhones. This could be explained by the fact that although iPhones and DWM3001CDK sensors are compatible, they do not use the same UWB chip.

C. Vector Generation for Direction Finding

As discussed in section II-A, the field of view in UWB-equipped devices is limited relative to device orientation. Let the orientation of a mobile terminal M_i be denoted as a vector ϕ_i, which is defined by the maximum ranging distance of M_i (e.g., according to our prior experiments, the maximum distance an iPhone can measure with UWB is about $40\,\mathrm{m}$) and the angle between the orientation of M_i and the x-axis of the indoor space. iPhones can also measure the angle between their orientation vector and the vectors formed with other devices. For example, if M_i can detect M_j, then the angle between ϕ_i and M_iM_j is referred to as the relative angle of M_j to M_i and is denoted as ${\theta_{ij}}'$. If M_j is to the left of M_i, ${\theta_{ij}}' < 0$, if M_j is to the right of M_i, ${\theta_{ij}}' > 0$ and otherwise, M_j and M_i are facing each other. Please refer to [19] for an illustration. The maximum relative angle that a UWB device can measure is also limited, so let it be denoted as β_i (for M_i). Because of the limited field of view, UWB devices cannot measure relative angles to devices behind them, and relative angle error becomes much higher as $|{\theta'_{ij}}^{(a)}|$ becomes closer to β_i [9], making direction finding challenging. One of the simplest solutions to this problem is trilateration, which requires that distances to at least three anchors are available to calculate the target's position. However, because iPhones' field of view is limited, if the anchor arrangement is too sparse, trilateration becomes challenging. This section will describe how generated vectors (discussed in section III-B) can be employed to estimate the relative angles of other anchors in the area using only one anchor's measurements:

1) Suppose M_i can estimate the relative angle (${\theta'_{iAj}}^{(m)}$) and distance ($r_{iAj}^{(m)}$) to an anchor A_j. As discussed in our previous paper [19], the absolute angle θ_{ij} of a positioning vector M_iM_j can be calculated as follows:
$$\theta_{ij} = 90^\circ - \gamma_i - {\theta'}_{ij}, \tag{1}$$
where γ_i is the bearing of M_i, i.e., the angle between ϕ_i and the y-axis of the indoor space in the clockwise direction, and ${\theta'}_{ij}$ is the relative angle to M_j from M_i's perspective. Please note that γ_i is assumed to have zero error, but it can be estimated with a high degree of accuracy using a method proposed in [21]. Thus, $M_iA_j^{(m)}$ can be computed as $\begin{bmatrix} r_{iAj}^{(m)} & 90^\circ - \gamma_i - {\theta'_{iAj}}^{(m)} \end{bmatrix}^T$.
2) Then M_i's origin vector can be generated as $OM_i^{(g)} = OA_j^{(a)} - M_iA_j^{(m)}$.
3) Next, to find directions to each other anchor $A_k \neq A_j$, $M_iA_k^{(g)}$ needs to be generated. This can be achieved using $OM_i^{(g)}$ as $M_iA_k^{(g)} = OA_k^{(a)} - OM_i^{(g)}$.
4) Lastly, the relative angle to A_k can be computed by finding the angle between ϕ_i and $M_iA_k^{(g)}$:
$$\begin{aligned} {\theta'_{ik}}^{(g)} = &\, sign(\phi_i \cdot rot90(M_iA_k^{(g)}))\times \\ &\cos^{-1}\left(\frac{M_iA_k^{(g)} \cdot \phi_i}{||M_iA_k^{(g)}||_2 \times ||\phi_i||_2}\right), \end{aligned} \tag{2}$$
where $sign(x)$ is a function that returns -1 if its input x is negative and returns 1 otherwise, $rot90(v)$ is a function that rotates its input vector v by 90° anticlockwise, i.e., $rot90(M_iA_k^{(g)}) = \begin{bmatrix} r_{iAk}^{(g)} & 90^\circ + \theta_{iAk}^{(g)} \end{bmatrix}$, $\cdot$ refers to the dot product operator, $\phi_i \cdot rot90(M_iA_k^{(g)})$ determines whether A_k is to the right of M_i's orientation (> 0) or to the left (< 0), and $||\cdot||_2$ refers to vector magnitude.

Thus, generated vectors can be used to reduce the number of anchors required to find directions to other anchors and find the target's position ($OM_i^{(g)}$). However, the accuracy of this method strongly depends on the accuracy of relative angle estimation. According to our experiments with iPhones in static and mobile settings [8], AoA error is much higher when ranging with DWM3001CDK sensors than when ranging with iPhones. Therefore, we propose the use of machine learning for AoA error reduction, as described next.

