Capítulo 6

Wireless sensing: Low-cost monitoring using Wi-Fi signals and IoT devices

Samuel Vieira Ducca, Henrique Freire da Silva, Artur Jordao Lima Correia, Cintia Borges Margi

Abstract

In this work, we discuss the emerging field of wireless sensing, focusing primarily on cost-effective Wi-Fi sensing implementations that leverage Channel State Information (CSI). We present the theoretical foundations of signal extraction and processing methodologies, alongside machine learning techniques for effective inference. To demonstrate practical applications, we analyze a real-world case study and provide a hands-on implementation for human presence detection using our open-source Wisensing-ESP32 framework and commodity ESP32 devices. Lastly, we conclude with an examination of current technical challenges and promising research directions in this rapidly evolving domain.

6.1. Introduction

Wireless communication technologies — once developed primarily for the purpose of data exchange — are increasingly being repurposed for sensing tasks, giving rise to the rapidly expanding field of Integrated Sensing and Communications (ISAC). This paradigm shift leverages the pervasive nature of wireless protocols such as Wi-Fi and Bluetooth, which have become integral to modern digital infrastructure, particularly with the widespread adoption of the Internet of Things (IoT). The omnipresence of these signals in our daily environments has enabled the development of cost-effective and unobtrusive sensing solutions that eliminate the need for specialized hardware. By opportunistically reusing existing communication waveforms, ISAC systems facilitate applications ranging from human activity recognition and health monitoring [1] to environmental mapping and object tracking. For instance, wireless sensing has demonstrated capabilities such as gesture recognition, including the interpretation of sign language [2], non-contact vital sign monitoring [3, 1], and even 3D reconstruction of physical spaces using reflected radio waves [4].

Among the various communication standards investigated for ISAC, Wi-Fi (IEEE 802.11) stands out as one of the most widely adopted protocols for sensing. Its ubiquity, low deployment cost, and compatibility with commercial off-the-shelf (COTS) devices make it an attractive platform for pervasive sensing applications. More importantly, Wi-Fi provides access to rich physical-layer information in the form of Channel State Information (CSI), which captures fine-grained characteristics of the wireless channel, including amplitude and phase per subcarrier. This capability allows Wi-Fi to serve as a high-resolution sensing method that can be implemented using the embedded antennas of low-power IoT devices, thus enabling a broad array of sensing applications in homes, vehicles, healthcare settings, and industrial environments.

This chapter provides a comprehensive overview of wireless sensing across multiple communication protocols, with a particular emphasis on low-cost Wi-Fi sensing. We begin by examining various wireless standards — including LoRaWAN, Bluetooth, and Wi-Fi — in terms of their capabilities and limitations for sensing applications (Section 6.2). Next, we explore the theoretical foundations of Wi-Fi-based sensing, including the structure and interpretation of CSI, as well as signal processing and machine learning techniques for feature extraction and inference (Section 6.3). To demonstrate practical applicability, we present a detailed case study in Section 6.4, where Wi-Fi sensing is employed as a low-cost solution for detecting hazardous animal crossings on rural roadways. This example illustrates how Wi-Fi can serve as a viable sensing alternative in resource-constrained scenarios.

To bridge the gap between theory and real-world implementation, we introduce Wisensing-ESP32 in Section 6.5, an open-source, end-to-end toolkit for real-time Wi-Fi sensing using low-cost ESP32 microcontrollers. This section walks the reader through the complete development pipeline — from raw CSI signal processing to training and deploying lightweight machine learning models for real-time person detection. Finally, in Section 6.6, we discuss the current technical challenges and emerging research directions in Wi-Fi sensing, such as scalability, resilience to environmental variability, and privacy concerns. We conclude with our final remarks in Section 6.7.

6.2. Wireless sensing using communication protocols

As both radar and wireless sensing systems rely on the propagation and reflection of electromagnetic waves, many of the physical factors that influence radar performance similarly affect the performance of sensing systems based on wireless communication protocols [5, 6]. A key performance determinant is the signal-to-noise ratio (SNR), which is influenced by system parameters such as antenna gain and transmitted power. These parameters govern the system's sensitivity and its ability to distinguish relevant data from background noise. The angular resolution is primarily determined by the ratio of the signal wavelength to the effective aperture size of the antenna array. Higher frequencies, which correspond to shorter wavelengths, can enable finer angular resolution when the array size is maintained or increased. However, they also introduce higher path loss and are more susceptible to blockage and scattering from obstacles, which limits their effectiveness in cluttered or non-line-of-sight (NLOS) environments. Another essential parameter is bandwidth, which directly impacts spatial (or range) resolution. Systems with greater bandwidth can resolve objects that are closer together by providing more detailed temporal information about reflected signals. However, accessing wider bandwidths often requires operation in less congested or higher-frequency spectrum bands, which may not be available or practical depending on regulatory and hardware constraints. These trade-offs make the selection of communication protocols and frequency bands highly application-dependent, with certain wireless technologies offering advantages in specific sensing and communication tasks while posing limitations in others.

Given these considerations, the performance of wireless sensing systems varies significantly across different communication protocols, each defined by unique frequency ranges, bandwidth availability, modulation techniques, and physical layer characteristics. In this section, we present a comparative overview of wireless sensing capabilities across various frequency bands and communication standards. Our analysis focuses on device-free (or passive) sensing, where the sensing system relies solely on ambient wireless signals without requiring active participation or signal transmission by the target. This paradigm is particularly relevant for unobtrusive monitoring applications, such as human presence detection, activity recognition, and environmental sensing, and poses unique challenges and opportunities compared to traditional active sensing approaches that utilize devices like smartphones or wearables.

6.2.1. LoRaWAN (Sub-1 GHz)

LoRaWAN [7] is a widely adopted communication protocol designed for low-power, long-range wireless transmission, operating in sub-GHz frequency bands typically ranging from 433 MHz to 915 MHz. Unlike conventional wireless communication standards such as Wi-Fi, which employ Orthogonal Frequency-Division Multiplexing (OFDM) to divide the available bandwidth into multiple subcarriers, LoRaWAN leverages Chirp Spread Spectrum (CSS) modulation. In this technique, the entire channel bandwidth — commonly 125 kHz — is utilized to transmit data through frequency-modulated chirp pulses. By sweeping the signal across the entire allocated bandwidth, CSS enables energy dispersion over time and frequency, improving resilience to interference and enhancing detection reliability in challenging propagation environments. Additionally, the low symbol rate and coherent detection techniques contribute to some robustness against Doppler effects, particularly in low-mobility scenarios.

The combination of lower operational frequencies and CSS modulation grants LoRaWAN superior resistance to environmental interference and extends its communication range, even in non-line-of-sight conditions. While this makes LoRaWAN a promising candidate for wide-area sensing applications, it introduces challenges for fine-grained sensing tasks due to its limited sensitivity and constrained data throughput. In particular, the protocol's strict duty cycle regulations significantly restrict the maximum achievable sampling frequency, thereby limiting its applicability in scenarios that require high temporal resolution.

Despite these constraints, LoRaWAN has demonstrated potential in some sensing applications. Notable examples include soil moisture detection using Received Signal Strength Indicator (RSSI) measurements [8] and human identification based on gait analysis [9]. However, existing studies often overlook the impact of the protocol's duty cycle limitations on the sensing performance, an aspect that warrants further investigation to better assess LoRaWAN's viability for sensing tasks in real-world environments.

6.2.2. Wi-Fi (2.4 to 6 GHz)

Wi-Fi (IEEE 802.11) has emerged as the predominant wireless protocol for sensing applications, primarily due to its widespread adoption, cost-effectiveness, and the ability to extract detailed physical-layer information through Channel State Information. Operating in the unlicensed 2.4 GHz and 5 GHz Industrial, Scientific, and Medical (ISM) bands — and more recently in the 6 GHz band with IEEE 802.11ax — Wi-Fi utilizes Orthogonal Frequency Division Multiplexing (OFDM) to divide its available bandwidth into multiple subcarriers. This multicarrier modulation technique enhances robustness against frequency-selective fading and multipath propagation, making Wi-Fi particularly suitable for indoor and cluttered environments where signal reflections are prevalent. The CSI extracted from Wi-Fi signals captures both amplitude and phase information for each subcarrier, offering a fine-grained view of the wireless channel's state. We discuss CSI and OFDM theoretical aspects in more detail in Section 6.3.1.

Given the widespread use of Wi-Fi in residential, commercial, and vehicular environments, one of the most extensively explored application domains is human activity detection and classification, particularly in indoor settings. For example, [10] developed a child presence detection system capable of identifying children inadvertently left in vehicles with over 99% accuracy, leveraging existing in-car Wi-Fi infrastructure. Other notable human-centric applications include gesture recognition — such as sign language interpretation with accuracy exceeding 81% [2] — and health monitoring tasks like respiration tracking and heart rate variability estimation [1].

Beyond human detection, Wi-Fi sensing has also been applied to a variety of contexts not directly related to the human body. In security and transportation, [11] employed polarized directional antennas to identify 14 types of materials concealed within luggage, achieving classification accuracy of up to 97%. In the agricultural domain, [12] utilized both amplitude and phase CSI for assessing wheat moisture content, observing improved performance when incorporating phase difference data. Similarly, [13] demonstrated the use of Wi-Fi sensing to evaluate the ripeness of fruits such as kiwis and avocados.

Collectively, these studies underscore the versatility and robustness of Wi-Fi CSI as a sensing method. Its integration with signal processing and machine learning techniques, further explored in Sections 6.3.3 and 6.3.4, enables a broad spectrum of applications in both human-centric and environmental sensing. Despite its numerous advantages, Wi-Fi also presents some limitations as a sensing platform. Wi-Fi operates primarily in congested, unlicensed ISM bands that are susceptible to interference from other devices such as Bluetooth, microwave ovens, and neighboring Wi-Fi networks. Furthermore, Wi-Fi devices are generally not optimized for sensing tasks; variations in hardware, driver implementations, and firmware across vendors can result in inconsistent CSI quality and sampling rates. These factors hinder reproducibility and generalizability of sensing models across different platforms.

Addressing these challenges is a key motivation behind ongoing standardization efforts, such as IEEE 802.11bf [14], which aims to unify sensing capabilities and improve interoperability across devices. We discuss this protocol in more detail in Section 6.6.3.

6.2.3. Bluetooth (2.4 GHz)

While most wireless sensing systems documented in the literature leverage Wi-Fi technology, comparatively fewer studies have explored device-free sensing using Bluetooth protocols [15]. Bluetooth, which operates in the 2.4 GHz ISM band, has been applied in some sensing tasks such as vehicle detection through signal attenuation [16] and occupancy estimation in indoor environments using RSSI measurements [17].

Unlike Wi-Fi — which supports the extraction of Channel State Information — Bluetooth is limited to scalar RSSI measurements. RSSI provides a coarse measure of signal intensity at a given time and is highly susceptible to multipath fading, noise, and temporal variability, making it inherently less robust for fine-grained sensing tasks. In contrast, CSI offers significantly more granular and stable channel information, enabling a broad range of advanced sensing applications, particularly for passive human detection and activity recognition.

Despite these limitations, Bluetooth may offer compelling advantages for wireless sensing, notably its low power consumption and widespread adoption in consumer devices. Emerging features in Bluetooth 5.1 [18], such as Angle of Arrival (AoA) and Angle of Departure (AoD) information, introduce spatial awareness capabilities that could enhance the accuracy of Bluetooth-based sensing [15]. However, these features remain underexplored in the context of device-free sensing, warranting further investigation.

6.2.4. Cellular Networks (sub-1 GHz to 5 GHz)

While millimeter wave (mmWave) commercial microwave links — commonly used as backhaul infrastructure for cellular networks — have been extensively investigated for environmental sensing applications such as rainfall estimation [19] and atmospheric water vapor measurement [20], the lower-frequency radio access links between cellular towers and end-user devices have received comparatively limited attention in the context of sensing. Nonetheless, a few studies have explored the feasibility of leveraging these sub-6 GHz cellular frequencies for environmental and activity monitoring. For instance, the work by Hunt et al. [21] demonstrated the potential of using RSSI metrics from 2.4 GHz cellular signals for estimating vegetation biomass. In another study, Chen et al. [22] utilized CSI from 2.165 GHz LTE signals to recognize hand gestures, illustrating that even conventional LTE infrastructure may be repurposed for fine-grained motion sensing.

Despite these promising results, the adoption of cellular technologies for sensing purposes remains limited due to several challenges. These include the higher cost of commercial-grade equipment, the proprietary nature of many cellular protocols, and the technical complexity associated with accessing and processing lower-layer cellular signal metrics. As a result, wireless sensing research has traditionally favored more accessible and cost-effective protocols such as Wi-Fi. Although emerging 5G networks operate partially in the sub-6 GHz band — mainly around 2 to 3 GHz — their sensing capabilities also extend towards the mmWave spectrum and will be discussed in greater depth in the subsequent section on mmWave-based sensing.

6.2.5. 5G and beyond (mmWave)

As discussed in the previous section, sensing using millimeter-wave frequencies has long been explored in specific applications, particularly in rainfall estimation via commercial microwave backhaul links in cellular networks. These systems exploit signal attenuation characteristics — analogous to RSSI-based methods — to infer the presence and intensity of precipitation, with particular sensitivity to small water droplets due to the short mmWave wavelengths [19]. However, these microwave links are typically static and sparsely deployed, which significantly constrains their applicability to localized and environmental sensing tasks.

The emergence and gradual global deployment of 5G New Radio (NR) has substantially broadened the availability and spatial coverage of mmWave communication signals. Operating across a wide frequency spectrum that includes both sub-6 GHz (FR1) and mmWave bands (FR2, up to 71 GHz), 5G enables a new class of ISAC applications by combining high bandwidth, large antenna arrays for massive MIMO, low-latency communication, and highly directional beamforming capabilities [23]. These features make 5G especially attractive for high-resolution sensing. Recent works have demonstrated mmWave-based systems capable of detecting human presence and activity [24], performing 3D environmental reconstruction [4], and monitoring health indicators such as respiration and heart rate variability [3]. Even non-traditional domains such as agriculture are seeing promising results, such as the non-invasive estimation of fruit sugar content using mmWave spectroscopy [25]. Despite the high directionality and attenuation associated with mmWave signals, some studies have shown the feasibility of sensing in non-line-of-sight (NLOS) scenarios, leveraging signal reflections and advanced CSI processing algorithms [26].

Nonetheless, significant barriers remain to widespread research and adoption of mmWave-based sensing. Commercial 5G NR equipment, particularly mmWave-capable base stations and user equipment, remains prohibitively expensive — often exceeding tens of thousands of dollars in cost. While Software-Defined Radios (SDRs) such as the USRP X310 or B210 series have been employed as more flexible and comparatively lower-cost platforms for prototyping and experimentation [27], they still entail substantial financial and technical overhead compared to Wi-Fi or Bluetooth-based systems. Moreover, extracting and interpreting raw physical layer signals from 5G NR transmissions requires a deeper expertise in signal processing and protocol stack decoding, further constraining entry to this field.

