2026

REAL-LIFE HEALTH APPLICATIONS OF WIRELESS SENSING USING EDGE DEVICES

Md Touhiduzzaman
Virginia Commonwealth University

REAL-LIFE HEALTH APPLICATIONS OF WIRELESS SENSING USING EDGE DEVICES

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

MD TOUHIDUZZAMAN

Doctorate in Computer Science - August 2021 to July 2026

Director: Dr. Eyuphan Bulut,

Professor, Department of Computer Science

Virginia Commonwealth University

Richmond, Virginia

July, 2026

Acknowledgements

I would like to thank my advisor, Dr. Eyuphan Bulut, for his endless support from the very beginning of my doctoral studies. I am truly grateful for his invaluable guidance, expertise, encouragement, and boundless patience in transforming my naive research capabilities into those required to complete this doctoral thesis. I am also thankful to Dr. Steven M. Hernandez for my early research endeavors in the field of wireless sensing.

I would also like to extend my thanks to Dr. Kemal Akkaya, Dr. Tamer Nadeem, Dr. Changqing Luo, Dr. Peter E. Pidcoe, and Dr. Jane Chung for their gracious approval to serve on my dissertation committee amid their many commitments.

I am deeply grateful to my ever-supportive parents, my wife, my sisters, and my sweet daughters, whose unwavering sacrifices and encouragements have been the driving force behind all my enthusiasm and passion for academic and professional excellence. Above all, I am forever grateful to the Almighty for everything in this life and beyond.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract

REAL-LIFE HEALTH APPLICATIONS OF WIRELESS SENSING USING EDGE DEVICES

By Md Touhiduzzaman

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2026.

Director: Dr. Eyuphan Bulut,
Professor, Department of Computer Science

Wireless sensing has emerged as a transformative paradigm that leverages ubiquitous radio-frequency signals for environmental perception and human activity recognition beyond their primary role in data communication. Unlike wearable sensors and camera-based systems, wireless sensing offers a non-intrusive, privacy-preserving, and cost-effective alternative that operates seamlessly under low-lighting conditions, through walls, and in non-line-of-sight (NLoS) scenarios. The growing availability of edge computing devices further enables real-time on-device data processing, making wireless sensing solutions highly suitable for in-home health monitoring and smart living applications.

This dissertation investigates real-life health applications of wireless sensing through four completed systems, each addressing a distinct challenge in the deployment of ambient intelligence for healthcare and activity monitoring.

The first contribution is *Wi-PT-Hand*, a low-cost, end-to-end device-free system

for physical rehabilitation tracking of hand and finger movements. Using commercial off-the-shelf WiFi modules and deep learning models optimized for edge deployment, the system performs real-time segmentation, classification, and repetition counting of hand exercises, addressing a critical need in post-stroke and neurological rehabilitation. The system was subsequently extended into a clinically validated, spatially dynamic platform incorporating a Faraday-enclosed multi-link hardware setup, a speed-invariant signal processing pipeline combining Dynamic Time Warping (DTW) alignment with drop-frame data augmentation, and a validation-weighted convolutional neural network (CNN) ensemble with temporal majority voting. Experiments with 40 healthy volunteers and 11 patients diagnosed with Parkinson's disease or stroke yielded classification accuracies of 94.2%, 91.4%, and 86.3% on internal validation, external test, and clinical cohort datasets, respectively, with sub-300 ms inference on a Raspberry Pi 4B.

The second contribution extends to daily activity monitoring in senior housing environments. By deploying ESP32-based transceivers and Raspberry Pi modules in real apartment and home environments, the system collected naturalistic long-duration CSI datasets from 11 senior participants over five to nine days each, including participants with Parkinson's disease diagnoses. The deployment addressed practical challenges including signal interference, environmental changes, and user variability, and used supervised contrastive learning and stacking ensemble models to achieve cross-person and cross-environment generalization.

The third contribution introduces *MockiFi*, a novel data augmentation framework using Conditional Neural Processes (CNPs) for zero-shot learning. The approach generates synthetic Channel State Information (CSI) data for unseen user-activity pairs from a single reference activity of a new individual, using activity-to-activity transformation patterns learned from known users. Classifiers trained exclusively on

MockiFi-generated data achieved accuracies within one percentage point of classifiers trained on real data, significantly reducing the burden of data collection and labeling in new deployments.

The fourth contribution, *AgnoGest*, addresses a fundamental interference bottleneck in real-world wireless gesture recognition, i.e., the problem of distinguishing subtle hand gestures from dominant concurrent full-body locomotion. AgnoGest is an edge-optimized, locomotion-agnostic multimodal sensing platform that fuses 60 GHz mmWave radar point-cloud data and dual-link WiFi CSI on a Raspberry Pi 4B. A triple-adversarial learning architecture using Gradient Reversal Layers (GRL) suppresses locomotion-specific signatures across all three sensor streams, while dual-attention gated fusion dynamically weighs each modality's contribution to gesture and locomotion classification. The system achieves an average gesture recognition accuracy of 87.64% across entirely unseen volunteers over nine locomotion conditions, demonstrating that multimodal fusion of mmWave and WiFi substantially outperforms either modality alone in dynamic, real-world settings.

Together, these efforts establish a comprehensive path toward scalable, privacy-aware, and user-adaptive wireless sensing solutions for healthcare, assisted living, and personalized activity monitoring, with particular emphasis on edge deployability, clinical validity, and multimodal robustness.

CHAPTER 1

INTRODUCTION

Wireless radio-frequency (RF) signals permeate almost every indoor environment today. Beyond their role in high-speed data communication, these signals carry rich information about the physical environment they traverse. As they propagate from a transmitter to a receiver, RF signals reflect off walls, furniture, and human bodies, creating measurable variations in amplitude, phase, and spatial distribution. By carefully analyzing these variations, it is possible to sense human activities, gestures, and physiological patterns without requiring any device to be worn by the monitored individual. This dissertation explores two complementary wireless sensing modalities for real-life health applications. The first and primary modality is WiFi Channel State Information (CSI), which leverages the ubiquitous 2.4 GHz and 5 GHz WiFi signals present in almost every modern home to perform device-free activity recognition using low-cost hardware. The second modality is millimeter-wave (mmWave) radar, operating at 60 GHz, which provides high spatial resolution and precise Doppler velocity measurements. While each modality has distinct strengths and limitations, this dissertation demonstrates that fusing them within a multimodal framework yields sensing systems that are more robust, more generalizable, and better suited to the complex, dynamic conditions of real-world health monitoring. In general, this dissertation specifically focuses on three interconnected research directions:

1. WiFi sensing solutions for real-life health applications, including monitoring of physical therapy exercises and daily activity tracking of older adults.

2. Generation of synthetic training data for WiFi sensing models to reduce the

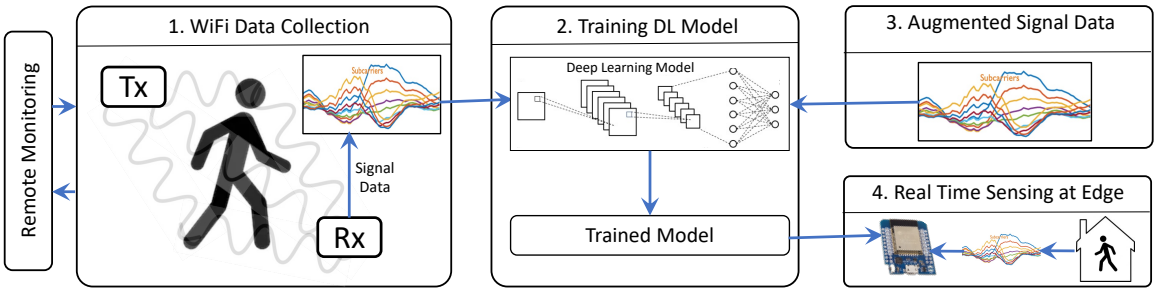


Fig. 1.: Summary of the dissertation.

data collection burden in new environments and for new individuals, enabling zero-shot learning.

3. Multimodal wireless sensing combining mmWave radar and WiFi CSI to address the locomotion-agnostic gesture recognition problem, where fine-grained hand gestures must be recognized reliably even when the user is simultaneously performing full-body locomotion such as walking, turning, or squatting.

Fig. 1 illustrates the relevance and connectivity of the technical topics covered in this dissertation.

1.1 Wireless Sensing Approaches

This dissertation uses two wireless sensing modalities throughout its chapters. This section provides a brief overview of each and explains the motivation for combining them.

1.1.1 WiFi Sensing

WiFi sensing provides a compelling alternative to traditional sensor-based systems, particularly when sensors such as wearables must be physically attached to participants. Although wearables can deliver precise and specialized sensing data,

they come with various limitations that WiFi sensing can effectively address. By analyzing CSI extracted from standard WiFi frames at the subcarrier level, fine-grained motion-induced perturbations of the wireless channel can be detected and classified without any device on the body of the monitored person.

1.1.1.1 Benefits of WiFi Sensing over Wearable Sensors

1. **Uninterrupted Monitoring:** Wearable devices are not always worn consistently. For instance, a smartwatch equipped with an accelerometer may be used to detect sudden falls, which is crucial for elderly care. However, such devices are often removed for charging at night or during fall-prone tasks like bathing, reducing their effectiveness. WiFi sensing, in contrast, can continuously and passively monitor activities without requiring any device on the body.

2. **Improved Comfort and Mobility:** Wearable sensors can feel intrusive or uncomfortable, leading to limited daily usage. Full-body motion capture wearable suits such as the Rokoko suit [1] and the Xsens MVN link suit [2] require wearing the complete suit and therefore limit each suit to a single volunteer [3]. Moreover, these often need wired power and data connections. Alternatively, multi-sensor solutions like the Polhemus G4 motion tracking system [4] and TEA CaptivL7000 physiological monitoring systems [5] can monitor high-precision limb movements and biosignals like ECG and EMG, but require belt-worn hubs and wired sensors that restrict natural movement. WiFi sensing eliminates the need for physical wearables, preserving user mobility and comfort.

3. **Cost-Effective for Multi-Person Scenarios:** Sensor-based systems typically focus on one individual at a time, which can become expensive and complex in environments with multiple people. WiFi sensing, although it introduces addi-

tional modeling challenges, inherently supports multi-person scenarios without scaling hardware costs proportionally.

1.1.1.2 Advantages of WiFi Sensing over Camera-Based Systems

1. **Enhanced Privacy:** Cameras are often perceived as invasive, especially in personal spaces like homes. Video data may unintentionally capture sensitive information. WiFi sensing, by contrast, involves signal-level analysis tailored to specific activities, with no visual data recorded.

2. **Line-of-Sight Independence:** Camera systems require a clear line-of-sight to detect and classify activities. WiFi signals propagate in all directions and can penetrate walls and furniture, enabling sensing in both line-of-sight and non-line-of-sight (NLoS) conditions.

3. **Lighting Agnostic:** Camera performance depends heavily on lighting conditions. In poorly lit environments or during nighttime, camera accuracy drops significantly. WiFi sensing operates independently of ambient lighting, making it robust for use at any time of day.

1.1.2 mmWave Radar Sensing

Millimeter-wave (mmWave) radar operates in the 30 GHz to 300 GHz frequency band, with the 60 GHz band being widely available in commercial off-the-shelf radar chips such as the Texas Instruments IWR6843ISK used in this dissertation. mmWave radars use Frequency Modulated Continuous Wave (FMCW) signals to construct precise spatial and velocity information about moving targets in their field of view. The processing pipeline converts raw analog-to-digital converter (ADC) samples into a sparse 3D point cloud through a sequence of signal processing steps. A Range

FFT converts the beat signal from each chirp into range bins, locating targets by their distance from the radar. A Doppler FFT across successive chirps within a frame estimates the radial velocity of each detected point. A Constant False Alarm Rate (CFAR) algorithm then identifies significant peaks against the noise floor, and Angle of Arrival (AoA) estimation from the phase differences across the antenna array provides the angular position of each point. The resulting 3D point cloud, with associated radial velocity measurements, provides a high-resolution spatial-temporal snapshot of the physical environment at up to 10 frames per second. Compared to WiFi CSI, mmWave radar sensing offers several distinct advantages:

1. **High Spatial Resolution:** mmWave radar achieves range resolutions on the order of centimeters (e.g., 7 cm with the IWR6843ISK) and sub-degree angular resolution, enabling precise localization of hand and finger movements within a user's personal space.

2. **Rich Velocity Information:** Doppler processing provides per-point radial velocity, which is a discriminative feature for fine-grained gesture recognition that is not directly available from WiFi CSI amplitudes.

3. **Compact and Self-Contained:** mmWave radar sensors are self-contained active sensing devices. Unlike WiFi sensing, they do not depend on an external transmitter, making them simpler to deploy in confined or custom enclosures.

4. **Deterministic Field of View:** The narrow, configurable field of view of mmWave radar naturally bounds the sensing region, reducing interference from out-of-range motion while improving signal-to-noise ratio for targets within range.

However, mmWave radar also has limitations that motivate its combination with

WiFi. Its narrow field of view means that targets that move outside the radar coverage (e.g., during full-body locomotion across a room) may be partially or fully occluded from the sensor. WiFi CSI, with its wide propagation area and NLoS sensitivity, complements the radar by providing continuous signal coverage across larger spaces.

1.1.3 Benefits of Combining WiFi and mmWave

WiFi and mmWave sensing occupy complementary positions in the design space of wireless sensing. WiFi offers broad, NLoS-robust, cost-free infrastructure coverage but has limited spatial resolution and is sensitive to multipath interference from concurrent full-body motion. mmWave radar provides precise spatial resolution and rich Doppler velocity signatures but is constrained to a narrow field of view and is affected by occlusion during locomotion. By fusing both modalities within a unified multimodal framework, it is possible to build systems that exploit each sensor's strengths while compensating for the other's weaknesses. In this dissertation, the AgnoGest system (Chapter 6) demonstrates this principle concretely: the mmWave branch extracts high-resolution point-cloud and velocity features for fine-grained gesture discrimination, while the dual WiFi CSI links provide continuous ambient coverage and NLoS robustness for locomotion modeling. A triple-adversarial architecture with Gradient Reversal Layers and dual-attention gated fusion combines these modalities to achieve locomotion-agnostic gesture recognition that no single modality could achieve alone.

1.2 Possibilities of Industry Applications

The commercial sector has shown significant interest in wireless sensing technologies, with multiple consumer-grade products now available. For instance, Google Soli [6] has demonstrated gesture sensing using radar, while Linksys Aware [7] enables WiFi-based motion sensing in home networks. Startups like Emerald Innovations [8]

and Origin Wireless [9] offer solutions for health monitoring and home automation using WiFi CSI analysis. Simultaneously, the evolution of edge computing has fueled the development of machine learning on-device. Hardware platforms such as Google's Edge TPU [10], Nvidia Jetson Nano [11], and Kendryte's K210 [12] are designed to run ML inference at the edge with low power and latency. These platforms are increasingly adopted in consumer and industrial products for smart security, gesture recognition, and health monitoring tasks. Despite advancements in hardware, end-to-end on-device learning remains an active research area, with most solutions focusing on inference rather than full model training. This presents a valuable opportunity for further collaboration between academia and industry. As wireless sensing technologies mature, combining them with real-time edge intelligence opens the door to scalable, private, and efficient solutions in healthcare, smart homes, security, and industrial automation. All works presented in this dissertation are carried out using ESP32 microcontrollers and Raspberry Pi devices, with machine learning models optimized in terms of memory and computational resource consumption to be runnable on such edge devices.

1.3 Dissertation Organization

The remainder of this dissertation is organized as follows:

1. Firstly, we provide background information and review existing literature in Chapter 2 related to WiFi sensing, mmWave sensing, multimodal wireless sensing, real-life deployment applications, and CSI data augmentation.

2. Next, our study of applying WiFi sensing to hand physiotherapy exercises is presented in Chapter 3. This chapter covers both the original Wi-PT-Hand system for small-scale hand movements and its clinically validated extension for

spatially dynamic, multi-direction rehabilitation exercises with real-time edge inference.

3. Chapter 4 describes our deployment and practical application of WiFi sensing tools to care for older adults in low-income senior housing apartments, including participants with Parkinson's disease diagnoses and experiments with cross-person and cross-environment generalization.

4. After this, in Chapter 5, we present our novel data augmentation approach using Conditional Neural Processes to enable zero-shot learning for WiFi sensing, reducing the need for training data collection in new environments and for new individuals.

