

MobLoc: CSI-Based Location Fingerprinting With MUSIC

Stepan Mazokha , Fanchen Bao , George Sklivanitis , *Member, IEEE*, and Jason O. Hallstrom 

Abstract—Many CSI-based localization methods have been proposed over the last decade. Fingerprinting has been one of the highest achieving approaches due to its capacity to capture environmental characteristics that are not readily captured using classic localization mechanisms such as multilateration. However, oftentimes the proposed methods are limited by reliance on large-scale training datasets. Further, methods are rarely evaluated on nonstationary devices, which are the most common in real-world environments. In our work, we address these challenges by introducing MobLoc. We adopt MUSIC pseudospectrum-based fingerprinting, which can benefit from, but does not heavily rely upon a large number of packets for each fingerprint. To evaluate our method, we leverage a publicly available dataset of passively collected CSI measurements, DLoc (Ayyalasomayajula et al., 2020), where an emitter sends signals in motion. We also benchmark MobLoc against a series of state-of-the-art localization methods. The results demonstrate that our method outperforms SpotFi (Kotaru et al., 2015), EntLoc (Chen et al., 2019), and AngLo (Chen et al., 2020), and falls very short of achieving DLoc accuracy. On the DLoc dataset, MobLoc achieves 0.33 m median (and 0.82 m, 90th percentile) localization error in a simple environment and 1.15 m median (2.59 m, 90th percentile) localization error in a complex environment. However, despite MobLoc not exceeding DLoc's accuracy, we consider its performance as a tradeoff for computational resources required to deploy the method in a real-world environment. We anticipate that this advantage will enable the adoption of MobLoc in city-scale localization systems, where the cost of computational resources is key.

Index Terms—Channel state information, fingerprinting, MUSIC, probability density estimation, WiFi localization.

I. INTRODUCTION

THERE has been continuous interest in WiFi localization over the past decade. This area holds importance to a variety of applications, including mobile navigation [5], [6], [7], mobility intelligence [8], [9], [10], and more. Advancements in the extraction of channel state information from common off-the-shelf devices have enabled higher accuracy localization methods that extract novel features from raw data [11], [12], [13], [14]. Meter-level and even decimeter-level localization accuracy have been demonstrated within complex environments [3], [4], [15]. Compared to purely algorithmic methods such as multilateration, CSI has been used to extract uniquely identifying features for fingerprinting (Section III).

However, a series of challenges hinder the adoption of CSI-based localization methods in production applications. From hardware limitations [16], [17], [18], [19] to measurement calibration procedures [17], [20], sensor deployments are an

expensive and laborious process. Yet, one of the most important challenges lies in the amount of data required to build radio maps and classification models for localization.

First, the extraction of high-quality fingerprints during the live localization stage often requires a significant number of packets for each location. For instance, Chen et al. [3] proposed a fingerprinting method, EntLoc, which optimally requires 50 packets for each location to build a fingerprint. However, consider an application such as Mobintel [21], where localization relies on passively emitted probe requests. Here, the devices of interest usually remain in motion and emit only a few packets at a time during a probe request burst [22], [23]. Such limitations can considerably reduce localization accuracy, which we confirm later in this article.

Second, the environment under observation must be exhaustively surveyed. While this initial objective can be satisfied, feature degradation requires continuous surveying of the space [4], [24], which is an expensive and resource-intensive endeavor. In addition to the above, the majority of the proposed localization methods are evaluated on proprietary datasets and are not publicly available; proposed methods are rarely open-sourced.

In this work, we attempt to address these issues through a new localization method, MobLoc. First, we use a fingerprinting approach due to its ease of use and potential for scalability. To build fingerprints, we introduce a Multiple Signal Classification (MUSIC) pseudospectrum-based method, designed to work well with a small number of packets for each location. Second, we leverage a publicly available dataset collected by the WCSNG Group at UC San Diego, DLoc [1]. This presents an opportunity to evaluate MobLoc on a sparse set of measurements, where the emitting device is constantly in motion. Third, we benchmark the performance of our method using the DLoc

Manuscript received 13 August 2023; revised 30 October 2023; accepted 15 November 2023. Date of publication 29 November 2023; date of current version 22 December 2023. This work was supported in part by the City of West Palm Beach, the Knight Foundation, in part by the Community Foundation of Palm Beach and Martin Counties, in part by ERC for Smart Streetscapes, funded through NSF Award EEC-2133516, and in part by NSF awards EEC-2030234, CNS-2117822, CNS-1753406, and ITE-2226392. (Corresponding author: Jason O. Hallstrom.)

Stepan Mazokha, Fanchen Bao, and Jason O. Hallstrom are with the FAU Institute for Sensing and Embedded Network Systems Engineering (I-SENSE), Boca Raton, FL 33431 USA (e-mail: smazokha2016@fau.edu; fbao2015@fau.edu; jhallstrom@fau.edu).

George Sklivanitis is with the FAU Center for Connected Autonomy and AI, and FAU I-SENSE, Boca Raton, FL 33431 USA (e-mail: gsklivanitis@fau.edu).

Digital Object Identifier 10.1109/JISPIN.2023.3336609

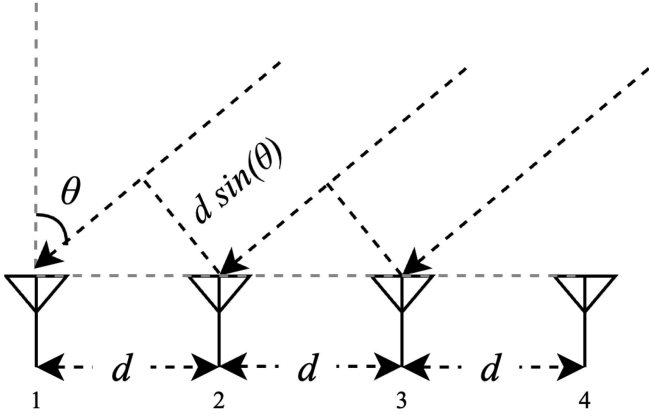


Fig. 1. Uniform linear array.

dataset against a series of state-of-the-art localization methods: SpotFi [2], EntLoc [3], and AngLoc [4]. Finally, we open-source MobLoc, as well as our implementation of the above-mentioned methods [25]. With this, we hope to encourage continued experimentation and adoption of our work by the community.

The rest of the article is organized as follows. Section II describes the preliminaries behind fingerprinting localization. Section III outlines recent advancements in the space. Section IV introduces the MobLoc design and delves into the details of its implementation. Section V introduces the experimental setup, benchmarking, experimental results, and performance analysis. Finally, Section VI presents our conclusions and plans for future work.

II. PRELIMINARIES

In this section, we introduce the basic principles of channel state information, hardware errors, and the fundamentals of direction-of-arrival estimation.

A. Channel State Information

Channel state information (CSI) represents a more capable input for wireless localization compared to RSSI. Some of its limitations are similar to RSSI since both are generated from the same signal source. However, CSI provides amplitude and phase information for each of the OFDM subcarriers used to transmit messages. This opens new avenues for localization via the estimation of direction-of-arrival and time-of-flight properties of the signal. We provide a high-level overview of CSI estimation below.