Looking forward, integrated sensing is positioned to become a core feature of sixth-generation (6G) wireless networks [28]. While still in the early stages of standardization and conceptual development, 6G envisions operation across a wide frequency range—from 400 MHz to 7 GHz (low-band), 7 GHz to 24 GHz (upper mid-band), and 24 GHz to 70+ GHz (mmWave and sub-THz bands). Anticipated enhancements include more intelligent beamforming, holographic MIMO surfaces [29], and advanced modulation techniques the could increase both sensing accuracy and energy efficiency. However, despite its potential, 6G sensing remains largely speculative at this stage. Widespread deployment at a reasonable cost is likely several years to a decade away, meaning that while 6G may eventually redefine the ISAC paradigm, it is not expected to provide accessible solutions for sensing applications in the near future.

6.3. Wi-Fi sensing

Among the wireless communication protocols previously discussed, Wi-Fi (IEEE 802.11) has emerged as the most promising platform for low-cost sensing, owing to its widespread deployment in IoT devices and its inherent ability to provide detailed physical-layer information. In particular, Wi-Fi offers access to CSI, which captures the fine-grained characteristics of the wireless channel and serves as a rich source of data for sensing applications. In this section, we review the fundamental theoretical concepts underpinning Wi-Fi sensing, including the computation and processing of CSI, common sensing architectures, and widely adopted machine learning techniques used to infer environmental and activity-related information from Wi-Fi signals.

6.3.1. Channel State Information (CSI)

Channel State Information characterizes the wireless communication link between transceivers by quantifying the amplitude and phase variations induced by multipath propagation and other channel impairments [30]. CSI enables channel equalization, MIMO precoding, and, more recently, advanced sensing applications [14]. To capture the time-varying nature of the wireless channel, we model the Channel Frequency Response (CFR) $H(f,t)$ as a summation over multiple propagation paths. Each path contributes a complex amplitude $\alpha_l(t)$ and a delay-induced phase rotation governed by $\tau_l(t)$ [31], according to Equation 1.

$$H(t,f) = \sum_{l} \alpha_l(t) e^{j2\pi f \tau_l(t)} \tag{1}$$

The IEEE 802.11 standard [32] (since the *g* amendment) adopts Orthogonal Frequency-Division Multiplexing as the modulation scheme. This technique facilitates channel estimation by partitioning the available bandwidth into multiple orthogonal subcarriers (tones) that carry modulated signals, effectively decomposing the data stream across the frequency domain and providing an individual channel estimate for each subcarrier. The subcarrier spacing is defined as $\Delta f = 1/T_s$, where T_s is the symbol period; this spacing establishes orthogonality and prevents mutual interference. In an OFDM system, subcarriers fall into one of three categories:

- **Data Subcarriers**: carry user data.
- **Pilot Subcarriers**: provide reference signals for channel estimation, synchronization, and tracking. These subcarriers use Binary Phase Shift Keying (BPSK) modulation.
- **Null Subcarriers**: act as guard bands to reduce interference with adjacent channels and comply with spectral masks.

Figure 6.1 illustrates a simplified spectral occupancy profile of a standard 20 MHz Wi-Fi channel with 64 subcarriers, where each subcarrier k follows the channel central frequency $f_k = f_c + k\Delta f$ with $\Delta f = 312.5\,\text{kHz}$.

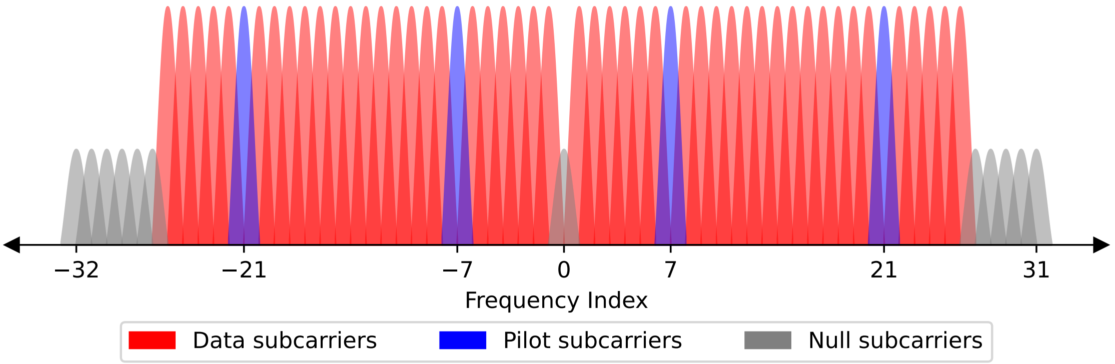

Figure 6.1: OFDM subcarriers on a 20 MHz Wi-Fi channel.

The physical layer implements a structured preamble that supports synchronization and channel estimation. This preamble includes a Short Training Field (STF), a Long Training Field (LTF), and a Signal (SIG) field [32]. The training fields embed predefined symbols in the data and pilot subcarriers depicted in Figure 6.1. The STF employs 12 evenly spaced pilot tones to achieve signal detection, Automatic Gain Control (AGC) convergence, timing synchronization, and coarse frequency offset estimation, while the LTF uses all 52 available tones to perform fine frequency acquisition and channel estimation. The SIG field conveys essential transmission parameters, such as the modulation and coding scheme and the Physical layer Service Data Unit (PSDU) length.

Wi-Fi systems support various modes—non-HT, HT, VHT, and HE—to meet different performance and compatibility requirements. Each mode tailors the preamble by incorporating appropriate versions of the STF, LTF, and SIG fields. In non-HT mode, the legacy preamble consists of the L-STF, L-LTF, and L-SIG fields to ensure backward compatibility. Additional fields follow in HT, VHT, and HE modes to accommodate higher throughput, multiple spatial streams, and denser deployment environments. Figure 6.2 presents the structure of a mixed-mode Physical layer Protocol Data Unit (PPDU) that includes legacy support.

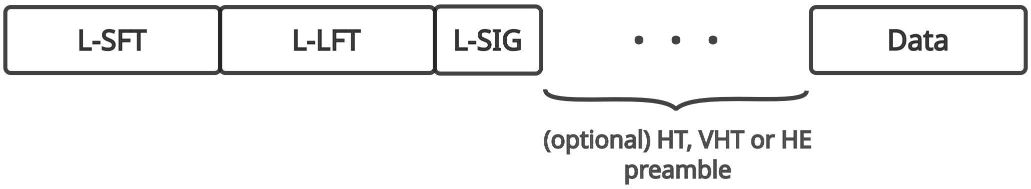

Figure 6.2: IEEE 802.11 PPDU. Adapted from [32] © 2021 IEEE.

The design of the preamble LTF allows the receiver to convert time-domain training signals into the frequency domain, using the Discrete Fourier Transform (DFT), and estimate the channel response on each OFDM subcarrier. The received signal at the kth subcarrier on the nth OFDM symbol is given by Equation 2, where $Y_{n,k}$ represents the detected signal, $X_{n,k}$ denotes the transmitted signal based on the LTF for the given mode, and $N_{n,k}$ is the additive Gaussian noise at the receiver [30].

$$Y_{n,k} = H_{n,k}X_{n,k} + N_{n,k} \tag{2}$$

This per-subcarrier, per-frame CSI estimation, directly derived from the training symbols included in every received frame, enables precise tracking of multipath effects and rapid channel dynamics. By continuously extracting CSI, Wi-Fi systems generate a real-time map of the propagation environment that supports advanced sensing applications, such as motion detection, occupancy estimation, and other context-aware functionalities.

6.3.2. Sensing arrangements

Wi-Fi sensing can be performed using three different sensing arrangements [14], as illustrated in Figure 6.3: (i) monostatic, (ii) bistatic and (iii) multistatic. We describe each of the arrangements as follows:

1. **Monostatic sensing:** in this configuration, the sensing transmitter and receiver are the same device. The measurement process is akin to radar, where the device measures the echoes of its own transmission to collect sensing data. This approach has limited use in current Wi-Fi sensing systems that operate in sub-7 GHz frequency bands. However, it will be supported in the upcoming IEEE 802.11bf protocol [14], utilizing the 60 GHz band.

2. **Bistatic Sensing:** the sensing transmitter and receiver are separate devices, commonly an Access Point (AP) and a Client Station (STA). This is the most commonly used sensor arrangement for Wi-Fi sensing using low-cost IoT devices.

3. **Multistatic Sensing:** this extends bistatic sensing by incorporating multiple sensing transmitters or receivers. Typically, this setup involves an AP and several STAs, enabling more complex sensing measurements across multiple devices.

6.3.3. Data collection and processing

CSI data represents the channel at each subcarrier as a complex number. Extracting meaningful environmental features from these raw measurements involves several signal processing techniques. We demonstrate the extraction of amplitude, phase, power delay profile (PDP), and spectrograms from a series of 400 CSI measurements sampled at 100 Hz, allowing for direct comparison. Figure 6.4 illustrates our measurement scenario — a cow crossing the 12 m line-of-sight (LOS) between two transceivers — and serves as the basis for all subsequent analysis.

Basic features include the amplitude (A_k) and phase (ϕ_k), defined in Equations 3 and 4. They represent the signal gain, that is often an attenuation, and the transmission lag.

$$A_k = \sqrt{(\Re(H_k))^2 + (\Im(H_k))^2} \tag{3}$$

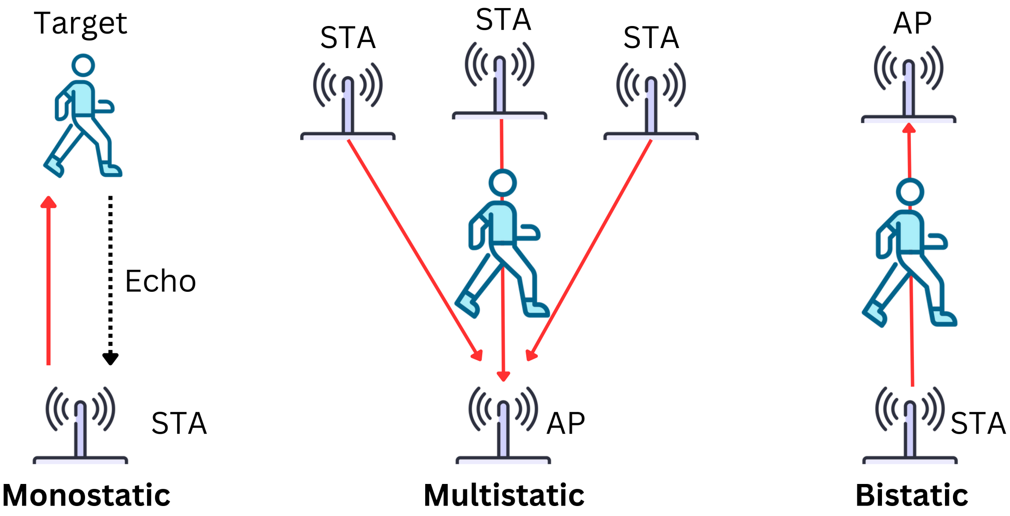

Figure 6.3: Diagram illustrating monostatic, multistatic and bistatic Wi-Fi sensing.

$$\phi_k = atan2(\Im(H_k), \Re(H_k)) \tag{4}$$

Figure 6.5 displays heatmaps for the absolute amplitude (converted to dB using $A_{db} = 20\log(A)$) and unwrapped phase for each subcarrier at every sampled frame. For the extracted amplitude, there is a noticeable disturbance around frame 200. This indicates a break of LOS by the cow crossing, increasing the attenuation of propagating signals.

Figure 6.4: Cow crossing on pasture background. [33]

A complementary view involves the temporal difference of amplitude and phase. Calculating the difference between consecutive samples, $\Delta x[t] = x[t] - x[t-1]$, emphasizes relative changes in the signal. Figure 6.6 presents heatmaps for the relative amplitude and phase across the samples.

In this example, amplitude remains relatively consistent. Most frames show minimal variation, while some capture the event with more distinct values. A positive change suggests lower attenuation and fewer obstructions between transceivers, whereas a nega-

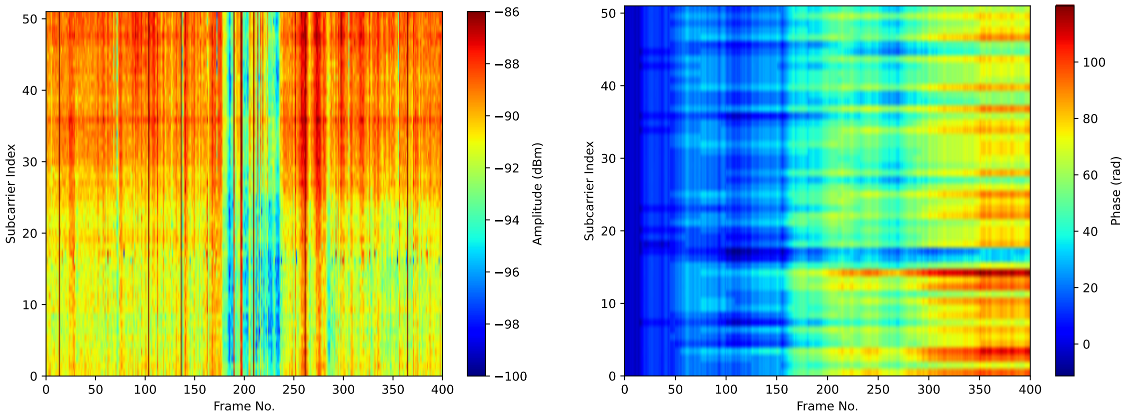

Figure 6.5: Heatmap of (a) absolute amplitude (dBm) and (b) phase (rad).

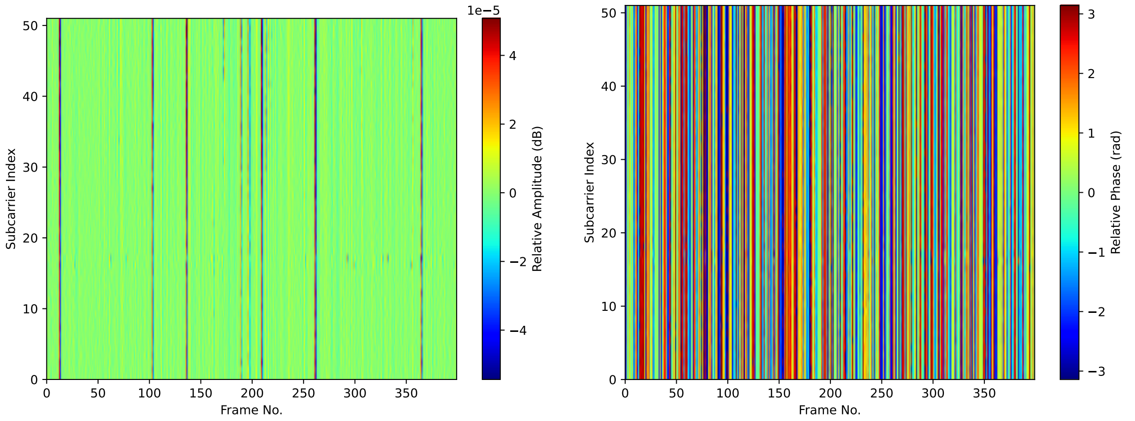

Figure 6.6: Heatmap of relative (a) amplitude and (b) phase (rad).

tive change may signal a blocked LOS. In contrast, phase fluctuates from frame to frame without clearly reflecting the target's effect.