5. Chapter 6 presents AgnoGest, our multimodal wireless sensing system that combines 60 GHz mmWave radar and dual-link WiFi CSI on a Raspberry Pi 4B edge device to perform locomotion-agnostic gesture recognition through a triple-adversarial framework with dual-attention gated fusion.

6. Finally, we conclude this dissertation in Chapter 7 with our remarks on these works and future directions.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Preliminary Theory on WiFi Sensing

To understand how wireless sensing with WiFi is technically feasible, we need to analyze the underlying communication principles. The following three subsections discuss the fundamental constructs central to most WiFi sensing systems.

2.1.1 Orthogonal Frequency-Division Multiplexing (OFDM)

OFDM is a widely used modulation scheme in modern wireless protocols, including IEEE 802.11n/ac/ax. It partitions the available spectrum into several orthogonal subcarriers, each modulated with a portion of the data stream. This enhances resistance against frequency-selective fading and intersymbol interference.

Each subcarrier in the frequency domain is modeled using a sinc function, defined as $sinc(f) = \frac{\sin(\pi f)}{\pi f}$. To convert these frequency-domain representations into time-domain signals suitable for transmission, the Inverse Fast Fourier Transform (IFFT) is applied, generating a time-localized composite signal, i.e., an approximate rectangle function. The *sinc* function is used as it allows each subcarrier to be placed orthogonally to one another in the frequency domain. Moreover, such subcarriers are spaced such that their peaks align with the nulls of other subcarriers, thereby preserving orthogonality. Each OFDM symbol, often lasting $3.2\mu s$ with a $0.8\mu s$ guard interval [13], can be modulated using schemes like Binary Phase-Shift Keying (BPSK,) Quadrature Phase-Shift Keying (QPSK), or Quadrature Amplitude Modulation (QAM), and represented as complex-valued symbols in the form of $s = I + jQ$, where I and Q

denote the in-phase and quadrature components, respectively. Each OFDM symbol represents binary data using complex I/Q samples, where the in-phase component (I) corresponds to the real part and the quadrature component (Q) corresponds to the imaginary part. For instance, a 16-QAM modulation scheme can encode 4 bits of data within a single complex symbol [14].

In a real-life environment, when wireless signals travel from a transmitter to a receiver, they often take multiple paths due to reflections off walls, furniture, or other obstacles. These multiple paths can cause constructive or destructive interference, which leads to a phenomenon called frequency-selective fading. Since OFDM transmits data across multiple subcarriers, each at a slightly different frequency, not all subcarriers are affected the same way by interference. Some subcarriers may experience signal boosts (constructive interference), while others may be weakened (destructive interference). This diversity across subcarriers makes OFDM more resilient to interference compared to single-carrier systems.

To help the receiver detect and correct for the varying effects of the wireless channel on each subcarrier, OFDM includes a few special subcarriers known as pilot subcarriers. These carry known reference signals instead of actual data. Since the receiver knows what these pilot symbols should look like, it can use them to estimate how the wireless channel is affecting different parts of the spectrum. This process is called subcarrier equalization, and it enables the system to correct distortions in the signal and recover the original transmitted data more accurately.

In a standard WiFi channel that uses a 20 MHz bandwidth, OFDM divides the channel into 64 subcarriers, each spaced 312.5 kHz apart. These subcarriers are centered around a carrier frequency. For example, WiFi Channel 1 is centered at 2.412 GHz, and its subcarriers span from 2.401 GHz to 2.423 GHz. The subcarriers are indexed relative to the center: subcarrier 0 is the direct-current (DC) subcarrier

(which is typically unused to avoid hardware distortion), subcarriers -21, -7, 7, and 21 are designated as pilot subcarriers, and all other subcarriers between -26 and +26 carry actual encoded data. The remaining subcarriers at the edges are null subcarriers, which serve as guard bands to reduce interference with adjacent channels. This layout helps isolate signals, maintain orthogonality between subcarriers, and ensures data is transmitted reliably even in challenging wireless environments.

In summary, OFDM works by splitting the data into many parallel streams and transmitting them over multiple narrowband subcarriers. These subcarriers are mathematically orthogonal to one another, meaning they can overlap in frequency without interfering, which allows OFDM to make very efficient use of the available spectrum. With the help of pilot subcarriers, the receiver can estimate and compensate for the channel effects on each subcarrier, making OFDM highly effective for modern wireless communication systems like WiFi.

2.1.2 Received Signal Strength Indicator (RSSI)

RSSI is a coarse-grained metric indicating the total power received over a wireless link, typically measured in dBm. Unlike CSI, which provides subcarrier-level resolution, RSSI offers a scalar estimate of signal strength, integrating over the full bandwidth.

Given a received signal $r(t)$ at time t, the instantaneous power is, $P(t) = |r(t)|^2$. RSSI is generally computed as an average over time or packet duration:

$$\text{RSSI}_{\text{avg}} = 10 \log_{10} \left(\frac{1}{T} \int_0^T |r(t)|^2 dt \right)$$

In practical systems, if P_{recv} is the received power and N_{floor} is the noise floor, then RSSI is often approximated as, RSSI = $(P_{recv} - N_{floor})$. Although less precise than CSI, RSSI can still capture environmental changes due to motion or obstructions

with lower complexity, making it useful for lightweight sensing applications having hardware or computational constraints.

2.1.3 Channel State Information (CSI)

OFDM systems typically use CSI to describe amplitude and phase variations across subcarrier frequencies as wireless signals are transmitted between a transmitter and a receiver. Channel estimation is the process used to detect variations across the subcarriers in OFDM systems through the transmission of a set of known shared pilot symbols in a comb-type pilot pattern [15] where the pilot subcarriers remain the same over time. The signal vector detected at the receiver $\mathbf{y}$ is modeled as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\eta} \tag{2.1}$$

where $\mathbf{x}$ is the transmitted signal vector, $\mathbf{H}$ is the channel matrix containing complex coefficients for each subcarrier i, and $\boldsymbol{\eta}$ denotes additive white Gaussian noise vector. For subcarrier i, the channel frequency response (CFR) can be denoted as:

$$h_i = A_i e^{j\phi_i} \tag{2.2}$$

It contains both real ($\mathcal{R}(h_i)$) and imaginary ($\mathcal{I}(h_i)$) parts. The amplitude A_i and phase ϕ_i can be recovered from these real and imaginary components for subcarrier i as:

$$A_i = \sqrt{\mathcal{R}(h_i)^2 + \mathcal{I}(h_i)^2}, \quad \phi_i = \text{atan2}(\mathcal{I}(h_i), \mathcal{R}(h_i)) \tag{2.3}$$

However, due to the complexity of any given environment, the signal received is not only a result of a direct line of sight (LoS) transmission, but is also affected by the environmental multipath — the multiple physical paths that the signal travels from the transmitter (TX) to the receiver (RX). Considering this kind of multipath

propagation, the channel response becomes a sum over multiple paths:

$$h_i = \sum_{m=1}^{N} A_m e^{\frac{-2\pi j f_i d_m}{c} + j\phi_m} \tag{2.4}$$

Here, A_m, d_m, ϕ_m denote the amplitude, distance, and phase shift of the m-th path, f_i is the subcarrier frequency, and c is the speed of light. Each multipath component influences the received signal in several ways. First, it undergoes *amplitude attenuation* due to interactions with environmental surfaces. Second, it experiences *time delays and phase shifts* based on the propagation distance of the path. Although the physical lengths of these paths remain consistent across all subcarriers, each subcarrier may still encounter different levels of *frequency-selective fading*. This occurs due to the constructive or destructive combination of signals arriving at slightly different frequencies.

OFDM systems are well-equipped to handle this type of fading because each subcarrier operates at a distinct frequency (f_i). When certain subcarriers suffer from destructive interference, others are likely unaffected, allowing data to still be transmitted reliably. This frequency diversity adds robustness to communication in environments with rich multipath propagation.

In the context of *device-free human sensing*, signal paths can be broadly categorized into two sets. The first set, denoted by Ω_s, represents *static paths* that reflect off stationary objects like walls and furniture and do not change over time. The second set, Ω_d, includes *dynamic paths* that are influenced by the movement of human targets in the environment.

The overall channel frequency response at subcarrier i can be modeled as:

$$h_i = \underbrace{\sum_{m \in \Omega_s} A_m e^{-2\pi j f_i d_m / c + j\phi_m}}_{h_{\text{static}}} + \underbrace{\sum_{n \in \Omega_d} A_n e^{-2\pi j f_i d_n / c + j\phi_n}}_{h_{\text{dynamic}}} \tag{2.5}$$

Here, A_m and A_n denote the amplitude, d_m and d_n are the distances of propagation, ϕ_m and ϕ_n are the respective phase shifts, and c is the speed of light.

Since the static component h_{static} remains constant over time, a carefully designed neural network, i.e., a machine learning model, can be leveraged to nullify it from the total signal utilizing the dynamic component h_{dynamic}. This step is crucial because the *LOS* path, which typically has the highest amplitude, is included in the static part. Its removal prevents it from overshadowing the more subtle variations caused by human motion.

The resulting dynamic signal is now primarily shaped by human activity, improving the system's sensitivity to motion. This also enhances the robustness of wireless sensing models to static environmental changes, enabling more accurate and reliable activity recognition [16].

For time-domain analysis, Channel Impulse Response (CIR) for time n is derived via IFFT:

$$H_n = \sum_{m=0}^{N-1} h_m e^{-j2\pi nm/N} \tag{2.6}$$

On the other hand, this CIR can be transformed back to the frequency domain through FFT, as:

$$h_m = \sum_{n=0}^{N-1} H_n e^{j2\pi nm/N} \tag{2.7}$$

CSI can thus be retrieved, consisting of both the static and dynamic paths, i.e.:

$$h_i = h_{\text{static}} + h_{\text{dynamic}} \tag{2.8}$$

CSI captures variations in both amplitude and phase across multiple subcarri-

ers and antennas, enabling fine-grained motion tracking. Compared to RSSI, CSI provides:

- Higher spatial and temporal resolution;
- Fine-grained tracking of environmental dynamics across subcarriers;
- Greater robustness for activity recognition and localization; and
- Richer data structure for deep learning-based modeling.

That is why we have used mainly CSI for all WiFi sensing works presented in this dissertation. Throughout these works, we collect a sequence of CSI measurements over a sliding time window of length w. These measurements are organized into an $S \times w$ matrix, denoted as $\mathbf{H}[t]$, where S represents the number of subcarriers and each column corresponds to a CSI sample collected at a specific time frame. The matrix is defined as:

$$\mathbf{H}[t] = \begin{bmatrix} h_1[t-w+1] & h_1[t-w+2] & \cdots & h_1[t] \\ h_2[t-w+1] & h_2[t-w+2] & \cdots & h_2[t] \\ \vdots & \vdots & \ddots & \vdots \\ h_S[t-w+1] & h_S[t-w+2] & \cdots & h_S[t] \end{bmatrix} \tag{2.9}$$

Here, each $h_s[t]$ represents the CSI value for subcarrier s at time t. After applying signal preprocessing techniques (e.g., filtering, normalization, or transformation), the matrix $\mathbf{H}[t]$ is used as input to a machine learning model for WiFi sensing tasks such as activity recognition, localization, or motion detection.

2.1.4 WiFi Sensing Applications

WiFi sensing has enabled a wide range of innovative sensing applications since its emergence. Based on our review of existing literature, we organize this section

into seven key research areas: (1) activity and people recognition, (2) gesture and fine-grained movement detection, (3) behavior and contextual sensing, (4) health monitoring and rehabilitation tracking, (5) localization, (6) other fields of applications, and (7) security and privacy preservation in WiFi sensing.

2.1.4.1 People and Activity Monitoring

WiFi-based Human Activity Recognition (HAR) has emerged as a compelling alternative to traditional sensing approaches such as wearable sensors and vision-based systems. Wearable devices, although capable of capturing high-resolution motion data via accelerometers and gyroscopes, require user compliance and may be considered uncomfortable or intrusive [17]. Camera-based methods, such as those using depth cameras or RGB video feeds, suffer from privacy concerns and often require direct line-of-sight and favorable lighting conditions [18].

WiFi sensing leverages variations in CSI induced by human movement to perform HAR passively. Systems like E-eyes [19] and WiTrace [20] showed that a variety of gross body movements (e.g., walking, falling, sitting) can be detected using commodity WiFi hardware. These systems primarily operate in constrained laboratory environments and often depend on expensive equipment such as Intel 5300 NICs [21].

Environment-agnostic CSI modeling is another key area of development, with techniques including antenna ratio analysis and voting schemes across subcarriers to mitigate environmental variance [22]. Other recent studies have explored deep unsupervised domain adaptation to reduce cross-environment training data dependency [23]. Despite promising results, many of these solutions are yet to be validated in long-term, real-life deployments.

There are a few studies [24, 25, 26, 27] exploring possibilities of deploying a series of wireless sensors like ZigBee, XBee, Arduino, temperature sensors, contact

sensors, LPG sensors, GSM modules, wearable smartwatch, and so on forming a Wireless Sensor Network (WSN) to remotely monitor wellbeing of the elderly people. However, none of these studies utilizes ubiquitous ambient WiFi signals to track and monitor people.

A significant number of studies [28, 29, 19, 30, 31, 20] explore the usage of several WiFi capable devices to collect CSI data and then recognize several activities, even with centimeter-level accuracy in passive gesture tracking. However, only a handful of the studies [**wifederated**, 32, 33, 34] explore the usage of low-cost devices like ESP32 for WiFi sensing.

2.1.4.2 Gesture and Fine-Grained Movement Detection

WiFi sensing has also been applied to detect finer movements such as gestures and limb articulation. This includes hand, finger, or object-interaction gestures, often demanding higher temporal and spatial resolution. Work by Venkatnarayan et al. [35] demonstrated multi-user gesture recognition using commercial WiFi. Tang et al. [36] proposed a parallel LSTM-FCN network for hand gesture classification based on CSI signals.

WiFi-based gesture tracking systems have also evolved through fusion with filtering and transformation techniques. DeepSeg [37] utilizes convolutional neural networks to jointly segment and classify actions. Transtrack [38] integrates Hampel filtering, PCA, and recursive Otsu thresholding for segmenting whole-body activities. Wi-Gym [39] employs a variety of techniques including Gabor filters and dynamic time warping (DTW) to assess physical performance in gymnastics.

However, these works primarily target large-scale gestures or full-body motions. Segmentation and counting of small-scale activities such as finger movements remain significantly underexplored in these frameworks. Additionally, most systems lack

modularity and scalability when introducing new activity classes.

2.1.4.3 Behavior and Contextual Sensing

Beyond direct activity recognition, WiFi sensing has been used for behavior and environmental context detection. Mehmood et al. [40] proposed a WiFi-based solution for real-time occupancy estimation, including applications for smart transportation. FallDeFi [41] presents a real-time fall detection framework using commodity WiFi, showcasing the use of CSI phase variations. Additionally, many systems are sensitive to transient environmental changes or hardware limitations. Works such as Anti-Fall [42] and RT-Fall [43] attempt to build resilience by using signal statistics like standard deviation of phase difference or band-pass filtered CSI values. Still, adaptive techniques remain an open area of research for behavior modeling. These can help ensure the safety of the elderly or sick individuals without sacrificing their privacy within their own homes.

CSI amplitude and phase differences have also been used for environmental analysis tasks, including soil moisture tracking [44] and in-baggage object detection [31]. These applications demonstrate the breadth of WiFi sensing but are typically focused on static environmental conditions rather than active human behavior.

2.1.4.4 Health Monitoring and Rehabilitation Tracking

There is growing interest in leveraging wireless sensing for health-related monitoring, including respiration monitoring, physical rehabilitation and elderly care. In the case of respiration and breathing monitoring, several studies [45, 46, 47, 48, 49] exist with systems detecting the status of respiration and counting breathing of a specific person, even when there are multiple person present in the environment. Other than these, most market-ready solutions today rely on either Inertial Measurement

Unit (IMU)-based wearables or camera systems such as Kinect [50] and RGB-D sensors [51]. However, these solutions have downsides, including high cost, privacy issues, and constrained mobility.

Several studies explore mmWave or RF radar for rehabilitation tasks. MARS [52] uses mmWave-based sensing for posture assessment and guided rehabilitation. These systems show promise in capturing movement even beyond line-of-sight, but often require expensive radar hardware and are not yet suitable for home-based applications.