In wideband, multiple-input, multiple-output (MIMO) OFDM system releases (802.11a/g/n/ac), IQ samples are transmitted through a wireless channel through a series of frequencies—OFDM subcarriers. CSI is used to compensate for propagation and asynchrony effects associated with the transmission channel. It is estimated through a process of *channel sounding*: transmission of a known training sequence (LTS) in the preamble of every OFDM burst for channel estimation [17]

$$Y^i = H^i \cdot X^i + n^i. \quad (1)$$

In (1), X^i , Y^i , H^i , and n^i are the i th transmitted OFDM symbol, received OFDM symbol, CSI, and additive Gaussian noise, respectively [26]. We can express H^i as a 2-D matrix:

$$H^i = \begin{bmatrix} h_{0,0}^i & \cdots & h_{0,N-1}^i \\ \vdots & \ddots & \vdots \\ h_{M-1,0}^i & \cdots & h_{M-1,N-1}^i \end{bmatrix}. \quad (2)$$

Here, CSI is expressed through M receiving antennas and N equidistant subcarriers [15]. Each element of this matrix is a linear combination of multipath components, as presented in

$$h_{m,n}^i(f) = \sum_{l=0}^L \alpha_l e^{-j2\pi f \tau_l} \quad (3)$$

where α and τ denote attenuation and propagation delays, respectively, for L multipath components, given a subcarrier frequency f [27].

B. CSI Hardware Errors

CSI provides a set of complex values representing amplitudes and phases for each OFDM subcarrier. Often, these measurements are affected by hardware and transmission channel errors. We survey key issues here and provide details of our mitigation techniques in Section IV-C.

1) *Sampling Frequency Offsets*: Sampling frequency offsets are caused by the receiver's analog-to-digital converter. They are caused by a mismatch between the transmitter's oscillation frequency and receiver's sampling frequency. The result is a set of phase offsets that are proportional to indices of CSI subcarriers [17].

2) *Symbol Timing Offsets*: Symbol timing offsets are the most significant cause of errors in CSI measurements. They arise due to a lack of receiver knowledge concerning the beginning of the transmitted OFDM symbols. This challenge can be addressed by starting an OFDM header with known periodic sequences—short-training fields. However, due to intercarrier and intersymbol interference and carrier phase rotation, this error is not entirely eliminated [17], [28].

3) *Interantenna Phase Offsets*: Interantenna phase offsets are introduced due to the independence of oscillator circuits across antenna radio units. As a result, captured samples often lack synchronization. Such offsets are semideterministic, and they only change between reboots of a receiver [20]. Multiple researchers have attempted to exploit the differences to implement calibration techniques for CSI measurements [20], [29].

C. MUSIC

The MUSIC superresolution algorithm [30] has been a staple of WiFi-based localization methods for more than a decade [2], [17], [31]. Most commonly, it is used to estimate DoA of the emitted signal. To understand how it works, we must review the nature of signal propagation in an environment.

Consider a transceiver system in which a signal is emitted with L propagation paths. Let us assume that a receiver contains a uniform linear array of M antennas. Importantly, each of the

antennas is positioned at most $\frac{\lambda}{2}$, where λ is the wavelength of the signal carrier frequency at distance d . This is required to avoid spatial aliasing. For the k th propagation path, denote θ_k as the DoA angle with respect to the normal of the antenna array of the receiver.

Consider signal attenuation at each of the antennas of the receiver. For the second antenna, signal attenuation is the same as for the first antenna, but with an additional phase shift caused by the additional distance the signal had to travel to reach the second antenna. Similar effects are present for all other antennas. This effect is illustrated in Fig. 1.

Therefore, each propagation path is dependent on DoA and attenuation, as expressed in the following equation:

$$\Phi(\theta_k) = \exp(-j2\pi d \sin \theta_k \lambda). \quad (4)$$

Using this formulation, we can formulate a steering vector for each of the propagation paths—a vector of phase shifts at the receiving antenna array

$$a(\theta_k) = [1\Phi(\theta_k)\dots\Phi(\theta_k)^{(M-1)}]^\top. \quad (5)$$

As a result, we obtain a steering matrix with dimensions $M \times L$ of the antenna array, which incorporates all propagation paths

$$A = [a(\theta_1), \dots, a(\theta_L)]. \quad (6)$$

Let us now consider the formulation of the received signal:

$$x = A\Gamma + n \quad (7)$$

where $\Gamma = [\gamma_1 \dots \gamma_L]^\top$ is a vector of complex attenuations for each of the L propagation paths, and n is noise with zero mean and σ_n^2 variance.

We can use this formulation to obtain an $M \times M$ correlation matrix R_{xx} :

$$R_{xx} = E[xx^*] = AR_{ss}A^* + \sigma_n^2 I \quad (8)$$

where x^* is the Hermitian transpose of x , A is the steering matrix introduced in (6), A^* is the Hermitian transpose of A , $R_{ss} = E[ss^*]$ is the source correlation matrix, and I is the identity matrix.

The correlation matrix R_{xx} consists of M eigenvalues $\lambda_1, \dots, \lambda_M$ that are respectively associated with M eigenvectors $E = [e_1, e_2, \dots, e_M]$. Here, eigenvalues are sorted in nondecreasing order. Therefore, the smallest $M - D$ components correspond to noise, and the remaining D components represent incoming signals. This provides an opportunity to formulate an E_n noise subspace from $M - D$ eigenvectors.

We can use the noise subspace E_n in combination with antenna array steering vectors to formulate pseudospectrum P for a range of possible DoA angles θ_k

$$P(\theta_k) = \frac{1}{a(\theta_k)^H E_n E_n^H a(\theta_k)}. \quad (9)$$

Finally, we can perform a peak search on the obtained pseudospectrum vector. The highest peak is expected to yield the signal's DoA angle.

III. RELATED WORK

Direct algorithmic localization with DoA (i.e., trilateration) is challenging. Environments often contain various obstacles that prevent consistent line-of-sight connectivity. Since exhaustive modeling of all possible propagation paths is not usually feasible, researchers have proposed alternative localization methods, such as fingerprinting.

Compared to trilateration, fingerprinting does not rely on algorithmic location estimation. Instead, the objective is to build a radio map by extracting uniquely identifying features for the majority of the tiles during an offline mapping stage, and later performing feature matching during an online localization stage. Still, this method suffers from a number of limitations.

Fingerprinting requires an extensive data collection stage, which is often performed through war-driving—a process of surveying the environment to obtain a sufficient number of measurements for as many of the tiles as possible. However, the quality of fingerprints deteriorates over time due to gradual environmental changes (i.e., weather changes, movement of obstacles, etc). This motivated numerous efforts to consider approaches for updating the radio map on a regular basis.

