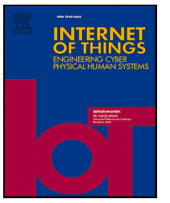
Internet of Things 26 (2024) 101152

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Research article

On the Generalization of Deep Learning Models for AoA Estimation in Bluetooth Indoor Scenarios

Ivan Pisa *, Guillem Boquet, Xavier Vilajosana, Borja Martinez

Wireless Networks Group, Universitat Oberta de Catalunya, Barcelona, 08018, Spain

ARTICLE INFO

Keywords:
Angle of Arrival
Angle estimation
Artificial Neural Networks
Bluetooth
Direction finding
Generalization

ABSTRACT

Indoor positioning remains as one of the major challenges of the Internet of Things (IoT), where assets are usually deployed over environments where the Global Navigation Satellite Systems (GNSS) are denied. Angle-of-arrival (AoA) estimation techniques are among the most prominent candidates to address this challenge and multiple techniques based on conventional signal processing, such as the Multiple Signal Classification (MUSIC) algorithm, have been extensively studied. However, they require a deep knowledge of the elements involved in the AoA system, such as the characterization of the receivers' antennas. On the other hand, Deep Learning (DL) models are considered as promising solutions, as they can provide AoA estimations without resorting to a deep knowledge of the physical system. Instead, they only need to be trained with labeled AoA signal measurements. Nevertheless, in the training process, non-desired effects like overfitting appear, yielding models with poor generalization capabilities. Although these models have already been considered in the literature, the problem of DL models generalization has not been addressed in depth. Therefore, this manuscript first compares the performance of DL models to the MUSIC algorithm as a baseline, to then analyze their generalization to different situations. These include changing the position of the receivers within the same area, considering measurements at different time instants and deployments in new scenarios, specifically in locations different from the training one. The assessment showed that the generalization of the DL models is weak compared to the MUSIC algorithm, especially when applied to environments not previously observed.

1. Introduction

Motivated by the recent standardization of Bluetooth Direction Finding technologies [1], IoT paradigm is considering Angle of Arrival (AoA) / Angle of Departure (AoD) methods [2] as its main indoor positioning technologies. This is evidenced by the wide variety of products being commercialized in the market that basis their functionality on AoA techniques [3].

AoA-based location consists of first deducing which direction the emitted signal is coming from (from the receiver point of view), and then geometrically translating the angle information into an estimate of the position.

The physical principle underlying AoA is the phase shift of the incident signal as received at each antenna of an array conforming the receiver. Then, this phase difference is processed to derive the angle estimations at the receiver (see Fig. 1), which determine the direction of the incident signals [4]. In general, multiple locators are considered to estimate the asset's position in spatial (Cartesian) coordinates by applying triangulation techniques, or simply to extend the AoA-based location to larger areas [5].

* Corresponding author.
E-mail addresses: ipisad@uoc.edu (I. Pisa), gboquet@uoc.edu (G. Boquet), xvilajosana@uoc.edu (X. Vilajosana), bmartinezh@uoc.edu (B. Martinez).

https://doi.org/10.1016/j.iot.2024.101152
Received 29 January 2024; Received in revised form 26 February 2024; Accepted 28 February 2024
Available online 29 February 2024

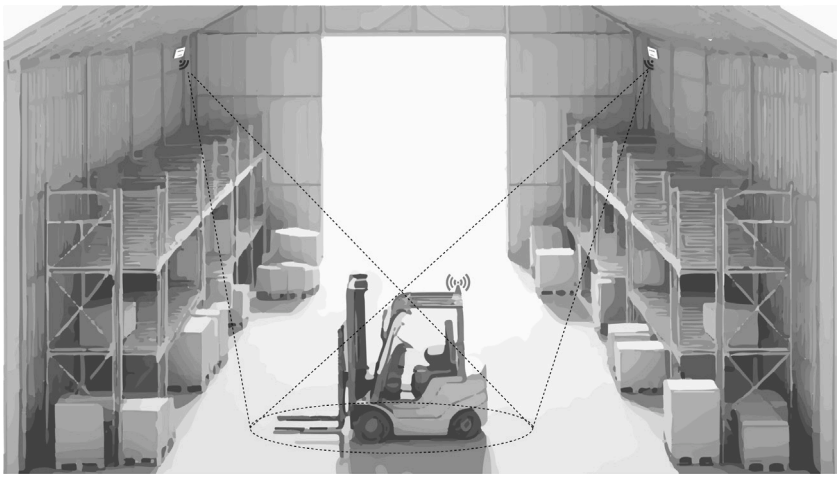


Fig. 1. AoA-based location Example.

Hence, the correct estimation of the AoA is crucial for indoor positioning systems. However, multiple signal effects —such as multipath components, polarization, spread delays, jitter, and noise— may generate ambiguities or distort the phase difference of received signals [4]. These noise sources decrease the accuracy of the derived AoA estimates and thus propagate the errors in the reported position. Notably, multipath effect can directly mislead the AoA estimation due to the impinging signal replicas received. In addition, most of the conventional techniques rely on critical aspects that may compromise the cost, scalability, and operation of the nodes where they are deployed. First, the need of a calibration of the array of antennas to obtain its steering vector [6] (basically a multi-directional calibration of the phase response of the antenna, which will be explained in detail in Section 3). Second, they involve costly computations such as the inversion of matrices and the eigenvector decomposition, which may require a large amount of computational resources when deployed at large scale.

In such a context, the ability and success of Deep Learning (DL) models and techniques in the modeling non-linear and complex relationships in other contexts have attracted their attention as candidates to mitigate the aforementioned problems [7,8]. First, DL models can seamlessly cope with most of the signal effects that affect conventional AoA approaches [9]. Second, once trained, they can obtain AoA estimates from input data without the necessity of characterizing the antenna array, nor processing the non-idealities [10]. Moreover, the computational cost can be reduced in the production stage, since the processing of AoA estimates by DL models do not require complex computations like matrix inversion, eigen-decomposition, etc. Instead, they provide AoA estimations by means of simple matrix multiplications.

Nevertheless, DL models face many problems as well. To begin with, they cannot be completely immune to multipath. In addition, they suffer their own issues such as overfitting, a critical issue regarding their generalization capability [11]. As we will discuss in this article, generalization is arguably the most important barrier to the potential adoption in real-world applications (industrial facilities, logistic centers, etc.). Translated into the AoA estimation process, this means that DL models may provide really accurate AoAs for the scenario for which they are trained, but not for unseen ones. This is a major issue that needs to be tackled if a scalable and commercial AoA estimation solution is desired. Literature has shown that most of the time, the collection of more data to train the DL models solve the problem. However, the efforts in terms of human resources and time invested in the AoA data gathering process cannot be neglected. In some scenarios, such as the manufacturing industry, it may even be impossible, as they operate uninterruptedly. This is the central issue addressed in this article. To that goal, this work first reviews the AoA DL-based SOA, the proposed DL models for AoA estimation available in the SOA as well as traditional methods in terms of accuracy. Then, the efforts are focused on the assessment of DL model's generalization capability, with the main objective of determining if these models can arise as scalable and reliable AoA estimators.

The main contributions of this manuscript are:

- To the best of our knowledge, this work is the first to bring to light (in this area) that DL model training cannot be considered unbounded, and may in fact be impossible in some scenarios. This may be especially relevant for industrial applications.
- By stating that training is constrained, we shift the main focus from accuracy to generalization. This work demonstrates that generalization is indeed a problem, since the performance of DL models that behave as expected in the training position decays dramatically when used in other positions and/or scenarios.
- Our results suggest that the models are learning the environment of the locator, rather than its physical characteristics which (by design) allow the angle of the incident signal to be discerned. This work can be a starting point for further research to find training methods that can solve this problem.

The remainder of this article is organized as follows. Section 2 summarizes the related work in the AoA topic, Section 3 presents the physical system on which our AoA assessment is based, Section 4 defines the models considered for the comparative and Section 5

describes the experimental setup. Section 6 presents the experimental results and discusses the generalization of the DL models compared to conventional AoA estimators. Finally, Section 7 concludes this manuscript.

2. Related work

The spread of the IoT has reignited the challenge of indoor positioning, in search of new technologies able to provide accurate positions in environments where the well-known Global Positioning Systems (GPS) do not succeed, as GPS satellite signals are blocked by building walls. As an alternative approach, the dual-function simultaneous use of different radio technologies such as Wi-Fi, BLE, ZigBee, RFID, etc., have been considered for that purpose. On top of that, several techniques can be applied. For example, (tri)lateration uses at least three measured distances (based on RSSI or time of flight) to well-known reference points to estimate an unknown position [12]. AoA is based on triangulation instead, that is, obtaining the unknown Euclidean position of an emitter from the angles of the arriving signals at some reference points [5]. Deployed at strategic points, the receivers (also known as anchors and locators) must be able to determine the direction of the incident electromagnetic wave, which has traditionally been carried out by means of complex signal processing techniques. Nowadays, however, the incursion of DL models is changing this approach. AoA estimates can be derived considering only raw Analog-To-Digital Converter (ADC) measurements (IQ samples), without resorting to classical signal processing techniques. In this context, this section summarizes the main works related to: (i) Conventional approaches, i.e., non-data-driven methods, and (ii) DL approaches, relying only on raw input data as the basis of AoA estimators.

2.1. Conventional AoA-based estimators

Conventional algorithms usually exploit inner features from signals and systems [13]. For instance, the MUSIC algorithm considers the orthogonality between signal and additive noise to generate a pseudo-spectrum from which the AoA can be estimated [6] (this algorithm is explained in detail in Section 4). Some of its major issues are related to its computation complexity, which increases with the resolution of the spatial spectrum discretization [6], and the strong dependency on the receiver's antenna characterization due to the requirement of its calibration to get the steering vector [14] that is, the phase delays a plane wave experiences at an array of antennas. Besides, MUSIC sweeps all the possible directions from which a signal can be received. This sweep becomes critical when MUSIC is applied over a dual polarized 2D resolution space (azimuth (θ) and elevation (ϕ)). To alleviate this computational cost, the adoption of MUSIC variations such as the Fast MUSIC algorithm is proposed [15]. However, the problem of the antenna calibration still persists [16]. This issue also applies for the methods —such as CAPON— which determine the AoA through the calculation of a spatial pseudo-spectrum and relying on the antenna calibration and steering vector computation [17].