On the WiFi sensing side, recent studies such as DeepSeg [37] and U-Shape networks [53] have demonstrated the potential for action segmentation and recognition, though largely for whole-body actions. Few have addressed segmentation and counting of small-scale physiotherapy movements, which demand higher temporal fidelity.

Additionally, ensemble learning techniques, as shown in works like StackFi [54], spatially distributed setup [55], and ensemble learning for WiFi counting [56], indicate that combining multiple models can enhance robustness across dynamic user scenarios. However, lightweight deployment on edge devices and segment-specific counting remain largely unaddressed in current research.

2.1.4.5 Localization

Localization is one of the earliest and most studied applications of WiFi sensing, aiming to track the physical position of a target using ambient WiFi signals. Traditional approaches often require the tracked subject to carry a transmitting or receiving device. For example, the system in [57] uses static WiFi anchors to determine the target's location. Other methods like relative positioning use mobile or semi-static anchors for relative localization [58, 59]. UAV-based localization also emerges as a unique direction where drones equipped with WiFi antennas act as both transmitters

and receivers [60, 61].

Device-free localization removes the requirement for the target to carry any WiFi device. Systems like CSDict [62] and the model proposed by Zhou et al. [63] construct databases of environmental CSI fingerprints across various locations to match and infer a target's position. These fingerprinting techniques, while effective, can be sensitive to changes in the environment, such as furniture movement or structural modifications.

To mitigate environmental sensitivity, recent works explore domain adaptation [64, 65] and transfer learning [66] strategies. These approaches aim to enhance the generalizability of learned models across different settings without requiring recollection of training data in every new environment.

In parallel, occupancy estimation and indoor crowd counting have gained attention for applications like retail analytics [67], security monitoring [68], and emergency evacuation planning [69]. While mobile crowd estimation techniques rely on detecting continuous motion [70], stationary crowd sizes can also be inferred by analyzing minor movements such as fidgeting [71].

2.1.4.6 Other Fields of Applications

Beyond the well-established applications of WiFi sensing with extensive research backing, several emerging use cases have garnered relatively limited attention in research. For instance, EmoSense [72] employs WiFi sensing to infer a subject's emotional state by analyzing physical movements, offering a means of monitoring individual mental health, a capability with promise in diverse applications beyond traditional methods. WiEat [73] utilizes movement patterns to identify eating behaviors, supporting health and dietary management by recognizing subtle actions associated with food consumption. WiFi sensing extends to tracking qualities in food and agricul-

ture, such as assessing fruit ripeness [74] and monitoring wheat moisture levels [75] to prevent spoilage, enabling non-invasive assessment of crop conditions. Liquid temperature detection [76], level monitoring [77], and identification [78] have been achieved through WiFi sensing by measuring the dielectric properties and resonance frequency response of liquids subjected to vibration, demonstrating the potential for remote liquid analysis. Additionally, vibration detection via WiFi sensing [79] is being explored for detecting anomalies in industrial equipment, presenting opportunities for predictive maintenance and fault detection in manufacturing settings. Further research directions include unobtrusive emotion recognition systems leveraging WiFi and vision-based methods [80], innovative eating recognition systems using head and face motions recorded by in-ear sensors [81, 82], and the application of wireless sensor networks to monitor agricultural land and display sensed parameters [83], opening new possibilities for comprehensive environmental monitoring.

2.1.4.7 Security and Privacy Preservation in WiFi Sensing

Privacy implications have been raised regarding WiFi sensing, especially when individuals may be tracked without consent. As highlighted in [34], ambient WiFi signals can be exploited to breach location privacy in non-public or personal environments. The study [84] uses a scheduling scheme over spatially distributed transmitters, preventing eavesdroppers from performing accurate WiFi sensing while allowing legitimate users to function correctly. An adversarial deep learning approach is presented in [85] to modify WiFi CSI data, ensuring privacy preservation for sensitive human behaviors while maintaining accuracy for other activities. A privacy-conscious WiFi-based crowd monitoring system is proposed in [86] that uses probe request bursts for crowd counting, ensuring strict privacy standards through anonymization techniques.

2.2 Preliminary Theory on mmWave Radar Sensing

Millimeter-wave (mmWave) radar is an active RF sensing modality operating in the 30–300 GHz frequency band. Unlike WiFi sensing, which passively exploits ambient communication signals, a mmWave radar transmits its own Frequency Modulated Continuous Wave (FMCW) chirp signals [87] and analyzes the returned reflections from the environment. This self-contained design makes it independent of any external wireless infrastructure. Commercial off-the-shelf chips, such as the Texas Instruments IWR6843ISK operating at 60 GHz, achieve range resolutions on the order of centimeters and sub-degree angular resolution with a compact chip-level form factor and energy-efficient design [88, 89]. These properties make mmWave radar well suited for sensing applications including gesture recognition [90], sound and vibration detection [91], and fine-grained human motion tracking [92, 93, 94].

The processing pipeline of an mmWave FMCW radar converts raw analog-to-digital converter (ADC) samples into a structured 3D point cloud through a sequence of signal processing stages [95, 96]. The following subsections describe each stage and the derived features that serve as inputs to subsequent machine learning models.

2.2.1 Range FFT

The radar receiver mixes the transmitted chirp signal with the reflected echo to produce an intermediate frequency (IF), or beat, signal. For a target at distance R, the IF signal frequency f_{if} is directly proportional to the round-trip time of flight. The Range FFT converts N_{adc} time-domain ADC samples per chirp into frequency-domain range bins:

$$x_{range}(k) = \sum_{n=0}^{N_{adc}-1} x_{adc}(n)\, e^{-j\frac{2\pi}{N_{adc}}nk}. \tag{2.10}$$

The physical distance R of the target corresponding to bin k is then recovered as:

$$R = \frac{c \cdot f_{if}}{2 \cdot S}, \tag{2.11}$$

where c is the speed of light and S is the chirp ramp slope. Each range bin therefore represents a thin shell in the radar's field of view at a specific distance from the sensor.

2.2.2 Doppler FFT

To estimate the radial velocity of each detected target, a second FFT is applied across N_{chirp} successive chirps within a single radar frame. This two-dimensional (2D) FFT captures the inter-chirp phase shift $\Delta\phi$ caused by target motion between consecutive chirps:

$$X_{doppler}(m) = \sum_{q=0}^{N_{chirp}-1} X_{range}(q)\, e^{-j\frac{2\pi}{N_{chirp}}qm}. \tag{2.12}$$

The radial velocity v_r of the target is derived from the measured Doppler frequency f_D as:

$$v_r = \frac{\lambda \cdot f_D}{2} = \frac{\lambda \cdot \Delta\phi}{4\pi \cdot T_c}, \tag{2.13}$$

where λ is the wavelength of the carrier signal and T_c is the chirp cycle duration. The resulting Range-Doppler map simultaneously encodes the distance and radial velocity of all targets in the field of view.

2.2.3 CFAR Detection and Angle of Arrival Estimation

To isolate physically meaningful reflections from background clutter and thermal noise, a Constant False Alarm Rate (CFAR) algorithm [97] is applied to the Range-Doppler map. CFAR adaptively identifies local power peaks by comparing each cell's power level to the average noise floor estimated from a window of surrounding refer-

ence cells:

$$P_{target} > \alpha \cdot \frac{1}{N_{ref}} \sum_{i=1}^{N_{ref}} P_{noise,\,i}, \tag{2.14}$$

where α is a detection threshold factor and N_{ref} is the number of reference cells. Only cells exceeding this adaptive threshold are retained as valid target detections.

For each CFAR-detected point, the Angle of Arrival (AoA) is estimated from the phase difference ω measured across the receive antenna array elements separated by spacing d:

$$\omega = \frac{2\pi d \sin(\theta)}{\lambda} \quad \Longrightarrow \quad \theta = \arcsin\left(\frac{\lambda\,\omega}{2\pi d}\right), \tag{2.15}$$

where θ is the angular position of the target relative to the radar boresight. Combining range R, radial velocity v_r, and angle θ yields a spatially structured 3D point cloud (x, y, z) with associated radial velocity v_r for each detected scatterer.

2.2.4 Velocity Feature Extraction

The raw per-point radial velocities in the point cloud mix the bulk motion of the body (e.g., full-body locomotion during walking) with the local motion of specific body parts (e.g., fine-grained hand gestures). To disentangle these two components, the mean differential velocity v_d is computed per radar frame from the point cloud $P = \{p_1, \ldots, p_N\}$:

$$v_d = \frac{1}{N} \sum_{i=1}^{N} |v_i - v_r|, \quad \text{where} \quad v_r = \frac{1}{N} \sum_{i=1}^{N} v_i. \tag{2.16}$$

By centering the velocity distribution around the global radial mean v_r, this calculation effectively suppresses uniform locomotion contributions from all points moving together with the body. Consequently, v_d highlights high-frequency localized bursts of motion, such as those generated by hand movements, serving as a critical discriminative feature for gesture recognition even in the presence of concurrent full-body

locomotion [89].

2.2.5 Point-Cloud Representation and Voxelization

The raw output of the mmWave pipeline is a sparse, variable-size set of detected 3D points per frame. Deep learning models operating on such data require a fixed-size, structured representation. Voxelization discretizes the continuous 3D coordinate space within a physical bounding box into a regular volumetric grid of dimensions $N_z \times N_y \times N_x$. Each point $p_i(x_i, y_i, z_i)$ is mapped to an integer grid cell (i_x, i_y, i_z) as:

$$i_x = \left\lfloor \frac{x_i - x_{min}}{\Delta_x} \right\rfloor, \quad i_y = \left\lfloor \frac{y_i - y_{min}}{\Delta_y} \right\rfloor, \quad i_z = \left\lfloor \frac{z_i - z_{min}}{\Delta_z} \right\rfloor, \tag{2.17}$$

where the cell resolutions $\Delta_x = (x_{max} - x_{min})/N_x$, $\Delta_y = (y_{max} - y_{min})/N_y$, and $\Delta_z = (z_{max} - z_{min})/N_z$ are defined by the physical extent of the sensing region. Points falling outside the bounding box are discarded as out-of-range artifacts.

Multiple points can map to the same voxel cell. Per-cell features are computed by aggregating all contributing points, with two channels commonly used: (i) an occupancy channel recording a binary flag indicating whether the cell contains at least one point, and (ii) a velocity channel recording the mean differential velocity v_d (Eq. (2.16)) of all points within the cell. This dual-channel representation captures both the spatial structure and the local motion dynamics of the gesture within a compact, fixed-dimension tensor that is compatible with standard 3D convolutional neural network architectures [95]. Compared to point-cloud processing methods such as PointNet++ [98], voxelization offers deterministic inference latency and hardware-efficient 3D convolution kernels, making it more suitable for real-time deployment on resource-constrained edge devices.

2.2.6 Applications of mmWave Radar Sensing

mmWave FMCW radar sensing has been applied to a wide variety of human-centric and environmental sensing tasks, spanning from fine-grained gesture detection to large-scale crowd monitoring.

2.2.6.1 Gesture and Hand Motion Recognition

The high spatial and Doppler resolution of mmWave radar makes it particularly well suited for detecting subtle hand and finger movements. Lien et al. [90] introduced Project Soli, which uses a 60 GHz radar to recognize micro-gestures for touchless interaction with consumer devices. Palipana et al. [99] demonstrated that sparse mmWave point clouds are sufficient for recognizing a rich vocabulary of mid-air gestures. The RadHAR framework [95] established that voxelized point-cloud representations processed by 3D convolutional neural networks yield strong performance on gesture and activity datasets, while more recent systems like MHTrack [100] use transformer-based encoders to isolate hand micro-motions even while the user is walking or turning.

2.2.6.2 Human Activity Recognition and Localization

At the activity level, mmWave radar provides fine-grained sensing categories including human tracking, motion recognition, and biometric measurement [89]. Recent systems such as mmRidge [101] use high-resolution Doppler and spatial frequency measurements to identify emergent crowd flow structures, while mmSnap [102] achieves instantaneous target position estimates through Bayesian one-shot fusion in a self-calibrated radar network. Single-radar configurations have also enabled privacy-preserving crowd analytics and occupancy estimation [103].

2.2.6.3 Health Monitoring

mmWave radar's sensitivity to minute vibrations and positional changes enables contactless monitoring of physiological signals. Wang et al. [104] demonstrated contactless Heart Rate Variability (HRV) monitoring by decomposing minute chest wall movements from a 60 GHz radar stream. Biomedical radar technology is showing broader transformative potential in clinical monitoring applications, including respiration, gait analysis, and fall detection [105].

In physical rehabilitation contexts, mmWave sensing offers the ability to capture movement even in partial NLoS conditions. MARS [52] uses mmWave-based sensing for posture assessment and guided exercise rehabilitation, demonstrating clinical promise. However, dedicated mmWave radar systems often require expensive and specialized hardware, which remains a barrier to home deployment compared to low-cost WiFi-based alternatives.

2.2.6.4 Mapping, Localization, and Autonomous Systems

In automotive and industrial contexts, FMCW radar has been deployed for mapping and localization [106], human activity recognition in outdoor environments [107], facial expression tracking [108], and signal classification [109]. Deep learning approaches have further enhanced RF sensing capabilities in outdoor scenarios [110], and RF signals have demonstrated reliable differentiation between cars and pedestrians for autonomous driving applications [111]. Recent hardware and algorithmic advances have resolved historical trade-offs between sensing distance and resolution, opening new frontiers in military surveillance and high-precision automotive radar [112].

2.2.6.5 Multimodal Sensing with WiFi

WiFi is often fused with vision-based modalities [113, 114, 115] for activity or human detection, which is occasionally coupled with audio too [116]. RFID [117], IMU and other wearable inertial sensors [118, 119] are also fused with WiFi to perform multimodal sensing of activities and environmental metrics. A growing body of work recognizes that mmWave and WiFi sensing occupy complementary positions in the design space [120], and that their combination can overcome limitations of either modality alone. Multimodal benchmarks such as MM-Fi [121] synergize mmWave, WiFi CSI, LiDAR, and RGB-D camera data for robust human activity recognition. However, most existing multimodal systems do not address locomotion-agnostic gesture recognition or real-time edge deployment, both of which are addressed in Chapter 6.

2.3 Data Augmentation

Augmentation of training data can help increase the accuracy and efficiency of a machine learning model by providing a variety to training data. This is widely applied for image data but in recent studies [122, 123, 124, 125], we also see augmentation techniques for CSI data. Most of these techniques are borrowed from computer vision domain thus CSI data is first converted to spectrogram images for proper application of these techniques. Leveraging the principles of zero-shot [126] and few-shot learning [127], synthetic CSI samples are also generated to reduce the volume of the data needed to train models. This is usually achieved by leveraging additional information, such as the word and attribute embeddings from the text-domain as it is utilized in [128]. Some other studies have also used GANs [129] or their variants (e.g., conditional GAN [130]) to generate synthetic CSI data. In another approach [131],

analysis from online videos of human activities are utilized to teach human activity sensing capabilities without any field measurements.

2.4 Alternative Wireless Sensing Modalities

This dissertation employs WiFi CSI and mmWave radar as its two primary sensing modalities, described in Sections 2.1 and 2.2, respectively. Beyond these two, the broader wireless sensing research landscape includes several other RF and optical modalities, each offering distinct trade-offs in range, resolution, cost, and infrastructure requirements. This section summarizes the key alternatives most relevant to the health monitoring and activity recognition applications considered in this dissertation.

2.4.1 Ultra-Wideband (UWB) Sensing

Ultra-Wideband (UWB) radio transmits extremely short pulses spread over a wide frequency band (typically 3.1–10.6 GHz), resulting in very high timing resolution and centimeter-level ranging accuracy. This precision makes UWB well suited for indoor localization and device-free sensing tasks that require spatial granularity beyond what narrowband signals like WiFi can provide [132, 133]. In the context of device-free sensing, Bocus et al. [134] demonstrated that UWB can detect human presence and localize subjects in multipath-rich indoor environments with high accuracy. The fine-grained impulse response of UWB signals makes them particularly sensitive to subtle movements, including breathing and micro-gestures, without requiring the subject to carry any device. A more recent direction leverages UWB signals for Activities of Daily Living (ADL) recognition at a continuous, semantic level. ADL-CLIP [135] introduced a text-aligned RF representation for continuous ADL detection in home environments, aligning UWB signal embeddings with natural language activity descriptions through a contrastive pre-training strategy. This zero-

shot and few-shot capability for recognizing unseen activity categories from only their textual descriptions represents a promising frontier for scalable, annotation-efficient home monitoring. The key limitation of UWB compared to WiFi is that it requires dedicated UWB transceivers (which are not part of standard home infrastructure), adding hardware cost and deployment complexity.