Another approach is to analyze the CSI as a Channel Impulse Response (CIR) in the time domain, revealing the multipath propagation and power distribution. The PDP maps the received signal power over time delays, which helps differentiate propagation paths based on when signals arrive. It is computed as shown in Equation 5 for a channel with N subcarriers.

$$PDP[t] = \left|\mathscr{F}^{-1}\{H[f]\}\right|^2 = \left|\frac{1}{N}\sum_{k=-\frac{N}{2}}^{\frac{N-2}{2}} H_k e^{j2\pi\left(\frac{k}{N}+\frac{1}{2}\right)t}\right|^2, \tag{5}$$

Besides serving as a feature on its own, the PDP aids in denoising the signal by eliminating components beyond a given delay τ and converting the resulting CIR back to the frequency domain. Figure 6.7 shows the PDP for the CSI samples at frames 50 and 200.

Limitations arise in PDP analysis due to its granularity and the number of paths it

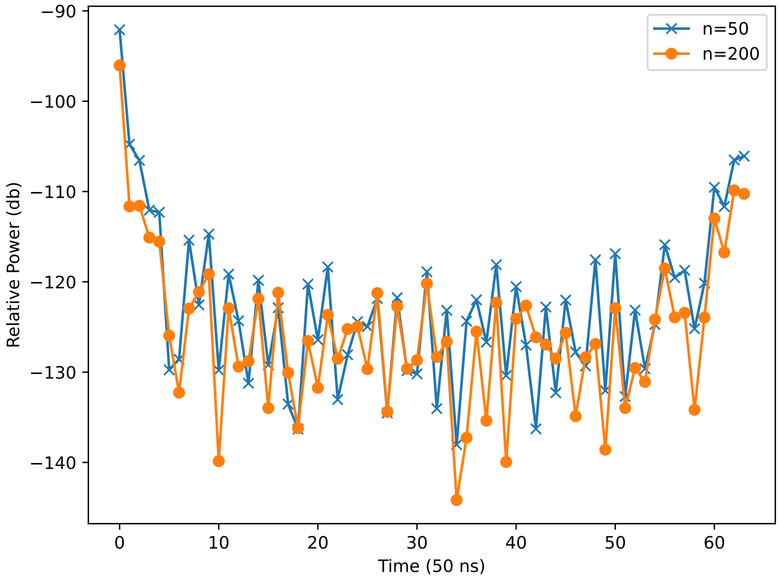

Figure 6.7: PDP feature for frames n=50 and n=200.

resolves. The system bandwidth determines the resolution, and closely spaced multipath components may merge within a single delay interval. This merging reduces clarity and can cause the PDP to underrepresent the actual number of propagation paths. For Wi-Fi at the 2.4 GHz band, only 20 MHz and 40 MHz channels exist, resulting in delay resolutions of 50 ns and 25 ns, respectively. These resolutions correspond to distance resolutions of approximately 15 m and 7.5 m, based on the speed of light. Since the data collection took place outdoors with transceivers positioned 12 m apart, the delayed paths mainly include unwanted components, such as ground reflections and noise.

Tan et al. [34] proposes the use of the 5 GHz band and switching between multiple adjacent channels within coherence time. This approach calls for techniques like the Non-uniform Discrete Fourier Transform (NDFT) instead of a conventional Inverse Discrete Fourier Transform (IDFT), as it requires uniformly spaced channels.

Analyzing CSI as a spectrogram offers a time–frequency representation of the channel dynamics. By applying a Short-Time Fourier Transform (STFT) across the temporal sequence of CSI measurements, it becomes possible to observe how frequency components evolve over time. This view captures transient variations in the channel caused by motion, environmental changes, or other dynamic events [35]. Spectrograms help identify patterns and signatures associated with specific activities or disturbances in the environment, making them particularly useful in applications such as gesture recognition and activity monitoring [35, 36]. The spectrogram is defined in Equation 6, where *m* is the time bin, *k* is the frequency bin, *W* is the window size, *S* is the slide, *x* is the function analyzed (e.g. amplitude, phase), and *w* is the window function used as a filter.

$$S[m,k] = \sum_{n=0}^{W-1} w[n]x[Sm+n]e^{-j\frac{2\pi kn}{W}} \tag{6}$$

Figure 6.8 demonstrates the application of the spectrogram over the amplitude of a single subcarrier (index 13) with $W = 64$, $S = 32$ and a Hamming window. The spectrogram shows that around 2 seconds the amplitude change expressed on Figure 6.5 creates a non-zero frequency component that indicates the movement of the animal.

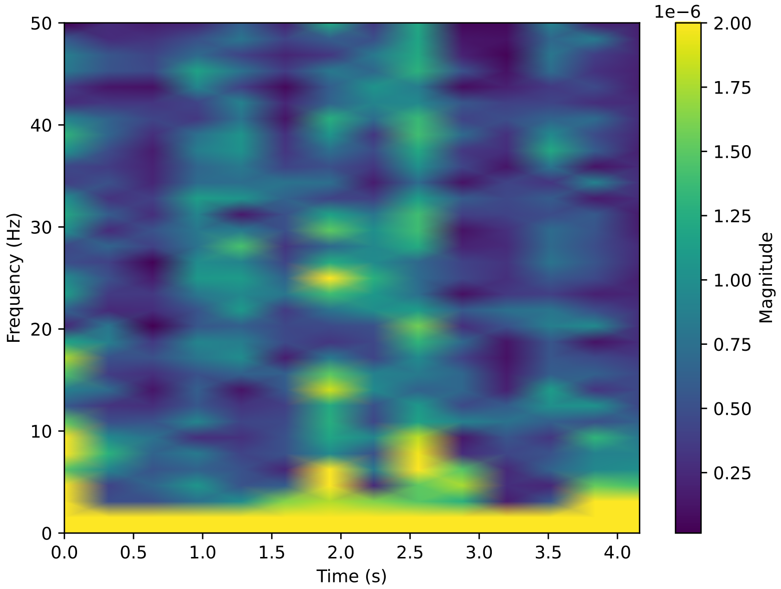

Figure 6.8: Amplitude spectrogram with Hamming window.

Beyond the previously discussed features, CSI supports the extraction of advanced spatial and temporal characteristics such as Time of Flight (ToF), Angle of Arrival (AoA), and Angle of Departure (AoD). These features enhance localization and motion tracking by estimating the propagation delay and direction of signal components. Techniques like the MUSIC algorithm [37, 38] exploit the spatial diversity available in MIMO systems to resolve multiple signal paths with high angular or temporal precision. Extracting these features requires CSI from multiple antennas coupled with careful calibration and synchronization. Thus, their effectiveness depends on access to high-resolution spatial measurements and proper array geometry.

6.3.4. Machine learning algorithms for Wi-Fi sensing

In this section, we introduce and discuss the fundamentals of machine learning and its application in Wi-Fi sensing. Additionally, we highlight emerging trends and open challenges that could shape the next generation of applications in Wi-Fi-related tasks.

The field of machine learning comprehends a wide variety of techniques, including supervised and unsupervised learning, reinforcement learning, and deep learning. Below, we cover popular machine strategies and highlight the most notable advances in their application to Wi-Fi sensing.

According to our literature review, the most popular machine learning technique applied to Wi-Fi sensing is supervised learning. In this category, previous studies apply both parametric and non-parametric models. In the first, the learning process consists of finding a hyperplane from data and their respective labels that separate instances given certain optimization constraints (i.e., minimum error or maximum margin). Successful parametric techniques applied in Wi-Fi sensing include decision trees, random forests, gradient boosting, and Naive Bayes [30]. For example, [39] employ a random forest classifier to estimate sleep duration of college students with per-minute granularity, by analyzing the Wi-Fi packet traffic of multiple user-owned devices throughout the night and applying a moving average to estimate the user's bedtime and wake up times. They achieve performance statistically indistinguishable from a commercially available wearable device for tracking sleep, obtaining an overall error rate between 7 and 30 minutes.

An important characteristic of parametric techniques is that the training data are unnecessary during the inference stage (i.e., the testing). Models belonging to the nonparametric, on the other hand, require all training data (or part of them) in memory. It turns out that these models often rely on distance functions (i.e., euclidean or cosine distance) to classify new instances of data. Overall, this process involves assigning the category of an unseen instance to the closest data points in the training set. Given the typical low-cost purpose of Wi-Fi sensing applications, nonparametric strategies, such as k-Nearest Neighbor, may hinder the applicability of real-time sensing in resource-constrained IoT devices due to the aforementioned memory requirements.

Differently from standard nonparametric models, the Support Vector Machines (SVM) model stores only a small set of examples – the support vectors, the examples closest to the separating plane. Due to its simplicity and positive results, SVM occupies an important place in the history of machine learning and pattern recognition, including Wi-Fi sensing. In the work by [40], for example, the authors employ a multi-class SVM to classify 10 different moisture levels in wheat using Multi-Channel and Multi-Scale Entropy (MCSEn) of Wi-Fi CSI data, achieving 91% to 99% accuracy in both NLOS and LOS scenarios. They also explored three different SVM kernel functions, namely the Gaussian Radial Basis Function (RBF), the Linear Kernel Function and the Polynomial Kernel Function. The Gaussian RBF kernel function provided the best overall performance, maintaining high accuracy in low and high moisture content samples.

Beyond the traditional models mentioned above, neural networks and deep learning represent a growing and promising family of machine-learning techniques for Wi-Fi sensing. These models operate by successively applying linear transformations followed by non-linear activation functions. Formally, each transformation corresponds to a layer consisting of a set of small structures called neurons. Given randomly initialized weights, neurons compute a simple dot-product on input data. The learning in neural networks involves updating the weights towards a minimum of a function often using iterative optimization mechanisms such as the Stochastic Gradient Descent algorithm.

Previous studies confirm that neural networks with a large number of layers may uncover patterns in data more effectively, leading to significant and notable results [41]. Models within this setting correspond to deep models, forming the foundation of deep learning. Neural networks and deep learning have emerged as popular and effective strategies for Wi-Fi sensing, mainly in the form of Convolutional Neural Networks (CNNs). For example, [2] combine both CSI amplitude and phase information from human activity, converting the Wi-Fi sensing task to a image classification task by using a CNN. Their approach is also tailored for transfer learning techniques, which allows using pre-trained models for cross-domain sensing with only a small number of samples for the new target domain. In a single environment, their model achieved over 97% accuracy for human gait identification, whereas the accuracy for new users (new domain) decreased to 77%. Even though this is a promising technique that outperforms other types of models, there is still a considerable loss in accuracy when employing transfer learning to recognize new subjects.

Despite promoting notable advances in Wi-Fi sensing tasks, models based on neural networks are not without drawbacks. The computational demand is among the most notable limitations involving modern and top-performance deep learning models [42, 43]. To deal with this issue, different efforts focus on improving the computational cost of deep models. Among the most promising and aligned with Wi-Fi sensing are knowledge distillation and quantization. As the name suggests, knowledge distillation distills the knowledge of a large, often computationally expensive model into a smaller, more efficient model. In contrast, quantization focuses on reducing the data type precision, e.g., from the typical 32-bit floating points to 8-bit integers. According to modern literature on Wi-Fi sensing, quantization emerges as the most popular technique for deploying deep models on low-resource devices such as the ESP32 [30].

A parallel and growing line of research towards more efficient deep models focuses on compression through pruning [44, 45]. The idea behind pruning is to identify and remove the least important structures (neurons, layers or both) from neural networks while preserving their predictive ability. According to existing studies, state-of-the-art pruning methods cut over 75% of computing with minimal loss in model predictive ability [44]. Therefore, these techniques emerge as promising candidates for enabling deep learning on low-resource devices and Wi-Fi sensing tasks.

Another family of methods for efficient machine learning in Wi-Fi sensing tasks involves dimensionality reduction and feature selection [46, 47]. This category of methods consists of reducing the number of attributes from data before using a model. While these techniques share a similar purpose, their process is different. Specifically, dimensionality reduction projects the original data space (spanned by the attributes) onto a low-dimensional space. In contrast, feature selection permanently eliminates some attributes from data. Due to their effectiveness, previous works on Wi-Fi sensing employ both dimensionality reduction and feature selection [30]. A commonly employed feature selection technique is to select the subcarriers based on statistical properties, such as selecting the highest variance, before continuing with Principal Component Analysis (PCA) to reduce the dimensionality of the CSI data. For instance, [33] employed a PCA explaining 98% of the data variance to reduce CSI features from 26000 to just 35 per sample.

As a final note, while the interest in applying neural networks and deep learning to Wi-Fi sensing applications continues to grow, to the best of our knowledge, no work explores the use of foundation models or general-purpose AI models [42, 43]. It turns out that existing techniques for reducing the computational cost of deep learning still fail to make foundation models efficient enough for direct application on low-cost Wi-Fi sensing hardware. Therefore, we believe that studying the challenges of applying foundation models to Wi-Fi sensing is an encouraging research direction and can pave the way for a new chapter in the field.

6.4. Case study: Low-cost animal and pedestrian crossing detection in rural roads using WiFi sensing and deep learning.

In this section, we present a case study demonstrating the use of Wi-Fi sensing as a cost-effective alternative for detecting and classifying dangerous roadway crossings in rural areas. The case study draws upon the foundational work conducted in the Master's dissertation by [33], and is further supported by two complementary conference publications addressing the system's networking [48] and machine learning components [49].

6.4.1. Context and motivation

Traffic accidents are the leading cause of death for individuals aged 5 to 29, resulting in over 1.35 million fatalities annually [50]. Among these incidents, collisions between vehicles and wildlife represent a significant share, posing serious environmental, economic, and public health concerns across both developed and developing nations. In Brazil alone, tens of millions of vertebrates are killed on roadways each year. This scale of mortality has prompted growing concern among conservation researchers, with studies suggesting that vehicle collisions may pose a greater threat to certain endangered species than even illegal hunting [51]. The situation is similarly concerning in the United States, where official estimates attribute over 26,000 human injuries, 365 million animal deaths, and more than 8 billion dollars in annual damages to wildlife-vehicle collisions [52]. Notably, over 89% of these incidents take place on rural two-lane roads, where limited infrastructure and lower traffic volumes contribute to underinvestment in monitoring and safety solutions.

To mitigate these accidents, animal detection systems are frequently deployed. These systems aim to detect animal presence near roads and activate warning signs to alert oncoming drivers. Technologies such as LiDAR sensors [53] and imaging cameras [54] have been used with some success, but their high costs and limited scalability make them impractical for widespread deployment over long rural roads. In this context, Wi-Fi sensing emerges as a promising alternative. By leveraging the channel state information extracted from the embedded antennas of low-cost IoT devices, a Wi-Fi sensing-based system can detect environmental changes — such as the movement of animals — without relying on expensive hardware or dedicated sensors. However, at the time this research was conducted, most Wi-Fi sensing studies were conducted in controlled environments such as enclosed labs. As such, they did not yet reflect the practical challenges of developing these solutions in complex, real-world environments like rural roads, which were addressed in this research by [33].