Another conventional algorithm is the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT), which performs the AoA estimation by means of exploiting the rotational invariance of the signal subspace [18]. In that manner, the AoA estimates are obtained with a lower computational cost than MUSIC. Notwithstanding, ESPRIT fails in the AoA estimation process when coherent and highly-correlated signals are present [19]. Therefore, the removal of the correlation between signals is required, which entails an increment in the computational cost [19]. One algorithm tackling the reduction of computational complexity of MUSIC and ESPRIT corresponds to the Propagator Direct Data Acquisition (PDDA) method. It is based on the computation of the propagation vector, which reflects the cross-correlation between the received signal at the first element and at the other elements of an array [19]. Thus, PDDA does neither perform any kind of matrix decomposition nor require prior knowledge of the impinging signals [20]. However, it depends on the calculation of the propagator vector. Consequently, the dependency of the model on the antenna characterization is still high.

Other algorithm that aims to reduce the computational costs is the Subtracting Signal Subspace (SSS), which is also based on the correlation matrix decomposition. This algorithm adopts the signal subspace of the correlation matrix to generate a spatial pseudo-spectrum [21], as an alternative of selecting the noise subspaces. The main issue here is that SSS requires the computation of the spatial spectrum, the steering vector, and the antenna calibration. The former can be a highly time-consuming task, while the latter require the same accurate characterization of the antenna as MUSIC and PDDA do. Other approaches resort to the estimation of the AoA directly from the received measurements. This is the case of the work presented in [22], where the AoA estimations are obtained through the comparison of the received phase with the phase shift relative to a reference profile. In detail, results there show that AoA estimations are really accurate under controlled environments such as an anechoic chamber, however, authors faced a loss of accuracy when using the proposed approach in a real scenario. To solve this, they apply a filtering method to reduce non-desired and uncontrolled effects.

2.2. DL-based AoA estimators

Starting at [23], DL methods are adopted to estimate the AoA through the use of DL models like Fully Connected Neural Networks (FCNNs) and Convolutional Neural Networks (CNNs) [24]. Their usage is motivated by some well-known benefits: (i) their good performance in the modeling of non-linear relationships between received signals and the AoA labels, (ii) the easiness of the model generation, and (iii) the decoupling of the estimation process from hardware elements such as the receiver's antenna. In some cases, DL models are considered to estimate the pseudo-spectrum to later determine the AoA from it, mimicking the steps of classical techniques. Alternatively, DL models can directly estimate the AoA from raw measurements of the RF received signal. In this regard,

BLE raw data are the integers representing the magnitudes of In-phase (I) and Quadrature (Q) components of the impinging signals, which are transmitted adopting a quadrature modulation [1,2,25].

In terms of the pseudo-spectrum estimation process, FCNNs and Autoencoders have been considered in [26] to determine the AoA of different impinging signals. The autoencoder structures act as spatial filters which feed parallel FCNN layers acting as classifiers. The outputs of these layers are later mix to generate a spatial pseudo-spectrum and determine the direction or AoA, as performed with the conventional methods. Following this principle, FCNNs are adopted in [27] to estimate the probability density spectrum of AoAs for a receiver with a Field of View (FOV) equal to 120 degrees. Later, a post-processing is performed to determine the pairs of angles defining the position of an emitter. In this case, results show that a high resolution can be achieved whenever the Signal to Noise Ratio (SNR) values are over 18 dB. More complex DL models are also adopted to estimate the aforementioned pseudo-spectrum. In [28], CNNs are used to estimate it considering the measurement vector of received signals. Then, the peaks search is performed as in the MUSIC algorithm to estimate the θ and ϕ angles, i.e, the angles defining from where a signal is received in a 3D space. As a result, this model outperforms the FCNNs accuracy of [27], since reliable performances are obtained for lower SNR values. Other CNN proposals process the pseudo-spectrum as an image to classify from which direction the impinging signal is received [29]. Results there show an 87% accuracy for the same scenario from which CNN's training data are gathered. The most important point is that this performance is achieved in the presence of simulated noise and Rayleigh channel fading [30].

Regarding the direct estimation of the AoA, in [31], DL models are adopted with Gaussian Processes (GPs) and Regression Trees (RT) to estimate the AoA in scenarios affected by multipath effects. Since ANNs, GPs and RT require a training stage to either train the model, or derive the kernel to perform the estimates, a training process is periodically performed considering an arbitrary period [31]. Results show improvements between 20% and 50% in θ accuracy for a FOV of 120° with respect to MUSIC algorithm when the multipath effect is less severe (Signal-to-Noise Ratio (SNR) > 30 dB). For lower SNR values, a consistent performance is also achieved. Other examples considered two FCNNs to firstly classify the number of impinging signals and, with them, perform the estimation of the angles [32]. The training process of these FCNNs is performed considering data synthetically generated and corrupted with Additive White Gaussian Noise (AWGN). The performance of the approach is later assessed over a perfect array geometry without offsets, showing that under these specifications, the FCNNs can outperform the MUSIC algorithm when low SNR values are observed. However, the performance of the approach over real array geometries is not available. FCNNs and CNNs are also considered in [33] to implement an AoA estimator. To do so, they are trained separately and tested with AoA labels and the covariance matrix of the received raw BLE measurements. They consist of Line-of-Sight (LOS) and Non-line-of-sight (NLOS) data captured with a Uniform Linear Array (ULA) of four elements. In terms of the AoA labels, it is worth noting that θ angles are in the range $[-74°, 74°]$, i.e., using a FOV of 148°. As a result, approximately the 80% of the accuracy errors are placed in the vicinity of 2.5° when the models are tested over the training dataset.

Finally, in other cases, the FCNN and CNN models are not considered to directly estimate θ and ϕ angles, but to classify the region from where the impinging signals are coming. This is the case of the work presented in [34], where five different configurations of FCNNs and CNNs models are proposed to predict the θ angle from IQ measurements of the BLE signal. Tested over a simulated rectangular room with different levels of furniture, all the configurations perform well. Nevertheless, the best one corresponds to the CNN. In it, each anchor (receiver) has its own CNN devoted to estimating its AoAs. Consequently, as many AoA estimates as anchors are obtained. Later, all the estimates are processed by the Least Squared module to derive the final position of the emitter. Results show that this approach yields errors of 0.95 m, whereas if the conventional PDDA algorithm is considered instead, it yields errors of 1.5 m.

2.3. Motivation: the generalization problem

In summary, it is quite prevalent in the literature that DL models can outperform in accuracy the conventional AoA estimation algorithms. Nevertheless, most of the works reviewed test the DL models' behavior over measurements coming from the same dataset and scenario considered in the training process. In addition, most of them use synthetic data and few experiment in real-world scenarios. However, what it is crucial in the adoption of DL models as AoA estimators is their generalization capability, because in real-world applications, it is seldom possible to train the system in the operating environment. A first step in this direction is performed in [34], where an analysis of the DL models' generalization is performed over a simulated scenario. However, and to the best knowledge of the authors, there are still many open questions regarding the adoption of DL models as AoA estimators in real-world IoT applications. For that reason, the present work attempts to take the following open questions a step further:

- **Positional generalization**: Do DL models trained on data from one specific locator generalize to neighbor locators? If this is the case, DL models trained with measurements from a small set of representative locators could be deployed in any other. This is imperative for scalability.
- **Temporal generalization**: Do DL models trained on data from one specific locator at a given time generalize to the same locator in the future? The main idea is to determine whether DL models drift in time or not. Correcting a drift would require data from different instant times in the training process. This issue is especially relevant for some applications, as gathering more tagged data may be impossible after the initial deployment.
- **Scenario Generalization**: Do DL models behave similarly when deployed in scenarios other than where they have been trained? A model generalizes in a broad sense when it performs well in different scenarios. This is one of the crucial questions regarding the adoption of DL models as AoA estimators. Solutions able to generalize between scenarios are the ones that could provide an efficient, scalable and low-cost solution. Despite its relevance and potential impact, this issue has been largely ignored in the literature.

Other important aspects must also be taken into account, although these are issues not strictly related to generalization. In particular, the amount of data considered in the training process is one of the core aspects of DL models, and it may critically affect the final performance. If not enough data are considered, the model would be underfitted, providing inaccurate AoA estimations. On the other hand, too much data may overfit the model. In this case, the model memorizes the pairs of input and output data, but with very poor results on new inputs never seen before.

Moreover, training in a real environment may be very limited or even impossible, so the volume of data required for training and the generalization in different setups play a critical role. The ultimate question would be whether a model can be trained in a controlled setup in a reasonable amount of time, with enough data to learn the physical model of the locator, but not too much data to avoid overfitting to the environment. Under these conditions, the models would be easily portable and therefore highly scalable.

Taking all this into account, we address these questions for AoA in real indoor scenarios. To achieve this, a comparison of the performance between different DL models is performed, taking the MUSIC algorithm as the baseline. MUSIC is selected due to its adoption and performance offered in current works [6,14–16,31,33]. Therefore, the generalization capability of DL models is assessed in comparison with the performance of MUSIC, especially under new conditions.

3. System model

Throughout this work we consider a BLE AoA-based localization system consisting of a set of deployed locators, the BLE receivers, with known fixed position. For evaluation purposes, the mobile BLE emitter is placed at known (labeled) reference positions. Each locator obtains the input samples through sensing the BLE signal, from which the incidence direction has to be estimated. At the end of the localization process, a Cartesian position is usually obtained from the angular estimates. Here, however, the evaluation is focused on the AoA estimation process.