A variety of CSI-based fingerprinting methods have been presented in the last decade. Chapre et al. [32] proposed raw CSI fingerprinting. In MIMO-CSI, fingerprints are composed of amplitude and phase difference means across 30 subcarriers over the total number of transmit and receive antennas. To achieve the overall minimum 1.13-m distance error, the authors use 100 000 samples collected at each location. Wang et al. proposed two deep learning approaches to CSI-based fingerprinting—PhaseFi [33] and DeepFi [34]. Here, the authors use the output values of a deep learning model as features for fingerprinting the environment. The method achieves a mean error of 1.08 m in a living room environment. Xiao et al. [35] considered probability distributions of CSI amplitudes as a fingerprinting feature. They leverage Kullback–Leibler divergence as a fingerprint comparison metric and achieve a distance error of 1.3 m in 50% of the test spots. Wu et al. [36] proposed a deep neural network-based fingerprinting system—DNNFi. Compared to DeepFi, DNNFi uses a single model for localization instead of multiple autoencoders for different reference points, yet achieves similar performance. Shi et al. [37] proposed a probabilistic fingerprinting method and adopt a Bayes estimator for comparing online and offline fingerprints. The method achieves 0.96 m of median distance estimation accuracy. Finally, Chen et al. proposed a fingerprinting solution using two methods: EntLoc [3] and AngLoc [4]. The former leverages amplitudes of CSI subcarriers and relies on 50 packets for each location to produce a mean error of 1.84 m. The latter considers both amplitudes and phases of CSI measurements to produce ToF-DoA estimates using the JADE-MUSIC algorithm [38]. The method also uses bivariate kernel regression to combine ToF-DoA pairs with entropy and produces a mean error of 1.18 m. As we focus on the development of our localization solution, we select EntLoc, AngLoc, and DLoc—the latest and most architecturally similar fingerprinting methods—to benchmark our method's performance.

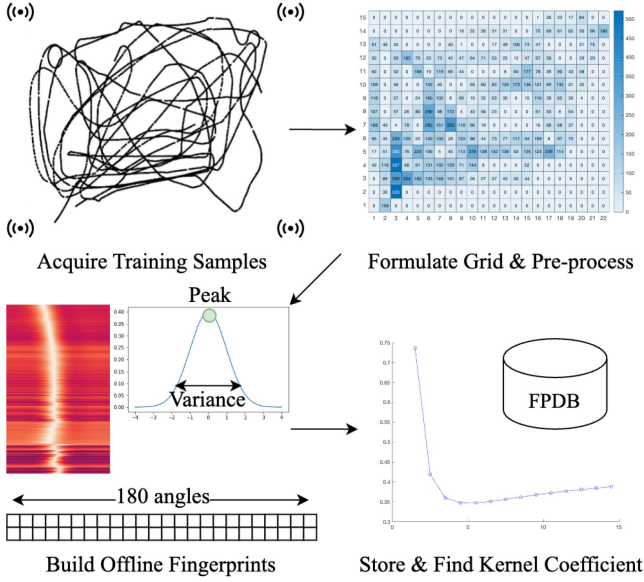


Fig. 2. MobLoc steps: Offline.

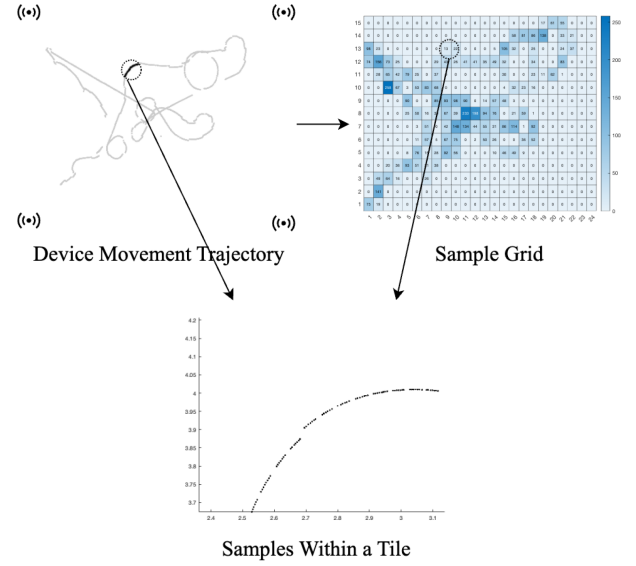


Fig. 4. Grid formulation example.

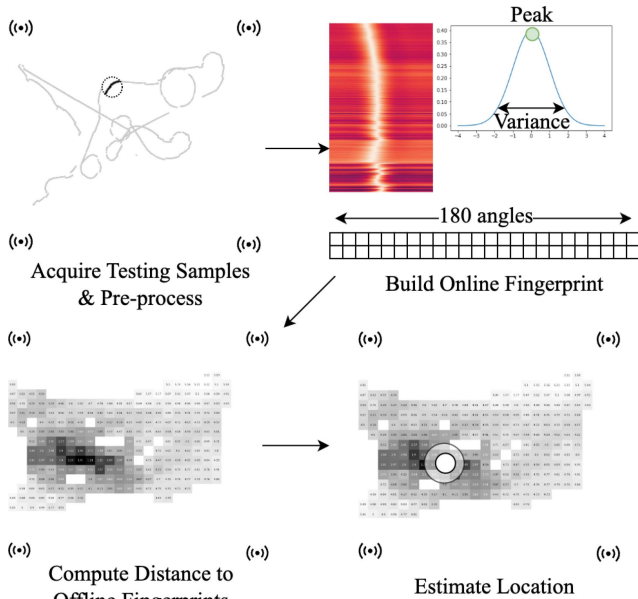


Fig. 3. MobLoc steps: Online mode.

IV. SYSTEM DESIGN

In this section, we present the design of the MobLoc localization system.

A. Overview

The MobLoc system design is illustrated in Figs. 2 and 3. The system is designed following a standard fingerprinting pattern and consists of two modes of operation.

During the offline mode, illustrated in Fig. 2, the system receives a training set of labeled CSI measurements, performs preprocessing, and splits the environment into a grid, where each

tile is mapped to a known set of coordinates. Next, MobLoc constructs fingerprints for each of the tiles with the available CSI measurements. These fingerprints form a database (i.e., a radio map) and are used to estimate the kernel coefficient that is required for the weighted kernel regression step during the later stage of operation.

During the online mode, illustrated in Fig. 3, the system receives a batch of CSI measurements from the testing dataset, performs preprocessing, and constructs a new fingerprint of a similar structure to those stored in the database. The system then calculates the distances between the online and offline fingerprints to evaluate the similarity of the live and prerecorded signals. The resulting distance map is used for weighted kernel regression to predict the device's coordinates, discussed in Section IV-E.

B. Grid Formulation

Each offline fingerprint must be associated with a unique area of the environment. To achieve this, we build a grid of T tiles of equal size, as illustrated in Fig. 4. We use ground truth coordinates of the device's movement trajectory to map CSI data to these tiles, producing a grid of samples. Each tile is assigned an index and ground truth location coordinates, later used for localizing devices. CSI measurements from each of these tiles are used to formulate the fingerprints.

It is important to note that real-world environments rarely provide sufficient data coverage for all available grid tiles. Measurements are often obtained from devices that are in motion. This limits the number of packets available in each of the grid tiles. Further, it makes extracting fingerprinting features more challenging since each emitted packet is sent from a slightly different position, causing phase and amplitude shifts with significant impact.

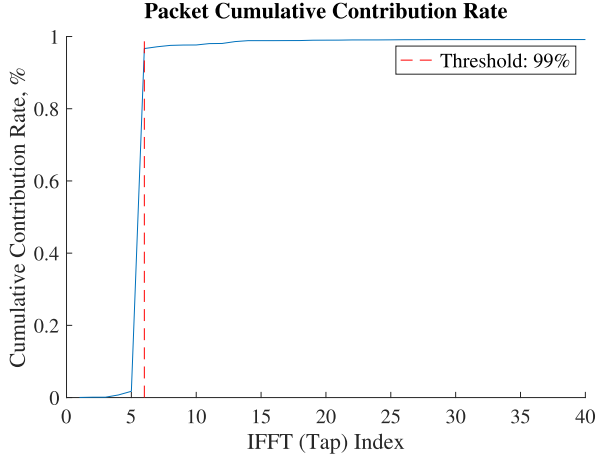


Fig. 5. Tap filtering threshold.