D. Machine Learning Model Design for AoA Error Reduction

While a significant amount of research has been conducted on improving AoA estimation accuracy using machine learning, to the best of our knowledge, the direct use of AoA measurements for this purpose has not been explored. Thus, this paper is among the first to propose a machine learning solution to tackle the issue of high AoA error between iPhones and DWM3001CDK sensors based on AoA measurements. To achieve this, we installed six DWM3001CDK sensors as anchors in a $621.5\,\mathrm{m}^2$ space and collected distance and relative angle measurements on two UWB-equipped iPhone 11s acting as targets for 50 minutes. Then, a training dataset was created using a sliding window of length WS and a stride of size ST, and different window sizes and strides were tested. For example, suppose $WS = 10$, $ST = 5$, and the following input sequence of distances and relative angles was collected by M_i towards anchor A_j over 102 seconds:

$$\begin{bmatrix} r_{iAj}^{(m,0)} & r_{iAj}^{(m,1)} & \cdots & r_{iAj}^{(m,100)} & r_{iAj}^{(m,101)} \\ \theta'^{(m,0)}_{iAj} & \theta'^{(m,1)}_{iAj} & \cdots & \theta'^{(m,100)}_{iAj} & \theta'^{(m,101)}_{iAj} \end{bmatrix} \tag{3}$$

For simplicity, suppose one relative angle was measured per second from the same anchor A_j. Then the input X and output y for the training dataset were generated using the following steps:

1) First, nil relative angles were removed. For simplicity, suppose in this example, there are no nil measurements.
2) Secondly, the sequence was filtered by applying the stride, and the amount of time between the previous and current measurement was added (in this case, because of the stride, 5 seconds should be added to each item in the sequence (except for the first measurement)):

$$\begin{bmatrix} r_{iAj}^{(m,0)} & r_{iAj}^{(m,5)} & \cdots & r_{iAj}^{(m,100)} \\ \theta'^{(m,0)}_{iAj} & \theta'^{(m,5)}_{iAj} & \cdots & \theta'^{(m,100)}_{iAj} \\ 0 & 5 & \cdots & 5 \end{bmatrix} \tag{4}$$

The stride is necessary because measurements from DWM3001CDK are collected about 60 times a second [8], so a stride was applied to reduce repeated measurements. As for the amount of time between measurements, its purpose was to make sure the model would learn that a longer time interval resulted in a larger change in the next measurement.
3) Let the set of measurements (distance, relative angle and time difference) collected at time t be denoted as $x^{(t)}$, i.e., $x^{(t)} = (r_{iAj}^{(m,t)}, \theta'^{(m,t)}_{iAj}, \Delta t)$, where $\Delta t = 5$ for all $t > 0$ and $\Delta t = 0$ for $t = 0$. Then, a sliding window was applied to the sequence to produce new input sequences for training (X) as follows:

$$X = \begin{bmatrix} x^{(0)} & x^{(5)} & \cdots & x^{(45)} \\ x^{(5)} & x^{(10)} & \cdots & x^{(50)} \\ & \vdots & & \\ x^{(55)} & x^{(60)} & \cdots & x^{(100)} \end{bmatrix} \tag{5}$$

In the experiments, two combinations of features were attempted: (1) including the measured distance in $r_{iAj}^{(m,t)}$ in $x^{(t)}$, i.e., $x^{(t)} = (r_{iAj}^{(m,t)}, \theta'^{(m,t)}_{iAj}, \Delta t)$, and (2) not including the distance, i.e., $x^{(t)} = (\theta'^{(m,t)}_{iAj}, \Delta t)$. The purpose of this choice was to observe whether including the distances in the training dataset was beneficial.
4) The task was to predict the next relative angle correctly, i.e., the expected output for these input sequences (y) would be constructed as follows:

$$y = \begin{bmatrix} \theta'^{(a,46)}_{iAj} & \theta'^{(a,51)}_{iAj} & \cdots & \theta'^{(a,101)}_{iAj} \end{bmatrix}^T \tag{6}$$

For example, the first element in y is $\theta'^{(a,46)}_{iAj}$ because the last reading in the first sequence in X was $x^{(45)}$, so the goal is to predict the next relative angle after the last angle in the input sequence. Note that each sequence corresponds to a single anchor A_j, i.e., although one phone could receive measurements from many anchors simultaneously, the sequences were generated for each anchor individually. This way, because the anchors were placed in different environments, all anchors would be treated equally instead of informing the model about their locations, bringing diversity to the AoA sequences. This procedure would be repeated for each anchor from which a target received measurements.