3.1. System hardware

BLE locators equip an antenna array that consists of two or more antennas, typically PCB patches, arranged in a linear, rectangular or circular layout. This array is used to extract the phase difference out of received BLE signals.

In this work, a custom Uniform Circular Array (UCA) with 6 patch antennas is used (see Fig. 2). In addition, each patch is vertically or horizontally polarized, so the system consists of 12 virtual antennas. Bluetooth Direction Finding proposes a switch-then-sample mechanism to access each antenna conforming the array, instead of sampling all the antennas at the same time. This approach lowers the hardware cost, requiring only one analog-to-digital converter (ADC). Thus, the array is controlled by an RF switch which is in charge of selecting which antenna (patch) samples the Constant Tone Extension (CTE) of the received BLE packet [1,2]. Technically, the CTE appears as an unmodulated offset of the carrier at the end of the packet or, equivalently, an unwhitened sequence of modulated 1's from the point of view of digital modulation. In practice, this leads to a fixed and constant frequency during the CTE, making the consecutive phase differences sampled in time (switch mechanism) equivalent to an instantaneous phase difference under some assumptions described next.

3.2. Fundamental array signal model

Let $s_0(t-\tau)$ be the transmitted CTE in baseband at the transmitter. As any other wireless communication system, the signal is modulated on top of a carrier signal of frequency f_c at the transmitter and then downconverted to baseband at the receiver side upon reception. In baseband, at the receiver's reference position in the far-field zone, the signal model when using the narrowband assumption $s_0(t) = s_0(t-\tau)$ reads

$$s(t) = \alpha\, s_0(t)\, e^{-j2\pi f_c \tau}\,, \tag{1}$$

where α is the signal's attenuation and τ the propagation time. This narrowband assumption is the basis of direction finding. The time of flight of a signal will be slightly different to reach antennas that are slightly separated and, under this assumption, the delay in time is only translated into a phase shift of the original signal.

Let the antenna array have M antennas, the phase difference ψ_m due to the additional propagation from the reference to the mth antenna position depends on the array geometry and the impinging direction $\{\theta, \phi\}$ of the transmitted signal. The geometry impact to (1) is then modeled as

$$s(t) = \alpha_m\, s_0(t)\, e^{-j2\pi f_c \tau}\, e^{j\psi_m(\theta,\phi)}\,, \tag{2}$$

explicitly showing that the received signal contains the angle information. For the UCA layout shown in Fig. 2, the phase difference is geometrically determined by [35]

$$\psi_m(\theta,\phi) = \frac{2\pi}{\lambda}\, r\, \sin\phi\, \cos\left(\theta - 2\pi\frac{m}{M}\right), \tag{3}$$

where λ denotes the signal's wavelength, r the radius, and $m = 1, \ldots, 6$ indicates the antenna number.

Let a_m be the mth antenna response, simultaneously accounting for the specific gain α_m and phase difference ψ_m. The received signal through the antenna in the presence of an additive noise $n(t)$ is then expressed as

$$x_m(t) = a_m(\theta,\phi)\, s(t) + n_m(t)\,. \tag{4}$$

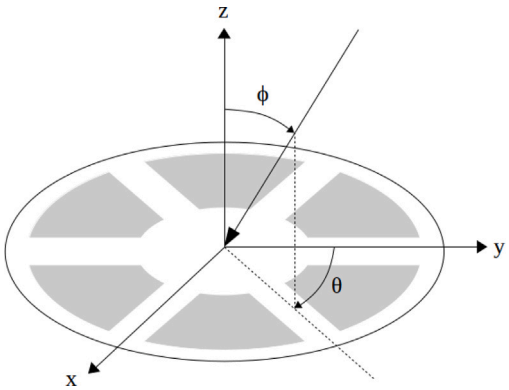


Fig. 2. Graphical description of an impinging signal received by a 6 patch antenna UCA. The incident direction is defined by θ, ϕ, the angles of arrival.

In vector form for the M antennas, the received signal $\mathbf{x}$ (size $M \times 1$) is composed of the UCA response vector, referred to as steering vector, $\mathbf{a}$ ($M \times 1$) multiplied by the source signal and the receiver noise vector $\mathbf{n}$ ($M \times 1$):

$$\mathbf{x}(t) = \mathbf{a}(\theta, \phi)\, s(t) + \mathbf{n}(t)\,. \tag{5}$$

Note that for a unique received signal, the measurement vector equals to $\mathbf{x}(t) = \mathbf{i}(t) + j\,\mathbf{q}(t) \in \mathbb{C}$, which is composed of the in-phase and quadrature components, i.e., the IQ samples given by the ADC's receiver.

Finally, the goal of direction finding is to estimate $\{\theta, \phi\}$ from the measurement $\mathbf{x}$.

3.3. Switch sampling model

As already mentioned, in the particular case of Bluetooth 5.1 Direction Finding specification, $\mathbf{x}(t)$ is not simultaneously acquired by each antenna, but there is a switch-then-sample mechanism to temporally multiplex a single ADC. Therefore, $\mathbf{x}(t)$ is obtained via sequential sampling following the predefined antenna switching pattern.

For a Round-Robin pattern that repeats N times, the acquisition of the signal in the first lap is defined by

$$\mathbf{x}_s^{(1)}(t) = \begin{bmatrix} x_1(t + T_s) \\ \vdots \\ x_M(t + M\,T_s) \end{bmatrix}, \tag{6}$$

where T_s is the sampling interval time and the superscript $^{(1)}$ denotes lap number one. The extended result to the end of the CTE is then defined by the matrix of size $M \times N$,

$$\mathbf{X}_s(t) = \left[\mathbf{x}_s^{(1)}(t) \;\; \ldots \;\; \mathbf{x}_s^{(N)}\left(t + (N-1)\,T_L\right) \right], \tag{7}$$

where $T_L > T_s$ is the lap duration limited so that $N \times T_L$ does not exceed the CTE duration. Note that the maximum number of measurements from each antenna decreases as the number of antennas rises because of the CTE's finite length.

The CTE duration used in our experiments is 160 μs (maximum allowed by the BLE standard), where the first 4 μs are a guard period and then 8 μs are reserved for the reference period. The switching and sampling slots are 2 μs each, which are consecutively alternated from the end of the reference period until the end of the CTE. The sampling slot is conformed by two slots of 1 μs each, where the former is idle, and the latter contains the mth antenna IQ sample. Thus, one effective sample requires 4 μs. In this work, M = 12 (6 antennas and 2 polarizations) and N=3, thus the size of $\mathbf{X}_s$ is 12×3.

3.4. Model equivalence

To directly apply traditional signal processing techniques to $\mathbf{X}_s$, we want the data vector with sequential sampling $\mathbf{x}_s$ to be equivalent to the simultaneous sampling case $\mathbf{x}$. In other words, the following equality must be satisfied:

$$x_m(t + m\,T_s + n\,T_L) = x_m(t) \quad \forall\; n, m\,. \tag{8}$$

Let $\Delta_T = m\,T_s + n\,T_L$ be any specific acquisition time of the mth antenna for notation convenience, so that we rewrite (4) as

$$x_m(t + \Delta_T) = a(\theta, \phi)\, s(t + \Delta_T) + n_m(t + \Delta_T)\,. \tag{9}$$

Now note in (9) that the source narrowband signal is constant for the specific assumption that the channel remains constant over a 152 μs period from the start of the reference period to the last sampling slot, thus $s(t + \Delta_T) = s(t)$. Additionally, the noise in consideration is AWGN, thus $n_m(t + \Delta_T) = n_m(t)$ from the statistical perspective. With these assumptions, (9) becomes

$$x_m(t + \Delta_T) = x_m(t) \tag{10}$$

and thus $\mathbf{x}_s(t) = \mathbf{x}(t)$, implying that we can use (7) for the goal of estimating $\{\theta, \phi\}$.

3.5. Channel frequency offset

Because BLE is aimed at mass-market use, the available hardware uses low-cost local oscillators, resulting in a significant Carrier Frequency Offset (CFO) Δ_{f_c} in the received signals with respect to the transmitted ones. Using the same assumptions of the equivalent model derivation, the CFO effect is modeled into (10) as

$$x_m(t + \Delta_T) = x_m(t)\, e^{j2\pi\Delta_{f_c}\Delta_T}\,. \tag{11}$$

Contrary to what the equivalent model assumes, the received signals will not be in baseband, resulting in a phase progression between sampling slots even though the source signal is constant over the CTE. That is, phase measurements between slots in a CTE are subject to the CFO, which changes over time.

Reducing the angle of arrival estimation error involves estimating the CFO and adjusting the offset as the first step of the angle estimation process. Many CFO compensation algorithms for BLE can be found in the literature. For example, an optimization of a cost function similar to the least squares method is proposed in [36]. The MUSIC algorithm, the maximum likelihood estimator and the Fourier transform have also been proposed as methods to estimate the CFO [37]. Samples collected during the reference period can be utilized to create a linear progression model for the phase at the reference antenna. This model can then estimate phase values at upcoming time points, enabling the computation of phase differences with respect to other antennas [38]. The reference samples can be ignored and directly calculate the phase difference between two adjacent antennas if the array exhibits a specific symmetry that ensures that the two antennas consistently measure the same phase variation [22]. In this work, we adjusted the CFO by averaging the phase drift between slots of the preceding Round Robin cycles for each antenna, which is an algorithm used by Texas Instruments [37].

3.6. Array calibration

The UCA's antenna steering vector is an ideal representation of how the received signal's phase is modified due to the physical distribution of the antenna. In addition to providing spatial diversity, the presence of neighboring antennas inside the array causes extra signal distortion because of mutual coupling. Since each antenna contributes new and unknown signal components to the original signals, mutual coupling has an impact on the performance [39]. The impact of mutual coupling on the array of antennas is described by the complex symmetric square matrix if the array structure and antenna response are symmetric [40]. An additional source of mismodelling for antenna arrays is hardware imperfections [41]. Similarly, the effect of path imbalance and switch leakage can be summarized into one additional matrix.