2.4.2 LoRa-Based Sensing

Long Range (LoRa) is a low-power wide-area network (LPWAN) technology that uses chirp spread spectrum modulation to achieve communication ranges of several kilometers at low data rates. Its signal propagation characteristics also make it applicable to through-wall and long-range human sensing. Youssef et al. [136] explored LoRa for long-range sensing, demonstrating the ability to detect human respiration and walking with high accuracy at distances of up to 25 meters through walls. Subsequent work by Xie et al. [137] extended this capability, achieving human walking detection at 120 meters and respiration sensing at 75 meters through multiple concrete walls by combining LoRa with advanced signal processing. Nie et al. [138] further applied deep learning to LoRa RF signals to achieve high accuracy in classifying multiple daily activities, identifying individuals, and estimating room occupancy. The long-range and low-power characteristics of LoRa devices make them especially suitable for through-wall sensing in large-scale or outdoor scenarios, potentially complementing shorter-range WiFi and mmWave systems in multi-scale sensing deployments.

2.4.3 Visible Light Communication (VLC) Sensing

Visible Light Communication (VLC) exploits modulated light from LEDs as both a communication medium and a sensing signal. Since optical signals travel at the speed of light and do not interfere with radio-frequency signals, VLC can coexist with

existing wireless infrastructure. Li et al. [139] demonstrated the use of VLC for 3D skeletal pose reconstruction, providing fine-grained human body segment localization from ambient LED lighting. However, VLC requires line-of-sight between the LED and the photodetector, and its range is constrained to the illuminated area, making it unsuitable for large indoor spaces with physical obstacles or for monitoring activities that occur outside the light cone. These constraints limit its applicability compared to RF-based sensing modalities in practical health monitoring deployments.

2.4.4 RFID-Based Sensing

Radio Frequency Identification (RFID) uses radio waves to read the state of passive or active tags attached to objects or environments. Although RFID is primarily an identification technology, its sensitivity to the electromagnetic environment has enabled a range of passive sensing applications. Xie et al. [140] demonstrated sub-millimeter resolution vibration measurements using commodity RFID readers. Hou et al. [141] used RFID phase readings for non-contact breathing monitoring. Feng et al. [142] and Wang et al. [143] showed that RFID signals can support person identification and indoor localization, respectively. The primary limitation of RFID sensing is its short range [144]: passive RFID tags typically operate within one to two meters of the reader, making the technology well suited for close-range item-level or point-of-care sensing but impractical for room-scale activity monitoring.

2.4.5 Other Wireless Sensing Technologies

Beyond the modalities discussed above, several additional technologies have been explored for wireless sensing. Bluetooth [145] has been applied to passive human sensing [146], but its nominal 10 m range can cause data loss when monitored devices move out of the Bluetooth coverage area, and its reliance on RSSI is susceptible to

abrupt fluctuations caused by frequency hopping. Wireless sensing is increasingly being integrated with communications, particularly with the advent of 6G [147, 148], where the unified sensing-communication framework promises higher data rates, finer spatial resolution, and native support for device-free sensing as a first-class network function. Metasurfaces [149] are being developed as reconfigurable intelligent surfaces that can focus and steer radio signals to improve positioning accuracy in indoor sensing applications. A brain-inspired wireless system [150] using salt-sized, implantable wireless sensors represents a novel approach to continuous neural and physiological monitoring with wireless transmission. Overall, each of these alternative wireless sensing modalities offers unique capabilities in terms of range, resolution, infrastructure requirements, and deployment cost. WiFi and mmWave radar, as used in this dissertation, strike a practical balance between sensing fidelity, edge deployability, and hardware accessibility for real-world health monitoring applications.

2.5 Real-time Sensing with Edge Learning

Real-time systems utilizing edge learning with resource-constrained devices are essential for various smart living applications [151, 152]. Sensing technologies are central, with wearable and environmental sensors commonly used. However, acceptability hinges on perceived intrusiveness, making RF sensing via WiFi and radar a promising, privacy-preserving alternative [153]. Multimodality enhances HAR by integrating various sensing modalities, addressing challenges in modeling spatial-temporal dependencies through advancements like CNNs, LSTM networks, and transformer networks. Fusion approaches and comprehensive datasets are also vital [154, 121, 96]. Real-time processing is crucial for responsive systems in smart homes, healthcare, and security, requiring lightweight computational frameworks and optimization techniques. To this end, WiFine [155] presents a lightweight system that achieves highly accurate, real-

time WiFi gesture recognition directly on resource-constrained edge hardware, as a Raspberry Pi 4B, within 0.19 seconds by fusing enhanced raw signal amplitudes and phase differences into a compact mini neural network. Wecar [156] proposes an architecture that offloads parameter-efficient continual learning to edge devices, e.g., Jetson Nano, and uses a dual-phase knowledge distillation scheme to deploy heavily compressed, real-time activity models onto memory-constrained WiFi receivers like ESP32 microcontrollers. Studies explore stereo-depth cameras [157], optimized SVMs, skeleton extraction, and integrated offline or online learning to improve performance [153].

Edge learning addresses the limitations of resource-constrained devices by employing Federated Learning (FL) [158], Knowledge Distillation (KD) [159], and Transfer Learning (TL) [160] to optimize ML models [161]. The shift towards Edge Computing (EC) is driven by privacy concerns and bandwidth optimization, with federated learning enabling collaborative training without centralizing sensitive data. Techniques like TinyTL [162] and incremental learning [163], along with quantization [164] and model pruning, further reduce memory and computational demands, allowing for on-device learning and re-training of DNNs. The integration of these domains promises to advance smart living applications [165], providing enhanced support for healthcare, safety, and overall quality of life.

CHAPTER 3

WI-PT-HAND: LOW-COST PHYSICAL REHABILITATION TRACKING FOR HAND MOVEMENTS USING WIFI SENSING

3.1 Introduction

Rehabilitation is the process of recovering a patient's health condition to its normal state after a period of illness. This is a critical process for patients affected by central nervous system disorders such as Parkinson's disease (PD) and cerebrovascular diseases (e.g., stroke). For example, stroke affects nearly 800,000 individuals each year in the U.S. and for approximately 600,000 of them, this is their first event [166]. Many survivors experience persistent difficulty with daily tasks as a direct consequence [167]. These include being less active, being less stable, moving slower as well as adopting inefficient movement patterns resulting in increased physical demands. Thus, more than two-thirds of stroke survivors receive rehabilitation services after hospitalization. According to recent studies, patients can recover up to 91% functional ability if they start the rehabilitation within three months of the stroke [168]. Similarly, rehabilitation can also minimize secondary complications [169]. Thus, maintaining physical activity and performing rehabilitation treatment as early as possible is crucial for the recovery of a patient.

In current practice, rehabilitation treatment for a patient is usually performed under the supervision of a physical therapist who provides guidance to the patient when performing specific exercises and makes sure that each exercise is performed properly for a successful treatment. While such a practice with one-on-one attention from a physical therapist will ensure a quality treatment for the patients, it limits

the application of rehabilitation treatment to a certain environment and comes with several costs (e.g., treatment cost, trip cost to the facility, dedication of specific time). Thus, alternative solutions which allow patients to perform such rehabilitation activities at home or outside of a dedicated rehabilitation facility can avoid such costs and provide a ubiquitous, and easily accessible solution. On the other hand, these alternative solutions should not reduce the quality of the treatment and must continue to give feedback to patients to ensure that rehabilitation exercises are performed properly and as prescribed. Note that clinical visits and expert based sessions can still be performed as needed but the count of such visits will get much smaller.

The existing at-home systems mainly depend on wearable sensors or cameras located in the patients' houses [170, 171]. However, such solutions come with several issues. For example, wearable solutions come with several burdens such as wearing the device, setting it up, and recharging the devices. Moreover, some patients may not be comfortable with wearing them. Similarly, camera-based solutions can trigger privacy concerns as they can record more than what is needed and they may require certain lighting conditions. Alternative to these existing systems, in this chapter, we propose an end-to-end at-home WiFi sensing based device-free physical rehabilitation tracking (*Wi-PT-Hand*) system focused on recognizing and measuring hand-based exercises. There are a few other solutions that also rely on the analysis of radio-frequency (RF) signals; however, these systems require placements of more costly equipment (e.g., mmWave radar[52]). On the contrary, our goal is to leverage the ubiquitously available WiFi signals and edge devices in most indoor environments and houses for rehabilitation activity tracking, thus providing a low-cost and non-intrusive solution. Moreover, to provide an end-to-end solution, we target a system that can automatically (i) segment the received signal between periods of activity and static periods (i.e., no movement), (ii) classify the specific rehabilitation movement

type (e.g., wrist, finger) for each segmented activity, and (iii) count the number of repetitions of the same activity performed within the current activity segment.

The main contributions of this work can be summarized as follows:

- We design and develop a low-cost, and non-intrusive end-to-end system that can be deployed on edge devices at homes for tracking hand movements.

- We consider segmentation, classification, and counting components together (for the first time for small-scale hand and finger movements) and develop solutions based on Bayesian optimizers and a hierarchical Dense Neural Network (DNN) model that are appropriate for edge devices.

- Through experimental evaluation, we analyze the performance of the proposed complete system as well as its individual components in various scenarios (e.g., different users/environments, hand conditions) and show its practicality and robustness.

- Building upon the above system, we further develop a clinically evaluated enhancement targeting spatially dynamic, multi-direction hand rehabilitation exercises. The enhanced system employs a Faraday-enclosed multi-link hardware design, a speed-invariant signal processing pipeline combining Dynamic Time Warping (DTW) alignment with drop-frame data augmentation, and a validation-weighted ensemble of per-link one-dimensional convolutional neural network (CNN) classifiers supporting real-time inference on a Raspberry Pi 4B edge device. The system is validated through experiments with 40 healthy volunteers and 11 clinical patients diagnosed with Parkinson's disease or stroke, achieving classification accuracies of 94.2%, 91.4%, and 86.3% on internal validation, unseen volunteer test, and clinical cohort datasets, respectively.

The rest of this chapter is organized as follows. In Section 3.2, we discuss the related work on both sensor-based rehabilitation tracking and WiFi sensing. In Section 3.3, we present the details of the proposed Wi-PT-Hand system. We evaluate the proposed system through experiments in Section 3.4. In Section 3.5, we describe the clinically evaluated enhancement of the system for spatially dynamic hand movements, including the updated hardware, the speed-invariant activity classification pipeline, real-time inference, and experimental validation with clinical patients.

3.2 Related Work

Our study spans multiple research fields such as rehabilitation tracking, WiFi sensing, and human activity recognition (including segmentation, classification and counting of activities) especially for small-scale activities like wrist and finger movements. Therefore, in addition to the already discussed literature in Chapter 2, our analysis of related work is divided into the following two subsections.

3.2.1 Sensor-Based Rehabilitation Tracking

Thanks to the ubiquity of Internet of Things (IoT) devices, there have been many smart health and assistive solutions developed to track the activities of patients during the rehabilitation process. These solutions can be categorized into two major approaches. In the first approach, wearable sensors attached to the body of the patient are considered [172, 173, 174, 17]. These sensors usually contain an Inertial Measurement Unity (IMU) that can measure accelerometer, gyroscope, and magnetometer readings and can obtain different directional and angular movements of the limbs of the patients. However, patients may feel uncomfortable with wearing these sensors and may not deal with charging them as needed.

In the second approach, the studies use camera-based solutions (e.g., Kinect [50],

RGB camera [18], depth camera [51]) and through the analysis of collected frames, detailed patient movements and poses can be detected. While such systems are non-intrusive as they do not require a device worn by the patient, they have certain limitations and drawbacks. For example, they can only track a person when they are in sight of the camera. These systems can also pose a privacy risk to users as they detect not only people's movements but also their faces and the environment they live in. Finally, they can be costly to deploy, especially for large areas where multiple cameras need to be deployed to obtain sufficient coverage.

In addition to these major approaches, there is also a relatively new but growing number of studies that rely on RF signals and radar imaging. These studies [175, 52, 176] rely on specific signals (e.g., FMCW [175], mmWave [52, 176]) and deep learning based analysis of signal features. These solutions can be more effective than previous solutions as even the activities that are performed out of sight can be detected because RF signals can penetrate through some obstacles and reflect from some other objects in the environment. On the other hand, these solutions require special equipment (e.g., mmWave radar), which can be costly. Different from these systems, our solution relies on WiFi signals which can be found in most indoor areas or can be produced through low-cost devices easily. Additionally, the proposed system can use low-cost microcontrollers to capture the signals, preprocess them, and make predictions directly on-board.

3.2.2 Segmentation and Counting Works on WiFi Sensing

Activity recognition using WiFi sensing has recently been studied a lot with a much focus on detection of different set of activities (e.g., gestures [35], falls [41]) and environmental changes (e.g., soil moisture [44], occupancy counting [40]). We refer the interested readers to the surveys on WiFi sensing (e.g., [177]) and in this part, we

specifically cover the WiFi sensing studies that not only aim to recognize the activity but also consider segmentation and counting of activities towards building a complete solution.

We start with summarizing these studies in Table 1 together with their issues and comparison to the proposed work. A device-free fitness assistance system is proposed in [178] for equipment-based exercises. While this study discusses segmentation, and counting components as well, they are not well defined. Auto-correlation based similarity analysis is considered for detecting workout and non-workout times as well as for counting the repetitions in general but no procedure is defined to find out the exact start and end times of activities. Moreover, no proper evaluation is performed to show the usability of their work. Above all, their system is shown to work only with very large-scale activities like equipment-based exercises. In contrast, we focus on very small-scale hand movements, such as wrist and finger movements, which are more challenging to identify.

The DeepSeg [37] system uses a CNN model to identify an activity session's start, motion, end, and static states and thereby segments the CSI series obtained from WiFi sensing over five fine-grained activities like several hand gestures and five coarse-grained activities like running. They divide the entire CSI series into continuous bins of CSI frames and predict which bin is what out of the four states. In this way, the start and end points of a very fine-grained action, like a finger movement, would not be specific and can offset these points by the size of the bin in the worst case. Moreover, the start and end states can be too short to be missed for small-scale activities like finger movement, which reduces the portability of the model for very fine-grained activities like hand physiotherapy actions.

Table 1.: Comparison of Existing Physical Therapy Monitoring and Tracking Systems

References	Year	Techniques Used	Segm.	Class.	Count.	Issues
Fitness Asst.[178]	2018	Auto-correlation, DNN, Spectrogram	◑	●	◑	Incomplete seg. and counting
DeepSeg [37]	2020	CNN to classify states and actions	●	●	○	No counting
RT-Fall [43], Anti-Fall [42]	2016	Band-pass filter, sliding standard deviation of CSI phase-difference	◑	●	○	Only for fall detection
Temporal Unet [179]	2019	Convolution using Unet on CSI amplitude standard deviation	◑	●	○	Incomplete segmentation and no counting
Tang et al. [36]	2021	Butterworth filter, LSTM-FCN	○	●	○	Only for hand gestures
Chen et al. [180]	2021	Butterworth filter, PCA, adaptive sliding window, CNN	●	●	○	Focuses on whole-body activities
Transtrack [38]	2022	Hampel, PCA, recursive Otsu, meta-transfer SVM	●	●	○	Whole-body segmentation and classification
Wi-Gym [39]	2023	Hampel filter, PCA, FFT, Gabor, SVM, DTW, Fuzzification	●	●	○	Whole-body actions without counting
U-Shape Nets [53]	2024	DTW, FCN, UNET, UNET++	◑	●	○	Whole-body datasets only
Wi-PT-Hand	2024	PCA, Local Averaging, Bayesian Optimizer, hierarchical DNN	●	●	●	Not speed-invariant and so not ready for clinical usage
Clinical Enhancement of Wi-PT-Hand	2026	Hampel, PCA, speed invariance, 1D-CNN, multi-link WiFi, ensemble, weighted counting, real-time, edge	●	●	●	N/A

● Fully addressed, ◑ Partially Addressed, ○ Not addressed.