Thus, the machine learning task at hand was to minimise the difference between the predicted and ground truth relative angles for each target M_i ($i \in [1, ..., N]$) and each anchor A_j ($j \in [1, ..., N_A]$), so the MSE (Mean Squared Error) loss was used for training. Specifically, the loss would be calculated as follows for each timestamp t and then averaged over all timestamps in the training set:

$$\frac{1}{N}\frac{1}{N_A}\sum_{i=1}^{N}\sum_{j=1}^{N_A}(\theta'^{(a,t)}_{iAj} - \theta'^{(predicted,t)}_{iAj})^2 \tag{7}$$

Although each input sequence was generated for a single anchor and a single target, the loss was averaged for all anchors and targets because they were treated equally, meaning each anchor-target pair represents a unique sample, so the loss represents averaging over all samples.

The dataset was used to train three machine learning models, as advised by previous papers, e.g., [9], and their descriptions can be found in Table I. Different values of hyperparameters specified in the table were tested, and the best hyperparameters for the proposed dataset are discussed in section IV.

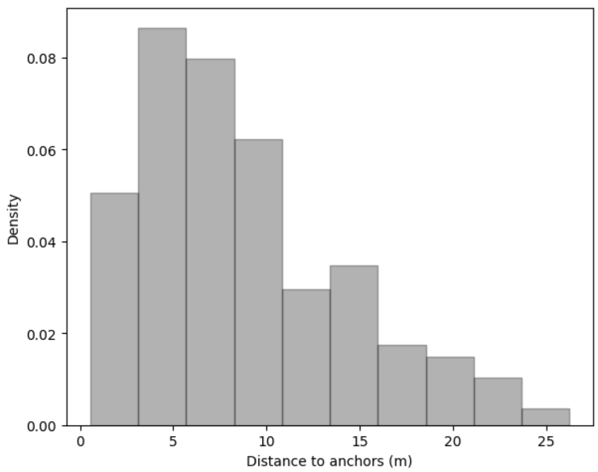

Fig. 3. Distribution of ground truth distances from the phones to anchors.

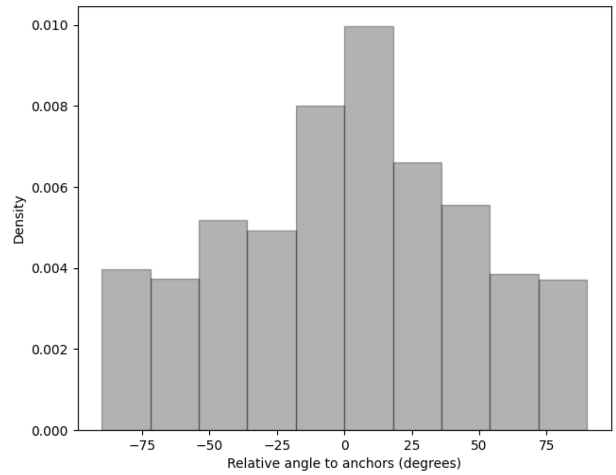

Fig. 4. Distribution of ground truth relative angles from the phones to anchors.

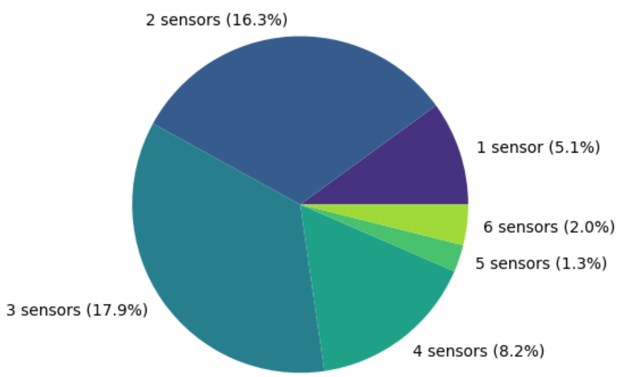

Fig. 5. Frequency of distance measurements by the number of sensors in range in any one second.

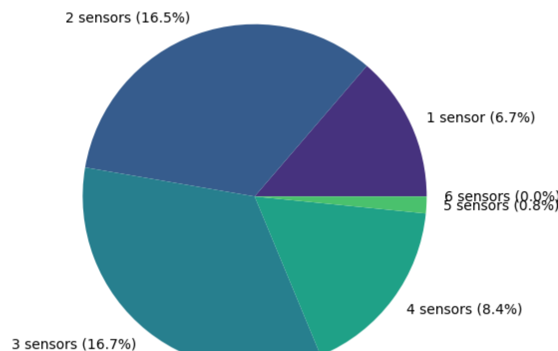

Fig. 6. Frequency of relative angle measurements by the number of sensors in range in any one second.

TABLE I
MACHINE LEARNING MODELS TRAINED TO REDUCE AOA ERROR AND THEIR HYPERPARAMETERS.

Model	Description	Hyperparameters
Random Forest Regressor (RF)	An ensemble of decision trees trained on random samples of the dataset, whose results are averaged to prevent overfitting.	Number of trees: 10, 50, 100.
Multi-Layer Perceptron network (MLP)	A neural network of densely connected layers.	Number of layers: 1, 2; number of hidden units: 25, 50, 100.
XGBoost (XGB) [22]	An ensemble of decision trees where each tree is added sequentially in order to correct the errors of the previous tree using gradient descent.	Number of parallel estimators: 1, 10, 50, 100.