For both mismodelling factors, the matrices can be estimated by analyzing the antenna array response under controlled conditions, then subsequently employ the matrices to rectify signal measurements or angle estimates. This process is known as array calibration. Existing compensation techniques exhibit a trade-off between computational complexity and efficiency [40]. They can be divided into distinct methods based on self-calibration, mutual impedance, active antenna pattern and S-parameter. Self-calibration algorithms require prior knowledge of the coupling matrix. Mutual impedance methods involve complex modeling and simulation of both the antenna and the nearby structure in the array setup. The S-parameter method, while less accurate, allows for straightforward measurement of the parameters using a network analyzer [39].

In this work, we utilized an active antenna pattern method to investigate the radiation pattern of each antenna element. This involved exciting individual antennas while terminating the others for each BLE channel, with the angle space discretized in 1-degree steps. The antenna patterns obtained at each discretization step were employed to construct the array response vector for each specific channel. This method is used by Texas Instruments in their RTLS Toolbox in SimpleLink CC13x2/26x2 SDK and explained in [42]. It should be noted that this is a time-consuming process, which also requires an anechoic chamber.

3.7. Multipath reception

BLE is typically used in indoor settings, resulting in severe multipath effect. Coherent signal reception due to multipath is a major problem here, as it changes the phase of the signal in reception. It cannot be estimated without assuming the reflection geometry, signal polarization, reflection coefficient or others [43]. It is usually mitigated by exploiting frequency diversity. In BLE, this means receiving several packets in different BLE frequency channels, which takes additional time [36]. Spatial Smoothing is a renowned solution [44]. On the other hand, it can be countered by identifying the LOS component [45] or, as proposed recently, by analyzing synchronized measurements obtained from multiple distinct arrays [46]. In this work we focus on single packet reception, as the traditional method relying on MUSIC do not cope with multipath. Therefore, if neural network models can outperform its performance, it would represent a clear competitive advantage.

4. Competing AoA estimators

This section presents the two AoA estimators that will be evaluated. These estimators are intended to be representative of the two approaches to be compared: a conventional algorithm based on signal processing —MUSIC— and a deep learning, FCNN-based model.

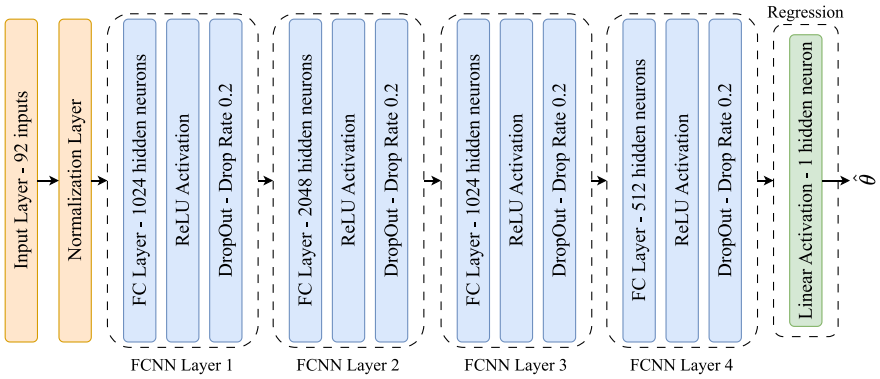


Fig. 3. FCNN-based model.

4.1. MUSIC

The widely-used MUSIC algorithm, which serves as the baseline to assess the generalization and performance of the DL models considered in Section 4.2. The MUSIC algorithm departs from the IQ samples from the received impinging signal to estimate the sample correlation matrix [47], defined as the complex matrix:

$$\hat{\mathbf{R}}_x = \frac{1}{N}\mathbf{X}_s\mathbf{X}_s^H \tag{12}$$

where H denotes the hermitian transpose. In this work, the size of $\hat{\mathbf{R}}_x$ is $M \times M$. Note that $\hat{\mathbf{R}}_x$ corresponds to an estimation of $\mathbf{R}_x = E\{\mathbf{X}\mathbf{X}^H\}$. Then, the eigenvector decomposition of the sample correlation matrix is carried out:

$$\mathbf{R_x} = \mathbf{Q_s}\mathbf{D_s}\mathbf{Q_s}^H + \mathbf{Q_n}\mathbf{D_n}\mathbf{Q_n}^H, \tag{13}$$

where $\mathbf{Q_s}$ and $\mathbf{Q_n}$ denote the matrices of the eigenvectors representing the signal and noise spaces, respectively. $\mathbf{D_s}$ and $\mathbf{D_n}$ are the diagonal matrices containing their eigenvalues. Finally, MUSIC computes a pseudo-spectrum from which the AoA estimation candidate θ, ϕ can be obtained:

$$\hat{\theta}, \hat{\phi} = \underset{\theta,\phi}{\operatorname{argmax}} \left(\frac{1}{\mathbf{A}^H(\theta,\phi)\mathbf{Q_n}\mathbf{Q_n}^H\mathbf{A}(\theta,\phi)} \right) \tag{14}$$

MUSIC has a high computational cost from an IoT device perspective [48]. To start with, it requires from an eigenvalue decomposition (13). Most importantly, it requires a sweep for all angles θ and ϕ at the desired resolution in (14). In addition, MUSIC relies on a complex calibration of the antenna array which, as explained in Section 3.6, involves characterizing the array response $\mathbf{A}(\theta, \phi)$ for all angles and for all BLE channels.

4.2. Deep learning

The key advantages of DL models are the lightweight computations in operation, and its capability in providing an estimate directly from raw inputs, thus, avoiding the characterization of the array (Section 3.6) and the estimation of the frequency offset (Section 3.5).

Among the different topologies of neural networks available in the literature, this work focuses on FCNNs. This choice is based on the advantage that FCNNs can process raw measurements, while CNNs require the computation of $\mathbf{R_x}$ in (12). As reported in [33], FCNNs show good performance when tested in the same scenario. This makes FCNNs a good candidate for our evaluation purposes, as we are focused on the generalization assessment.

Fig. 3 shows the internal structure of the proposed FCNN model, which is based on the DeepAoANet-FC network presented in [33]. The proposed model differs from the DeepAoANet-FC in the input and output data, sharing the same internal structure. Whereas DeepAoANet considers the sample covariance matrix as the input data, our model directly estimates the AoAs from raw BLE measurements. In terms of the outputs, our model directly returns the AoA estimations instead of multiple values from which infer the final AoAs. Focusing on the proposed FCNN-based model internal structure, it consists of fully connected layers devoted to sequentially mapping the input measurements to the selected output. The first layer corresponds to the Input layer, which considers 92 input measurements: the raw IQ samples from the CTE (45 real and 45 imaginary parts), the average Received Signal Strength Indicator (RSSI) of the CTE and the BLE channel. The 45 IQ samples contain 8 samples from the reference period (sampled at 1 μs) and 37 samples obtained from the rest of the CTE duration (6 patches using vertical polarization and 6 using horizontal, 3 full laps and one extra sample at the end to complete the 148 μs available). See Table 1 for more details.

The Input layer is followed by a normalization layer devoted to scaling input measurements towards zero mean and unit variance. This is added to ease the training process and to avoid the masking of considered variables [49, Chapter 5]. The normalization layer

Table 1
Data considered as inputs of the FCNN-based model.

Data	Description
Date and Time	BLE Timestamp
Transmitter Id	Transmitter's identifier
Receiver Id	Receiver's MAC address
Average RSSI	Average RSSI in dBm (1 dB resolution)
BLE Channel	BLE Channel
I samples	List of 45 real IQ components
Q samples	List of 45 imaginary IQ components
Label name	Point where the emitter is placed
Label position	True known emitter's XYZ point
Label azimuth (θ)	True azimuth angle (from receiver's viewpoint)
Label elevation (ϕ)	True elevation angle (from receiver's viewpoint)

is followed by four learnable fully connected (FC) layers with 1024, 2048, 1024 and 512 hidden neurons each. Besides, each FC layer considers a non-linear Rectified Linear Unit (ReLU) activation function and a DropOut layer, which seeks for the improvement of the model's generalization [49, Chapters 6 and 7]. This layer drops the output of the previous layer with a probability equal to the drop rate. Along the model, the drop rate is set to an arbitrary value equal to 0.2, which proved helpful in training [33]. Finally, a one-hidden-neuron fully connected layer with a linear activation function provides the output.

As a final remark, it is worth noting that this work does not put the efforts on getting and leveraging the power of the best DL models. It is focused instead on the generalization assessment in new and non previously observed scenarios. For this reason, simple and well-known configurations have been chosen for both the DL model and the MUSIC baseline. Notice that, for simplicity, efforts in this work are focused on estimating θ angle.

5. Experimental setup

In order to evaluate the generalization of DL models, a set of experiments have been specifically devised. The experimental setup is detailed next.

5.1. Scenarios

Our ultimate goal is to evaluate the generalization in new scenarios (outside controlled environments) and under realistic operating conditions. With that aim in view, two real-world scenarios are used: (i) a testing scenario where the environment conditions are controlled, and (ii) an operation/production scenario, where the conditions may change due to the presence of furniture, electronic devices and people.

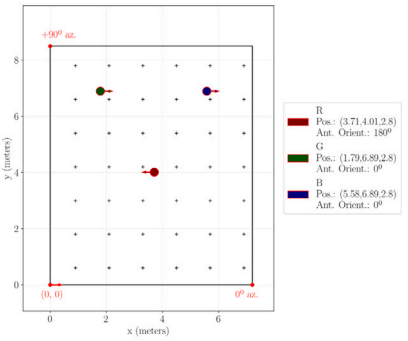
The testing scenario is a rectangular room of dimensions 7.2 (width) × 8.5 (length) × 3 (height) m. The deployment consists of three different locators considering a 360° FOV, and distributed following a triangular configuration as shown in Fig. 4(a). As shown in the figure, one locator (locator R) is in a central position, while the other two (locators G and B) have been placed closer to the walls and windows, trying to force multipath and other interactions with different materials. This is the reason for the non-symmetrical arrangement. The BLE emitter is placed at tagged points on a rectangular grid, with a total of 42 positions, uniformly distributed along the whole room (see Fig. 4(b)).