The RT-Fall [43] (and its predecessor, Anti-Fall [42]), also uses stateful transition between activities to segment and detect a fall of various natures. It uses a band-pass filter to denoise the CSI data. Then, the standard deviation of the phase difference is used to identify a person's specific action which ends with a fall. However, this system is highly dependent on the environment since the phase difference and amplitude of the CSI data vary broadly between environments. In addition, that study focuses on only fall activities.

The Temporal Unet [179] proposes a customized mapping function from CSI series to action label using the convolution of CSI series windows to detect several movements in various environments. The purpose is not entirely segmentation, but their approach produces segments of actions convincingly, which is not fully explored and a proper segmentation methodology is not described. In addition, no repetition counting approach is provided in this study. The action scale is again based on whole-body movements. The study in [36] shows promising classification results with fifty different types of gestures. However, it does not study how to detect the start and end time of gestures, i.e., segmenting, nor how many times the gesture is performed. In [180] a real-time activity recognition system is studied using dynamic-sized sliding windows. Their approach looks mostly whole-body movements such as sitting, and walking and only one hand-based small scale movement (i.e., waving) is included. Moreover, no solution for counting is proposed.

Similarly, in more recent works [38, 53, 39], some solutions for segmentation and classification of whole-body movements are studied. For example, the study [38] applies Hampel and band-pass filters to denoise principal components and then takes recursive Otsu thresholding to separate gesture-induced signals from the background noise inspired by image segmentation. In [53], authors compare action recognition, the start and end point detection of action, and action segmentation using three U-

shape networks, i.e., FCN, UNet, and UNet++, on several publicly available data sets of whole-body actions. Finally, in Wi-Gym [39] a gymnastics exercise assessment system is developed by comparing how seven gymnastic actions are performed by a person and a trainee through a segmentation and classification procedure. All these studies, however, do not consider small-scale movements and no solution for counting the activities performed is proposed. Overall, to the best of our knowledge, there is no study that considers small-scale hand and finger movements and provides an end-to-end solution with segmentation, classification, and counting components together.

3.3 Proposed Wi-PT-Hand System

In this section, we introduce our end-to-end wireless sensing based solution for tracking physical therapy hand movements. The proposed system aims to detect (i) start and end of different movement/activity segments, (ii) the type of the movement performed in each segment, and finally (iii) the number of repetitions of that activity performed within that segment. In Fig. 2, we provide an overview of the proposed system and illustrate the steps required for reaching out these goals. Next, we elaborate on each of these steps in the following subsections. The notations used throughout the chapter together with their descriptions are given in Table 2.

3.3.1 CSI Data Collection

WiFi sensing uses ambient WiFi radio signals to detect and sense the physical properties of the environment. These RF signals propagate over multiple unique physical paths from a transmitter to a receiver. These multi-paths then cause variations in the RF signal as these signals reflect off of surfaces and propagate through surrounding objects such as furniture, walls and people within the environment, as discussed in Section 2.1.3. To collect CSI data, we use the WiFi-enabled ESP32

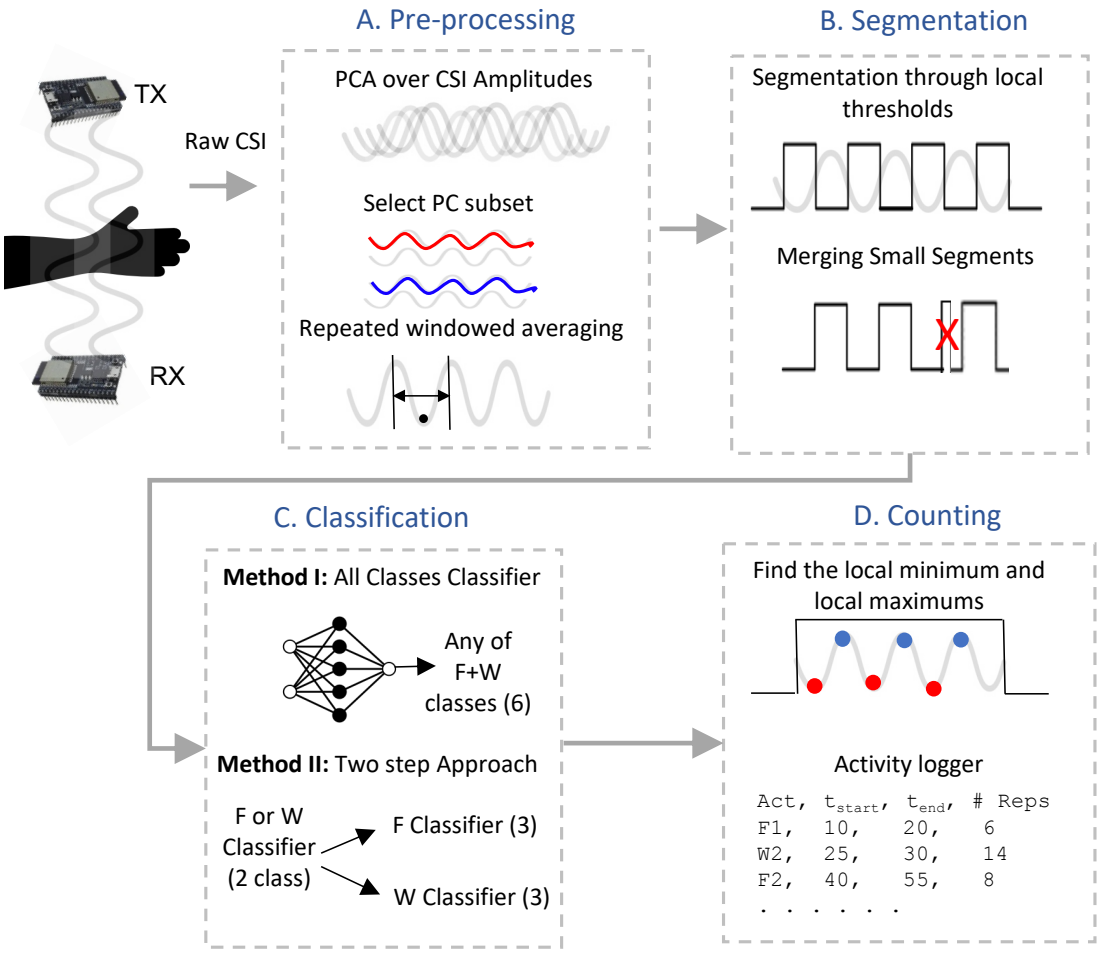


Fig. 2.: Wi-PT-Hand System Overview.

microcontrollers and our recently developed open-source ESP32-CSI Toolkit [181]. These ESP32 microcontrollers provide a small-size, low-cost, and standalone solution compared to other existing methods [21, 182, 183] (which require a host laptop with an updated Network Interface Card (NIC)); thus, they can be easily deployed anywhere. We transmit frames from an ESP32 transmitter and then collect WiFi CSI from a separate ESP32 receiver at a fixed rate of 100Hz. We then extract the amplitudes of the CSI data to be used in the next steps of the proposed system.

3.3.2 Data Preprocessing

Once the CSI data is collected, we then preprocess it through the steps given in Algorithm 1. That is, we first run Principal Component Analysis (PCA) over the amplitudes of the CSI data (line 1). Then, from the set of principal components (PCs) computed by PCA, we select a smaller subset of ρ components starting from the ρ_s^{th} component (line 2). These components are considered to be the most significant representative consecutive subset for the data and will be determined by the optimizer. Later on, we take the average of the selected PCs to further reduce the dimension of the data and make the impact much more visible (line 2). Finally, we slide a window over the entire time series data and perform a moving average over this data for η times recursively using a window size of ω. As discussed in Section 3.3.6, the values of η and ω are optimally chosen along with a few other parameters using a hyperparameter optimization approach in our system.

Table 2.: Notations and their descriptions

Notation	Description
H	CSI series
ρ_s	First selected PC from the beginning
ρ	Number of total selected PCs
η	Number of repetitive averages
ω	Window size
τ	Threshold used for the separation of activity and non-activity segments
σ	True labels per CSI frame
P_{pca}	PCA output for CSI amplitudes
P	Average over all selected PCs of P_{pca}
P_ω	Preprocessed CSI data over a window size ω
ψ	Window multiplier
γ	Smoothing factor
$\mathcal{A}$	Action label for a CSI frame or segment
$\mathcal{A}'$	Non-action label for a CSI frame or segment
χ	Array of activity and non-activity labels per CSI frame
κ	Set of counts per activity segment
$\lvert X \rvert$	Size or length of any array X
$\overline{X[a:b]}$	Mean of values within range $[a, b]$ in any array X

Fig. 3 shows how Algorithm 1 preprocesses the series of principal components of the CSI time series data step by step. The output of PCA over the amplitudes of the CSI data consists of 64 principal components as shown in Fig. 3a. For this specific dataset, our hyperparameter optimization method chooses the third, fourth, and fifth principal components (PCs) as the best subset from all PCs, as depicted in Fig. 3b. By taking the average of amplitudes in these selected PCs, the impact gets more visible while having some noise, as shown in Fig. 3c. Furthermore, with repetitive and recursive averaging for η times over sliding windows of size ω, the noise is filtered as plotted in Fig. 3d. After this preprocessing part, we prepare to segment and label this result for the activity and non-activity portions.

Algorithm 1: Preprocessing of data (H, ρ_s, ρ, η, ω)

$P_{pca} \leftarrow PCA(H)$

$P \leftarrow \overline{P_{pca}[\rho_s : \rho_s + \rho - 1]}$ `// Select a subset of PCs and take the average`

for $n \leftarrow 1$ **to** η **do**

  for $i \leftarrow 1$ **to** $|H|$ **by** 1 **do**

    `// Select a sliding window`

    $a \leftarrow \max(1, i - \omega/2)$

    $b \leftarrow \min(|H|, i + \omega/2)$

    $P_\omega[i] \leftarrow \overline{P[a : b]}$ `// Moving average`

  $P \leftarrow P_\omega$

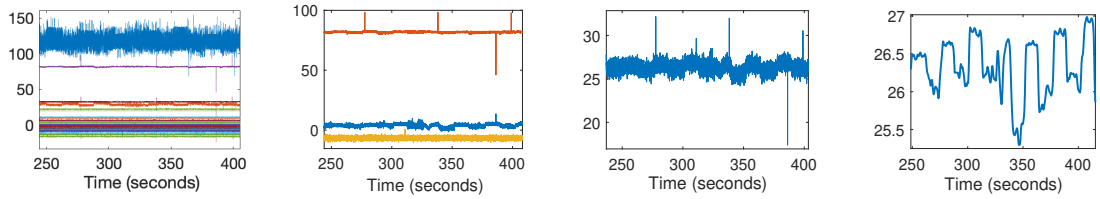
return P_ω

3.3.3 Activity Segmentation

The trending nature of fluctuation of P_ω obtained after the preprocessing steps can be seen very much related to the actions performed. Thus, we have designed our segmentation method based on this observation. To this end, we consider two different approaches. In the first approach, we pick the overall-average (OA), τ, of the series P_ω, as a threshold to separate the activity and non-activity segments using the following equation:

$$\forall i \in [1, |P_\omega|], \chi_i = \begin{cases} \mathcal{A}, & \text{if } P_\omega[i] < \tau \\ \mathcal{A}', & \text{otherwise} \end{cases}, \text{where } \tau = \overline{P_\omega} \tag{3.1}$$

Here, $\chi_i = \mathcal{A}$ and $\chi_i = \mathcal{A}'$ represent an activity and non-activity CSI frame, respectively. The average value, $\tau = \overline{P_\omega}$ is considered as the threshold over the series



(a) All principal components (P_{pca}) (b) Selected PCs for $\rho = 3$ (c) Selected PCs's avg. (P) (d) Repetitive avg.(P_ω)

Fig. 3.: Step by step impacts of Algorithm 1.

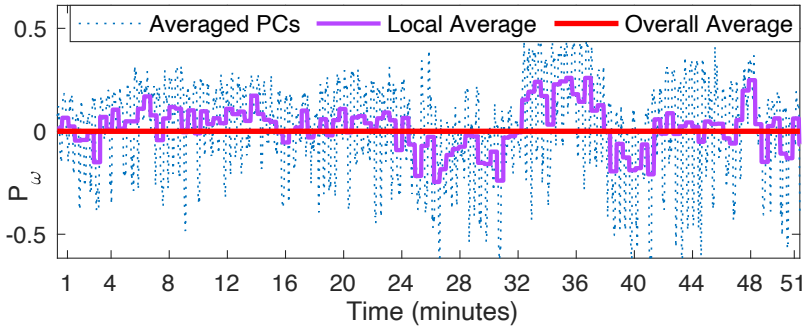
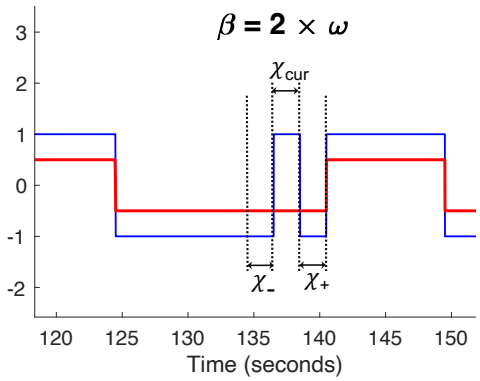
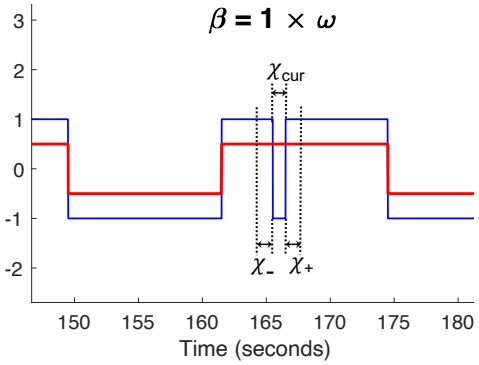


Fig. 4.: Sinusoidal trending nature of the principal components of the CSI data, motivating choice of Local Average as the threshold, rather than Overall Average as the threshold.

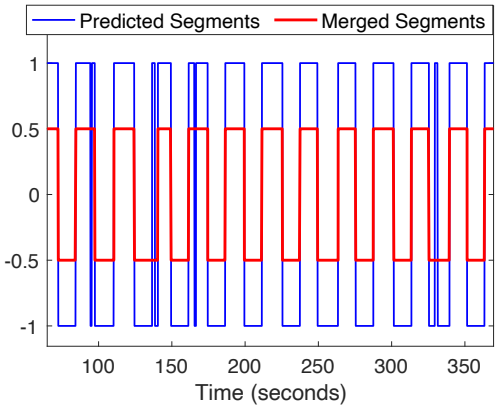
P_ω, which is basically a flat line along the mean of the data series. It identifies the portion of the data above this threshold as non-activity and the portion of the data below the threshold as activity, which aligns with our observation. While this overall average value can provide some understanding towards the separation of activity and non-activity periods, due to many factors (e.g., environment, hardware, etc.), there usually occurs local variations in the data; thus OA method often fails to identify such local trends of the time series data with such a global average value. This can be observed in Fig. 4, which clearly shows that the globally calculated overall average cannot be used for identifying all activity segments while a local average might be



(a) Merging divided non-activity segment.



(b) Merging divided activity segment.



(c) Fixing Multiple Isolated Segments.

Fig. 5.: Examples depicting Algorithm 3 in action.

more successful to differentiate activity and non-activity periods. To this end, as a second approach, we propose to use a threshold obtained from a local average (LA) over a larger window of size $\psi \times \omega$, and we aim to recognize local changes in the series by comparing this local average in a small and a large window. The details of this approach is given in Algorithm 2. We first calculate the thresholds in large non-overlapping windows (lines 2-4). Then, if the local average obtained in the smaller window (that the current data point is in) is less than the local average in the larger window (line 7), we recognize it as part of an activity segment; otherwise, it is considered within a non-activity segment.