6.4.2. Related work: monitoring technologies and challenges

To be effective in real-world conditions, detection systems for roadway safety applications need to meet two key requirements: (i) reliable accuracy in identifying dangerous crossings and (ii) scalability over long distances. In safety-critical scenarios like animal-vehicle collisions, achieving high sensitivity — minimizing false negatives — is especially important. Missed detections could lead to a false sense of security for both drivers and road operators, increasing the risk of accidents. While reducing false positives is also desirable to avoid unnecessary driver alerts, the priority remains ensuring that real threats are consistently detected.

Environmental conditions play a significant role in how well different detection technologies perform. Vision-based systems, such as cameras, can struggle in low-light conditions or during adverse weather events like fog, rain, or snow [55, 53]. Other common sensing technologies, like Doppler radar and infrared sensors, can also be affected by environmental noise, such as moving foliage [56]. These challenges are typical on rural roads, meaning detection systems need to be resilient and adaptable to maintain reliable performance under varying and often harsh outdoor conditions.

Scalability is another critical factor, especially for rural areas where resources are limited. High-cost equipment or infrastructure-heavy solutions are often not feasible across the many kilometers of road where monitoring may be needed. As such, the overall cost per kilometer becomes a key consideration. Remote locations also frequently lack reliable energy and network infrastructure, so detection systems must be designed to operate efficiently in terms of both bandwidth and power. This includes using communication protocols suited to low-connectivity environments and considering whether data is processed at the edge or in the cloud—both of which come with trade-offs when handling input from large numbers of distributed sensors.

A wide range of animal detection systems have been proposed in the literature, many of which demonstrate high levels of accuracy—particularly those based on computer vision and LiDAR. However, these solutions often fall short when it comes to cost-effectiveness and scalability, limiting their practicality for deployment along the extended stretches of rural highways. For example, while LiDAR sensors can achieve detection accuracies exceeding 99% for animals, vehicles, and pedestrians, they offer a relatively short effective range of around 30 meters and can cost as much as US$3,900 per unit [53]. On the other hand, more affordable and scalable options, such as Doppler radar and infrared sensors, generally lack the capability to classify detected objects and do not deliver the level of accuracy needed for dependable real-world monitoring.

Given these limitations, this M.Sc. research [33] aims to address both core challenges: ensuring accurate detection and classification while maintaining scalability and cost-efficiency. The proposed system is designed to balance these three critical factors, making it suitable for large-scale deployment in rural road environments.

Table 6.1 provides an overview of current state-of-the-art solutions for traffic, animal, and pedestrian detection, highlighting their performance characteristics and adaptability to low-light conditions. To the best of our knowledge, no existing approach in the literature offers a combination of high-accuracy detection and classification at a low cost, particularly in settings with limited infrastructure.

6.4.3. Method

To develop an affordable IoT-based system for monitoring rural roads, our approach addresses three key aspects: (i) the wireless sensor network (WSN) architecture, (ii) data acquisition and machine learning-based event detection, and (iii) model optimization for deployment on resource-constrained devices.

The first aspect focuses on ensuring reliable and timely data transmission across extensive road networks equipped with hundreds of sensor nodes operating under limited

Table 6.1: Summary of existing sensing methods for animal, vehicle and pedestrian detection on roads. Adapted from [33].

Work	**Method**	**Subjects**	**Detection Accuracy**	**Classification Accuracy**	**Low Cost**	**Operates in Low Light**
[54]	Camera	Vehicles, animals, people	98%	97 - 99%	-	-
[57]	Camera	Animals	98%	-	-	-
[53]	LiDAR	Vehicles, animals, people	> 99%	> 99%	-	✓
[58]	PIR	Vehicles, people	76 - 94%	-	✓	✓
[56]	Doppler	Animals	80%	-	✓	✓
[16]	Bluetooth	Vehicles	98%	-	✓	✓
[59]	WiFi	Vehicles	> 99%	91%	-	✓
Ours [33]	**WiFi**	**Vehicles, animals, people**	**> 95%**	**> 95%**	✓	✓

energy and connectivity conditions. We evaluate both cloud-centric and edge-computing architectures through simulations in the COOJA simulator [60], taking into account traffic load, sensor density, road length, and processing latency associated with low-cost hardware [48]. Performance is assessed using metrics such as Packet Delivery Rate (PDR) and Round Trip Time (RTT), enabling a comparative analysis of architectural efficiency in rural scenarios. Additionally, we utilize OMNeT++ [61] to examine the coexistence of Wi-Fi sensing under the IEEE 802.15.4 communication standard, ensuring reliable transmission of both sensing and communication packets. The insights from these simulations were used as basis for the architecture detailed in Section 6.4.5.

The second aspect leverages Wi-Fi sensing to detect hazardous road-crossing events using the built-in antennas of low-cost IoT devices. Adapting the methodology presented in [59], we collect CSI data from ESP32 boards for various entities — such as large and small animals, pedestrians, vehicles, and ambient noise — across diverse environments. CSI amplitude features are used to train a Transformer-based deep learning model, which is then evaluated using standard performance metrics to assess its effectiveness in identifying and distinguishing potentially dangerous scenarios.

The third aspect focuses on optimizing the trained models for deployment on low-power hardware. Due to hardware constraints and software compatibility limitations, we design a lightweight Multi-Layer Perceptron (MLP) model to perform initial detection directly on the ESP32 devices, while offloading more computationally intensive classification tasks to a Transformer model running on a Raspberry Pi-based edge node. Through targeted feature engineering and dimensionality reduction techniques, we significantly reduce the computational and memory requirements of both models, enabling real-time inference within the operational limits of the target hardware.

In the following sections, we discuss each of these three aspects in detail.

6.4.4. Networking

Deploying large-scale IoT-based roadway monitoring systems introduces significant challenges due to the limited computational capabilities of low-cost sensing hardware, which may be insufficient for real-time data processing and classification. To address this, we ([48]) explored alternative computing architectures, specifically cloud and edge processing paradigms. Using the COOJA simulator, we modeled an IoT network spanning up to 10 kilometers of roadway across three distinct topologies (as illustrated in Figure 6.9): (i) *cloud processing*, where data is relayed to an internet-connected sink node; (ii) *edge processing*, in which data is transmitted to nearby local processing units; and (iii) *cloud processing with retransmission*, which incorporates intermediary retransmitter nodes equipped with more powerful antennas to aggregate and forward sensor data over fewer hops. These topologies were then evaluated to determine their suitability for scalable and time-sensitive road monitoring applications.

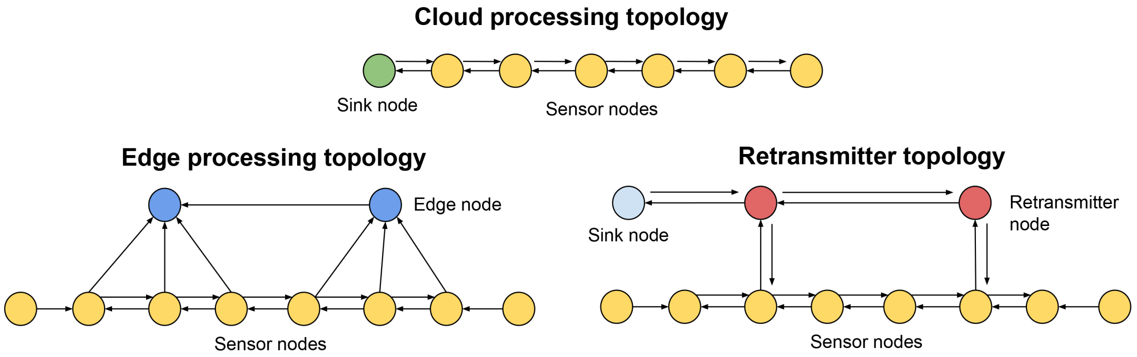

Figure 6.9: Diagram illustrating the cloud, retransmitter and edge processing network topologies. Adapted from [48] © 2022 IEEE.

Among all of the simulated scenarios, we focus our evaluation on the most demanding, analyzing the packet delivery rate on a network monitoring a distance of 10 km. Figure 6.10 presents a summary of the results. While the retransmitter-based topology significantly improves upstream data transmission compared to the cloud-based approach —achieving a PDR exceeding 77% — it offers minimal benefit for downstream communication, which remains below 12%. In contrast, the edge processing topology demonstrates superior scalability and reliability, with bidirectional PDR surpassing 99%. It is important to note, however, that these simulations did not account for potential interference at the physical layer between WiFi sensing and IEEE 802.15.4 communication.

Given that IEEE 802.15.4 and IEEE 802.11 (WiFi) both operate in the 2.4 GHz frequency band, simultaneous use could result in cross-technology interference, potentially degrading network performance. To assess the feasibility of protocol coexistence within the proposed system architecture (Section 6.4.5), we conducted a series of simulations using OMNeT++[1].

Initial results revealed that IEEE 802.15.4 performance was significantly impaired when operating on overlapping frequencies with WiFi, with PDR dropping below 40%.

[1] Simulation parameters at: `https://github.com/SamuelDucca/csi-animal-crossing`

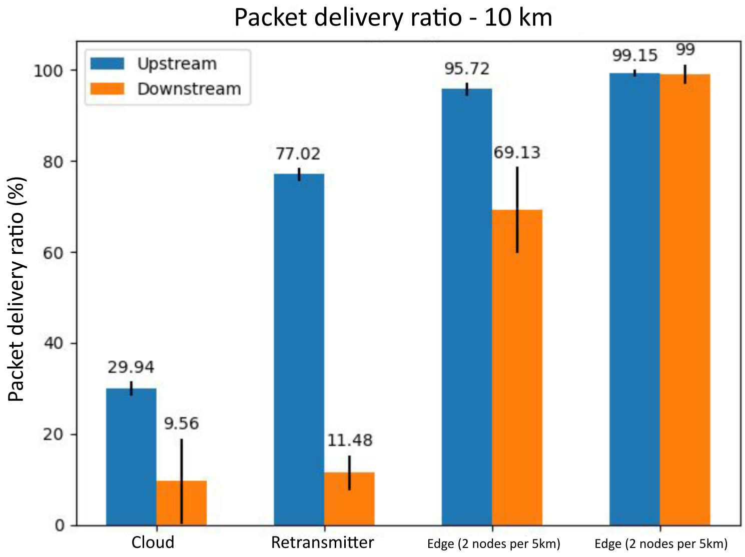

Figure 6.10: Packet delivery ratio for cloud, retransmitter and edge topologies. [33]

In contrast, WiFi communication remained largely unaffected by IEEE 802.15.4, maintaining a PDR near 49% across all tested channels. By isolating the WiFi sensing on a non-overlapping frequency, IEEE 802.15.4 PDR improved to over 99%, while WiFi PDR remained unchanged. The relatively low WiFi PDR happened due to network saturation caused by all sensors transmitting on the same channel.

To address this, we distributed Wi-Fi sensing across channels 1 to 11 to mitigate congestion and reassigned IEEE 802.15.4 to channel 26 (2480 MHz), which does not overlap with WiFi. As shown in Figure 6.11, this approach led to a significant improvement in performance: WiFi PDR exceeded 94%, while IEEE 802.15.4 maintained close to 99%. Additionally, a reduction in the standard deviation of WiFi PDR values indicates more uniform performance across the network, underscoring the effectiveness of frequency planning in enhancing coexistence.

After conducting the network simulations to identify the most suitable topology and validate the coexistence of the sensing and communication protocols, we finalized the system architecture detailed in the following section.

6.4.5. Architecture

Our proposed architecture integrates cost-effective ESP32 microcontrollers for roadside detection and communication of crossing events, supported by Raspberry Pi-based edge computing nodes for classifying the detected events. Specifically, the ESP32-S3 is employed for Wi-Fi sensing, while the ESP32-H2 manages IEEE 802.15.4 communications. Sensing nodes are deployed in pairs on both sides of the roadway at 50-meter intervals, ensuring coverage within WiFi and 802.15.4 radio ranges. Edge processing units — based on the Raspberry Pi 4 Model B — are distributed approximately every kilometer to minimize communication latency and handle local inference tasks. The use of IEEE 802.15.4 also supports resilient mesh networking, allowing alternate routing paths in case of in-

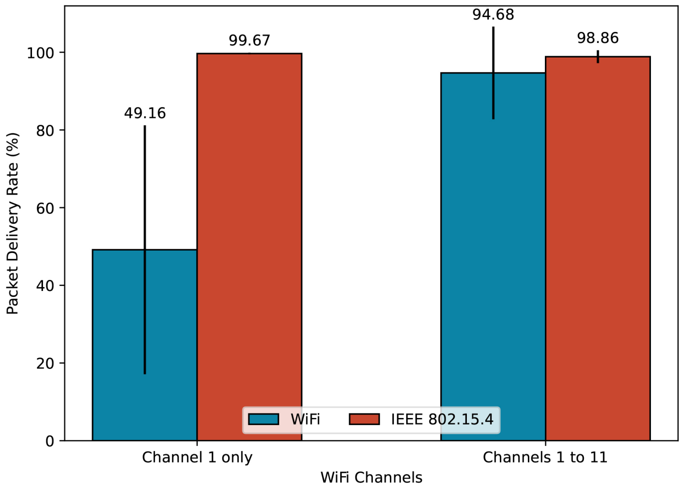

Figure 6.11: WiFi and IEEE 802.15.4 packet delivery rate with and without distribution of WiFi sensing over multiple channels. [33]

dividual node failures. In regions with complex terrain or reduced line-of-sight due to curves, the sensing node density may need to be increased to maintain effective coverage.

As shown in Figure 6.12, the system operates through a three-stage process:

1. **Local Event Detection:** Wi-Fi CSI data is processed in real time on the ESP32-S3 using a lightweight MLP model optimized to minimize false negatives. When a crossing is detected, visual alerts (e.g., LED signals) can be triggered immediately to warn oncoming vehicles.

2. **Data Transmission:** Upon detection, the ESP32-H2 module forwards event data through an ad-hoc IEEE 802.15.4 network to the nearest sink node. Its integration as a radio co-processor [62] ensures communication tasks do not interfere with sensing operations.

3. **Edge Classification:** The sink node, powered by a Raspberry Pi 4, executes a more complex Transformer model to classify the crossing event (e.g., pedestrian, animal, or vehicle) detected by the sensor nodes. If deemed a false positive, the edge node can instruct the sensor to deactivate the warning system. Conversely, validated events may be transmitted via satellite or cellular networks to notify the road operator. This enables informed decision-making and timely response planning, such as dispatching a tow truck or deploying an animal recovery team, depending on the nature of the detected event.