On the other hand, the production scenario is located in a social area of the Universitat Oberta de Catalunya's main headquarters. Furniture such as sofas, tables, and chairs are present (see Fig. 4(c)), as an example of a more realistic operation. The evaluation area is a rectangular perimeter of $10 \times 12 \times 2.7$ m, part of the 500 square meters second floor of the headquarters. The deployment consists of four locators in a rectangular arrangement (locators R, G, B and O), In this case, the selected positions do not aim to force multipath effects, but represents a real deployment looking for the best coverage of the area. A grid of 42 points uniformly distributed is used for testing purposes (see Fig. 4(d)) and, therefore, ensuring a fair comparison of the DL models' performance between the testing and the production scenarios.

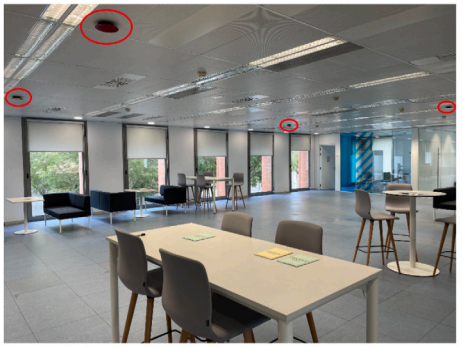
5.2. Dataset

Training data are obtained from the testing scenario. A total of four datasets collected in four different days, with more than 145 000 raw BLE measurements each, are considered to train the DL model. Noise and multipath effects are intrinsically present in the BLE measurements. Each dataset considers the data defined in Table 1. For more details regarding the raw BLE measurements, readers are referred to the repository, where the datasets are publicly available [50].

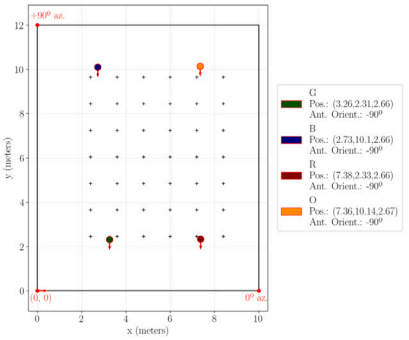
(a) Test Scenario



(b) Test Configuration



(c) Production Scenario



(d) Production Configuration

Fig. 4. Testing and Production scenarios and their configurations. In each scenario picture (left) the position of locators is circled in red. The emitter in a sample position is circled in blue. On the right, the crosses mark the known positions used as true labels in each scenario.

5.3. DL training

The training process of the DL model consists in the optimization of a loss function, which is defined accordingly to the DL model's purpose. The symmetric and differentiable loss function is defined as

$$L(\theta, \hat{\theta}) = \left| \mathrm{atan2}\left(\sin(\theta - \hat{\theta}), \cos(\theta - \hat{\theta}) \right) \right| \tag{15}$$

where atan2 is the 2-argument arctangent and returns the minimal difference in the interval $(-\pi, \pi]$, and the sin and cos representation are used to provide a smooth function that respects the modular nature of angle comparison.

The training process is carried out considering the ADAM optimization algorithm with a learning rate equal to $1 \cdot 10^{-4}$ [49, Chapter 8]. The maximum number of epochs is set to 10 000, however, the training process can be stopped earlier by the early stopping whenever overfitting is detected.

To perform the training process, three splits of the datasets are considered: train, validation, and test. Each split represents the 80%, 10% and 10% of the dataset, respectively. In addition, it is random, ensuring that all the data casuistic is present in each split. The train split is used to fit the model and, therefore, optimize the loss function. In our case, we have determined the splitting distribution through a test and error process. It is worth noting that this splitting is also consistent with the usual distribution for DL models, as mentioned in [49, Section 5.4]. DL models are prone to over-fit the training data, which occurs when the performance of the model against the validation dataset starts to drop while it is still improving against the train dataset. To avoid this, different techniques can be considered. Here, an L2 extra penalty equal to $5 \cdot 10^{-5}$, the DropOut, and early stopping techniques are adopted [49, Section 7.2]. Finally, the model performance is computed with non previously observed measurements defining the test split, a subset equivalent to 10% of the whole data.

6. Results: AoA estimation performance

This section evaluates quantitatively the problem of generalization outlined in 2.3. A specific methodology has been defined for that purpose, which is articulated around three tests:

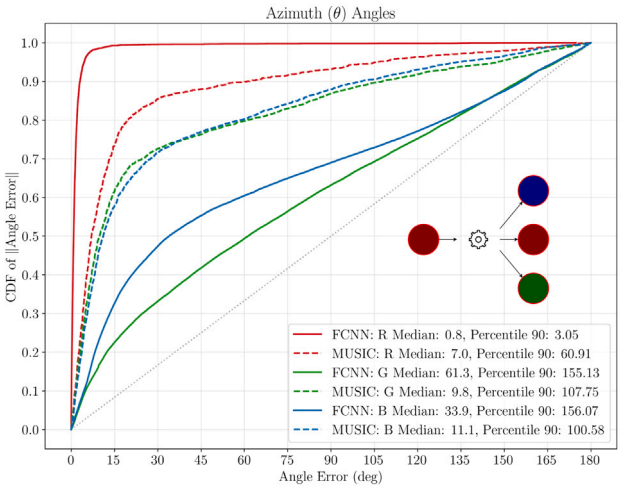


Fig. 5. CDF of each locator adopting the FCNN-based model trained only with data from R. The goal is to evaluate the performance of the model in neighboring locators. MUSIC performance is provided as the baseline for comparison purposes. The extra legend depicts this approach, where colored circles denote each locator and the gear stands for the DL training process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1. Positional generalization: determine whether DL models trained with samples from a particular locator (at some specific position) can be used to estimate the AoAs of neighboring locators.
2. Temporal generalization: DL models trained with measurements from one day against measurements from other days.
3. Portability: contrasting DL models trained in one scenario against measurements in different scenarios.

Note that MUSIC is considered as the baseline approach, since it is agnostic to changes of locator position within a site, as well as to changes of site.

6.1. Positional generalization

The objective of the positional assessment is to determine if DL models trained with data from one specific locator generalize in others. This would entail that a single model could be deployed on other locators —in the same scenario— to independently estimate their respective AoAs, and therefore, simplify the training process.

Here, the assessment is performed considering the grid configuration presented in Fig. 4(b). The FCNN-based model is trained with measurements from a unique day (campaign 1) coming only from the R locator, and its performance is determined with the test split from the three locators: (i) R, (ii) G, (iii) B.

From Fig. 5, the Cumulative Density Function (CDF) of the AoA estimation errors shows that the proposed FCNN-based model (trained on R) generalize for the test split from the same R locator. For instance, it shows that more than an 80% of the AoA estimation errors are in the vicinity of 0°. Specifically, the median and percentile 90 of errors equal to 0.8 and 3.05 degrees, respectively, which means that AoA estimations provided by the FCNN-based model are quite accurate, outperforming the MUSIC algorithm. This result validates our setup, since the result of this model is comparable to those presented in the literature. For example, similar results are observed in [34] when DL models are considered individually. In this case, however, the authors propose the adoption of fully connected networks for each BLE channel and locator. Our approach with a single FCNN for all channels yields similar results, while notably simplifying the training since a unique model is trained instead of as many as channels. However, the performance drops drastically when the model trained on the R locator is used for G and B locators. Fig. 5 shows that the performance of G and B falls well below MUSIC. In this case, the MUSIC algorithm shows that the median of the errors between the estimated angles and the real ones equals to 9.8° and to 11.1°, for locators G and B, respectively. Their percentile 90 equal to 107.75° and 100.58°. In contrast, for the DL model, the highest median of errors equals to 61.3° and the percentile 90 to 155.13°, which are yielded by the G locator. In terms of the B locator, these values equal to 33.9° and 156.07°, respectively. It is clear that under these conditions —the model applied to a different physical locator and placed at another position— MUSIC outperforms the DL methods. In this case, it is arguable that placing the locators close to the wall makes them prone to experience multipath. Indeed, this is precisely the purpose of this experimental arrangement.

In contrast, if specific FCNN-based models are trained with data from the locator where they are deployed, one can observe in Fig. 6 that MUSIC is outperformed also for those locators more prone to multipath. Now, Locators G and B show estimation errors whose medians are 0.4° and 0.7° (in contrast to 61.3° and 33.9°) and whose percentile 90 are 2.00° and 2.31° (compared to 107.75° and 156.07°), respectively. This result can be summarized in that each locator learns its environment —either its physical position, its own hardware characteristics or both.