Algorithm 2: Segmentation of Activities (P_ω, ω, ψ)

$\omega_m \leftarrow \psi \times \omega$ `// larger window`

`// Find local thresholds by averaging in non-overlapping large windows`

for $i \leftarrow 1$ **to** $|P_\omega|$ **by** ω_m **do**

  $s \leftarrow max(1, i - \omega_m/2), \quad e \leftarrow min(|P_\omega|, i + \omega_m/2)$

  $\tau_s[s:e] \leftarrow \overline{P_\omega[s:e]}$

`// Assign activity labels by comparing smaller window and larger window averages`

for $i \leftarrow 1$ **to** $|P_\omega|$ **by** ω **do**

  $s \leftarrow max(1, i - \omega/2), \quad e \leftarrow min(|P_\omega|, i + \omega/2)$

  if $\overline{P_\omega[s:e]} < \tau_s(\frac{s+e}{2})$ **then**

    $\chi[s:e] \leftarrow \mathcal{A}$ `// Activity`

  else

    $\chi[s:e] \leftarrow \mathcal{A}'$ `// No activity`

return χ `// Activity/non-activity labels`

The second approach is more capable of recognizing the activity segments throughout a long data series compared to the first overall average based approach. However, both of these methods can produce isolated segments with small durations, usually on the edge of the actual segments. This can be within an activity segment (as shown in Fig. 5a) or within a non-activity segment (as shown in Fig. 5b). To address this problem, we develop a merging process for such segments as described in Algorithm 3. That is, starting from the smallest possible segment size ω to the maximum possible segment size $\omega \times \gamma$, we check all possible segments of size β over the entire time

Algorithm 3: Merging Isolated Segments (χ, ω, γ)

```
1  for β ← ω to γ × ω by ω do
2      for s ← (β + 1) to (|χ| − 2β + 1) by 1 do
3          χ_cur ← χ̄[s : s + β − 1]
4          χ_− ← χ̄[s − β : s − 1]   // preceding segment
5          χ_+ ← χ̄[s + β : s + 2β − 1]   // succeeding segment
6
               // If the current segment does not match with neighbors, it
            is converted.
7          if ((χ_cur = A || χ_cur = A′) & (χ_− = A|| χ_− = A′) &
8          (χ_+ = A|| χ_+ = A′) & (χ_cur ≠ χ_−& χ_cur ≠ χ_+)) then
9              χ[s : e] ← χ_+
10 return χ
```

frame. If each of the current, preceding and succeeding windows of the same size can be uniformly considered as either an activity ($\mathcal{A}$) or non-activity ($\mathcal{A}'$) segment, then the current segment assignment is considered to be valid. But if the current segment does not match with the preceding and succeeding segments (line 6), which also requires them to be the same segment type (i.e., $\mathcal{A}$ or $\mathcal{A}'$), the current segment is converted to its neighbors' type (line 7). Note that the rationale behind starting with smaller possible segment sizes (line 1) is this bottom-up approach allows merging smaller falsely identified segments first, leading it to generate larger mergeable segments.

Fig. 5 illustrates examples of this merging process. In Fig. 5a, a misidentified activity segment is removed as it is surrounded by non-activity segments of same size, while in Fig. 5b a misidentified non-activity segment (which divides the activity segment) is converted to an activity segment as it is surrounded by activity segments.

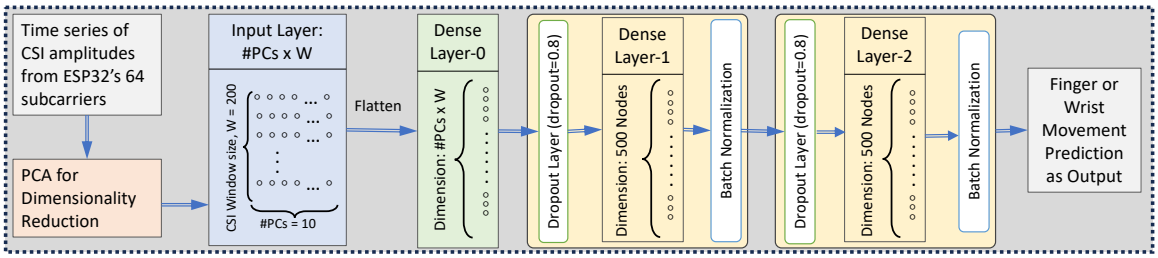


Fig. 6.: Activity Group Binary (Finger or Wrist) Classifier DNN Model Specifications

out of the used three DNN models.

Finally, in Fig. 5c, the result of running the merging algorithm on multiple segments over a time frame can be seen. All the isolated and falsely identified segments are merged or removed properly. Note that the parameters used both in Algorithm 2 and Algorithm 3 are optimized through an optimizer as we will discuss later. The segmentation accuracy is defined as the number of activity or non-activity CSI frames that are correctly identified and the optimizer uses the corresponding F1-loss for this.

3.3.4 Activity Classification

After the segmentation of the CSI time series into activity and non-activity segments is completed, next, we start the activity classification process. That is, for each activity segment, we need to recognize which specific activity is performed. Here, considering the two different subcategories of the hand-based movements, namely, wrist-based and finger-based movements, we propose a hierarchical classification model. That is, we first aim to recognize the subgroup of the action performed and then we aim to recognize the actual activity type within that subgroup. To this end, for the first step, we develop a binary classification model, and for the second step, we develop two different intra-group activity recognition models. The goal of this hierarchical approach is to develop a more scalable solution so that new categories of activities can easily be added later without having to retrain the entire model again

each time.

For each of these models, we design a machine learning classifier model $\mathcal{M}$ using a Dense Neural Network (DNN) architecture with four dense layers and we set the number of output neurons equal to the number of classes to be recognized (e.g., two classes for the binary model that aims to recognize the subcategory). We apply a dropout layer between each dense layer to prevent overfitting. Finally, we use the Adam optimizer to optimize the categorical cross entropy loss function,

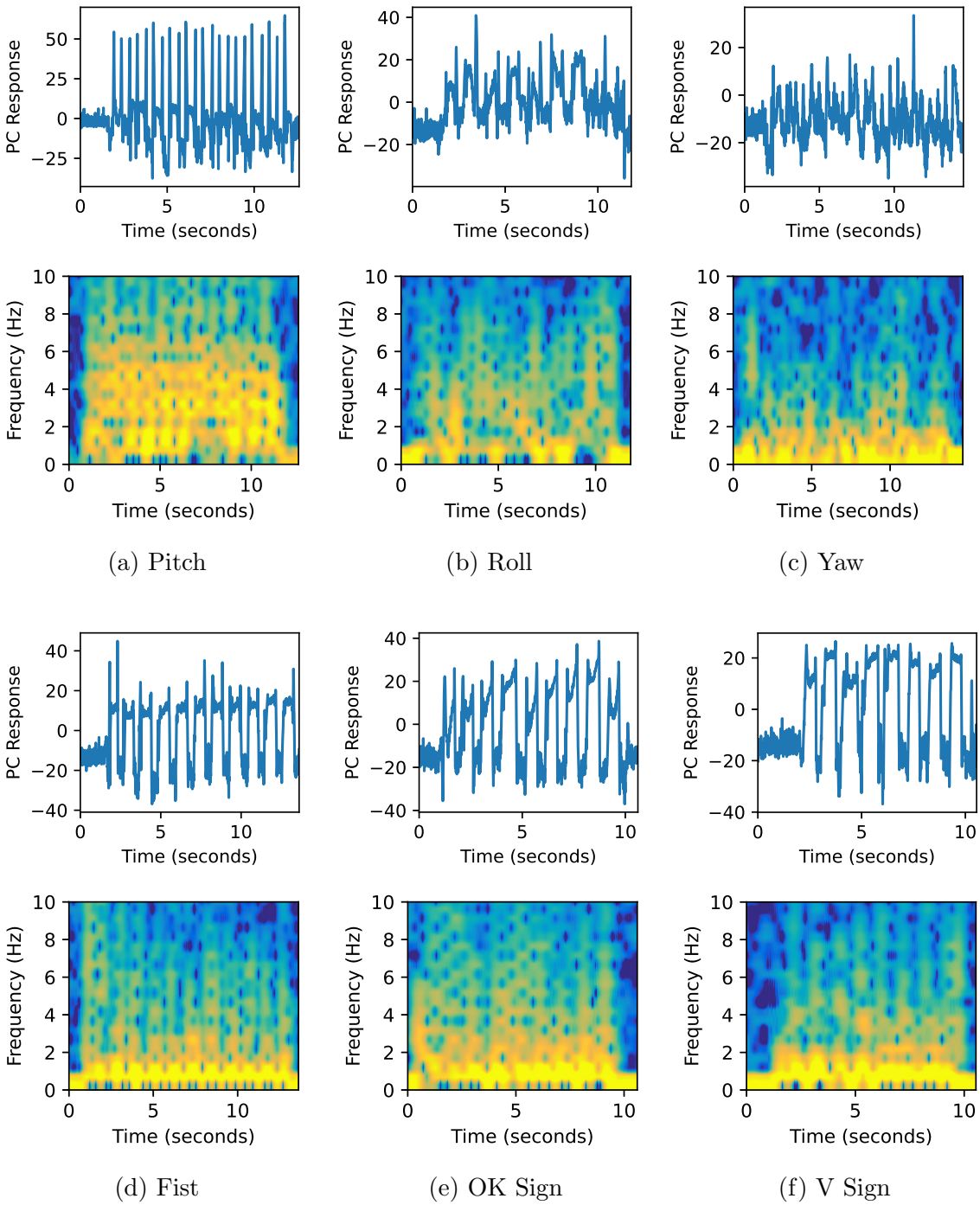
$$\mathcal{L}(x, y) = \frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_i[c] \log \left(\mathcal{M}(x_i)[c] \right)$$

where $\mathcal{M}(x_i)$ is the model prediction for input CSI x_i and y_i is the one-hot encoded true class for the i-th CSI measurement. Training is performed on batches of size 1024 leveraging only the first 10 principal components which are precomputed using PCA. An optimized model for the initial binary classification model is depicted in Fig. 6 along with the evaluated hyperparameters. The other two models for specific and intra-group wrist and finger action classification are similar to this model, except for different values of the hyperparameters. Note that more complicated models can be considered for classification tasks in this case, however, our motivation is to keep the models as lightweight as possible to fit them into edge devices.

3.3.5 Activity Repetition Counting

Once the CSI data series is segmented into activity segments and each activity segment is classified as one of the finger or wrist activities, the next step is to count how many times the same activity is repeated within each activity segment. To this end, we use the corresponding activity segment's internal periodic behavior and similarity.

In Fig. 7, we illustrate this periodic behavior through the line plots and spec-



(a) Pitch (b) Roll (c) Yaw

(d) Fist (e) OK Sign (f) V Sign

Fig. 7.: Line plots and spectrograms for the first principal component response showing unique repetitive patterns of different hand and finger movements.

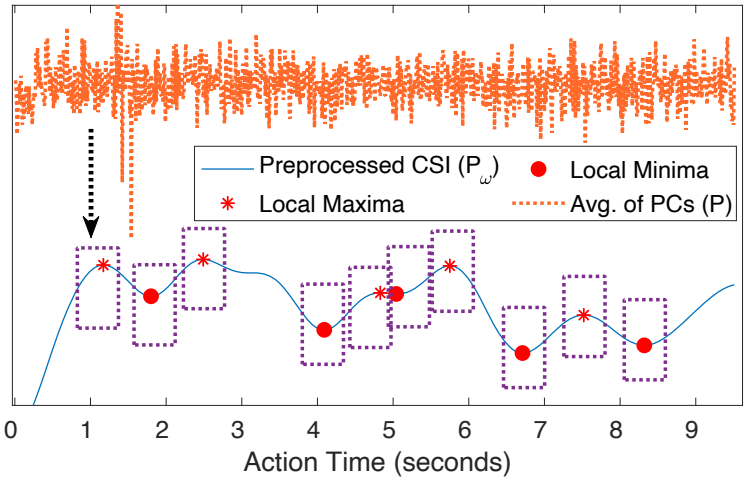


Fig. 8.: Counting 10 repetitions of an action by identifying the total local optima within a specified activity segment after an optimized number of repetitive window averaging.

trograms for the first PC's response for six different activities that we also use for our evaluations. Each subfigure clearly shows that there is a repetitive signal pattern which is also confirmed to closely match with the number of repetitions of the activity. Motivated by this relation, we then use the total number of local optima i.e., local maxima and local minima, within the segmented region to count the number of repetitions of the activity. Here, note that the raw CSI data, even after averaging over a selected number of principal components (shown as Avg. of PCs, P, in Fig. 8) can include so many local optima points looking like noise. The repetitive averaging (η times) method over an optimal window size (ω) can help obtain the signal pattern (i.e., P_ω) that matches with the actual counts. However, this averaging should be optimized for the counting specifically (i.e., preprocessing steps used for segmentation should be performed again for counting). Moreover, as it is shown in Fig. 7, the repeating patterns of different activities can be different thus the optimization model for getting the activity counts should be uniquely developed for each activity. Bring-

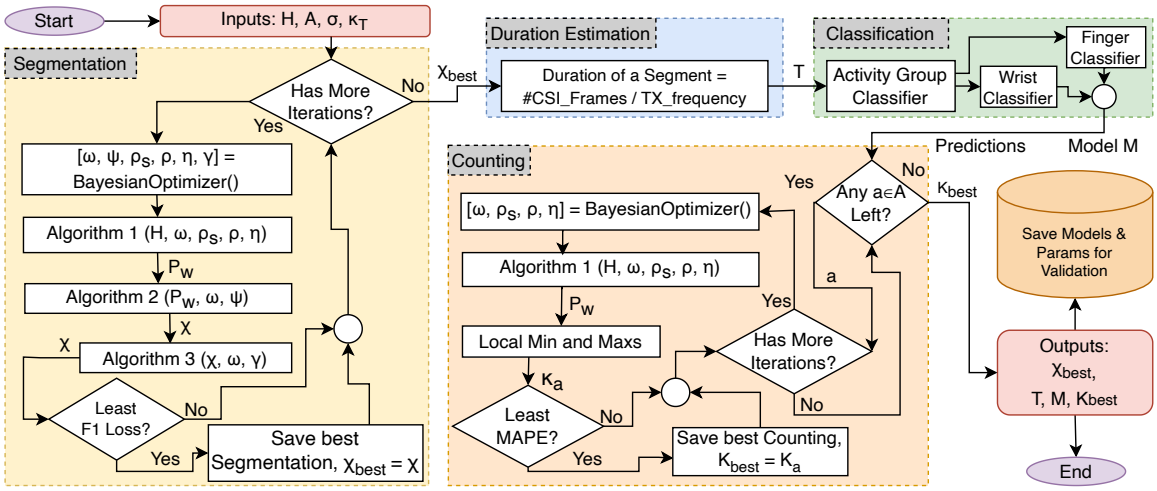


Fig. 9.: Flowchart of the full system with optimization iterations.

ing all these together, we define the counting process (i.e., finding the total number of local optima in P_ω) after an optimized preprocessing (Algorithm 1) performed for each of the experimented actions.

For counting error, which is used during optimization of the model, we use the mean absolute error (MAE), which is defined as:

$$\mathrm{E}_c = \frac{1}{n}\left(\sum_{i=1}^{n} |\mathrm{Y}_i - \hat{\mathrm{Y}}_i|\right), \tag{3.2}$$

where Y_i and $\hat{\mathrm{Y}}_i$ are the actual and predicted number of repetitions of the same activity in the input CSI segment i, respectively.

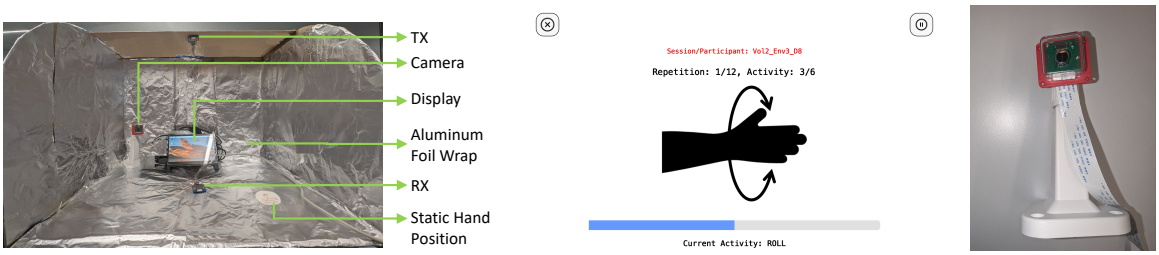
3.3.6 Hyperparameter Optimizer

Next, we elaborate on the process used to optimize the values of the hyperparameters used at different parts of the proposed system. The optimization process is illustrated through a flow chart shown in Fig. 9, where each of the segmentation, classification and counting processes is handled separately, but one's output is used in the other one as input.