C. Preprocessing

CSI measurements are often affected by a variety of hardware issues, as summarized in Section II-B. Failure to address them significantly impacts the accuracy of direction-of-arrival estimates. Therefore, we must perform a series of preprocessing steps.

1) **Tap Filtering:** First, we reduce the impact of noise on the captured signal. To address this, we adopt tap filtering. The method is outlined in Algorithm 1. For each packet, we first perform an inverse fast Fourier transform (IFFT) to convert signal measurements to the time domain. The values in the resulting array are referred to as *taps*. Next, we use (10) to calculate the cumulative contribution rate for each tap to determine which of them can be eliminated

$$C_k = \frac{\sum_{i=1}^k u_i}{\sum_{i=1}^K u_i}. \quad (10)$$

Here, k is the index of the tap for which C_k is calculated, K is the overall number of taps, and u_i is the value of the i th tap.

Next, filtration is performed by removing the taps with $C_k > C$, where C is determined by evaluating the cumulative contribution rate plot, as displayed in Fig. 5. Here, the x -axis denotes the indices of the IFFT taps, and the y -axis denotes the cumulative contribution of each tap. The red dashed line demonstrates the tap, after which the signal reaches the threshold, and the taps should be truncated.

Finally, the truncated series of taps is converted to a frequency domain with a fast Fourier transform (FFT), resulting in a successful filtration of noise from the packet.

2) **Sampling Frequency Offsets:** Next, we address sampling frequency offsets introduced due to the transmitter's synthesis rate not being perfectly matched to the receiver's sampling rate. In CSI, this offset appears as a phase shift proportional to subcarrier indices, can be identified using linear regression, and is defined as ρ_{sfo} in

$$\rho_{sfo} = \operatorname{argmin}_{\alpha\omega} \sum_k (\varphi + 2\pi f_\delta k\alpha + \omega)^2. \quad (11)$$

Algorithm 1 Tap Filtering

Return:

N-element CSI array;

Inverse transform length $nfft$;

Noise filtering threshold C ;

Ensure:

Obtain CSI taps by applying IFFT on CSI array;

Calculate C_k for each tap using (10);

Trim taps with $C_k > C$;

Apply FFT on the filtered taps to obtain CSI_{filtered} ;

return CSI_{filtered}

Here, α and ω variables can be estimated using polynomial fitting, f_δ is the intersubcarrier frequency delta, k is the index of the subcarrier, and φ is an unwrapped phase of the packet CSI.

3) **Symbol Timing Offsets:** Next, we remove symbol timing offsets. There is asynchrony between transmitters and receivers. However, a receiver must recognize when a data frame begins. To address this, transmitters send pilot frames containing an LTS sequence known to both parties. The receiver performs autocorrelation using the known sequence and uses the identified start of the sequence to understand when to start decoding data. Yet, the identification process is imperfect due to the short size of the LTS sequence.

To address this and estimate the symbol timing offset ρ_{sto} , we apply the following equation:

$$\rho_{sto} = \frac{-2\pi k \operatorname{argmax}(|fft(csi)|^2)}{N} \quad (12)$$

where $\operatorname{argmax}(|fft(csi)|^2)$ is the absolute value of the highest tap of an inverse fast Fourier transform of a packet of CSI samples, and k is the index of the subcarrier.

4) **Interantenna Phase Offsets:** Interantenna offsets contribute significantly to DoA estimation errors. They are usually caused by a lack of synchronization across clocks associated with receiver antennas. According to [20], [29], such offsets are semideterministic and should be computed once per initialization (i.e., reboot) of a device.

To calculate these offsets, one must have access to CSI measurements collected in a wired setup, which protects the transmitted signals from environmental impacts [31], [39]. Alternatively, offsets can be estimated when an emitter is positioned at 0° to the receiver to achieve equal distances between the TX antenna and each of the RX antennas [19].

Given the availability of such CSI measurements, one can estimate interantenna offset ρ_{iao} with the following equation:

$$\rho_{iao} = \text{csi}_{A_i} \times \text{csi}'_{A_{i+1}} \quad (13)$$

where csi_{A_i} is a vector of complex CSI values of a single packet from the i th antenna, and $\text{csi}'_{A_{i+1}}$ is a conjugate transpose of a single packet of CSI from an adjacent antenna $i + 1$.

D. Building a Fingerprint

We build our fingerprinting method under two constraints:

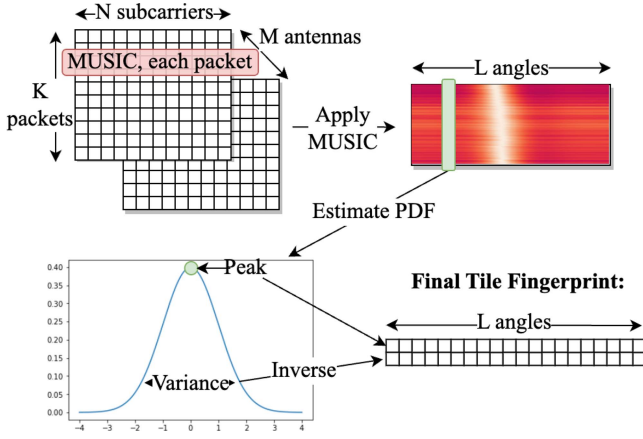


Fig. 6. Building a fingerprint from an access point.

- 1) *Data scarcity*: Real-world scenarios limit the number of measurements available for building fingerprints.
- 2) *Presence of multipath effects*: It is challenging to identify the true line-of-sight signal path between transmitters and receivers, which limits the accuracy of DoA estimation algorithms in real-world conditions.

Given these constraints, we leverage the entire DoA pseudospectrum produced by the MUSIC algorithm to extract fingerprinting features. Since MUSIC can perform even on a single packet of multiantenna CSI measurements, it can withstand data scarcity in producing this pseudospectrum. Further, the use of the entire pseudospectrum, instead of selecting the highest peak, removes the pressure of correct identification of the signal's line-of-sight propagation path, which is the cornerstone of most other localization methods. While the classic implementation of MUSIC relies on the highest peak to identify DoA, we use the entire pseudospectrum to associate the underlying behavior with a tile in the environment.

The process of building a pseudospectrum-based fingerprint from an access point is illustrated in Fig. 6. The algorithm begins by obtaining K CSI packets across M antennas for each tile in the environment grid. Next, the standard MUSIC algorithm is applied to each packet to produce a $K \times L$ -shaped pseudospectrum heatmap, where L represents a number of examined DoA angles. Here, each row represents a pseudospectrum for a single packet, while each column represents a single DoA angle. Each cell in the heatmap represents the power of the pseudospectrum. Next, we estimate the probability density function for each of the DoA angles in the pseudospectrum, as presented by the green vertical line covering the MUSIC pseudospectrum heatmap in the figure. Next, we find the peaks and variances of each of the DoA angle distributions and store them in the final fingerprint array. Importantly, we add a small constant 0.001 to ensure nonzero variance values and inverse them in the fingerprint to later use them as weights for similarity evaluation; i.e., the higher the variance, the less impact on the estimation of distance. The use of peak and variance values for each angle of the pseudospectrum allows MobLoc to overcome a key limitation on the number of packets required to build a fingerprint. Compared

Algorithm 2 Building a Fingerprint

Return:

CSI array $K \times M \times N$, where $K > 0, M > 0, N > 0$;

Ensure

for $i = 1 \dots K$ **do**

Obtain MUSIC pseudospectrum P_i for CSI_i ;

end for

Normalize P between 0 and 1;

Initialize $FP = \text{zeros}(2, L)$;

for $i = 1 \dots L$ **do**

Obtain PDF_i for angle i across all packets K ;

Find maximum of the PDF peak = $\max(PDF_i)$;

Estimate variance with offset $w = \sqrt{\text{var}(P_j) + 0.001}$;

Assign $FP(:, i) = [\text{peak}, \frac{1}{w}]$;

end for

return Fingerprint FP

to EntLoc and AngLoc, which require at least 50 packets per location, this method works with a minimum of two packets per fingerprint and is key to scenarios where collecting an extensive fingerprinting dataset is not feasible (e.g., pedestrian and vehicular tracking in cities).