E. Equipment and Tools

To collect measurements for the dataset, two UWB-equipped iPhone 11s were used as targets and six DWM3001CDK sensors by Qorvo were used as anchors. DWM3001CDK is an integrated board with the DW3110 UWB transceiver and nRF52833 with Bluetooth 5.2 [13]. The module has a ranging accuracy of $10\,\mathrm{cm}$ [23]. Unlike other chips in the recently released DW3XXX family, DW3110 does not support PDoA, meaning it cannot estimate relative angles to other devices. However, it supports TWR and TDoA [13]. iPhones use the U1 UWB chip, which is compatible with the DW3110 UWB transceiver used by the DWM3001CDK module. U1 supports both ranging and AoA estimation. Both U1 and DW3110 rely on the HRP (High-Rate Pulse) repetition frequency mode [4], [24]. The Qorvo Nearby Interaction app was used to collect measurements [25]. The app was modified so that UWB sessions would be terminated if the ground truth absolute relative angle exceeded $90°$ to avoid phase ambiguity issues.

IV. EXPERIMENTAL RESULTS

This section will describe the experimental procedures and AoA accuracy achieved by the proposed machine learning solutions.

A. Experimental Setup

Six DWM3001CDK sensors were placed in tripods at the same height of $150\,\mathrm{cm}$ in a $621.5\,\mathrm{m}^2$ area, which is located on the fourth floor of the Jockey Club Innovation Tower on the Hong Kong Polytechnic University campus. They acted as anchors in the experiments, meaning their positions were fixed. The anchors' ground truth coordinates were recorded using the Velodyne VLP-16 with a 9-axis IMU. Two iPhone 11s were used as targets and were attached to a trolley at the same height. They were carried on the trolley around the area in random paths at a walking pace (at a median speed of $60\,\mathrm{cm/s}$) for 50 minutes, and the phones collected measurements simultaneously. We found that at most one phone could establish a UWB session with one DWM3001CDK sensor, so the phones were never connected to the same sensor at the same time. An example of a ground truth path along which the phones were carried is illustrated in Fig. 7. The red points represent the locations of the anchors, while the arrows on the red line illustrate the orientations of a phone.

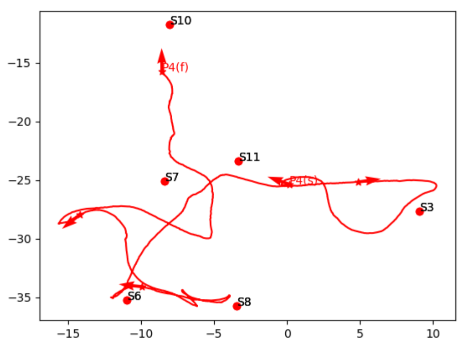

Fig. 7. An example of a ground truth path along which measurements were collected.

B. Dataset Analysis

This section will describe the characteristics of the dataset collected during the experiments. Fig. 3 demonstrates the distribution of ground truth distances between a target and an anchor. It shows that the distances ranged between $0.54\,\mathrm{m}$ and $26.2\,\mathrm{m}$, and the majority were between 5 and $10\,\mathrm{m}$. Similarly, Fig. 4 illustrates the distribution of ground truth relative angles, which ranged between $-90°$ and $90°$ and were mostly between $-50°$ and $50°$. Relative angles outside of this range were not explored due to phase ambiguity but could be part of future research. Overall, 100,022 filtered records were collected during the experiments, i.e., with non-nil distances and relative angles. Then these measurements were converted into training sequences with three different window sizes (10, 50, 100) and strides (1, 5, 10), as described in section III-D. Thus, dataset sizes were different for each combination of the window size and stride, which are listed in Table II.

Fig. 5 and Fig. 6 depict the number of anchors to which an iPhone could estimate the distance and relative angle to in any one second, respectively. It can be observed that there were more distance measurements than relative angle estimates, which makes sense because relative angle estimation is highly dependent on the phone's orientation. Fig. 5 also shows that trilateration would be possible the majority of the time because the targets were connected to at least 3 sensors. However, in areas with a sparse anchor arrangement, direction finding with one or two anchors in range could be useful. For reference, in our experiments, there was roughly 1 sensor per $100\,\mathrm{m}^2$ (6 sensors in a $621.5\,\mathrm{m}^2$ area).

TABLE II
DATASET SIZES FOR DIFFERENT WINDOW SIZES (WS) AND STRIDES (ST) USED FOR TRAINING THE MACHINE LEARNING MODELS.