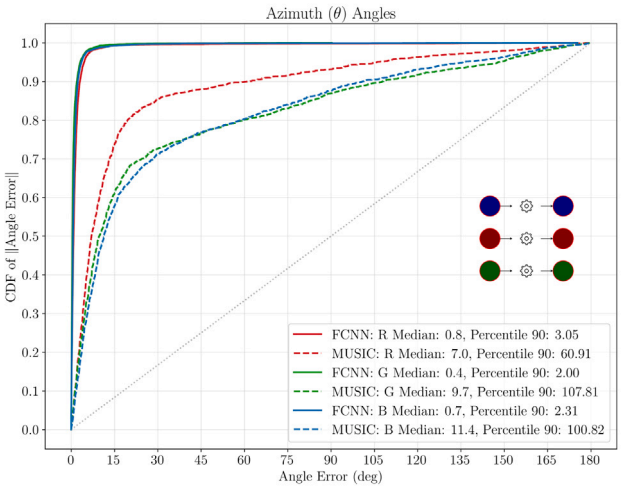


Fig. 6. CDF of estimation errors when specific FCNN-based models are considered, that is, models trained with data from the locator they are in charge of. In the extra legend, colored circles denote each locator and gears stand for each DL model training process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

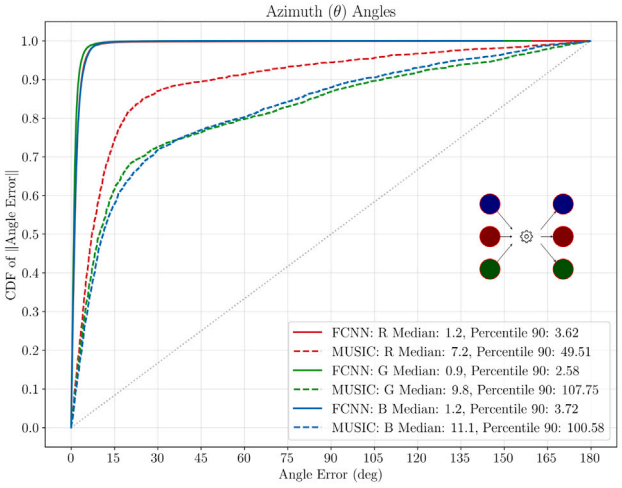


Fig. 7. CDFs of the AoA estimation errors for a unique FCNN-based model trained with data from all locators in the testing scenario. The extra legend summarizes the approach, where colored circles define each locator and the gear stands for the DL model training process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Alternatively, we wonder whether a single DL model trained with data from all locators at the site can achieve better performance. This could mean that signals received by a locator contain information that can be exploited by others. In Fig. 7 a unique FCNN-based model trained with measurements from all locators is considered. In this case, the model provides estimation errors whose median equals to 1.2° and their percentile 90 to 3.72° for the worst case (locator B). This result is only slightly worse than Fig. 6. However, increasing diversity could have beneficial effects against overfitting, thus improving the generalization.

6.2. Temporal generalization

The goal is to determine if DL models generalize in time, that is, if the model drifts in time for the same locator and place. Should this be the case, the DL model could be used through the time without the necessity of gathering more labeled data and modeling the temporal evolution of the environment (re-training). This is of utmost importance for certain domains. Collecting more training data is not always feasible, and may be even impossible, as in industrial facilities or commercial environments which cannot stop their activity.

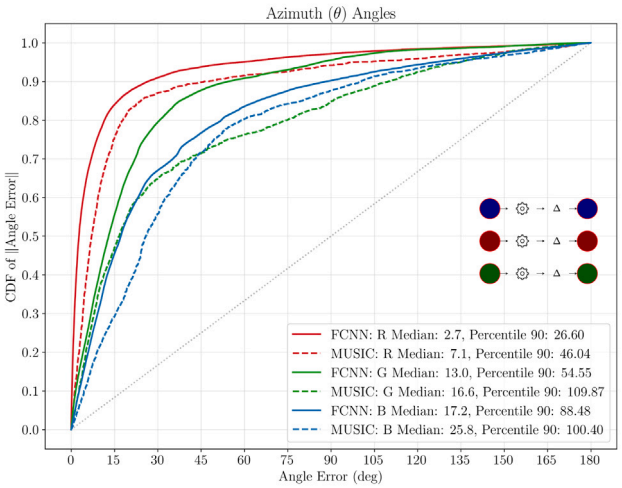


Fig. 8. CDF of estimation errors for locators in the testing scenario, considering individual FCNN-based models for each locator. Note that the models are trained with measurements from each locator on the first day and tested on the same locator on the second day. In terms of the extra legend, the Δ represents the test of the model on the second day.

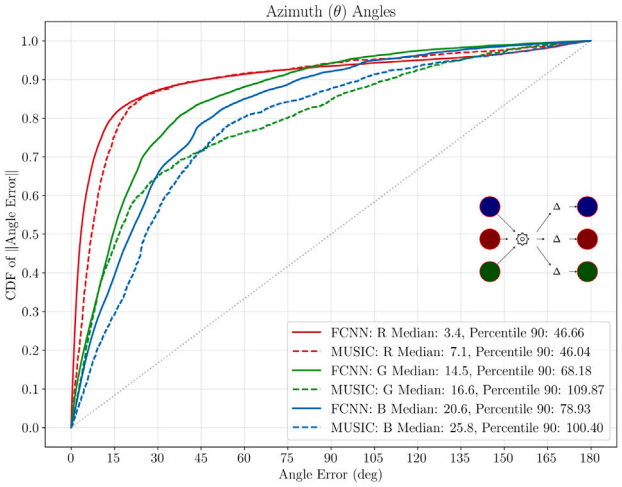


Fig. 9. CDF of estimation errors for locators in the testing scenario, considering a general FCNN-based model as AoA estimator. Note that a common model is trained with data from all locators on the first day and is tested on the second day. Here, Δ represents the test of the model on the second day.

For this assessment, two options are considered based on the results of the previous subsection: (i) specific FCNN-based models for each locator, or (ii) a common FCNN-based model trained with data from all available locators. Using a common model trained on a single locator is discarded because of the poor results obtained in the previous section (Fig. 5).

The first case, where specific models for each locator are considered, results in Fig. 8. There, it is shown that exists a loss of accuracy, especially for the DL models. In the previous section, it was observed that for R, the median of errors of the FCNN model against data from the test split of the same day was 0.8° (see Fig. 6). Now, if it is tested with measurements from the same locator, but gathered in a different day, the median of errors is increased until 2.7°. A similar trend, yet more pronounced, is observed for the remaining locators. Now, the estimation errors of G and B equal to 13.0° and 17.2°, respectively, whereas in the same day split these errors were equal to 0.4° and 0.7°, respectively. Still, FCNN models are able to outperform the MUSIC algorithm.

Some drift is also observed if a shared FCNN-based model is considered for all locators instead of locator-specific models. As it is shown in Fig. 9, the FCNN-based model yields higher errors than in the case where it is tested with measurements from the same day as the training data (shown in Fig. 7). Focusing on the locator returning the worst performance, i.e., Locator B, the model yields errors with a median of 20.6° and a percentile 90 of 78.93°, whereas previously, these values equal to 1.2° and 3.72°, respectively. Nevertheless, the performance achieved is still better than the MUSIC's one.

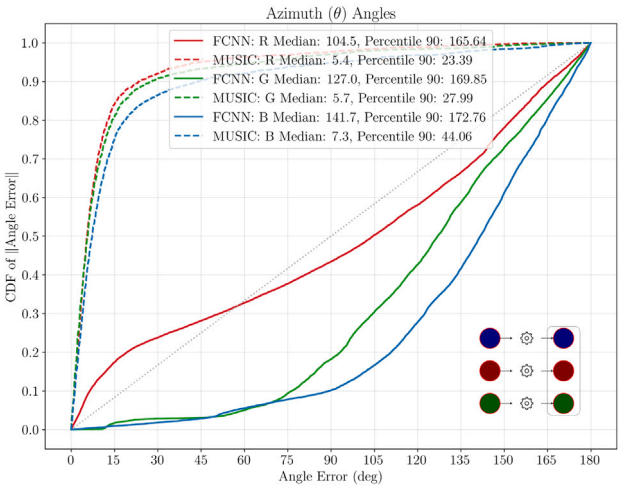


Fig. 10. CDF for specific FCNN models trained with data from the testing scenario, but adopted in the production environment. Each model is used in the same physical locator, which has been reinstalled at the new location. This is depicted on the extra legend as a rectangle containing the locators.

Both cases corroborate that although a loss of accuracy is observed, FCNN models generalize in time when applied in the same scenario where they have been trained, at least in terms of improving MUSIC's accuracy.

6.3. Scenario generalization

The ultimate goal of this study is to assess the reliability of DL models in new scenarios, that is, environments not previously seen during the model training. In addition, we were looking for realistic conditions, with furniture and elements of everyday use. For that purpose, the production scenario shown in Fig. 4(c) was selected.

Three strategies were devised: (i) use the same physical locator mounted in the new scenario, but with its corresponding model trained in the old scenario (testing). (ii) train a common model with data from all the locators in the testing scenario, and (iii) transfer only the most representative FCNN-based model.

The first experiment is equivalent to Fig. 6 in that the model is deployed on the locator in which it has been trained, but now locators are not installed in the same position where they were trained, but in a new scenario. The production scenario considers four locators, R, G, B and O, so for this first approach we will use only data from the three locators already trained. Fig. 10 shows the results when the FCNN-based models from the testing scenario are used at their respective locators in their new positions. Recalling our equivalent experiment in Section 6.1 (Fig. 6), it is now clear that each FCNN-based model learned the specific conditions of the testing scenario, rather than learning its own hardware characteristics. Even more, what models have learned is self-defeating in the new location.

The second experiment evaluates a common model, trained with data from all locators in the testing scenario, when it is deployed on all locators in the production scenario. This is the same model that was used to evaluate the positional generalization in Fig. 7 and the temporal generalization in Fig. 9, but used in the production environment. The objective now is to assess whether the diversity introduced by training the model with 3 different locators (at 3 different positions) is representative for this new scenario.

From Fig. 11, the model that was able to generalize in different positions in the same scenario (Fig. 7) and also in time (Fig. 9), has a very poor performance when deployed at the new scenario. Again, for the worst case, the median is now 109.7° (compared to 1.2° and 20.6°, respectively). In fact, while the two previous experiments outperformed the MUSIC (1.2° vs. 11.1° and 20.6° vs. 25.8°, respectively) now the performance is under the MUSIC's result (109.7° vs. 7.4°)

In the third experiment, we deploy on the production scenario the (single) model for the R locator trained on the testing scenario. This is the same model as the first experiment in Section 6.1 (Fig. 5). There we saw that R was not representative of the other two, but being aware that G and B were placed in multipath-prone positions. We now try to assess whether R is representative of the positions in the new scenario. From Fig. 12, it can be noticed that the general performance is poor. The medians range from 18.8° at best to 104.5° at worst. Thus, none of them outperforms MUSIC. Surprisingly, the worst case is R, the locator where the model we are using was trained. This confirms once again that the models learn their physical position in space, not the intrinsic characteristics of the receiver.