6.4.6. Cost Analysis

This section presents a comparative evaluation of our WiFi sensing-based solution for detecting dangerous roadway crossings against conventional camera-based monitoring systems. While total deployment costs typically encompass equipment, infrastructure, installation, and maintenance — which often vary in different scenarios — we focus our

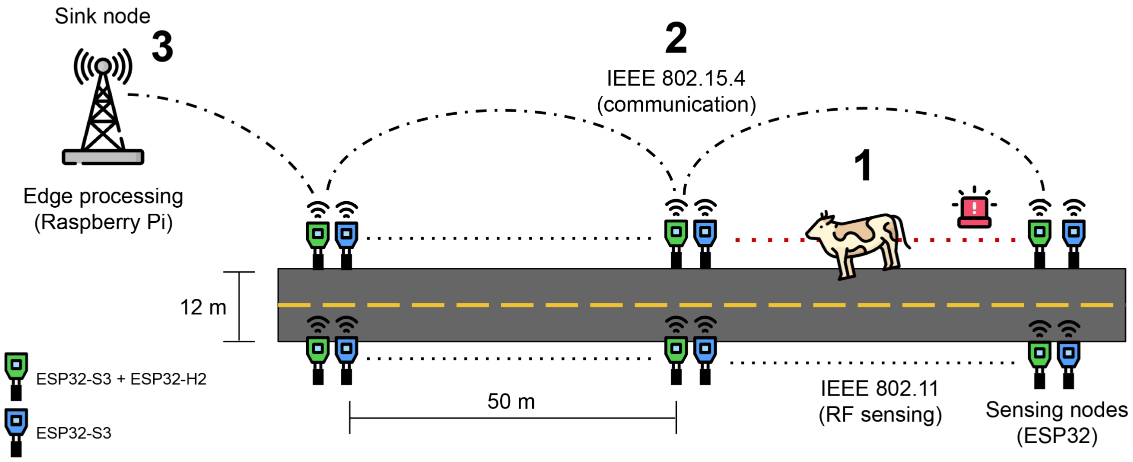

Figure 6.12: Diagram illustrating the system architecture for the (1) detection, (2) data transmission and (3) classification of roadway crossings using WiFi sensing [33].

analysis on three primary factors: (i) hardware cost, (ii) power consumption, and (iii) bandwidth usage, as summarized in Table6.2.

Table 6.2: Total power consumption, internet bandwidth usage and cost per kilometer for each sensing method [33]

Method	Power Consumption (W/km)	Internet Bandwidth (kbps/km)	Cost (USD/km)
Video Monitoring	64	8650	3000
WiFi Sensing (ours)	68.28	<1	407.4

The proposed WiFi sensing system demonstrates a significant cost advantage, with an estimated deployment cost of USD 407 per kilometer — approximately 7.3 times lower than camera-based systems, which average at USD 3000 per kilometer. Despite the reduced cost, power consumption remains comparable, but the WiFi system has the advantage of operating independently of internet connectivity. Furthermore, additional infrastructure commonly required for video surveillance — such as tall mounting poles and networking hardware — is unnecessary for our solution, further reinforcing its economic efficiency.

6.4.7. Data collection and processing

Before training the models, we construct a dataset through multiple field experiments conducted across diverse environments, including gravel, dirt, and paved roads, as well as pasture areas. WiFi Channel State Information (CSI) is collected from cows, dogs, pedestrians, and vehicles using two ESP32 boards positioned 12 meters apart, as shown in Figure 6.13.

We then extract the amplitude from the complex CSI matrix, capturing the signal strength of each subcarrier in every recorded frame. To mitigate distortions introduced by

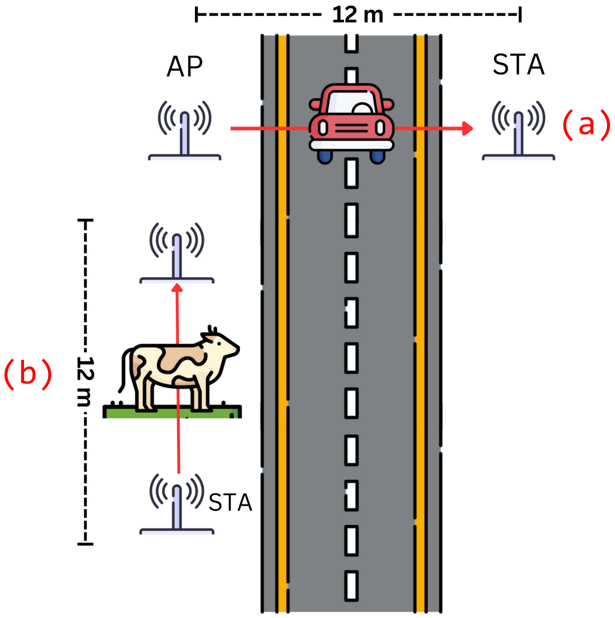

Figure 6.13: Data collection layout for vehicles (a), animals and pedestrians (b).

AGC, we apply an RSSI-based normalization technique, as described in [63]. Subcarriers that do not carry meaningful information for CSI-based sensing are excluded, resulting in a refined set of 52 active subcarriers.

Next, the non-zero amplitude values are converted to decibels, while null entries are assigned a value of zero post-conversion to avoid computational anomalies such as negative infinity. A running mean filter is then applied across each frame to suppress noise and reduce the influence of outliers, enhancing signal consistency without distorting meaningful variations. Additionally, zero-amplitude values — typically artifacts of signal acquisition errors — are excluded from the running mean calculation to preserve data integrity.

Each sample in the assembled dataset consists of 52 subcarriers over 500 consecutive frames (equivalent to a 5-second interval), yielding a feature vector of 26,000 elements per sample. The resulting datasets, focused on animal crossing detection, have been made publicly available via Zenodo [64] (open access) and IEEE DataPort [65].

6.4.8. Detection and classification of events

After assembling the dataset, we employ feature engineering techniques to improve model performance and efficiency. First, we add 104 statistical features to each sample, consisting on the mean and standard deviation of each of the 52 Wi-Fi subcarriers in that sample. Then, we use a Principal Component Analysis that explains from 98% of the data variance to reduce data dimensionality, obtaining a final set of 35 features per sample.

As outlined in Section6.4.5, our proposed system integrates two distinct machine learning models for roadway crossing detection and classification. The first is a lightweight Multi-Layer Perceptron (MLP), deployed on ESP32 devices, which performs binary detection of crossing events. This model comprises five fully connected layers with 35, 16, 8, 4, and 1 neurons, respectively, interleaved with 20% dropout layers to improve generalization. ReLU activation is used in the hidden layers, while the output layer employs a sigmoid function to support binary classification.

The second model is a Transformer-based neural network responsible for identifying the type of object involved in each detected crossing — such as a person, vehicle, or animal [49]. This model incorporates four Transformer layers with decreasing self-attention module sizes of 64, 32, 16, and 8, respectively.

Both models are trained using the TensorFlow framework with an 80/20 train-test split on the same dataset. To ensure compatibility with low-cost IoT hardware, we optimize the models using the LiteRT framework (formerly TensorFlow Lite Micro), applying post-training quantization and feature engineering techniques to reduce input dimensionality and improve inference performance. During this process, we convert the 32-bit floating-point model weights to 8-bit signed integers, substantially reducing the model size, before converting it to a C data array format that can be executed by the ESP32.

Figure 6.14 presents the confusion matrices for the Transformer-based classifier (a) and the MLP detection model (b). The MLP model achieves a high detection rate, with a false negative rate as low as 0.7% and an overall accuracy exceeding 95%, although it incurs a moderate false positive rate of 9.3%. The Transformer model, tasked with event classification, demonstrates strong performance across all classes. Even for the most challenging category (dog) accuracy remains above 90%. Misclassifications are primarily observed between dogs and persons, likely due to similarities in the WiFi CSI signatures produced by their motion. Nevertheless, the Transformer maintains a low false positive rate of 1.3% and a false negative rate of 1.0%, resulting in an overall classification accuracy of 95.8%. These results underscore its effectiveness in refining detections and mitigating false positives from the MLP model.

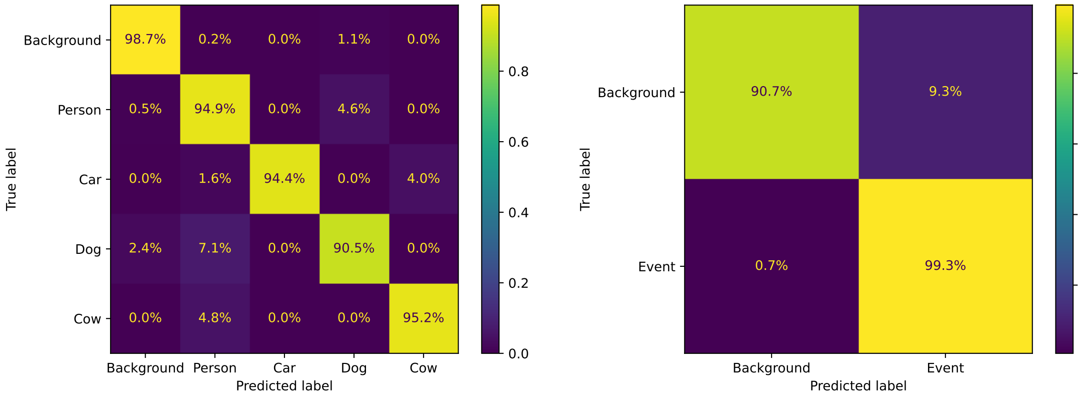

Figure 6.14: Confusion matrix for the Transformer classification model (a) and the MLP detection model (b), normalized by true labels. [33]

By applying our optimization techniques described in the beginning of this section, we enable both models to be executed in real time on their respective target hardware while preserving high accuracy levels above 95%. The MLP model achieves an average inference time of 0.17 ms on the ESP32, with a minimal memory footprint of just 5.6 KB.

In parallel, the Transformer model runs on the Raspberry Pi with an average inference time of 2.19 ms and occupies approximately 2.2 MB of memory.

6.4.9. Considerations

In summary, this case study demonstrates that Wi-Fi sensing, combined with optimized machine learning models and a robust wireless sensor network architecture, can serve as an effective and low-cost solution for monitoring hazardous roadway crossings in rural settings. Through simulations in both COOJA and OMNeT++, we show that an edge computing topology significantly enhances communication reliability, while careful frequency allocation ensures coexistence between IEEE 802.11 and IEEE 802.15.4 protocols. Furthermore, the lightweight MLP and Transformer models achieve high classification accuracy with minimal resource consumption, enabling real-time operation on commodity IoT hardware. Cost analysis reinforces the practicality of the approach, highlighting substantial savings compared to traditional camera-based systems.

While these results show promise for the feasibility and scalability of Wi-Fi sensing in remote and infrastructure-limited environments, the proposed system presents some limitations that warrant further investigation. First, the Wi-Fi sensing approach relies on consistent signal propagation, which may be affected by extreme weather conditions, dense vegetation, or varying terrain profiles. Additionally, the current dataset, while diverse, may not capture the full range of real-world scenarios, such as high-speed crossings or simultaneous events involving multiple entities. These limitations may be addressed by further data collection experiments, increasing the types of weather conditions and crossing events represented.

6.5. Hands-on experience: using Wi-Fi CSI data for person detection with Wisensing-ESP32

In this section, we utilize the Wisensing-ESP32 framework to demonstrate the complete development pipeline of a Wi-Fi sensing system for real-time person detection using low-cost IoT devices. This includes the processing of Channel State Information (CSI), construction of a dedicated dataset, and the subsequent training and optimization of machine learning models tailored for execution on resource-constrained hardware.

6.5.1. Motivation and overview

The inherently complex nature of Wi-Fi signals makes machine learning a common approach in Wi-Fi sensing applications, as these algorithms can learn to recognize subtle patterns in the signal. However, effective model training typically requires a substantial amount of annotated data. While publicly available datasets exist for well-established tasks such as human activity recognition, Wi-Fi CSI datasets are still scarce, and those available may not align with the requirements of specific applications. Consequently, it is often necessary to conduct custom data collection experiments and manually annotate the resulting data — a labor-intensive process further hindered by the lack of available tools for CSI-based dataset building.

As the field of Wi-Fi sensing progresses, several tools have emerged to facilitate CSI collection using ESP32 boards, which are widely used for low-cost sensing applications [30]. For instance, the ESP32-CSI-TOOLKIT [66] enables raw CSI data collection using ESP32 boards configured as access points or client stations, but does not offer support for data preprocessing or analysis. Similarly, the CSIKit library [67] provides basic amplitude extraction and limited filtering capabilities, but lacks comprehensive support for dataset preparation and model deployment.

To address these limitations, we introduce Wisensing-ESP32, an all-in-one solution for developing amplitude-based Wi-Fi sensing systems using ESP32 devices. The toolkit includes scripts for CSI amplitude extraction and preprocessing, as well as an interface for easy data annotation, dataset creation, and a Jupyter environment for training, evaluating and quantizing machine learning models optimized for ESP32 use.

Wisensing-ESP32 consists of three main modules: (i) data processing, (ii) model training, and (iii) onboard inference on the ESP32 microcontroller, as shown in Figure 6.15. The process begins with the user supplying raw Wi-Fi CSI in one or more .csv files. These files are handled by the Data Processing module, which extracts the CSI amplitude, applies filtering and normalization, and allows the user to annotate the data to create a labeled dataset. This dataset is then used in the Model Training module to develop a machine learning model for the intended application. Once trained, the model is optimized and converted into a format suitable for deployment on the ESP32. Finally, the Onboard Inference module loads the model onto the ESP32, enabling it to perform real-time CSI data collection, processing, and inference for detecting or classifying events using Wi-Fi signals.

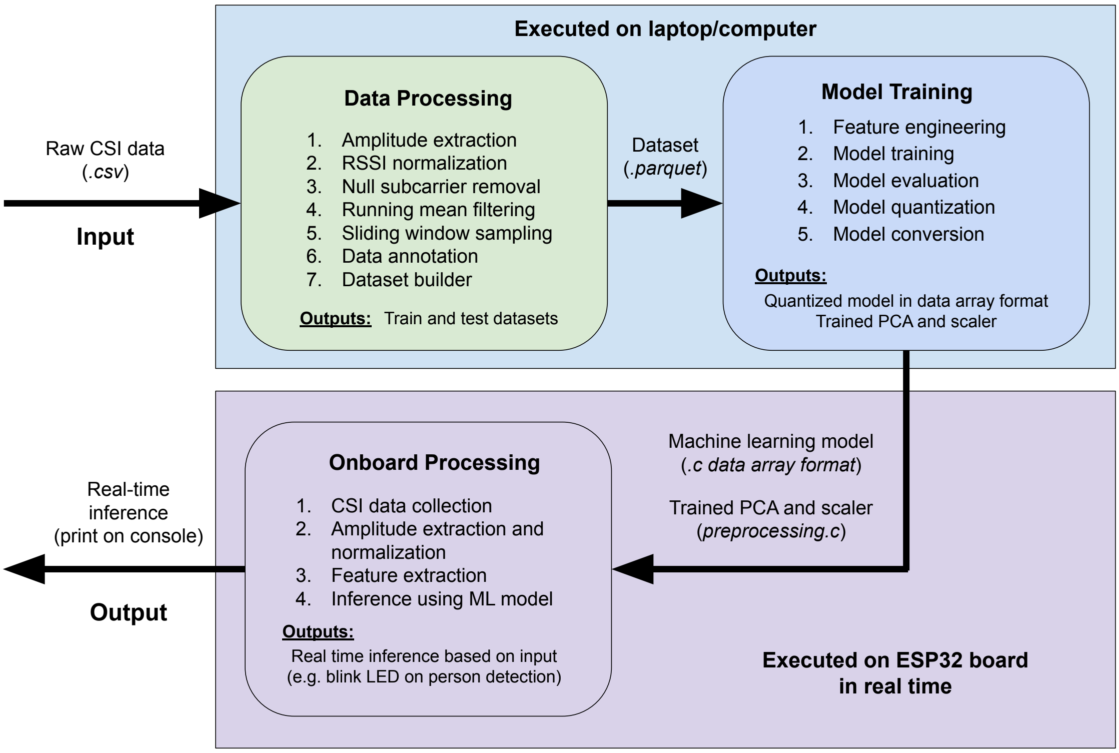

Figure 6.15: Diagram illustrating the three modules of Wisensing-ESP32: (i) data processing, (ii) model training and (iii) onboard processing.