To begin with, we get the CSI series (H), set of the actions ($\mathcal{A}$), true labels per CSI frame (σ), and true counts per action segment (κ). For segmentation part, we use Algorithm 1, Algorithm 2 and Algorithm 3 in sequence and predict the segmentation value, χ, which is basically activity or non-activity label for each of the CSI frame out of the provided CSI series, H. The values of parameters (e.g., window size (ω), window multiplier (ψ), group of principal components (defined by ρ_s and ρ), number of repetitive averages (η)) used in these three algorithms are optimized through a Bayesian optimizer, which uses the true segmentation labels per CSI frame, σ, and the F1-loss of each prediction made to reach the best set of parameter values through several iterations (e.g., 1000 as used in our evaluations). Note that as our system has a specific and stable transmission (TX) frequency, i.e., 100Hz, we can also calculate the duration of any activity or non-activity segment by dividing the total number of CSI frames within that specific segment by the TX frequency.

For the DNN models used for classification as described in Section 3.3.4, we leverage Tree-structured Parzen Estimator (TPE) as elaborated in [152] to optimize the hyperparameters. These parameters include CSI window size (50 to 1000 CSI frames per window) passed to the model, number of hidden neurons ([25-500]), learning rate [10^{-9}-10], whether or not to apply kernel or activation regularization and the per layer dropout percentage [0.0-0.9]. Note that for the proposed hierarchical model, we can train and perform hyperparameter optimization for each sub-model independently of one another. Thus, this allows for greater control of the training process of each sub-model and offers the ability to add or replace sub-models without affecting the prediction quality of the rest of the hierarchical system.

Finally, in the counting part, we use a Bayesian optimizer as in the segmentation section, for the set of input parameters (i.e., ω, ρ_s, ρ and η) required towards obtaining the best counting accuracy. Note that this is performed for each activity separately,



(a) (b) (c)

Fig. 10.: Data collection box with (a) the locations of all components, (b) a display to show animated instructions/predictions, (c) a camera to record the ground truths of segmentation, counting, and classification.

as the periodic behaviors of different actions can be different due to the intrinsic unique features of the activity. Each of these optimizations is run for 1000 iterations and the sets with the least MAE, as described in Eq. (3.2), are saved as κ_{best}. At the end of this optimization process, the best sets of hyperparameters as optimized by the Bayesian optimizer (for segmentation and counting) and the TPE (for DNN), are saved to provide validations with different data sets afterward.

3.4 Evaluation

In this section, we start by describing the experimental setup and the data sets collected. We then evaluate the performance of the proposed system through results regarding different individual components, namely, segmentation, activity classification, and counting. We then evaluate the system as a whole and compare it to an existing system.

3.4.1 Experimental Setup

To evaluate the proposed *Wi-PT-Hand* system, we performed our experiments using a pair of ESP32 microcontroller units. These devices are low-cost and can be

(a) (b)

Fig. 11.: Experimental design with TX/RX pairs illustrated for two different categories of activities (a) hand and wrist movements, (b) finger movements.

used to collect CSI data through the tool we developed recently [181]. One ESP32 is used in the TX role and the other in the RX role. The sampling rate is set as 100Hz. They are aligned vertically in a box (with one side open), where the user's hand performs the activities. The distance between the TX and RX devices was 35 centimeters. The hand is placed in the line of sight of the TX-RX pair. Fig. 10 shows the equipment setup. As shown in Fig. 10a, we wrapped the three surrounding walls and the bottom of the box with aluminum foil to mitigate external electromagnetic interference from the environment and to minimize the impact on the optimized algorithms and machine learning models. A Raspberry Pi supported display is placed inside the box to show instructions to the volunteer to perform specific actions during the data collection session using our annotator app [184] in collaboration with our server app [185] to save the data from ESP32 inside the same Raspberry Pi device. This display can also be used to show predictions in a practical deployment. Besides, a camera is placed to record the movements inside the box so that we can prepare the ground truths of segmentation, counting, and classification. Fig. 11 shows the movements performed inside our setup.

Table 3.: Information about the datasets

(a) Activity durations and repetitions in training ($D1$), validation ($D2$) and test (other volunteers) datasets.

Dataset	**Reps.**	**Action Dur.**	**Session Dur.**
D1	20	5 - 15 secs.	41.4 mins.
D2	4	5 - 15 secs.	8.1 mins.
Others	12	5 - 15 secs.	30 mins.

(b) Environments

No.	**Location**
1	Conference room
2	Open lab area
3	Living room
4	Bedroom
5	Office room

(c) Categorical Variations of the Volunteers

Category	**Variations**
Age group	19 - 67 years
Male	7 volunteers
Female	3 volunteers
Conditions	3 types

3.4.2 Hand Movement Data Sets

We have collected CSI measurements for two types of physical therapy procedures (i) wrist movements and (ii) finger movements. Throughout the evaluation section, we will mainly present the detailed results obtained for the first volunteer's data only. However, to show the generalizability of our system, we test the setup in five different environments as shown in Table 3((b)) and, as per Table 3((c)), with ten volunteers of age groups ranging from 19 to 67 years, of both male and female genders and with three types of hand conditions, i.e., normal condition, wearing gloves, and wearing a wrist and thumb support given by physiotherapist to real patients. Overall, we collected 20 different datasets[1].

[1]These datasets and our codebase can be found in https://github.com/MoWiNG-Lab/wipt.

3.4.2.1 Wrist Movements

The wrist movements are pitch (P), yaw (Y), and roll (R) of the palm around the wrist, as illustrated in Fig. 11a. These three movements are typically used to test the movement capability of the wrist along three axes during physical therapy, respectively. The volunteer repeated each hand movement multiple times for a specific duration measured in seconds. There were complete stops in between each successive action.

3.4.2.2 Finger Movements

We consider small-scale finger movements as the next set of actions. These finger movements are OK sign (O), Victory sign (V), and Fist (F) as illustrated in Fig. 11b. These finger movements are specifically selected to cover the exercise of almost all the joints in the fingers of a human hand.

The collected data is divided into two, to be used in training and validation of our algorithms using these hand movements. Table 3 provides an overview of these two sub data sets. In order to comply with a practical scenario, the actions were performed in random order, and the duration of each action was varied within a specific range of 5 to 15 seconds. The training dataset ($D1$) in this environment has 20 sessions of each of the six actions which includes several repetitions of that action. The validation dataset ($D2$) has a similar setup but it includes 4 sessions of each action. The training dataset has a total duration of 41 minutes and 25 seconds, and the validation dataset has an entire duration of 8 minutes and 6 seconds. The data sets collected in different environments and/or with different volunteers include 12 repetitions (each with 5-15 seconds) of each hand movement with a total duration of 30 minutes.

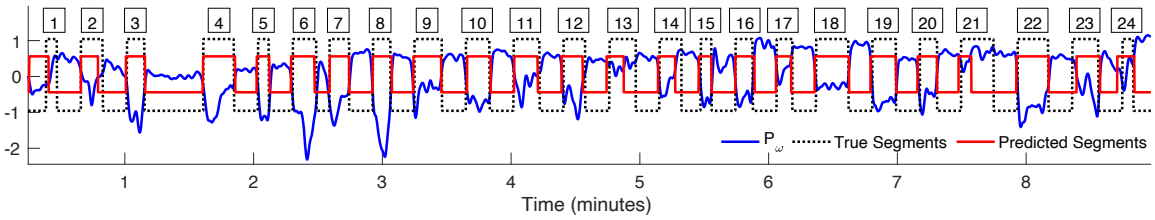


Fig. 12.: Segmentation results with Local-Average (LA) method optimized/trained with dataset D1 and validated with dataset D2 (i.e., D1/D2) yielding 90.27% per CSI frame segmentation accuracy and 100% segment detection accuracy.

3.4.3 Results

As Wi-PT-Hand system has different components (i.e., segmentation, classification, counting), as shown in Fig.2, we evaluate the performance of each component separately, as well as the entire system's performance. For the Bayesian optimizers used in segmentation and counting parts, we use the implementation available in Matlab 2022a version [186, 187, 188]. We also use Keras library [189] to implement the DNN models used in the classification part.

3.4.3.1 Activity Segmentation

Segmentation results are generated for both OA and LA methods in three different ways. In the first way, we use the dataset D1 for both training (of the optimizer) and validation; in the second way, we use the dataset D1 for training and D2 for validation; and finally, we use the dataset D2 for both training and validation. We consider these different ways to observe how the results change for our method in different scenarios and how generalizable our results are even within one volunteer's data.

Table 4 shows the segmentation results for all these different ways and different methods. Overall, LA method in general can provide better segmentation accuracy

than OA method, thanks to its varying and sliding window based approach that takes into account local changes in the CSI data. In addition to the segmentation accuracy, we also consider accuracy in terms of the percentage of the number of whole segments (including both the activity and non-activity segments) correctly identified compared to the total number of segments (shown under the "Total" column) in the validation data set. We consider the non-action segments being wrongly predicted as actions to be false-positive (FP) segments, while the action segments being wrongly predicted as non-actions are considered false-negative (FN) segments. If the number of total segments is S, the number of FP segments is S_{FP} and the number of FN segments is S_{FN}, then this accuracy is given as, $A_s = (1 - (S_{FP} + S_{FN})/S) \times 100\%$. The results in Table 4 show that LA method provides almost perfect accuracy in all scenarios and it is much better than what is obtained when OA method is used. We observe a clear difference (100% vs. 83.67%) for D1/D2 case as it is the most realistic scenario.

In Table 4, we also show the optimized values of each of the parameters. The optimizer finds the window size of 50 or 100 CSI frames as the most effective window size. The second and third principal components are found to be the most significant ones yielding the best results regarding segmentation. The number of repetitive averages is generally within the range of 3 to 5, while the merging factor performs better with lower values. It is worth mentioning that the most time-consuming part of our system is the Algorithm 3. Therefore, the lower the merging factor remains, the faster the system yields results.

Table 4.: Segmentation results through different methods and training/validation datasets

Method	Train/ Validate	ω	ψ	ρ_s	ρ	η	γ	Accuracy (%) by CSI-Frames	F1-Score	Segmentation Number			
										Accuracy(%)	FP	FN	Total
OA	D1/D1	50	-	2	2	3	3	81.92%	80.57%	90.45%	14	9	241
OA	D1/D2							84.65%	82.84%	83.67%	4	4	49
OA	D2/D2	50	-	2	2	3	4	92.07%	91.03%	100%	0	0	49
LA	D1/D1	100	19	2	2	5	2	91.41%	90.37%	97.51%	4	2	241
LA	D1/D2							90.27%	88.94%	100%	0	0	49
LA	D2/D2	50	27	2	27	10	7	94.89%	94.42%	100%	0	0	49

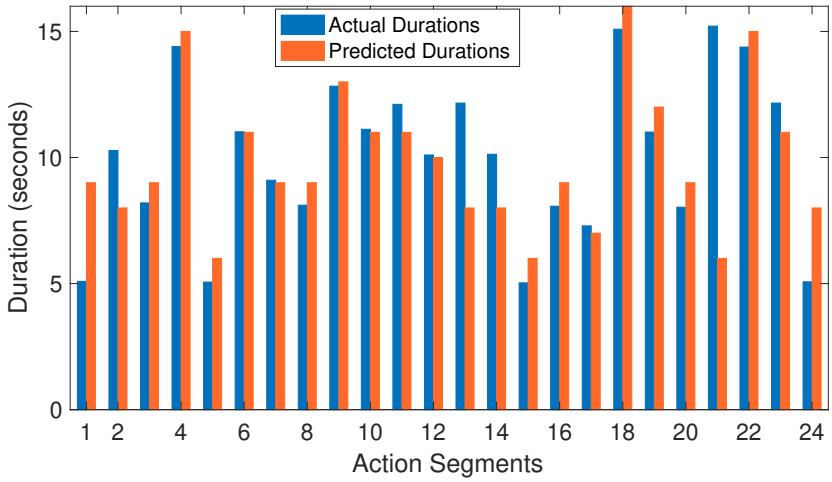


Fig. 13.: Actual and predicted durations per action segment using trained parameters with D1 and applying on the validation dataset D2 (i.e., D1/D2).

In Fig. 12, we show the segmentation results specifically in D1/D2 case to visualize how the segments obtained by the proposed LA method align with the actual segments. There are a total of 24 activity segments (and 25 static segments) and the LA method can recognize all of them at almost their exact boundaries. Therefore, even in the case of the most realistic scenario, the system gives 100% whole-segment recognition accuracy, while the accuracy per CSI frame detection is 90.27%. Note that, per CSI level accuracy will be affected even if the start and end of the segment is slightly missed (e.g., 2nd, 13th, 14th activity segment in Fig. 12) in the prediction, while it may not be very critical in practice. Thus, through an introduction of some flexibility around borders, this accuracy can be further increased.

Once the start and the end of a segment are known, along with the specific transmission frequency (e.g., 100Hz in our experiments), we can also calculate the duration of that activity or non-activity segment. In other words, we can estimate the time a participant performs a given activity using the segmentation predictions.

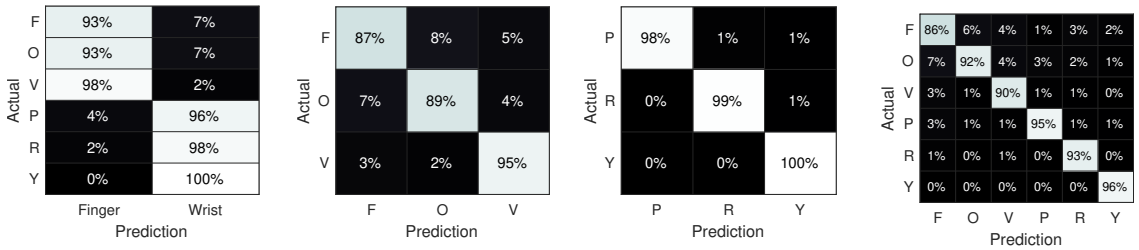


Fig. 14.: (a) Confusion matrix for binary wrist-finger classification. Total accuracy: 96.44%. Confusion matrix when using an individual ML model for (b) finger, total accuracy: 90.02%, and (c) wrist, total accuracy: 99.17%. (d) Confusion matrix obtained from the entire hierarchical model. Total accuracy: 91.22%.

The comparison of the actual and predicted duration is shown in Fig. 13, where, in general, we observe that the predicted durations align with the actual durations of the segments. Note that the proposed system can identify each individual segment in the validation dataset (as discussed in the segmentation results). However, as per the CSI frame level prediction, there are some false ones at the edges of the segments, therefore the duration estimation is not millisecond accurate. Also note that the Fig. 13 shows the comparison of duration results for only the activity segments but we also observe the same for the non-activity segments, since the CSI data collection is continuous and the non-activity segments complement the activity segments in our dataset.

3.4.3.2 Activity Classification

After identifying the start of a new activity segment, our system then performs classification to predict the type of the activity. As it is described in the proposed system architecture, we use a hierarchical solution for this task. That is, we begin with a binary classification to categorize the activity segment as either a wrist or

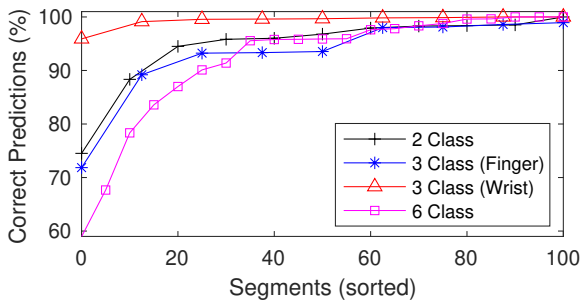


Fig. 15.: Percentage of samples correctly predicted per segment.

finger movement. Following this, we run a corresponding model to further classify the specific wrist or finger movement within that category.