	WS = 10	**WS = 50**	**WS = 100**
ST = 1	99676	98422	96822
ST = 5	19789	18517	17047
ST = 10	9806	8597	7362

C. Machine-Learning-Based AoA Estimation

1) Relative Angle Estimation: In order to find the optimal window size and stride for this dataset, the models listed in Table I were trained on every combination of $WS \in [10, 50, 100]$ and $ST \in [1, 5, 10]$. After shuffling, 80% of the dataset was used for training, and 20% was used for testing. The descriptive statistics of the relative angle prediction error of the RF, XGB and MLP models for different combinations of WS and ST as well as model and dataset hyperparameters can be found in Table III. The "distance" column in the table indicates whether measured distances were used in the input, as described in section III-D, the "parameters" column refers to the number of estimators for RF and XGB and the number of hidden units and layers for MLP (e.g., (50, 50) indicates two layers with 50 hidden units in each). The last four columns list the average angle error, 75th and 90th percentiles of the relative angle error and maximum error. The table is sorted in the ascending order of maximum predicted relative angle error and only lists the top three results for each model. The measured error serves as the baseline, i.e., the absolute difference between each measured and ground truth relative angle in the test set. Please note that the dataset was shuffled for each model combination to minimise bias towards a specific set of test data, hence the measured errors are slightly different in the table. The table indicates that XGB with 50 estimators achieved the lowest maximum error with $WS = 100$ and $ST = 5$ and when measured distances were included in the input, improving the 90th percentile measured angle error of $56.43°$ to $11.07°$. In addition, the maximum relative angle error was reduced from $191.89°$ to $58.98°$. The second best model in terms of maximum error was RF with 50 estimators, $WS = 100$ and $ST = 1$. Although its maximum error was slightly higher than that of XGB, its 90th percentile error was lower – $3.02°$. This means XGB was better at reducing the range of relative angle error at the expense of slightly worse error on average. As for MLP, its maximum error was slightly higher than the other models,

TABLE III

RELATIVE ANGLE ERROR OF THE TOP THREE PARAMETER COMBINATIONS FOR THE THREE MACHINE LEARNING MODELS COMPARED TO MEASURED RELATIVE ANGLE ERROR, SORTED BY THE MAXIMUM PREDICTED RELATIVE ANGLE ERROR IN ASCENDING ORDER FOR EACH MODEL.

Model	Parameters	WS	ST	Distance	Type	Mean (°)	75% (°)	90% (°)	Max (°)
RF	50	100	1	yes	Measured error	28.58	38.87	61.04	192.28
					ML error	1.25	1.22	3.02	64.89
	100	100	10	yes	Measured error	28.9	38.86	59.89	189.57
					ML error	4.61	4.94	10.69	70.13
	100	100	1	yes	Measured error	28.44	38.22	60.48	193.16
					ML error	1.21	1.19	2.88	71.66
MLP	(50, 50)	100	10	no	Measured error	29.12	38.73	62.43	193.89
					ML error	13.3	18.32	28.39	76.91
	(50, 50)	100	10	yes	Measured error	27.29	36.32	57.48	183.7
					ML error	8.46	11.12	18.64	90.17
	(25, 25)	100	10	no	Measured error	27.86	36.88	57.92	186.56
					ML error	14.97	21.1	33.01	91.94
XGB	50	100	5	yes	Measured error	27.98	37.23	56.43	193.89
					ML error	5.09	6.76	11.07	58.98
	50	100	10	yes	Measured error	27.68	36.76	54.75	197
					ML error	5.13	6.32	10.7	78.11
	1	10	10	yes	Measured error	28.79	37.14	60.23	187.06
					ML error	29.16	45.69	61.06	84.99

with 50 hidden units, 2 layers, $WS = 100, ST = 10$ and no distances yielding the best result of about 77°. The next two best results were both achieved by RF, with 100 estimators, $WS = 100$ and $ST = 1$ having very similar results to the RF model with 50 estimators, meaning more estimators did not result in a significant improvement. It can also be observed that $WS = 100$ resulted in the best performance in the majority of cases, meaning that longer sequences were helpful in predicting the ground truth angles. Thus, in general, it seems having a larger window size is more beneficial for model performance. In addition, the table shows that including measured distances in the input is shared by most of the top models, meaning it was a helpful feature.

2) Direction Finding: As discussed in section IV-C1, machine learning can effectively reduce AoA error using sequences of AoA measurements, which can be collected by the smartphone directly. However, it is unlikely that people carrying smartphones would need a precise relative angle to navigate to places of interest, so direction categories, such as to the left, to the right, etc., would be more practical. In addition, the AoA measurement error between iPhones and DWM3001CDK is inherently high, but it can be reduced if the relative angles are translated into directions instead. Another issue is that the machine learning models only predict the relative angles to one anchor at a time, so in this section, equation 2 is employed to generate relative angles to all other anchors using the relative angle to one anchor (source anchor). To allow for the high variance in AoA measurement error, three direction categories were created as follows:

- Left: $\theta'^{(m)} \in (-90°, -10°)$
- Right: $\theta'^{(m)} \in (10°, 90°)$
- Front: $\theta'^{(m)} \in [-10°, 10°]$

This section will outline the results on direction finding using the machine learning predictions and compare them to those generated from measured relative angles.