6.5.2. Required equipment

For this hands-on experience, the following equipment is required:

- Two ESP32 boards. They are only required for the last stage of the demonstration, where we detect a person walking through a room in real-time;
- A laptop computer with at least 25 GB of available disk space. We also recommend at least a quad-core CPU with support for AVX instructions and 8 GB of RAM;
- A Windows (7 or higher) or Linux (Ubuntu 16.04 or higher) operating system;
- One power source (such as a powerbank) for the ESP32 that is not connected to the computer;
- Two USB cables compatible with the ESP32 of your choice (there are both type-C and micro-USB models available). They will be used for programming and providing power for the boards during the experiments;
- A way of supporting the boards at least 70 cm above the ground during the experiments, such as a sufficiently high chair or table.

6.5.3. Setting up the tool

The Wisensing-ESP32 Github repository[2] contains all of the code required for this demonstration, as well as detailed instructions on how to install the dependencies and set up the tool. Alternatively, we also provide a virtual machine[3] containing the tool and its dependencies already installed. To use it, download and import the `wisensing-esp32.ova` file using Oracle VirtualBox 7.0[4] with Guest Additions. Once imported, before powering on the machine, modify the USB settings to allow access to the ESP32 board by right-clicking the virtual machine in the VirtualBox interface and navigating to *Settings >USB*. Select the option *Enable USB Controller* according to your USB port version (likely USB 3.0).

After powering on the virtual machine, use the `sbrc` user and the `sbrc` password to login. You can perform experiments with the Data Processing and Model Training modules without the ESP32 hardware, but the Onboard Processing module requires access to the board. Before running Onboard Processing, with the ESP32 board already connected to the computer, use the *Devices >USB* menu at the top of the window and select the device corresponding to your ESP32 board to be able to access the board through the virtual machine.

[2] Available at: https:// github.com/SamuelDucca/Wisensing-esp32

[3] Available at: https://drive.google.com/file/d/1ZLx-myHmZwEWzmHinmBKqMmYcd-gT0Ap/view

[4] Available at: https://www.oracle.com/virtualization/technologies/vm/downloads/virtualbox-downloads.html

6.5.4. Importing and processing data

After setting up the tool, navigate to the `DataProcessing` folder, where the data processing scripts are located. All data manipulation is done in a non-destructive way, so new files will be created after every processing step.

First, raw CSI files (in `.csv` format) must be placed in the `1_raw_data` folder. We provide three CSI files representing a person walking between two ESP32 boards placed at distances of 140, 150 and 200 cm. This data is sufficient for developing our person detection Wi-Fi sensing application, but you may also add data from your own experiments. We recommend using the ESP32-CSI-TOOL [66] to collect raw CSI data in a format that is compatible with Wisensing-ESP32.

To begin processing the raw data, navigate to the `DataProcessing` folder and run the following on the terminal:

```
python 1_format_and_preprocess.py
```

This script will format the raw CSI data, calculate its amplitude normalized by the RSSI and apply a running mean filter to reduce noise, as described in Section 6.4.7. The running mean window size can be configured in the `config.py` file, but we recommend using the default values in the course of this demonstration.

At this point, two new folders are created, containing the transformed data: `2_formatted_data` (raw formatted data) and `3_preprocessed_data` (extracted and filtered amplitudes). From this point onward, we use the `.parquet` format to save the files, reducing memory usage and improving read and write speed.

You can now visualize the processed CSI data (located in the `3_preprocessed_data` folder) as a heatmap plot, using the `single_plot.py` script. For instance, we can analyze the data from our 150 cm distance experiment by running:

```
python single_plot.py preprocessed-person_150cm_1.parquet
```

A matplotlib window will appear, showing the amplitude heatmap plot for this experiment, as illustrated in Figure 6.16. The narrow vertical blue/green strips indicate a drop in amplitude caused by a person walking between the ESP32 boards. You can zoom in the plot using the loupe in the bottom left corner to better visualize the amplitude variations. Close the window when you have finished analyzing the data.

6.5.5. Data annotation

In order to use the processed data for training a machine learning model, it is necessary to annotate it by assigning class labels to specific segments — i.e., windows of frames — corresponding to observed events. This annotation process is carried out using the `data_annotation.py` script, which uses the source file name and the desired class to annotate as arguments. For instance, to annotate the "person" class in the experiment analyzed previously, run:

```
python data_annotation.py preprocessed-person_150cm_1.parquet
person
```

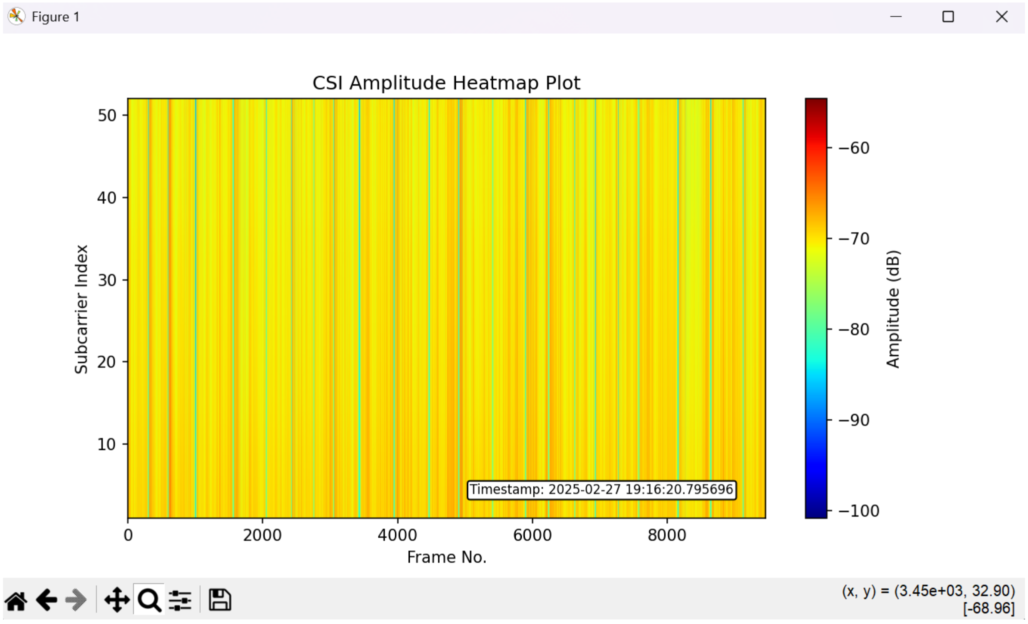

Figure 6.16: Matplotlib window showing the CSI amplitude heatmap plot of a person walking between two ESP32 boards placed 150 cm apart.

When the new matplotlib window opens, zoom in between frames 0 and 2000 using the loupe to better visualize the data. Then, annotate the data by *right clicking* the narrow amplitude drops caused by a person interfering with the signal, as shown in Figure 6.17. A blue dot with a horizontal bar will appear, showing the size of the window delimiting that sample. Avoid any juxtaposition between samples, as that could lead to data contamination and reduced performance due to duplicated samples in the dataset.

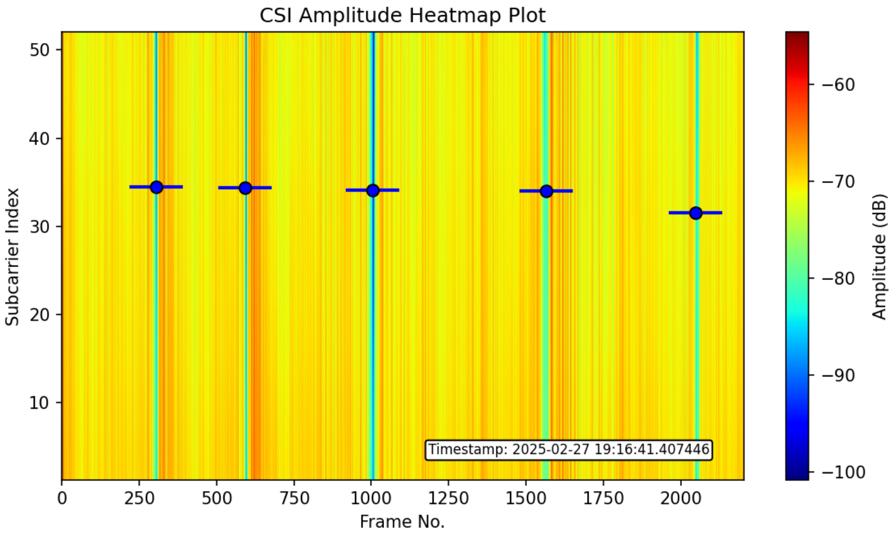

Figure 6.17: Matplotlib window showing the data annotation interface. The blue dot with a horizontal bar represents an annotated window.

In case you make a mistake, use CTRL+Z to undo the last annotation.

When you are done, simply close the window. The annotated frames will be automatically saved in the `slicing_source.txt` file, in a format suitable for input in the next scripts. We provide a `slicing_source_demo.txt` file with all samples from the three experiments already annotated, so manual annotation is not required to proceed.

6.5.6. Data slicing and dataset assembly

After annotating the data, we proceed by executing a script that segments the annotated windows into individual samples and applies a sliding window procedure, generating multiple samples per annotated event. The window size and stride can be adjusted in the `config.py` file; however, for the purposes of this demonstration, we recommend using the default settings.

Use the provided `slicing_source_demo.txt` file for slicing the data using the `2_slicing.py` script:

```
python 2_slicing.py slicing_source_demo.txt
```

This script will create a new `4_sliced_data` folder containing the separated train and test samples (80/20 split). Now, run the final `3_dataset_builder.py` script to assemble a dataset, using the dataset source file and the dataset name as arguments. The `dataset_source.txt` file is provided as an example. You can edit it to remove or add data from any experiment or class. Run:

```
python 3_dataset_builder.py dataset_source.txt my_dataset
```

A new 5_datasets folder will be created, containing both train and test datasets in `.parquet` format. In the following section, we will use these datasets to train the machine learning model.

6.5.7. Model training and quantization

With the dataset prepared, the Model Training module will be used to develop a machine learning model for detecting human presence based on Wi-Fi CSI data. This module is provided as a Jupyter Notebook that includes all the essential code and guidance for importing data, performing feature engineering, training the model with TensorFlow, applying quantization using LiteRT, and converting the final model into a C data array compatible with the ESP32.

The module pipeline begins with the application of feature engineering techniques to reduce data dimensionality and enhance model performance. Specifically, Principal Component Analysis (PCA) is used to project the original high-dimensional data into a lower-dimensional subspace, followed by feature normalization using a standard scaler. This preprocessed data is then used to train a Multilayer Perceptron (MLP) model. The choice of an MLP architecture ensures compatibility with the ESP32 platform, as not all TensorFlow operators are supported on this resource-constrained hardware. To assist the user in evaluating the model, the training script provides visualizations of accuracy metrics across training epochs, along with a confusion matrix summarizing model performance on the test dataset.

To open the Jupyter Notebook, navigate to the `ModelTraining` folder using the terminal and run the following:

```
jupyter notebook
```

This action will launch the Jupyter Notebook interface in your default web browser. To begin, open the `model_training_and_quantization.ipynb` file and run the provided code. If you're new to Jupyter Notebooks, it's advisable to review the official documentation[5] before moving forward.

Running this module will produce four output files: `preprocessing.c`, `preprocessing.h`, `model.c`, and `model.h`. To integrate your trained model and feature extraction pipeline into the next module on the ESP32, copy these files into the `OnboardProcessing/esp-wisense/examples/person-detection/main` directory.

6.5.8. Onboard processing using the ESP32

In this section, we will use the Onboard Processing module to program two ESP32 boards: one responsible for creating a Wi-Fi Access Point (`SoftAP`) and the other responsible for collecting the Wi-Fi data and conducting real-time person detection (`PersonDetection`).

Overview

In order to properly function, the client station (`PersonDetection`) requires a Wi-Fi connection to a suitable AP (`SoftAP`) to request for frame transmissions. This is necessary to estimate CSI at a well-defined rate, as channel evaluation depends on the regular reception of frames at the processing node, and establish a spatial reference for the sensed signals. We provide an ESP32 AP implementation compatible with this requirement, but regular Wi-Fi Access Points (e.g. commercial wireless routers) can be used to this end if they are not configured to rule out ICMP Echo Requests and if they adhere to IEEE 802.11g or later standards.

At a high level, the module performs four primary functions, highlighted in Figure 6.15: CSI collection, amplitude extraction and normalization, feature extraction, and model inference. To ensure scalability and time efficiency, CSI processing is performed on a per-frame basis up to the inference, distributing computational tasks over time in small increments. This approach gradually constructs the feature set for the model, reducing peak processing loads and optimizing resource utilization.

Firstly, whenever the client station receives a frame from the associated AP, the null and Direct Current (DC) subcarriers are discarded, reordered in ascending (frequency) order and sent, along with the RSSI of the frame, to another task on the system to handle the first steps of data processing. We then employ the same amplitude extraction and normalization techniques described in section 6.4.7, preparing base features for further refinement.

Following, the amplitude values of the given frame are normalized to zero mean and unit variance and then projected into the principal component space. These partial

[5] Available at: https://jupyter-notebook.readthedocs.io/en/latest/

components are aggregated across frame iterations to form a comprehensive feature set for each analysis window. Finally, the reduced features are quantized and fed into the model for inference. The output is then de-quantized and evaluated against a predefined threshold to determine the occurrence of an event.

Setting up the module

If you are not using the virtual machine, first setup the ESP-IDF framework (v5.4 stable) for proper use of this module. Follow the Espressif Get Started guide[6] to install the essential software. In case of Linux or macOS setup, make sure to install tools for your desired chip targets when running the install script.

Building the project

Navigate to the `OnboardProcessing/esp-wisense/examples` folder. There are two available example projects: Person Detection and SoftAP. Each will be programmed into a different ESP32 board.

Open the ESP-IDF terminal, or, if using the virtual machine, run:

```
get_idf
```

Navigate to the `person-detection` folder. Before compiling the project, define the compile target:

```
idf.py set-target <chip_name>
```

Where `<chip_name>` indicates the ESP32 board SoC (e.g. esp32, esp32s3, etc). Additionally, in order to establish Wi-Fi connections, the SSID and password must be provided. Open the configuration menu (`idf.py menuconfig`) and navigate to `Component config > WiSense` to view the available options. To build the project, run:

```
idf.py build
```

The initial build may take several minutes depending on the configuration of the host machine.