As a result, for each access point, we obtain a $2 \times N$ fingerprint. Later, for simplicity of storage, we merge fingerprints from all access points to produce a final fingerprint array of size $2 \times KN$.

The overall procedure is detailed in Algorithm 2.

E. Online Mode and Localization

Once the offline fingerprint database is created, real-time device localization can be performed. A consecutively captured series of packets is collected from a device of interest. Measurements are preprocessed, and an online fingerprint is extracted using the series of steps described above.

Next, we use weighted Euclidean distance, as shown in (14), to determine the similarity between an online fingerprint and each of the offline fingerprints in the database

$$D_t = \sqrt{\sum_{i=0}^L (x_i - x'_i)^2 / w_i}. \quad (14)$$

Here, D_t is the estimated distance, w_i is a weighting factor calculated in Algorithm 2 for the i th DoA angle from the offline pseudospectrum, and x_i and x'_i are peaks of the probability density functions for the offline and online pseudospectrum angles, respectively. Importantly, the variances of online fingerprints are not used since the number of available packets in online mode is less likely to accurately represent the volatility of a given DoA angle compared to its offline counterpart.

As a result, we obtain a 2-D array of distance values for the entire environment grid, as shown in Fig. 7. The value in the t th tile represents the similarity estimate D_t . We later use this map to estimate device location.

To predict the coordinates of a device of interest, we follow the process proposed in EntLoc [3] and use weighted kernel

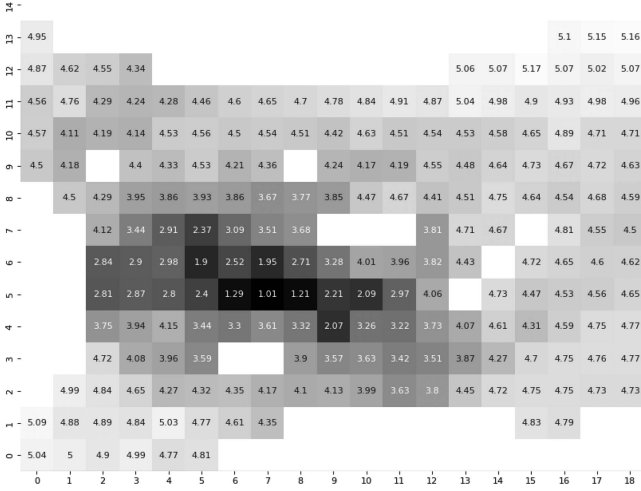


Fig. 7. Fingerprint distance map.

regression. In this approach, often known as weighted centroid estimation [40], [41], fingerprint similarity is used as a weighting factor. In other words, the coordinates of tiles with the highest weights bear the most impact on the predicted coordinates, as detailed the following equation:

$$l = \frac{\sum_{t=1}^T K_t l_t}{\sum_{t=1}^T K_t}. \quad (15)$$

Here, $l = (x, y)$ are coordinates of the predicted location, l_t are the center coordinates of the t th tile, $t \in [1, T]$ tiles, and K_t is a probability kernel of the t th tile that can be calculated as follows:

$$K_t = \exp(-\rho D_t) \quad (16)$$

where D_t is the previously estimated distance between the offline and online fingerprints of the t th tile, and ρ is the kernel coefficient. The latter is estimated separately for each environment using offline fingerprints with leave-one-out cross-validation.

V. PERFORMANCE EVALUATION

In this section, we present a performance evaluation of MobLoc. We begin by describing our experimental setup and benchmarks. Next, we compare MobLoc's performance to other state-of-the-art localization methods.

A. Experimental Setup

Collection of a multi-AP CSI dataset for fingerprinting is a challenging task. In addition to various hardware patching and calibration issues, one must ensure sufficient coverage of the surveyed area. Further, the emitter movement patterns must be similar to the behavior of real-world devices. Achieving these requirements is a lengthy process rife with challenges. Fortunately, there is a publicly available dataset, DLoc [1], which simplifies evaluation and enables benchmarking against other localization methods.

The DLoc dataset was produced by the WCSNG Group at UC San Diego in an indoor environment. Measurements

were collected in two spaces, emitted by a robotic system, MapFind, with a single antenna. The simple space (500 sq. ft.) contained three access points with a direct path available most of the time. The device's movement trajectory in this space is shown in Fig. 8(a). The x -axis and the y -axis denote the size of the surveyed area. The complex space (1500 sq. ft.) contained four access points, with one of them placed behind a wall to introduce non-line-of-sight data. The device's movement trajectory in this space is presented in Fig. 8(a). Similarly, the x -axis and y -axis denote the size of the surveyed area. For each environment, access points collected measurements for 20 min without reboots to avoid inconsistencies in interantenna phase offsets. The research team used common off-the-shelf Quantenna APs, each containing four equidistant antennas. Overall, the dataset provides access to over 100 000 labeled CSI and RSSI measurements.

B. Benchmarks

We seek to compare MobLoc's performance to other localization methods in similar conditions. However, this is a significant challenge. Unfortunately, the majority of publications do not provide access to the source code of the presented methods and use proprietary datasets to evaluate their performance. Therefore, we must implement the methods of interest and benchmark those implementations against MobLoc using the DLoc dataset.

For this experiment, we select four localization methods to compare with MobLoc:

- 1) *SpotFi* [2]: Introduced in 2015, the method has become a staple for CSI-based localization in recent years. The authors observe that the performance of the classic 1-D MUSIC algorithm is limited by the number of receiver antennas, which must be higher than the number of signal propagation paths. This restriction is rarely possible to overcome, which motivates SpotFi to consider 2-D MUSIC, which relies both on the number of antennas and the number of channel subcarriers. This produces 2-D direction-of-arrival and time-of-flight pseudospectrum heatmaps that can help separate DoA peaks across multiple propagation paths. Subsequent temporal analysis of these peaks and a novel peak likelihood estimation method are used to determine the signal direction of arrival.
- 2) *EntLoc* [3]: EntLoc is a CSI-based fingerprinting method. Its fingerprints comprise per-subcarrier autoregressive modeling-based entropy values of signal amplitudes. The method was tested on a custom dataset collected in two classroom environments with the emitter positioned in locations that were 1 m apart. For each offline fingerprint, EntLoc uses at least 50 packets to estimate entropy.
- 3) *AngLoc* [4]: AngLoc is another CSI-based fingerprinting method, a continuation of the EntLoc project. Compared to its predecessor, AngLoc leverages CSI amplitude and phase, producing two fingerprints. For the amplitude-based fingerprint, AngLoc reuses EntLoc's entropy estimation. For the phase-based fingerprint, AngLoc uses the