The directions were calculated for each AoA measurement. Direction accuracy was calculated by counting the number of observations where a given relative angle's direction coincided with the ground truth direction and dividing this number by the total number of observations. Table IV delineates the accuracy of directions produced from the following: (1) measured relative angles of the source anchors, which serve as the baseline; (2) relative angles to all anchors, generated from the measured angles, as in equation 2; (3) relative angles of the source anchor, which were enhanced by machine learning; (4) relative angles to all anchors, generated from the angles enhanced by machine learning, as in equation 2. The table lists the top three models in terms of direction accuracy for each model and each direction, i.e., front, left and right, as well as the average across all directions. It is sorted by the accuracy of directions generated for all anchors based on the relative angles enhanced by machine learning, in descending order. The meaning of the "distance" and "parameters" columns are the same as in Table III. Please note that these results were produced based on the relative angle predictions described in Table III. The table includes direction accuracy for the source anchors and other anchors separately to observe the effect of vector generation. It demonstrates that the direction accuracy of measured source angles was about 42% (for the front direction) and 69% (for the left and right directions). The lower accuracy for the front direction can be explained by the fact that the range of relative angles for the front (10° – (−10°) = 20°) was four times lower than the range for left and right directions (90° – 10° = 80°). Applying vector generation to these angles yielded a slightly better accuracy of about 64% to 86% for all anchors. As for machine learning results, although XGB achieved the lowest AoA error for individual anchors, as shown in Table III, RF was the most accurate in predicting all directions, as shown in Table IV. Overall, the table shows that unlike using generated vectors alone, with machine learning, all three directions could be predicted with almost 100% accuracy, although the accuracy for the front

TABLE IV

THE ACCURACY OF DIRECTIONS DERIVED FROM MEASURED RELATIVE ANGLES, THOSE IMPROVED WITH TOP THREE MACHINE LEARNING MODELS AND GENERATED VECTORS, SORTED BY THE ACCURACY OF PREDICTED DIRECTIONS GENERATED FOR ALL ANCHORS (IN DESCENDING ORDER).

Direction	Model	Parameters	WS	ST	Distance	Measured vector accuracy (source anchor, in %)	ML accuracy (source anchor, in %)	Generated vector accuracy (all anchors, in %)	ML accuracy (generated vectors, all anchors, in %)
Average	MLP	(100, 100)	100	10	yes	55.62	83.72	76.24	92.21
		(50, 50)	100	10	yes	58.5	82.17	77.8	91.11
		(100, 100)	100	5	yes	53.12	80.03	77.85	90.39
	RF	100	100	1	yes	55.42	97.81	75.23	**96.94**
		50	100	1	yes	55.31	97.88	75.78	96.85
		100	50	1	yes	54.6	97.01	75.65	96.65
	XGB	50	100	10	yes	56.56	88.66	77.29	94.26
		100	100	5	yes	54.59	91.53	76.95	94.05
		100	50	10	yes	54.55	86.79	76.17	93.99
Front	MLP	(100, 100)	100	10	yes	45.97	72.58	68.42	87.72
		(50, 50)	100	10	yes	48.95	65.03	71.88	84.66
		(100, 100)	100	5	yes	39.08	62.77	68.97	82.21
	RF	100	100	1	yes	43.47	95.79	64.25	**95.39**
		50	100	1	yes	43.61	96.11	64.68	95.07
		50	50	1	yes	42.33	94.65	63.79	95.01
	XGB	50	100	10	yes	43.07	75.18	71.53	89.93
		100	100	5	yes	42.28	83.89	68.11	89.16
		100	100	10	yes	44.78	84.33	65.52	88.97
Left	MLP	(100, 100)	100	10	yes	52.51	92.31	75.85	95.78
		(100, 100)	100	5	yes	54.5	92.21	78.99	95.62
		(50,)	100	1	yes	54.8	79.62	77.18	94.3
	RF	10	50	1	yes	54.88	98.49	77.17	**97.53**
		100	100	1	yes	54.5	98.95	76.23	97.47
		100	50	1	yes	54.01	98.3	77.27	97.47
	XGB	100	50	10	yes	53.31	94.2	76.49	97.13
		100	100	1	yes	52.43	93.43	76.41	96.72
		50	100	10	yes	57.62	96.34	74.71	96.46
Right	MLP	(50, 50)	100	10	yes	71.28	91.35	84.83	95.05
		(100, 100)	100	10	no	71.53	84.34	85.96	94.22
		(100, 100)	100	5	yes	65.79	85.1	85.57	93.34
	RF	50	100	1	yes	68.11	98.74	85.66	**98.03**
		100	100	1	yes	68.27	98.68	85.22	97.97
		100	50	1	yes	68.66	97.97	85.32	97.63
	XGB	100	100	10	yes	61.49	92.91	84.69	96.94
		100	50	5	yes	67.05	91.41	85.83	96.64
		100	100	5	yes	66.22	95.88	84.78	96.59