In summary, none of the three experiments generalized in the new scenario, falling in all cases below the baseline (remember that MUSIC is position agnostic).

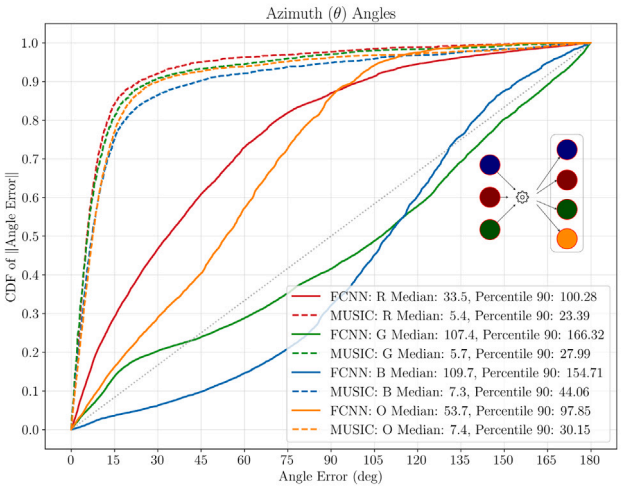


Fig. 11. CDF for a common FCNN model trained with measurements from all the locators in the testing scenario. The FCNN model is tested over the production scenario. The rectangle of the extra legend represents the production scenario.

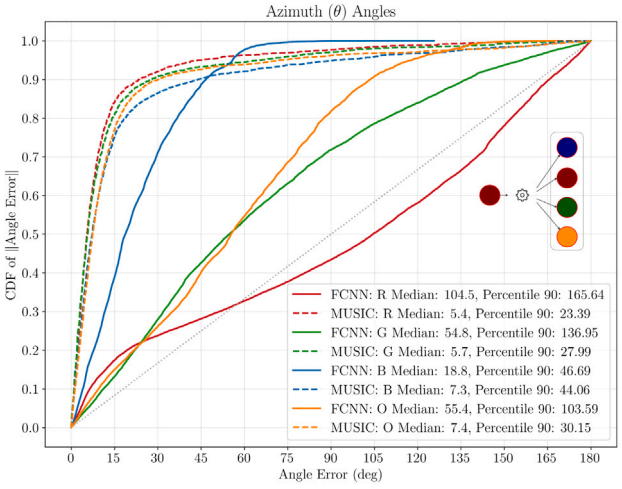


Fig. 12. CDF for individual FCNN models trained with data from locator R at the testing scenario. The models are tested over the production scenario, which is represented as a rectangle in the extra legend.

6.4. Amount of data

In an attempt to tackle FCNN's inability to generalize in unseen scenarios, the FCNN-based model can be trained considering more data gathered in the testing scenario. Thus, for this experiment, we trained the models with data from two different campaigns (performed on two different days (campaign 1 and campaign 2)).

In case of considering a common FCNN-based model, from Fig. 13 it is observed that the generalization improves (to be compared to Fig. 11). For example, the median for locator B is now 50.7° vs. the previous 109.7° (which was essentially random). However, this median is still far from the 7.3° achieved with the MUSIC algorithm.

If specific models for each locator are used instead, the medians range from 14.5° to 17.5° (see Fig. 14). These results are significantly better than those of Fig. 10. Perhaps more importantly, they present much less variability among locators. Still, all of them are below MUSIC performance, which ranges from 5.4° at best and 7.4° at worst.

This corroborates one of the well-known issues of DL: a model should be trained with large amounts of data that represent most of the scenario's dynamics where the DL model is deployed. This can be useful —up to a certain limit— when training in a scenario other than the final application. Yet too much data can induce the model to overfit, as found when training in the same scenario as the application.

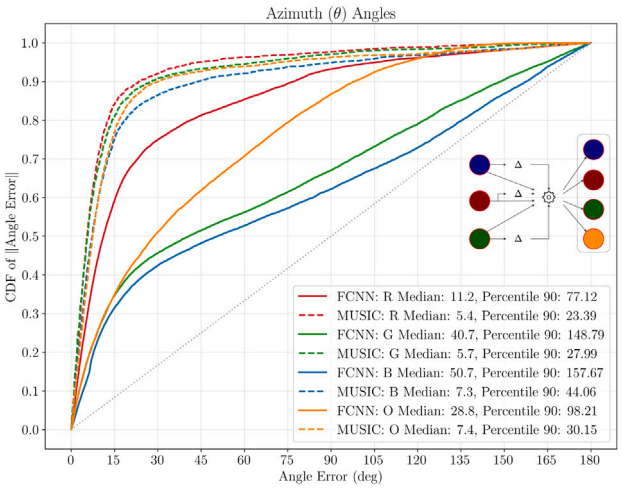


Fig. 13. CDF for a unique FCNN model trained with measurements from two different days and from all the locators in the testing scenario. The model is tested over the production scenario. Notice that Δ on the extra legend stands for measurements gathered on the second day.

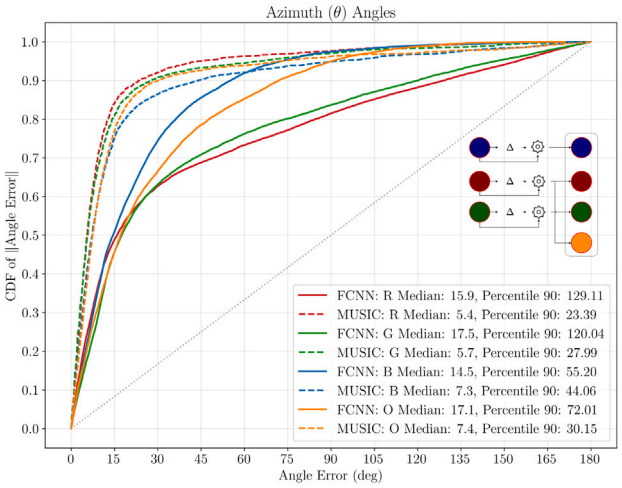


Fig. 14. CDF for individual FCNN models trained with measurements from each locator in two different days (see the Δ in the extra legend) and from the testing scenario. The models are tested over the production scenario.

These two conflicting constraints leave a narrow margin to fine-tune this application, in which MUSIC, the classical method, has proven to be most effective. Obviously, for our particular purposes, the fine-tuning process could be performed with measurements from the production scenario. However, this is not a viable solution for environments that cannot stop their operation. Therefore, this approach cannot be qualified as generalizable in a broad sense.

7. Conclusion

In this article, we explore different generalization possibilities of DL models as AoA estimators.

Our starting point consists of a conventional experiment with measurements gathered by a locator and divided into two different splits: (i) a labeled subset for training, and (ii) an unseen subset for testing purposes. With these conditions, our approach —trained and tested with these measurements— provides comparable results to those available in the literature. Not only this validates our software tools, models, and experimental setup in terms of estimation accuracy, but also in terms of the model's temporal generalization.

On this basis, the most important conclusion extracted from our experiments is that DL models give the impression of learning the particularities of the position where the locator has been installed —the environment—, instead of the intrinsic characteristics

of the receiver —the locator. This result has been proved performing two independent experiments. Firstly, adopting the model trained with measurements from a locator, but in a different position inside the same scenario. Secondly, considering the model of the same locator but in a new and unseen room —the environment. In both cases, results show a poor performance, being always outperformed by the conventional MUSIC algorithm.

Considering a less ambitious strategy, we have trained a unique DL model with measurements from different locators installed in the same room. This is performed to let the model extract more general features from the spatial diversity of these locators. Neither in this case, the model has been capable of generalizing in an unseen scenario.

Finally, we have proved that results of both strategies improve when the amount of measurements considered in the training process is increased, particularly when data from different days —trying to exploit their temporal diversity— are considered. However, this approach does not outperform the MUSIC performance in a new scenario.

Our results clearly show that DL models cannot be adopted as a general solution, i.e., as AoA estimators for any site. The training of the model with measurements captured in the environment where the locator is placed is a must. Therefore, this fact notably limits the possible applications, since in most cases the training in the place of operations might be very complicated or even impossible. Moreover, and from the insights of our results, we consider that the DL generalization issue should be always addressed in similar applications and environments. We have only put the focus on this issue. But, there is a need for additional research to improve DL models' ability to adapt to varying environments and circumstances.

CRediT authorship contribution statement

Ivan Pisa: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Guillem Boquet:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Xavier Vilajosana:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **Borja Martinez:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is co-funded by the European Union's Horizon 2020 (VEDLIoT project, No 957197); the Spanish Ministry of Science, Innovation and Universities (RF-VOLUTION, PID2021-122247OB-I00); and the Catalan Department of Research and Universities (SGR 00174 2021). Xavier Vilajosana is funded by the ICREA Academia Grant.