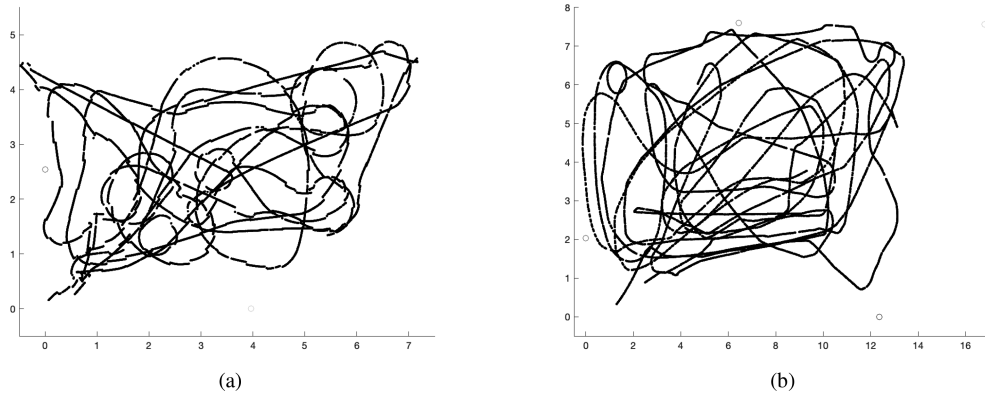


Fig. 8. DLoc movement trajectory (reproduced from [1], [42]). (a) Trajectory in a simple environment. (b) Trajectory in a complex environment.

TABLE I
MOBLOC VERSUS DLOC ACCURACY COMPARISON

Scenario	Median Error (m)		90 th -ile Error (m)	
	MobLoc	DLoc	MobLoc	DLoc
Simple	0.33	0.36	0.82	0.70
Complex	1.15	0.64	2.59	1.60

JADE-MUSIC implementation and records a direction-of-arrival and time-of-flight pair of the most likely pseudospectrum peak. Both fingerprints are combined using bivariate kernel regression to estimate a device's location.

- 4) *DLoc* [1], [43]: Finally, DLoc is a deep learning-based localization method. The authors adapt the deep learning method from computer vision. Instead of directly extracting features from raw CSI measurements, they train a deep learning model on figures of 2-D direction-of-arrival and time-of-flight heatmaps produced by SpotFi. The method achieves submeter localization accuracy and presents the best benchmark for MobLoc.

C. Experimental Results

We evaluate MobLoc performance in both the simple and complex environments from Ayyalasomayajula et al. [1]. For both, we use a 70%/30% data split. For fingerprinting methods (EntLoc, AngLoc, and MobLoc), we group measurements into a grid of tiles, as described in Section IV-B. We limit the number of samples to ten packets per tile. The latter is important because the dataset does not provide an even distribution of samples across all tiles.

Our results are summarized in Figs. 9 and 10 for simple and complex environments, respectively. The blue, black, red, pink, and blue lines display the cumulative distribution functions of localization error in meters for DLoc, MobLoc, SpotFi, AngLoc, and EntLoc, respectively.

From Fig. 9, we see that DLoc reaches 0.36 and 0.7 m at the median and 90th percentiles, respectively. MobLoc demonstrates similar performance. As summarized in Table I, MobLoc achieves 0.33 and 0.82 m localization accuracy at the median and 90th percentiles, respectively. In a complex environment, illustrated in Fig. 10, MobLoc continues outperforming SpotFi,

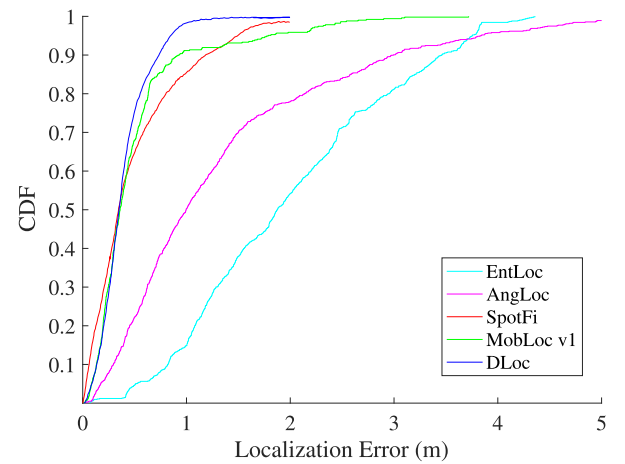


Fig. 9. MobLoc performance in simple environment.

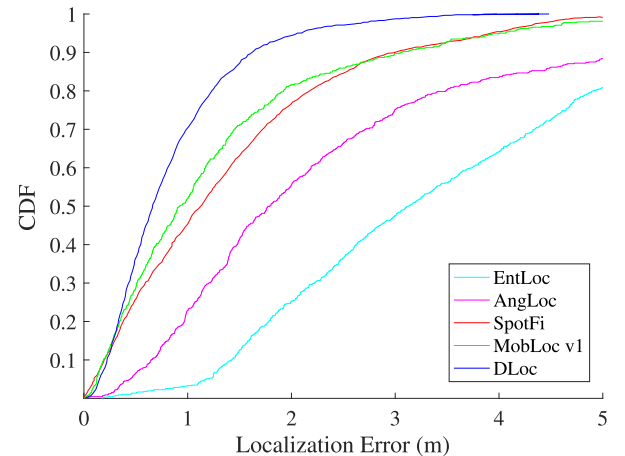


Fig. 10. MobLoc performance in complex environment.

EntLoc, and AngLoc, with 1.15 and 2.59 m at the median and 90th percentiles, respectively. However, MobLoc falls short of DLoc, which reaches 0.64 and 1.6 m at the median and 90th percentiles, respectively. Yet, it offers performance advantages, which we discuss in Section V-D.

TABLE II
MOBLOC VERSUS DLOC TIME MEASUREMENTS

Scenario		MobLoc, seconds		DLoc, seconds	
		Simple Env.	Complex Env.	Simple Env.	Complex Env.
Training		62.212	230.682	2508.225	6563.469
Testing	Building Features	0.576	0.716	0.274	0.385
	Localization	0.042	0.051	0.031	0.043

D. Performance Analysis

First, consider the limited performance of EntLoc. As discussed earlier, EntLoc relies solely on CSI amplitudes and builds fingerprints using per-subcarrier entropy values. The authors have demonstrated effectiveness in their article, which we confirmed in our testing. However, EntLoc relies on emitter stability and large numbers of measurements while building its fingerprints. According to the authors, the method demonstrates ideal performance when 50 packets are available for building each fingerprint. Yet, such quantities are rarely available in real-world situations, especially when the emitter device is in motion.

Next, consider the performance of AngLoc. As discussed, AngLoc combines entropy fingerprints from EntLoc and direction-of-arrival and time-of-flight pairs produced by the JADE-MUSIC algorithm. We can see that the method outperforms EntLoc, both in the original work, as well as in our implementation. However, the method's performance suffers from the way fingerprints are built. The JADE-MUSIC algorithm does not leverage the likelihood estimation method presented by SpotFi and instead selects the first available peak cluster. The selection of a single peak cluster does not capture the behavior of the entire pseudospectrum, which may contain additional valuable insights into signal propagation. This is the motivation for why MobLoc captures the entire pseudospectrum in each fingerprint.