Flashing and monitoring

After building the project, connect the ESP32 to the computer and flash the built image into the board to monitor its serial output:

```
idf.py flash monitor
```

This should automatically test all available ports in the machine to find the ESP32 board. In case it fails, use the optional argument `-p <PORT>`, where `<PORT>` denotes the serial port where the ESP32 is connected.

After flashing the board, to exit the serial monitor, type `CTRL+]`.

[6] Available at: https://docs.espressif.com/projects/esp-idf/en/v5.4/esp32/get-started/index.html

Now, the ESP32 board responsible for person detection is successfully programmed. Disconnect the board from the computer, navigate to the `softap` folder and repeat the building and flashing steps for the SoftAP using the second ESP32 board. Remember to configure the same SSID and password in both boards.

6.5.9. Experiment: Testing the person detection model

Once the boards are programmed, you can now proceed to the experiment. Place each ESP32 approximately 150 cm apart and at least 70 cm above the ground, as illustrated in Figure 6.18. The `PersonDetection` board should be connected to a laptop computer, while the `SoftAP` board may be connected to any available power source, such as a powerbank.

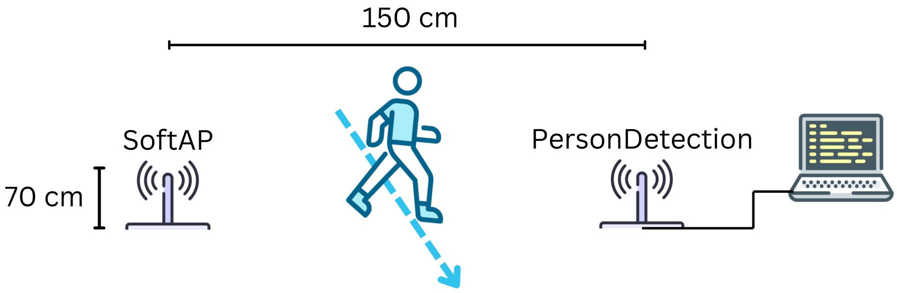

Figure 6.18: Diagram illustrating the person detection experiment with two ESP32 boards: SoftAP and PersonDetection.

Using the laptop, navigate to the `PersonDetection` project folder (`Wisensing-esp32/OnboardProcessing/esp-wisense/examples/person-detection`) and open the ESP-IDF terminal. Then, run the following command to begin monitoring the output:

```
idf.py monitor
```

Now, walk between the two ESP32 boards and analyze the output on the laptop screen. There should be a message indicating a person was detected. You can start experimenting with different situations to evaluate how the trained model performs, such as:

- Walking at different speeds;
- Walking in different directions;
- Walking closer to one board than the other;
- Stopping between the boards for a couple of seconds;
- Increasing the distance between the boards.

As the provided data used to train the machine learning model only contains a few different walking speeds and orientations, you are likely to find circumstances in which the model does not perform as expected. These situations can be addressed by collecting more representative data and re-training the model on a larger dataset. Truly environmental-independent Wi-Fi sensing is still and open area of research, as discussed in the following section.

6.6. Challenges and future trends

This section outlines several key challenges currently facing Wi-Fi sensing, with particular emphasis on the growing concerns surrounding user privacy in sensing applications. In addition, we examine emerging research directions and future trends that are poised to shape the development of this field, including opportunities for improving scalability, robustness, and standardization of Wi-Fi-based sensing systems using IEEE 802.11bf.

6.6.1. Privacy concerns

As Wi-Fi sensing technologies have matured, achieving capabilities such as human activity recognition through walls [68, 69] and in non-line-of-sight scenarios [70], increasing attention has been directed toward the privacy implications of these techniques [71]. Given Wi-Fi's pervasive deployment in residential, commercial, and public environments, concerns have emerged regarding potential misuse by malicious actors. Specifically, a bad actor equipped with passive receivers could exploit Wi-Fi signal leakage — radio signals that propagate beyond the intended spatial domain — to infer human presence, activities, or movements, even when the data payload is encrypted. Since Wi-Fi CSI and RSSI metrics can be captured without requiring association with the access point, privacy breaches could occur without detection or consent. Nonetheless, it is important to highlight that accurate through-the-wall sensing remains technically challenging. Many proposed systems demonstrate high performance only under controlled experimental settings and often experience significant degradation when deployed in environments that differ from those on which they were trained [70].

Despite these concerns, Wi-Fi sensing could offer privacy-preserving alternatives to more intrusive modalities such as camera-based systems. Many applications — such as elder and child care — have traditionally relied on computer vision, which inherently poses higher risks of misuse due to the detailed and identifiable nature of visual data. In contrast, Wi-Fi sensing leverages low-dimensional radio signal features that are typically less sensitive. For example, in the domain of human activity detection, Zhou et al. [72] demonstrated that adversarial learning techniques can be used to selectively detect desired events (e.g., falls or other health-critical situations) while discarding non-relevant activities. This selective detection method ensures that, even in the event of unauthorized access to the sensing system, the potential for privacy invasion remains minimal. Such applications showcase how Wi-Fi-based solutions can balance monitoring capabilities with enhanced user privacy.

In response to the increasing relevance of privacy in wireless sensing, recent research has explored technical countermeasures to mitigate eavesdropping risks. One such approach involves leveraging multiple spatially distributed transmitters to boost authorized sensing performance while simultaneously inhibiting unauthorized listeners [73]. While effective, this strategy entails significant modifications to existing infrastructure, limiting its scalability and practical deployment. Consequently, developing privacy-preserving methods compatible with current Wi-Fi infrastructure remains an open and pressing research challenge. Furthermore, it is crucial to consider additional signal metrics — such as RSSI and Angle of Arrival (AoA) — that may be passively accessible to an adversary and could be exploited to infer transmitter locations or user presence. Addressing these multidimensional privacy threats is necessary to ensure safe and ethical deployment of Wi-Fi sensing in real-world environments.

6.6.2. Scalable, distributed and efficient sensing

While Wi-Fi sensing has demonstrated promising results across a range of applications, several critical challenges remain underexplored—particularly with regard to scalability and the coordination of multiple sensing devices in a shared environment [74]. The use of multistatic sensing configurations, where multiple access points or devices cooperatively sense a common target, has the potential to significantly enhance detection accuracy, spatial resolution, and robustness. However, coordinating multiple transmitters and receivers introduces complex synchronization, calibration, and communication overheads. Additionally, as the density of Wi-Fi devices increases within a given area, spectrum congestion and mutual interference become prominent issues, potentially impairing the quality of sensing and communication. Despite these concerns, few studies rigorously evaluate the performance of densely deployed Wi-Fi sensing systems through comprehensive network simulations [33] or real-world testbeds, leaving a gap in our understanding of their feasibility and reliability at scale.

Beyond distributed sensing architectures, another growing research frontier involves developing models and systems that generalize across environments. Most existing models are highly sensitive to environmental characteristics, often exhibiting significant performance degradation when deployed in physical spaces that differ from the ones in which they were trained. This issue is largely due to the complex and environment-specific nature of radio wave propagation. Variations in room geometry, wall materials, furniture placement, and human presence can all alter the multipath profile of a wireless signal, leading to changes in the CSI or RSSI that are not accounted for by models trained in different conditions. As a result, a model that performs well in one environment may yield poor results when applied elsewhere, even for the same sensing task.

To address this limitation, recent studies have investigated domain adaptation and transfer learning approaches that aim to bridge the gap between source and target environments. These methods attempt to reuse knowledge learned in a reference setting while fine-tuning the model with a small amount of new data from the target environment [2]. In parallel, there is growing interest in designing lightweight and efficient machine learning architectures suitable for deployment on resource-constrained IoT hardware. Such approaches not only reduce energy consumption and cost but also enable real-time, on-device inference, which is essential for privacy-sensitive or latency-critical applications. Together, these advancements aim to make Wi-Fi sensing more adaptable, scalable, and practical for widespread real-world deployment.

6.6.3. IEEE 802.11bf

IEEE 802.11bf is an upcoming standard from the IEEE, expected to be finalized in 2025, that defines how Wi-Fi networks can be used for sensing applications [14, 75]. It introduces significant enhancements to the Medium Access Control (MAC) layer to enable robust and standardized Wi-Fi sensing capabilities. These enhancements include new frame formats and procedures for initiating and coordinating sensing operations in both single- and multi-user environments. Specifically, the standard defines sensing measurement exchanges that allow devices to trigger CSI or Time-of-Flight (ToF) estimations, typically through the transmission and reception of Null Data Packet (NDP) sequences. The MAC layer is extended to support periodic and on-demand sensing tasks with coordinated channel access, ensuring minimal interference with ongoing data communication. Additionally, the concept of sensing sessions is introduced, wherein devices can negotiate sensing parameters — such as measurement type, duration, and reporting interval — to enable consistent and synchronized data collection across the network. These mechanisms facilitate more deterministic and repeatable CSI capture, which is essential for reliable inference in sensing applications.

At the Physical (PHY) layer, IEEE 802.11bf extends existing PHY modes — such as those defined in 802.11n, 802.11ac, 802.11ax, and 802.11ay — to better support sensing operations across both sub-7 GHz and millimeter-wave frequency bands. In particular, IEEE 802.11bf supports operation in the 60 GHz mmWave band as defined by 802.11ad/ay, leveraging the large available bandwidths and highly directional transmissions to achieve high-resolution sensing. mmWave sensing enables fine-grained estimation of object location, motion, and orientation due to its short wavelength and reduced multipath spread. The standard incorporates improvements such as refined phase tracking, standardized CSI reporting formats, and optimized use of NDP-based channel sounding to collect high-fidelity measurements without unnecessary data payloads. While it does not alter subcarrier granularity, the PHY layer enhancements improve the accuracy and consistency of CSI and ToF data. Moreover, IEEE 802.11bf facilitates advanced spatial awareness through MIMO and support for Angle of Arrival (AoA) and Angle of Departure (AoD) estimation, enabling detailed environmental mapping and object tracking [76]. Sub-meter accuracy is achievable in favorable conditions, particularly in controlled indoor environments. Collectively, the MAC and PHY updates in IEEE 802.11bf lay the foundation for interoperable, low-latency, and high-precision sensing systems that operate seamlessly across legacy and next-generation Wi-Fi bands, including mmWave.

6.7. Conclusion

In this chapter, we provided an overview of the current landscape in Integrated Sensing and Communications (ISAC), highlighting the sensing capabilities enabled by various wireless communication protocols, including LoRaWAN, Bluetooth, and Wi-Fi. Among these, Wi-Fi has emerged as the most promising candidate for low-cost and pervasive sensing applications, owing to its ubiquitous deployment, affordability, and the ability to extract fine-grained CSI directly from commercial off-the-shelf hardware. Far from being a purely academic concept, Wi-Fi sensing has demonstrated practical viability, as illustrated through both the real-world case study on animal detection in rural environments (Section 6.4) and the hands-on demonstration using low-cost ESP32 devices (Section 6.5).

Despite its potential, Wi-Fi sensing still faces several challenges that must be addressed to support large-scale, privacy-preserving, and robust deployment. Notable research opportunities include improving the generalization of sensing models across diverse environments and developing secure frameworks that preserve user privacy without the need for extensive infrastructural modifications.

Looking ahead, the ongoing standardization efforts under IEEE 802.11bf represent a major milestone in the evolution of Wi-Fi sensing. By introducing enhancements to the MAC and PHY layers specifically tailored for sensing, this forthcoming amendment is expected to further catalyze research and adoption, enabling more reliable, interoperable, and scalable sensing systems. As these technological advancements continue to unfold, Wi-Fi sensing is well-positioned to become a key enabler of intelligent and context-aware applications across a wide range of domains.

6.8. *Acknowledgments

Artur Jordao would like to thank grant #402734/2023-8, National Council for Scientific and Technological Development (CNPq) and Edital Programa de Apoio a Novos Docentes 2023 (Processo USP nº: 22.1.09345.01.2). Cintia B. Margi is supported by CNPq fellowship #311687/2022-9. This study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil, process numbers #2023/11163-0 and #2022/07523-8, and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. This research was supported in part by Itaú Unibanco S.A. through the Programa de Bolsas Itau (PBI). Any opinions, findings, and conclusions expressed in this manuscript are those of the authors and do not necessarily reflect the views, official policy, or position of the financiers.

This work uses icons by Freepik, bsd and Kalashnyk in multiple figures.

References

[1] J. C. H. Soto, I. Galdino, E. Caballero, V. Ferreira, D. Muchaluat-Saade, and C. Albuquerque, *Minicursos do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, ch. 5 - Monitoramento de Sinais Vitais Utilizando Redes Wi-Fi. Porto Alegre, RS, Brazil: Sociedade Brasileira de Computação, 2022.

[2] Q. Bu, X. Ming, J. Hu, T. Zhang, J. Feng, and J. Zhang, “Transfersense: towards environment independent and one-shot wifi sensing,” *Personal and Ubiquitous Computing*, vol. 26, 2022.

[3] J. Zhang, R. Xi, Y. He, Y. Sun, X. Guo, W. Wang, X. Na, Y. Liu, Z. Shi, and T. Gu, “A survey of mmwave-based human sensing: Technology, platforms and applications,” *IEEE Communications Surveys and Tutorials*, vol. 25, 2023.

[4] J. Song, R. He, Z. Zhang, M. Yang, B. Ai, H. Zhang, and R. Chen, “3d environment reconstruction based on isac channels,” in *2024 International Conference on Ubiquitous Communication (Ucom)*, pp. 487–491, 2024.

[5] M. I. Skolnik, *Introduction to Radar Systems*. McGraw-Hill, 3rd ed., 2001.

[6] A. Liu, Z. Huang, M. Li, Y. Wan, W. Li, T. X. Han, C. Liu, R. Du, D. K. P. Tan, J. Lu, Y. Shen, F. Colone, and K. Chetty, "A survey on fundamental limits of integrated sensing and communication," *IEEE Communications Surveys and Tutorials*, vol. 24, 2022.

[7] LoRa Alliance, "LoRaWAN® specification v1.1." `https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/`, October 2017. Available online: `https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/` (accessed April 10, 2025).

[8] M. Škiljo, T. Perković, Z. Blažević, and P. Šolić, "Performance analysis of antenna-based soil sensing," *IEEE Journal of Radio Frequency Identification*, vol. 8, pp. 68–75, 2024.

[9] Y. Ge, W. Li, M. Farooq, A. Qayyum, J. Wang, Z. Chen, J. Cooper, M. A. Imran, and Q. H. Abbasi, "Logait: Lora sensing system of human gait recognition using dynamic time warping," *IEEE Sensors Journal*, vol. 23, no. 18, pp. 21687–21697, 2023.

[10] X. Zeng, B. Wang, C. Wu, S. Deepika Regani, and K. J. Ray Liu, "Intelligent wi-fi based child presence detection system," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Singapore), pp. 11–15, IEEE, 2022.

[11] C. Shi, T. Zhao, Y. Xie, T. Zhang, Y. Wang, X. Guo, and Y. Chen, "Environment-independent in-baggage object identification using wifi signals," in *Proceedings - 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems, MASS 2021*, IEEE, 2021.