was still slightly lower than the others (about 95%). It can be observed that among all the best combinations, ST was equal to 1, and measured distances were included in the input. In addition, the number of estimators was either 50 or 100 for RF and XGB, and WS was either 50 or 100. This suggests that in general, for better direction accuracy, the use of RF, larger window sizes, a larger number of estimators and a smaller stride are recommended. In terms of the average accuracy across all directions, the best outcome was achieved with $WS = 100, ST = 1$, RF with 100 estimators and measured distances included in the input. It also seems MLP showed the worst performance in direction accuracy compared to the other two models in general, meaning ensemble learning is better for this task. In addition, unlike the best parameter combination for MLP in Table III, for direction accuracy, having two layers with 100 hidden units was better.

3) Comparison with Existing Works: It is difficult to compare our experimental results to previous works because, to the best of our knowledge, no research has been done on improving AoA error between UWB-equipped iPhones and the DWM3001CDK module without CIR. However, we will still briefly present results achieved by related studies for comparison. For example, the authors of [9] achieved a $19°$ using a neural network trained on CIR data collected from DW3220 sensors, while the best model in Table III had a 90%-error of $3.02°$, even though the model had no access to CIR data and our training dataset was collected in a much larger area ($621.5\,\mathrm{m}^2$). [14] achieved a more similar result to ours - an error of $2.8°$ in noisy conditions using a CNN trained on CIR data from DW1000 sensors. The authors compared their results to two rule-based AoA estimation methods, i.e., PDoA and MUSIC, and found that the ML approach outperformed the traditional algorithms. This means that this study presents competitive results on improving AoA error in smartphone-based UWB positioning, where access to CIR is difficult.

V. CONCLUSION AND FUTURE WORK

In summary, UWB is a promising indoor positioning technology due to its high ranging accuracy and AoA support. AoA measurements can greatly reduce the number of anchors required in range for positioning and direction finding. However, our previous experiments showed that AoA estimates between UWB-equipped iPhones and the DWM3001CDK module are highly noisy, so this paper presented three solutions to this issue: a collaborative approach based on generated vectors in our positioning vector framework, a machine learning model that predicts the next ground truth AoA value based on prior noisy AoA sequences, and a hybrid of the two methods. We compiled a dataset of distance and AoA measurements between the DWM3001CDK sensors acting as anchors and two UWB-equipped iPhones acting as targets, trained three machine learning models on this dataset and compared their performance. It was found that machine learning combined with our generated-vector-based approach yielded the best results. In the future, it would be worth exploring the use of sequence models such as LSTMs or even transformers, but with a larger dataset. This could also potentially address the phase ambiguity issue and detect anchors in the back as well. In addition, future research could focus on transfer learning for AoA error reduction in different environments, UWB sensors and smartphones so that a universal solution to the issue is available. Lastly, while the purpose of this paper was primarily to improve the AoA error, it would be useful to finetune the models to improve positioning error, e.g., by incorporating both distance and relative angle error in the loss function.

Acknowledgment

This work is supported by the Hong Kong PhD Fellowship Scheme provided by the Research Grants Council (RGC) of Hong Kong.

References

[1] F. Furfari, A. Crivello, P. Baronti, P. Barsocchi, M. Girolami, F. Palumbo, D. Quezada-Gaibor, G. M. Mendoza Silva, and J. Torres-Sospedra, "Discovering location based services: A unified approach for heterogeneous indoor localization systems," *Internet of Things*, vol. 13, p. 100334, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660520301657

[2] J. Singh, N. Tyagi, S. Singh, F. Ali, and D. Kwak, "A systematic review of contemporary indoor positioning systems: Taxonomy, techniques, and algorithms," *IEEE Internet of Things Journal*, vol. 11, no. 21, pp. 34 717–34 733, 2024.

[3] Y. Sartayeva, H. C. Chan, Y. H. Ho, and P. H. Chong, "A survey of indoor positioning systems based on a six-layer model," *Computer Networks*, vol. 237, p. 110042, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128623004875

[4] A. Heinrich, S. Krollmann, F. Putz, and M. Hollick, "Smartphones with UWB: Evaluating the accuracy and reliability of UWB ranging," *arXiv preprint arXiv:2303.11220*, 2023.