References

[1] M. Woolley, Bluetooth core specification version 5.1 feature overview, 2021, Bluetooth Technology Website, (Accessed: 28 February 2023).
[2] M. Woolley, Bluetooth SIG, Bluetooth direction finding: A technical overview, 2021, Bluetooth Technology Website, (Accessed: 28 February 2023).
[3] G. Pau, F. Arena, Y.E. Gebremariam, I. You, Bluetooth 5.1: An analysis of direction finding capability for high-precision location services, Sensors 21 (11) (2021) 3589.
[4] H. Krim, M. Viberg, Two decades of array signal processing research: The parametric approach, IEEE Signal Process. Mag. 13 (4) (1996) 67–94.
[5] G. Ottoy, L. De Strycker, An improved 2D triangulation algorithm for use with linear arrays, IEEE Sens. J. 16 (23) (2016) 8238–8243, http://dx.doi.org/10.1109/JSEN.2016.2584647.
[6] B. Friedlander, A sensitivity analysis of the Music algorithm, IEEE Trans. Acoust. Speech Signal Process. 38 (10) (1990) 1740–1751, http://dx.doi.org/10.1109/29.60105.
[7] S. Dong, P. Wang, K. Abbas, A survey on deep learning and its applications, Comp. Sci. Rev. 40 (2021) 100379.
[8] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for IoT big data and streaming analytics: A survey, IEEE Commun. Surv. Tutor. 20 (4) (2018) 2923–2960.
[9] W. Zhu, M. Zhang, P. Li, C. Wu, Two-dimensional DOA estimation via deep ensemble learning, IEEE Access 8 (2020) 124544–124552, http://dx.doi.org/10.1109/ACCESS.2020.3005221.
[10] K. Kotrotsios, A. Fanariotis, H.-C. Leligou, T. Orphanoudakis, Design space exploration of a multi-model AI-based indoor localization system, Sensors 22 (2) (2022) 570.
[11] D. Jakubovitz, R. Giryes, M.R. Rodrigues, Generalization error in deep learning, in: Compressed Sensing and Its Applications: Third International MATHEON Conference 2017, Springer, 2019, pp. 153–193, http://dx.doi.org/10.1007/978-3-319-73074-5_5.
[12] B. Yang, L. Guo, R. Guo, M. Zhao, T. Zhao, A novel trilateration algorithm for RSSI-based indoor localization, IEEE Sens. J. 20 (14) (2020) 8164–8172, http://dx.doi.org/10.1109/JSEN.2020.2980966.
[13] E. Gentilho Jr., P.R. Scalassara, T. Abrão, Direction-of-arrival estimation methods: A performance-complexity tradeoff perspective, J. Signal Process. Syst. 92 (2) (2020) 239–256.

[14] F. Yan, M. Jin, X. Qiao, Low-complexity DOA estimation based on compressed Music and its performance analysis, IEEE Trans. Signal Process. 61 (8) (2013) 1915–1930.
[15] G. Ran, M. Xing-Peng, L. Shao-Bin, W. Yi-Ming, W. Xiu-Hong, A fast DOA estimation algorithm based on polarization Music, Radioengineering 24 (1) (2015) 215.
[16] B.R. Jackson, 2D direction of arrival estimation using uniform circular arrays with radiation pattern reconfigurable antennas, IEEE Access 10 (2022) 11909–11923.
[17] S. Kikuchi, H. Tsuji, A. Sano, Autocalibration algorithm for robust capon beamforming, IEEE Antennas Wirel. Propag. Lett. 5 (2006) 251–255.
[18] S.-F. Chuang, W.-R. Wu, Y.-T. Liu, High-resolution AoA estimation for hybrid antenna arrays, IEEE Trans. Antennas and Propagation 63 (7) (2015) 2955–2968, http://dx.doi.org/10.1109/TAP.2015.2426795.
[19] M.A. Al-Sadoon, N.T. Ali, Y. Dama, A. Zuid, S.M. Jones, R.A. Abd-Alhameed, J.M. Noras, A new low complexity angle of arrival algorithm for 1D and 2D direction estimation in MIMO smart antenna systems, Sensors 17 (11) (2017) 2631.
[20] H. Ye, B. Yang, Z. Long, C. Dai, A method of indoor positioning by signal fitting and PDDA algorithm using BLE aoa device, IEEE Sens. J. 22 (8) (2022) 7877–7887.
[21] M.A. Al-Sadoon, N. Abduljabbar, N.T. Ali, R. Asif, A. Zweid, H. Alhassan, J.M. Noras, R. Abd-Alhameed, A more efficient AOA method for 2D and 3D direction estimation with arbitrary antenna array geometry, in: Broadband Communications, Networks, and Systems: 9th International EAI Conference, Broadnets 2018, Faro, Portugal, September 19–20, 2018, Proceedings 9, Springer, 2019, pp. 419–430.
[22] N. Paulino, L.M. Pessoa, Self-localization via circular bluetooth 5.1 antenna array receiver, IEEE Access 11 (2022) 365–395.
[23] S. Jha, T. Durrani, Direction of arrival estimation using artificial neural networks, IEEE Trans. Syst. Man Cybern. 21 (5) (1991) 1192–1201.
[24] O. Bialer, N. Garnett, T. Tirer, Performance advantages of deep neural networks for angle of arrival estimation, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2019, pp. 3907–3911.
[25] M. Girolami, F. Furfari, P. Barsocchi, F. Mavilia, A bluetooth 5.1 dataset based on angle of arrival and RSS for indoor localization, IEEE Access (2023).
[26] Z.-M. Liu, C. Zhang, S.Y. Philip, Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections, IEEE Trans. Antennas and Propagation 66 (12) (2018) 7315–7327.
[27] M. Gall, M. Gardill, T. Horn, J. Fuchs, Spectrum-based single-snapshot super-resolution direction-of-arrival estimation using deep learning, in: 2020 German Microwave Conference, GeMiC, IEEE, 2020, pp. 184–187.
[28] L. Wu, Z.-M. Liu, Z.-T. Huang, Deep convolution network for direction of arrival estimation with sparse prior, IEEE Signal Process. Lett. 26 (11) (2019) 1688–1692.
[29] Z. HajiAkhondi-Meybodi, M. Salimibeni, A. Mohammadi, K.N. Plataniotis, Bluetooth low energy and CNN-based angle of arrival localization in presence of Rayleigh fading, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2021, pp. 7913–7917.
[30] J. Shen, B. Huang, X. Kang, B. Jia, W. Li, Localization of access points based on the Rayleigh lognormal model, in: 2018 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2018, pp. 1–6.
[31] A. Khan, S. Wang, Z. Zhu, Angle-of-arrival estimation using an adaptive machine learning framework, IEEE Commun. Lett. 23 (2) (2018) 294–297.
[32] J. Yu, W.W. Howard, D. Tait, R.M. Buehrer, Direction-of-arrival estimation with a vector sensor using deep neural networks, in: 2021 IEEE 93rd Vehicular Technology Conference, VTC2021-Spring, IEEE, 2021, pp. 1–7.
[33] Z. Dai, Y. He, V. Tran, N. Trigoni, A. Markham, Deepaoanet: Learning angle of arrival from software defined radios with deep neural networks, IEEE Access 10 (2022) 3164–3176.
[34] A. Koutris, T. Siozos, Y. Kopsinis, A. Pikrakis, T. Merk, M. Mahlig, S. Papaharalabos, P. Karlsson, Deep learning-based indoor localization using multi-view BLE signal, Sensors 22 (7) (2022) 2759.
[35] D. Alibi, U. Javed, F. Wen, D. He, P. Liu, Y. Zhang, L. Jiang, 2D DOA estimation method based on channel state information for uniform circular array, in: 2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services, UPINLBS, 2016, pp. 68–72, http://dx.doi.org/10.1109/UPINLBS.2016.7809952.
[36] M.L. Sollie, K. Gryte, T.H. Bryne, T.A. Johansen, Outdoor navigation using bluetooth angle-of-arrival measurements, IEEE Access 10 (2022) 88012–88033.
[37] S. Cloudt, Bluetooth Low Energy Direction Finding on Embedded Hardware by Mitigating Carrier Frequency Offset and Multipath Fading (Ph.D. thesis), Eindhoven University of Technology Eindhoven, The Netherlands, 2021.
[38] M. Cominelli, P. Patras, F. Gringoli, Dead on arrival: An empirical study of the bluetooth 5.1 positioning system, in: Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization, 2019, pp. 13–20.
[39] L. Yao, Bluetooth direction finding, 2018.
[40] S. Henault, Y. Antar, Unifying the theory of mutual coupling compensation in antenna arrays, IEEE Antenn. Propag. Mag. 57 (2) (2015) 104–122.
[41] E. Kornaros, S. Kabiri, F. De Flaviis, A novel model for direction finding and phase center with practical considerations, IEEE Trans. Antennas and Propagation 65 (10) (2017) 5475–5491.
[42] Bluetooth angle of arrival (AoA) antenna design – Application note, 2019, TIDA029A, Texas Instruments, Revised June 2023.
[43] S. Akhavan, M.A. Bahabadi, Y. Norouzi, H. Lolaee, Direction of arrival estimation using array of antennas for low-altitude targets in multi-path environment, IET Radar Sonar Navig. 10 (8) (2016) 1439–1445.
[44] R.T. Williams, S. Prasad, A.K. Mahalanabis, L.H. Sibul, An improved spatial smoothing technique for bearing estimation in a multipath environment, IEEE Trans. Acoust. Speech Signal Process. 36 (4) (1988) 425–432.
[45] M.C. Vanderveen, C.B. Papadias, A. Paulraj, Joint Angle and Delay Estimation (JADE) for multipath signals arriving at an antenna array, IEEE Commun. Lett. 1 (1) (1997) 12–14.
[46] M.L. Sollie, T.H. Bryne, K. Gryte, T.A. Johansen, Reducing ground reflection multipath errors for bluetooth angle-of-arrival estimation by combining independent antenna arrays, IEEE Antennas Wirel. Propag. Lett. (2023).
[47] R. Schmidt, Multiple emitter location and signal parameter estimation, IEEE Trans. Antennas and Propagation 34 (3) (1986) 276–280.
[48] N. Balamurugan, S. Mohan, M. Adimoolam, A. John, W. Wang, et al., DOA tracking for seamless connectivity in beamformed IoT-based drones, Comput. Stand. Interfaces 79 (2022) 103564.
[49] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, MA, USA, 2016, http://www.deeplearningbook.org.
[50] Wireless Networks Group, A Direction Finding (DF) Bluetooth Low Energy (BLE) indoor data set for angle of arrival (AOA) and position estimation, 2023, [Online]. Available at https://bitbucket.org/wineuoc/aoa-ble-dataset/src/master/. (Accessed: 22 March 2023).