Finally, consider the performance of MobLoc, which exceeds the performance of EntLoc, AngLoc, and SpotFi, but is still outperformed by DLoc. MobLoc leverages the best features of the fingerprinting presented in earlier works, such as measurement calibration, weighted kernel regression, and estimation of signal pseudospectrum. While the method can benefit from high quantities of measurements, it can perform even with two packets of CSI within a tile. Further, MobLoc strives to capture the behavior of the entire pseudospectrum produced by the MUSIC algorithm instead of a single peak, as in AngLoc. Additionally, it avoids the estimation of time-of-flight to increase performance ($O(n)$ versus $O(n * m)$). However, the MobLoc fingerprinting algorithm does not capture the environmental features as well as a multilayer perceptron network, which is the foundation of DLoc. Yet, DLoc requires a computationally intensive model training stage and is more challenging to scale. The latter requires significant amounts of training measurements, and it is often challenging to retrain to adapt the model to changing environmental conditions.

Experimental Comparison: To further evaluate the differences between MobLoc and DLoc, we conducted measurement tests for each of the relevant execution stages. We used the same

configurations adopted to evaluate accuracy. Tests were performed on a machine running Ubuntu 20.04, with 2 AMD EPYC 7313 16-core CPUs @ 3.0 GHz, 128 MB cache, 1024 GB RAM, and NVIDIA A100 PCIe 80 GB GPUs. Time measurements were averaged over ten consecutive runs. The findings are summarized in Table II.

First, we consider the training stage of both methods. For MobLoc, we capture the time required to build the database of fingerprints. As for DLoc, we capture the training time for the deep learning model. Our tests demonstrate that DLoc takes 3.7 times more time to train the model for a simple environment and 2.6 times more time to train the model for a complex environment compared to MobLoc. Such behavior is expected since training a deep learning model on visual data is considerably more resource-intensive compared to the largely analytical algorithmic techniques used in MobLoc.

Second, we consider testing both methods in the online mode for a single location. We split the evaluation into two parts: building features for DLoc and building fingerprints for MobLoc.

- 1) *Building features and fingerprints:* For both methods, this involves converting CSI measurements into a set of location-specific features. In MobLoc, we produce peaks and variances for each value in the DoA pseudospectrum, while in DLoc, CSI is converted into DoA-ToF heatmaps. Our results in Table II indicate that MobLoc takes 2.1 times more time to build online fingerprints for a simple environment and 1.8 times more time for a complex environment due to high kernel smoothing density.
- 2) *Localization:* In MobLoc, the produced peaks and variances are compared to the database of offline fingerprints to find the closest match. In DLoc, the produced 2-D images are propagated through the trained model to obtain the estimated location. Our results in Table II indicate that in both cases, localization speed is very similar.

It is important to note that while MobLoc is implemented in MATLAB, DLoc is implemented both in MATLAB (for producing 2-D heatmap imagery) and PyTorch (for training and testing the deep learning model). According to [44], there are slight differences in execution time for most common operations. This limits our capacity to provide an “apples-to-apples” comparison. However, even this imperfect analysis provides compelling evidence for the performance advantages of MobLoc.

VI. CONCLUSION AND FUTURE WORK

In this article, we presented a novel fingerprinting method and evaluated its localization performance against state-of-the-art passive, CSI-based localization methods. Our objective was to

produce a solution for passive device localization within the constraints of data scarcity and emitter mobility.

We leveraged a publicly available localization dataset, DLoc [1]. During our evaluation, we identified that state-of-the-art fingerprinting methods, such as EntLoc and AngLoc, lack performance when the emitter is constantly in motion.

MobLoc leverages the best features of prior fingerprinting methods (e.g., measurement calibration, weighted kernel regression, and estimation of signal pseudospectrum) to address these issues. It relies on MUSIC pseudospectrum-based fingerprints. MobLoc can benefit from high quantities of measurements, but can perform well close to the state-of-the-art—even with two packets of CSI within a tile. Additionally, we avoid the time-of-flight analysis used in JADE-MUSIC and SpotFi ($O(n * m)$ complexity), achieving $O(n)$ complexity. On the DLoc dataset, MobLoc achieves 0.33 and 0.82 m median and 90th percentile localization error, respectively, in a simple environment. It reaches 1.15 and 2.59 m median and 90th percentile localization errors, respectively, in a complex environment.

Future work on MobLoc will involve four tracks. The first will focus on collecting a larger indoor CSI dataset with access to raw OFDM IQ samples to provide more opportunities to introduce lower level fingerprinting methods and data augmentation strategies. The second will focus on improving MobLoc performance using faster DoA estimation methods, including our previous work with Hankel SVD [19], [45]. The third will consider strategies for adopting MobLoc in a production environment—the Mobintel testbed in downtown West Palm Beach [21], [46]. The final track will explore new learning-based methods to (hopefully) overcome the limitations of DLoc.