[12] W. Yang, X. Wang, A. Song, and S. Mao, "Wi-wheat: Contact-free wheat moisture detection with commodity wifi," in *2018 IEEE International Conference on Communications (ICC)*, (Kansas City, USA), pp. 1–6, IEEE, 2018.

[13] S. Tan, L. Zhang, and J. Yang, "Sensing fruit ripeness using wireless signals," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, (Hangzhou,China), pp. 1–9, IEEE, 2018.

[14] C. Chen, H. Song, Q. Li, F. Meneghello, F. Restuccia, and C. Cordeiro, "Wi-fi sensing based on ieee 802.11bf," *IEEE Communications Magazine*, vol. 61, p. 121–127, Jan. 2023.

[15] G. Iannizzotto, M. Milici, A. Nucita, and L. L. Bello, "A perspective on passive human sensing with bluetooth," *Sensors*, vol. 22, 5 2022.

[16] G. Maus and D. Brückmann, "A non-intrusive, single-sided car traffic monitoring system based on low-cost ble devices," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2020.

[17] M. Münch and F.-M. Schleif, "Device-free passive human counting with bluetooth low energy beacons," in *Advances in Computational Intelligence* (I. Rojas, G. Joya, and A. Catala, eds.), (Cham), pp. 799–810, Springer International Publishing, 2019.

[18] Bluetooth Special Interest Group, "Bluetooth Core Specification Version 5.1." `https://www.bluetooth.com/specifications/specs/core-specification-5-1/`, 2019. Accessed: 2025-04-10.

[19] H. Leijnse, R. Uijlenhoet, and J. N. M. Stricker, "Rainfall measurement using radio links from cellular communication networks," *Water Resources Research*, vol. 43, no. 3, 2007.

[20] N. David, P. Alpert, and H. Messer, "Novel method for water vapour monitoring using wireless communication networks measurements," *Atmospheric chemistry and physics*, vol. 9, no. 7, pp. 2413–2418, 2009.

[21] K. P. Hunt, J. J. Niemeier, L. K. da Cunha, and A. Kruger, "Using cellular network signal strength to monitor vegetation characteristics," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 2, pp. 346–349, 2011.

[22] W. Chen, K. Niu, D. Zhao, R. Zheng, D. Wu, W. Wang, L. Wang, and D. Zhang, "Robust dynamic hand gesture interaction using lte terminals," in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 109–120, 2020.

[23] Y. Chen, J. Zhang, W. Feng, and M. S. Alouini, "Radio sensing using 5g signals: Concepts, state of the art, and challenges," *IEEE Internet of Things Journal*, vol. 9, 2022.

[24] C. B. Barneto, M. Turunen, S. D. Liyanaarachchi, L. Anttila, A. Brihuega, T. Riihonen, and M. Valkama, "High-accuracy radio sensing in 5g new radio networks: Prospects and self-interference challenge," in *Conference Record - Asilomar Conference on Signals, Systems and Computers*, vol. 2019-November, 2019.

[25] R. Tavasoli, S. Sur, and S. Nelakuditi, "Sugarwave: A non-destructive estimation of fruit sugar content using millimeter-wave sensing," in *Proceedings - 2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems, MASS 2023*, 2023.

[26] P. Tosi, M. Henninger, L. G. De Oliveira, and S. Mandelli, "Feasibility of non-line-of-sight integrated sensing and communication at mmwave," in *IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC*, p. 331 – 335, 2024.

[27] M. Danneberg, R. Bomfin, A. Nimr, Z. Li, and G. Fettweis, "Usrp-based platform for 26/28 ghz mmwave experimentation," in *2020 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2020 - Proceedings*, 2020.

[28] N. González-Prelcic, M. Furkan Keskin, O. Kaltiokallio, M. Valkama, D. Dardari, X. Shen, Y. Shen, M. Bayraktar, and H. Wymeersch, "The integrated sensing and communication revolution for 6g: Vision, techniques, and applications," *Proceedings of the IEEE*, vol. 112, no. 7, pp. 676–723, 2024.

[29] C. Huang, S. Hu, G. C. Alexandropoulos, A. Zappone, C. Yuen, R. Zhang, M. D. Renzo, and M. Debbah, "Holographic mimo surfaces for 6g wireless networks: Opportunities, challenges, and trends," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 118–125, 2020.

[30] S. M. Hernandez and E. Bulut, "Wifi sensing on the edge: Signal processing techniques and challenges for real-world systems," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 46–76, 2023.

[31] Y. Liu, Z. Tan, H. Hu, L. J. Cimini, and G. Y. Li, "Channel estimation for ofdm," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1891–1908, 2014.

[32] IEEE, "Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp. 1–4379, 2021.

[33] S. V. Ducca, C. B. Margi, and A. J. L. Correia, "Low-cost animal and pedestrian crossing detection in rural roads using wifi sensing and deep learning," Master's thesis, Universidade de São Paulo, 2024.

[34] S. Tan, L. Zhang, and J. Yang, "Sensing fruit ripeness using wireless signals," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, 2018.

[35] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98–104, 2017.

[36] B. Tan, Q. Chen, K. Chetty, K. Woodbridge, W. Li, and R. Piechocki, "Exploiting wifi channel state information for residential healthcare informatics," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 130–137, 2018.

[37] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter Level Localization Using WiFi," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, (London United Kingdom), pp. 269–282, ACM, Aug. 2015.

[38] B. Yu, Y. Wang, K. Niu, Y. Zeng, T. Gu, L. Wang, C. Guan, and D. Zhang, "WiFi-Sleep: Sleep Stage Monitoring Using Commodity Wi-Fi Devices," *IEEE Internet of Things Journal*, vol. 8, pp. 13900–13913, Sept. 2021.

[39] C. Zakaria, G. Yilmaz, P. M. Mammen, M. Chee, P. Shenoy, and R. Balan, "Sleepmore: Inferring sleep duration at scale via multi-device wifi sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, Jan. 2023.

[40] W. Yang, E. Shen, X. Wang, S. Mao, Y. Gong, and P. Hu, "Wi-wheat+: Contact-free wheat moisture sensing with commodity wifi based on entropy," *Digital Communications and Networks*, vol. 9, no. 3, pp. 698–709, 2023.

[41] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, vol. 97, pp. 6105–6114, 2019.

[42] N. Maslej, L. Fattorini, R. Perrault, V. Parli, A. Reuel, E. Brynjolfsson, J. Etchemendy, K. Ligett, T. Lyons, J. Manyika, J. C. Niebles, Y. Shoham, R. Wald, and J. Clark, "Artificial intelligence index report 2024," 2024.

[43] Y. Bengio, S. Mindermann, D. Privitera, T. Besiroglu, R. Bommasani, S. Casper, Y. Choi, P. Fox, B. Garfinkel, D. Goldfarb, H. Heidari, A. Ho, S. Kapoor, L. Khalatbari, S. Longpre, S. Manning, V. Mavroudis, M. Mazeika, J. Michael, J. Newman, K. Y. Ng, C. T. Okolo, D. Raji, G. Sastry, E. Seger, T. Skeadas, T. South, E. Strubell, F. Tramèr, L. Velasco, N. Wheeler, D. Acemoglu, O. Adekanmbi, D. Dalrymple, T. G. Dietterich, E. W. Felten, P. Fung, P.-O. Gourinchas, F. Heintz, G. Hinton, N. Jennings, A. Krause, S. Leavy, P. Liang, T. Ludermir, V. Marda, H. Margetts, J. McDermid, J. Munga, A. Narayanan, A. Nelson, C. Neppel, A. Oh, G. Ramchurn, S. Russell, M. Schaake, B. Schölkopf, D. Song, A. Soto, L. Tiedrich, G. Varoquaux, A. Yao, Y.-Q. Zhang, F. Albalawi, M. Alserkal, O. Ajala, G. Avrin, C. Busch, A. C. P. de Leon Ferreira de Carvalho, B. Fox, A. S. Gill, A. H. Hatip, J. Heikkilä, G. Jolly, Z. Katzir, H. Kitano, A. Krüger, C. Johnson, S. M. Khan, K. M. Lee, D. V. Ligot, O. Molchanovskyi, A. Monti, N. Mwamanzi, M. Nemer, N. Oliver, J. R. L. Portillo, B. Ravindran, R. P. Rivera, H. Riza, C. Rugege, C. Seoighe, J. Sheehan, H. Sheikh, D. Wong, and Y. Zeng, "International AI safety report," 2025.

[44] Y. He and L. Xiao, "Structured pruning for deep convolutional neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2900–2919, 2024.

[45] H. Cheng, M. Zhang, and J. Q. Shi, "A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10558–10578, 2024.

[46] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *Journal of Machine Learning Research*, vol. 16, no. 89, pp. 2859–2900, 2015.

[47] B. Cancela, V. Bolón-Canedo, and A. Alonso-Betanzos, "E2E-FS: an end-to-end feature selection method for neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 7, pp. 8311–8323, 2023.

[48] S. V. Ducca and C. B. Margi, "Performance trade offs in iot-based traffic monitoring and incident detection systems," in *2022 Symposium on Internet of Things (SIoT)*, (São Paulo, Brazil), pp. 1–4, IEEE, 2022.

[49] S. V. Ducca, A. Jordão, and C. B. Margi, "Detection and classification of animal crossings on roads using iot-based wifi sensing," in *2023 IEEE Latin-American Conference on Communications (LATINCOM)*, (Panama City, Panama), pp. 1–6, IEEE, 2023.

[50] W. H. O. (WHO), *Global status report on road safety 2018: summary. (WHO/NMH/NVI/18.20)*. Geneva: World Health Organization, 2018.

[51] F. D. Abra, M. P. Huijser, M. Magioli, A. A. A. Bovo, and K. M. P. M. de Barros Ferraz, "An estimate of wild mammal roadkill in são paulo state, brazil," *Heliyon*, vol. 7, 2021.

[52] F. H. A. (FHWA), *Wildlife-Vehicle Collision Reduction Study: Report To Congress. U.S. Department of Transportation (FHWA-HRT-08-034)*. Federal Highway Administration (FHWA), 2008.

[53] J. Chen, H. Xu, J. Wu, R. Yue, C. Yuan, and L. Wang, "Deer crossing road detection with roadside lidar sensor," *IEEE Access*, vol. 7, 2019.

[54] N. Sharma and R. D. Garg, "Real-time computer vision for transportation safety using deep learning and iot," in *8th International Conference on Engineering and Emerging Technologies, ICEET 2022*, (Kuala Lumpur, Malasya), IEEE, 2022.

[55] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *7th International Conference on Learning Representations, ICLR 2019*, (New Orleans, United States), ICLR, 2019.

[56] F. Viani, A. Polo, E. Giarola, F. Robol, G. Benedetti, and S. Zanetti, "Performance assessment of a smart road management system for the wireless detection of wildlife road-crossing," in *IEEE 2nd International Smart Cities Conference: Improving the Citizens Quality of Life, ISC2 2016 - Proceedings*, (Trento, Italy), IEEE, 2016.

[57] P. K. Singh, K. N. Singh, M. K. Choudhury, J. Bharadwaj, R. J. Das, and D. Gogoi, "Advanced warning mechanism to reduce man animal conflict on roads using renewable energy," in *2022 IEEE International Power and Renewable Energy Conference (IPRECON)*, (Kerala , India), pp. 1–4, IEEE, 2022.

[58] N. Agafonovs, A. Skageris, G. Strazdins, and A. Mednis, "Imilepost: Embedded solution for dangerous road situation warnings," in *Proceedings - 1st International Conference on Artificial Intelligence, Modelling and Simulation, AIMS 2013*, (Kota Kinabalu, Malaysia), IEEE, 2014.

[59] M. Won, S. Sahu, and K. J. Park, "Deepwitraffic: Low cost wifi-based traffic monitoring system using deep learning," in *Proceedings - 2019 IEEE 16th International Conference on Mobile Ad Hoc and Smart Systems, MASS 2019*, (Monterey, USA), IEEE, 2019.

[60] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings - Conference on Local Computer Networks, LCN*, (Tampa, Florida), IEEE, 2006.

[61] "Omnet++ discrete event simulator." `https://omnetpp.org/`, 2024.

[62] "Openthread border router example." `https://github.com/espressif/esp-idf/tree/v5.2.2/examples/openthread/ot_br`, 2024.

[63] Z. Gao, Y. Gao, S. Wang, D. Li, and Y. Xu, "Crisloc: Reconstructable csi fingerprinting for indoor smartphone localization," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3422–3437, 2021.

[64] S. V. Ducca, "Dataset: Animal crossing wifi csi." https://doi.org/10.5281/zenodo.8266462, Aug 2023.

[65] S. Ducca, "Dataset: Channel state information data of animal crossings on rural roads." https://dx.doi.org/10.21227/0t6t-8s40, 2025.

[66] S. M. Hernandez and E. Bulut, "Lightweight and Standalone IoT Based WiFi Sensing for Active Repositioning and Mobility," in *21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM) (WoWMoM 2020)*, (Cork, Ireland), June 2020.

[67] G. Forbes, *CSIKit: Python CSI processing and visualisation tools for commercial off-the-shelf hardware.*, 2021.

[68] S. M. Hernandez and E. Bulut, "Adversarial occupancy monitoring using one-sided through-wall wifi sensing," in *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, 2021.

[69] Z. Wang, K. Jiang, Y. Hou, Z. Huang, W. Dou, C. Zhang, and Y. Guo, "A survey on csi-based human behavior recognition in through-the-wall scenario," *IEEE Access*, vol. 7, pp. 78772–78793, 2019.

[70] J. Geng, D. Huang, and F. D. la Torre, "Densepose from wifi," 2022.

[71] M. Duff, "Como o wi-fi sensing se tornou uma tecnologia funcional," *MIT Technology Review Brasil*, May 2024. Accessed: 2025-04-13.

[72] S. Zhou, W. Zhang, D. Peng, Y. Liu, X. Liao, and H. Jiang, "Adversarial wifi sensing for privacy preservation of human behaviors," *IEEE Communications Letters*, vol. 24, no. 2, pp. 259–263, 2020.

[73] S. M. Hernandez and E. Bulut, "Scheduled spatial sensing against adversarial wifi sensing," in *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 91–100, 2023.

[74] I. Ahmad, A. Ullah, and W. Choi, "Wifi-based human sensing with deep learning: Recent advances, challenges, and opportunities," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 3595–3623, 2024.

[75] T. Ropitault, C. R. D. Silva, S. Blandino, A. Sahoo, N. Golmie, K. Yoon, C. Aldana, and C. Hu, “Ieee 802.11bf wlan sensing procedure: Enabling the widespread adoption of wifi sensing,” *IEEE Communications Standards Magazine*, vol. 8, 2024.

[76] S. Blandino, T. Ropitault, C. R. D. Silva, A. Sahoo, and N. Golmie, “Ieee 802.11bf dmg sensing: Enabling high-resolution mmwave wi-fi sensing,” *IEEE Open Journal of Vehicular Technology*, vol. 4, 2023.