[5] S. Park and K.-Y. Jung, "Novel compact UWB planar monopole antenna using a ribbon-shaped slot," *IEEE Access*, vol. 10, pp. 61 951–61 959, 2022.

[6] V. Di Pietra and P. Dabove, "Recent advances for UWB ranging from Android smartphone," in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2023, pp. 1226–1233.

[7] F. Zhang, Z. Chang, J. Xiong, J. Ma, J. Ni, W. Zhang, B. Jin, and D. Zhang, "Embracing consumer-level UWB-equipped devices for fine-grained wireless sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 4, Jan. 2023. [Online]. Available: https://doi.org/10.1145/3569487

[8] Y. Sartayeva and H. C. Chan, "Dynamic positioning vectors for collaborative UWB-based positioning," in *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2024, pp. 546–555.

[9] T. Margiani, S. Cortesi, M. Keller, C. Vogt, T. Polonelli, and M. Magno, "Angle of arrival and centimeter distance estimation on a smart UWB sensor node," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–10, 2023.

[10] Y. Li, H. Zhao, Y. Liu, T. Wang, J. Yu, and Y. Shen, "High-accuracy 2-D AoA estimation using lightweight UWB arrays," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 3312–3317.

[11] M. Naseri, A. Shahid, and E. De Poorter, "Adapting UWB AoA estimation towards unseen environments using transfer learning and data augmentation," *Internet of Things*, vol. 27, p. 101298, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660524002397

[12] V. Tran, Z. Dai, N. Trigoni, and A. Markham, "DeepCIR: Insights into CIR-based data-driven UWB error mitigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 300–13 307.

[13] "DWM3001CDK DK software, sources, tools and developer guide," Nov 2022. [Online]. Available: https://www.qorvo.com/products/d/da008604

[14] M. Naseri, A. Shahid, G.-J. Gordebeke, S. Lemey, M. Boes, S. Van De Velde, and E. De Poorter, "Machine learning-based angle of arrival estimation for ultra-wide band radios," *IEEE Communications Letters*, vol. 26, no. 6, pp. 1273–1277, 2022.

[15] J. Fontaine, F. Che, A. Shahid, B. Van Herbruggen, Q. Z. Ahmed, W. Bin Abbas, and E. De Poorter, "Transfer learning for UWB error correction and (N)LOS classification in multiple environments," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4085–4101, 2024.

[16] Y. Tian, Z. Lian, P. Wang, M. Wang, Z. Yue, and H. Chai, "Application of a long short-term memory neural network algorithm fused with Kalman filter in UWB indoor positioning," *Scientific Reports*, vol. 14, no. 1, Jan 2024. [Online]. Available: https://doi.org/10.1038/s41598-024-52464-y

[17] H. Chen and A. Dhekne, "PnPLoc: UWB based plug & play indoor localization," in *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2022, pp. 1–8.

[18] S.-P. Huang, C.-B. Chen, T.-Z. Wei, W.-T. Tsai, C.-Y. Liou, Y.-M. Mao, W.-H. Sheng, and S.-G. Mao, "Range-extension algorithms and strategies for TDOA ultra-wideband positioning system," *Sensors*, vol. 23, no. 6, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/6/3088

[19] Y. Sartayeva, Y. Liu, Y. H. Ho, and H. C. Chan, "Positioning vectors for mobile ad-hoc positioning," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2023, pp. 512–517.

[20] C.-M. Zhong, R.-H. Lin, H.-L. Zhu, Y.-H. Lin, X. Zheng, L.-H. Zhu, Y.-L. Gao, J.-B. Zhuang, H.-Q. Hu, Y.-J. Lu, Z. Chen, and W.-J. Guo, "UWB-based AOA indoor position-tracking system and data processing algorithm," *IEEE Sensors Journal*, vol. 24, no. 19, pp. 30 522–30 529, 2024.

[21] Z. Chang, F. Zhang, J. Xiong, X. Xue, Z. Wang, B. Jouaber, and D. Zhang, "UWBOri: Enabling accurate orientation estimation with ultra-wideband signals," in *SWC 2024 : IEEE Smart World Congress*. Denarau Island, Fiji: IEEE, Dec. 2024. [Online]. Available: https://hal.science/hal-04754129

[22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[23] "UWB solutions compatible with Apple's U1 & U2 chip." [Online]. Available: https://www.qorvo.com/innovation/ultra-wideband/products/uwb-solutions-compatible-with-apple-u1

[24] P. Leu, G. Camurati, A. Heinrich, M. Roeschlin, C. Anliker, M. Hollick, S. Capkun, and J. Classen, "Ghost peak: Practical distance reduction attacks against HRP UWB ranging," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 1343–1359. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/leu

[25] "DWM3001CDK Ultra-Wideband (UWB) module development kit." [Online]. Available: https://www.qorvo.com/products/p/DWM3001CDK