REFERENCES

- [1] R. Ayyalasomayajula et al., “Deep learning based wireless localization for indoor navigation,” in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–14.
- [2] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, “Spotfi: Decimeter level localization using WiFi,” in *Proc. ACM Conf. Special Int. Group Data Commun.*, 2015, pp. 269–282.
- [3] L. Chen, I. Ahriz, and D. L. Ruyet, “CSI-based probabilistic indoor position determination: An entropy solution,” *IEEE Access*, vol. 7, pp. 170048–170061, 2019.
- [4] L. Chen, I. Ahriz, and D. L. Ruyet, “AoA-aware probabilistic indoor location fingerprinting using channel state information,” *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10868–10883, Nov. 2020.
- [5] Z. Zhang, S. He, Y. Shu, and Z. Shi, “A self-evolving WiFi-based indoor navigation system using smartphones,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 8, pp. 1760–1774, Aug. 2020.
- [6] F. Zafari, A. Gkelias, and K. K. Leung, “A survey of indoor localization systems and technologies,” *IEEE Commun. Surv. Tut.*, vol. 21, no. 3, pp. 2568–2599, thirdquarter 2019.
- [7] P. Roy and C. Chowdhury, “A survey on ubiquitous WiFi-based indoor localization system for smartphone users from implementation perspectives,” *CCF Trans. Pervasive Comput. Interaction*, vol. 4, no. 3, pp. 298–318, 2022.
- [8] U. D. Ulusar, G. Celik, and F. Al-Turjman, “Cognitive RF-based localization for mission-critical applications in smart cities: An overview,” *Comput. Elect. Eng.*, vol. 87, 2020, Art. no. 106780.
- [9] M. Golestanian, H. Lu, C. Poellabauer, and J. Kenney, “RSSI-based ranging for pedestrian localization,” in *Proc. IEEE 88th Veh. Technol. Conf.*, 2018, pp. 1–5.
- [10] F. Bao, S. Mazokha, and J. O. Hallstrom, “Mobintel: Passive outdoor localization via RSSI and machine learning,” in *Proc. 17th Int. Conf. Wireless Mobile Comput., Netw. Commun.*, 2021, pp. 247–252.
- [11] Z. Gao, Y. Gao, S. Wang, D. Li, and Y. Xu, “CRISLoc: Reconstructable CSI fingerprinting for indoor smartphone localization,” *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3422–3437, Mar. 2021.
- [12] F. Hejazi, K. Vuckovic, and N. Rahnnavard, “Dyloc: Dynamic localization for massive MIMO using predictive recurrent neural networks,” in *Proc. INFOCOM IEEE Conf. Comput. Commun.*, 2021, pp. 1–9.
- [13] W. Wei, J. Yan, X. Wu, C. Wang, and G. Zhang, “CSI fingerprinting for device-free localization: Phase calibration and SSIM-based augmentation,” *IEEE Wireless Commun. Lett.*, vol. 11, no. 6, pp. 1137–1141, Jun. 2022.
- [14] A. Foliadis, M. H. C. Garcia, R. A. Stirling-Gallacher, and R. S. Thomä, “CSI-based localization with CNNs exploiting phase information,” in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–6.
- [15] R. Yang, X. Yang, J. Wang, M. Zhou, Z. Tian, and L. Li, “Decimeter level indoor localization using WiFi channel state information,” *IEEE Sensors J.*, vol. 22, no. 6, pp. 4940–4950, Mar. 2022.
- [16] Y. Ma, G. Zhou, and S. Wang, “WiFi sensing with channel state information: A survey,” *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–36, 2019.
- [17] N. Tadayon, M. T. Rahman, S. Han, S. Valaee, and W. Yu, “Decimeter ranging with channel state information,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3453–3468, Jul. 2019.
- [18] F. Gringoli, M. Schulz, J. Link, and M. Hollick, “Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets,” in *Proc. 13th Int. Workshop Wireless Netw. Testbeds, Exp. Eval. Characterization*, 2019, pp. 21–28.
- [19] S. Mazokha, S. Naderi, G. I. Orfanidis, G. Sklivanitis, D. A. Pados, and J. O. Hallstrom, “Single-sample direction-of-arrival estimation for fast and robust 3D localization with real measurements from a massive MIMO system,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.
- [20] D. Zhang, Y. Hu, Y. Chen, and B. Zeng, “Calibrating phase offsets for commodity WiFi,” *IEEE Syst. J.*, vol. 14, no. 1, pp. 661–664, Mar. 2020.
- [21] S. Mazokha, F. Bao, J. Zhai, and J. O. Hallstrom, “MobIntel: Sensing and analytics infrastructure for urban mobility intelligence,” *Pervasive Mobile Comput.*, vol. 77, 2021, Art. no. 101475.
- [22] O. Waltari and J. Kangasharju, “The wireless shark: Identifying WiFi devices based on probe fingerprints,” in *Proc. First Workshop Mobile Data*, 2016, pp. 1–6.
- [23] L. Oliveira, D. Schneider, J. D. Souza, and W. Shen, “Mobile device detection through WiFi probe request analysis,” *IEEE Access*, vol. 7, pp. 98579–98588, 2019.
- [24] Z. Li and X. Rao, “Toward long-term effective and robust device-free indoor localization via channel state information,” *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3599–3611, Mar. 2022.
- [25] S. Mazokha, “Mobloc repository,” 2023. [Online]. Available: <https://github.com/I-SENSE/mobloc-paper>
- [26] P. Sure and C. M. Bhuma, “A survey on OFDM channel estimation techniques based on denoising strategies,” *Eng. Sci. Technol., Int. J.*, vol. 20, no. 2, pp. 629–636, 2017.
- [27] Y. Xie, Z. Li, and M. Li, “Precise power delay profiling with commodity Wi-Fi,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1342–1355, Jun. 2019.
- [28] D. Vasisht, S. Kumar, and D. Katabi, “Decimeter-level localization with a single WiFi access point,” in *Proc. 13th USENIX Symp. Networked Syst. Des. Implementation*, 2016, pp. 165–178.
- [29] A. Zubow, P. Gawłowicz, and F. Dressler, “On phase offsets of 802.11 AC commodity WiFi,” in *Proc. 16th Annu. Conf. Wireless On-demand Netw. Syst. Serv. Conf.*, 2021, pp. 1–4.
- [30] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.
- [31] J. Xiong and K. Jamieson, “Arraytrack: A Fine-Grained Indoor Location System,” in *Proc. USENIX Symp. Networked Syst. Des. Implementation*, 2013, pp. 71–84.
- [32] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, “CSI-MIMO: Indoor Wi-Fi fingerprinting system,” in *Proc. 39th Annu. IEEE Conf. Local Comput. Netw.*, 2014, pp. 202–209.
- [33] X. Wang, L. Gao, and S. Mao, “PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach,” in *Proc. IEEE Glob. Commun. Conf.*, 2015, pp. 1–6.
- [34] X. Wang, L. Gao, S. Mao, and S. Pandey, “CSI-based fingerprinting for indoor localization: A deep learning approach,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

- [35] Y. Xiao, S. Zhang, J. Cao, H. Wang, and J. Wang, "Exploiting distribution of channel state information for accurate wireless indoor localization," *Comput. Commun.*, vol. 114, pp. 73–83, 2017.
- [36] G.-S. Wu and P.-H. Tseng, "A deep neural network-based indoor positioning method using channel state information," in *Proc. Int. Conf. Comput., Netw. Commun.*, 2018, pp. 290–294.
- [37] S. Shi, S. Sigg, L. Chen, and Y. Ji, "Accurate location tracking from CSI-based passive device-free probabilistic fingerprinting," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5217–5230, Jun. 2018.
- [38] M. C. Vanderveen, C. B. Papadias, and A. Paulraj, "Joint angle and delay estimation (JADE) for multipath signals arriving at an antenna array," *IEEE Commun. Lett.*, vol. 1, no. 1, pp. 12–14, Jan. 1997.
- [39] Y. Zheng, M. Sheng, J. Liu, and J. Li, "OpArray: Exploiting array orientation for accurate indoor localization," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 847–858, Jan. 2019.
- [40] J. Wang, P. Urriza, Y. Han, and D. Cabric, "Weighted centroid localization algorithm: Theoretical analysis and distributed implementation," *IEEE Trans. wireless Commun.*, vol. 10, no. 10, pp. 3403–3413, Oct. 2011.
- [41] Z.-M. Wang and Y. Zheng, "The study of the weighted centroid localization algorithm based on RSSI," in *Proc. Int. Conf. Wireless Commun. Sensor Netw.*, 2014, pp. 276–279.
- [42] R. Ayyalasomayajula, "Wireless indoor localization dataset (WILD)," 2020. [Online]. Available: <https://wcsng.ucsd.edu/wild/>
- [43] R. Ayyalasomayajula, "DLOC repository," 2023. [Online]. Available: https://github.com/ucsdwcsng/DLoc_pt_code
- [44] A. Weiss and A. Elsherbeni, "Computational performance of MATLAB and python for electromagnetic applications," in *Proc. Int. Appl. Comput. Electromagn. Soc. Symp.*, 2020, pp. 1–2.
- [45] G. I. Orfanidis, D. A. Pados, G. Sklivanitis, E. S. Bentley, J. Suprenant, and M. J. Medley, "Single-sample direction-of-arrival estimation by Hankel-matrix decompositions," in *Proc. 56th Asilomar Conf. Signals, Syst., Comput.*, 2022, pp. 1026–1030.
- [46] S. Mazokha, F. Bao, G. Sklivanitis, and J. O. Hallstrom, "Urban-scale testbed infrastructure for data-driven wireless research," in *Proc. IEEE 4th 5G World Forum*, 2021, pp. 517–522.