To understand the performance of this approach, we begin by looking at the confusion matrices in Fig. 14a-c which represent the prediction accuracy of each model used in the hierarchical system. In Fig. 14a, the three finger activities: fist (F), OK sign (O), and victory sign (V) are clearly categorized as "finger" activities while pitch (P), roll (R), and yaw (Y) are correctly placed in the "wrist" category with accuracies all much greater than 90%. Once the category is recognized, we then look at the intra-category recognition for both finger and wrist activities separately in Fig. 14b which achieves an accuracy of 90.02% and Fig. 14c which achieves an accuracy of 99.17%. While there is a small amount of confusion between the predictions for the finger activities, this could be due to small pauses between each activity which can be further improved through combining predictions from proceeding time instances. Combining these three models into a hierarchical system, we achieve an overall prediction accuracy of 91.22%. The confusion matrix for this scenario is given in Fig. 14d. One key observation to be made with the use of the hierarchical approach is that if we add additional categories beyond finger and wrist (e.g., arm movements), the category-specific models do not need to be updated and as such, the prediction accuracy of these models remains the same.

Note that with a classical approach the activities could also be recognized through a single model, which we evaluated to have an accuracy of 89.89%. However, unlike the hierarchical approach, if we plan to add more categories (e.g., arm movements), we would need to retrain the model and due to model capacity, we would expect the per-class accuracy to suffer. In our case, we observe that fist accuracy decreases by 3%, OK sign increases by 3%, and the accuracy for the victory sign remains the same. However, if we compare the wrist activities, the six-class model decreases accuracy for all three classes, i.e., 16% for pitch, 1% for roll and 10% for yaw activities. For the yaw activity, this is particularly interesting since in the hierarchical approach, it achieves 100% prediction accuracy for both the binary wrist-finger classification phase as well as the wrist classification phase. This demonstrates that the model loses some pattern-matching capacity as the number of classes increases and demonstrates that the hierarchical approach can offer scalability when adding additional activity categories to the system.

As soon as we identify the beginning of a segment, we can immediately begin running our activity classification model pipeline on all incoming CSI samples. Obviously, not all of the activity classes achieve perfect accuracy. However, since we can also recognize the ending of the segment, we can capture statistics detailing the frequency that each class was predicted for a given segment which can then be passed through a majority voting scheme. To evaluate this, in Fig. 15, we look at each of the segments in our evaluation dataset and plot the number of samples correctly classified per segment. Overall, for all of the segments, more than 50% of samples are predicted correctly, which means that majority voting would allow each segment to be predicted correctly 100% of the time for all of the evaluated segments. Of course, a higher percentage of correctly predicted samples will ensure that the model can continue to generalize on future CSI data. For example, with the six-class classifier,

Table 5.: Comparison of different model sizes and accuracies obtained

Model	Model Size (MB)	Accuracy
2class	5.092	91.22%
finger3class	3.084	
wrist3class	8.731	
6class	3.157	89.89%

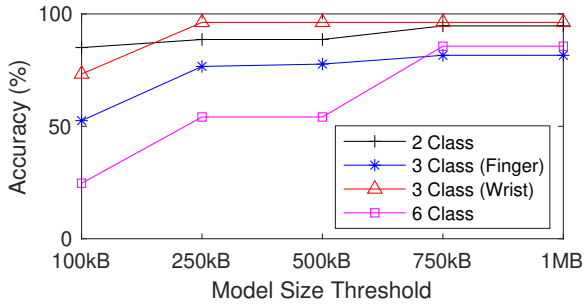


Fig. 16.: Cumulative accuracy achieved when given a threshold for maximum model size.

some segments achieve as low as 58.95% correct samples. On the other hand, the three-class wrist classifier correctly predicts more than 95% of all samples for each of the evaluated segments.

Model Size: In identifying hyperparameters for each of the classification models, we do not directly consider the model size. As such, the model sizes shown for each model in Table 5 vary without a direct correlation to the number of classes that are predicted by the model. Furthermore, since our goal is to run our model on a low-cost embedded microcontroller system, using a consistent and smaller model size allows for a faster inference rate as well as ensures that the model can load completely in memory. As such, we apply a threshold for each of the models in Fig. 16. As we increase the allowed model size threshold, we find that our optimal model hyperparameters can end up in a local-optimal value, thus we plot the cumulative

maximum accuracy for each model. We can see that the six-class classifier suffers the most from decreasing the allowed model size resulting in an accuracy of 54.15% even when given 500kB of model space compared to 77.69%, 96.23%, and 88.63% for the three-class finger classifier, three-class wrist classifier, and two-class activity-group classifier, respectively. Notice, by default, the ESP32 microcontroller we are using for CSI data collection is limited to a standard 520kB of RAM [152]. Furthermore, the hierarchical approach allows us to automatically switch in the corresponding model into RAM as needed compared to the six-class model approach which requires that all corresponding model weights are retained in memory even if they are not important for the prediction. For example, some weights of the model will be dedicated to finger-based movements while others are dedicated to wrist-based movements. However, outside of our proposed hierarchical approach, there is no obvious approach to splitting model weights per category for a pretrained model. This demonstrates that splitting our model training using our proposed hierarchical method allows us to achieve better predictions within the memory constraints inherited from embedded machine-learning environments.

3.4.3.3 Activity Repetition Counting

As the next step, for each recognized activity segment, the proposed system aims to find out the number of repetitions of that activity performed within that segment. Our simple yet efficient counting method is trained for each activity type separately. It looks for repetitiveness in the dataset using the local maxima and minima on the optimized CSI time series data. Table 6 shows the mean absolute error (MAE) of counts obtained for each of the actions using the optimized parameters on the training data, i.e., the sixth column, as well as using the unforeseen validation dataset on the seventh column.

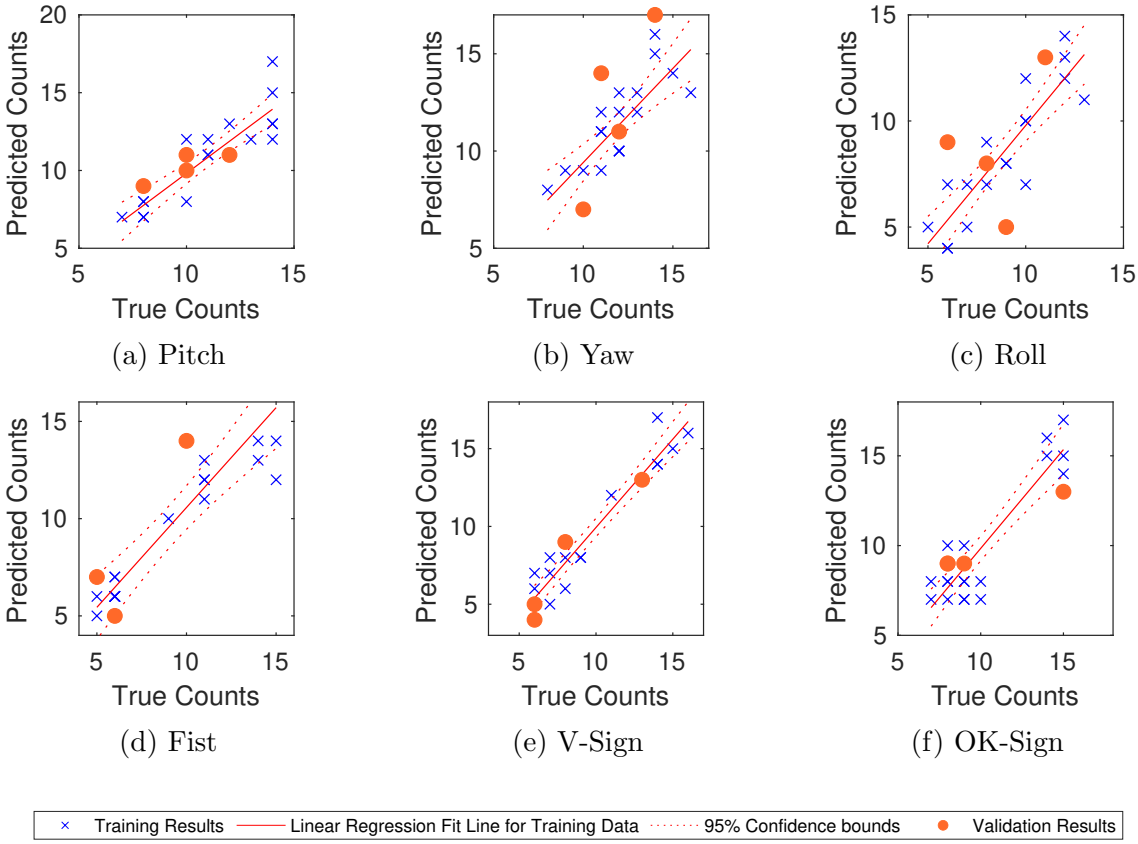


Fig. 17.: Linear regression on predicted vs. true counts per activity.

First of all, the results in Table 6 show that the optimized window size ω is the same, i.e., 50 CSI frames, for each type of activity. The number of optimized repetitive averages, η, is also very similar being mostly 6, but slightly different for the pitch and yaw actions (7 and 8, respectively). Note that the number of sinusoidal peaks mainly depends on these two variables, which is the base of our proposed counting method. On the other hand, the groups of the principal components, as defined by ρ_s and ρ, are found to be completely different from those for the segmentation. The second and third principal components are found to be the most effective in identifying the action or non-action segments as already shown in Table 4. However, the principal components from 21 to 60 are found to be effective for counting the

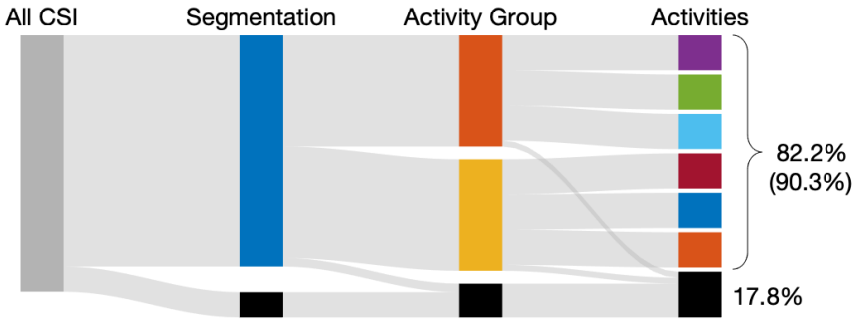
Table 6.: Per action counting results with optimal parameters (MAE 0.933 ± 0.88 for training and 1.37 ± 0.89 for validation).

Action	ω	ρ_s	ρ	η	Tr. Error	Val. Error
Pitch	50	25	32	7	0.95 ± 0.83	0.75 ± 0.50
Yaw	50	21	32	8	1.15 ± 0.99	2.50 ± 1.00
Roll	50	25	2	6	1.05 ± 0.94	0.75 ± 0.95
Fist	50	29	32	6	0.75 ± 0.78	2.25 ± 1.25
V-Sign	50	29	32	6	0.85 ± 0.87	1.00 ± 0.82
OK-Sign	50	32	4	6	0.85 ± 0.87	1.00 ± 0.82

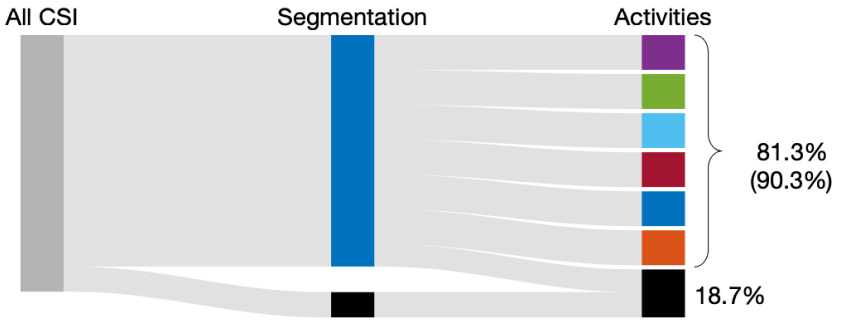
action repetitions. The group of such principal components varies for each of the six actions per the optimizer. The Eq. 3.2 is used to calculate the MAE of the counting method per action. These errors are listed in Table 6. Linear regression lines with 95% confidence interval are plotted in Fig. 17 per action counting model and show that the models do pretty well on a foreseen dataset. The confidence boundary gets thicker for the unforeseen D2 dataset using the D1 dataset's optimized parameters. It can be due to the lower number of samples in the D2 dataset. However, the validation errors are close to the training errors in the case of pitch, v-sign, and OK-sign actions, while deteriorating more in the case of yaw, roll, and fist actions.

3.4.4 Overall System Evaluation

So far we have evaluated the components of our proposed system independently from one another. However, since segmentation recognition, activity-group categorization, and activity-type classification tasks are calculated per CSI sample, we can calculate an overall accuracy for our proposed system. Additionally, we can identify the number of overall samples that failed any of the steps of our proposed system. We present two Sankey plots in Fig. 18a and Fig. 18b which show the percentage of samples that pass through each state of our system when using our hierarchical



(a) Hierarchical model



(b) Single six-class classifier model

Fig. 18.: Sankey plots comparing results of the two machine learning approaches.

model and the six-class classifier, respectively.

From Fig. 18a, we can see that there are four groups. First, we have all 100% of the collected CSI samples. Next, we have segmentation which was successful 90.27% of the time (per CSI frame accuracy in Table 4 for LA with D1/D2 case). The remaining 9.73% of samples are falsely identified as either activity or non-activity and as such, they are marked as failed for the remainder of the plot. Next, we have the binary activity-group classifier which fails 3.6% of samples. Finally, we have the activity classifier which fails around 10% and 0.83% of finger and wrist activities, respectively. Overall, 82.2% of all CSI samples successfully pass through all of the components in the system.

In the case of using the six-class classifier, as shown in Fig. 18b, we no longer separate the activity-group prediction from the actual activity prediction. The segmentation part is still successful 90.3% of the time and after classifying each of the six activities, we end up with 81.3% of activities passing successfully through our system which is lower than the number of samples that were successfully passed through with the hierarchical approach.

Notice, however, that it is not necessary for all samples to successfully pass through the system to achieve highly accurate rehabilitation exercise tracking statistics. By leveraging the segmentation start and stop points along with majority voting, we can further improve the successful throughput such that all 90.3% of the correctly segmented samples are successfully passed through the system. Specifically, while we may have a few anomalies in our predictions because we do not necessarily use the results of our predictions in real-time, we can use knowledge from the entire activity segment to inform our final predictions for all segments within the segment.

3.4.5 Generalizability of the Proposed System

In this part, we evaluate the generalizability and robustness of the proposed system using the other data sets we collected with different volunteers (including some having different hand conditions) and/or in different environments. We also evaluate how the proposed system performs with a new set of movements and discuss the benefit of the hierarchical design.

3.4.5.1 New Volunteers and Environments

As described in Table 3 and Table 4, we also collected data for the same types of hand movements from ten volunteers in five different environments. The summary of the results is given in Table 7. The first row of the table shows the summary of the

evaluation of the primary volunteer's actions, as elaborated in previous sections. The next rows show the results using the same optimal parameters or models from the first data set for each of the other data sets, respectively. It can be seen from Table 7 that once the system is optimized with one volunteer's data, it works well enough for any other volunteer even if they are in different environments with the same set of parameters.

Overall, we see 94.82% segment detection accuracy, average counting error of 1.72 ± 1.19 and a classification accuracy of 83.89%. Here, note that the classification accuracy is per CSI frame, but if we consider segment-level classification, the classification accuracy reaches 97.41%. This is computed by selecting a single class (i.e., highest predicted) per segment through majority voting. As it can be seen in Fig. 19, this majority voting yields 100% accuracy for action classification within each segment which makes the segmentation per CSI accuracy (S_c) the actual classification accuracy for the action segments. As the predicted non-action segments can also have

Table 7.: Validation with other volunteer and environment combinations.

Vol. No.	Env.	Acc. Per CSI S_c (%)	Segment Detect. S_d (%)	Duration Error (seconds)	Counting Error (MAE±Std.)	Classification Accuracy (%)				
						Binary C_b	Finger C_f	Wrist C_w	Overall C	Segment Level, C_s
1	1	90.27	100.0	1.11 ± 1.97	1.37 ± 0.89	96.44	90.02	99.17	91.22	99.15
1	1	87.53	94.48	1.35 ± 1.88	2.34 ± 1.87	94.52	86.07	98.74	87.34	98.42
2	4	84.55	93.10	1.37 ± 1.76	1.45 ± 1.33	92.17	83.56	90.12	80.04	96.92
3	1	88.43	93.10	1.42 ± 1.53	2.15 ± 1.03	94.15	86.87	97.35	86.72	98.46
4	1	81.65	94.48	2.12 ± 1.23	1.45 ± 1.11	90.63	82.11	88.36	77.25	95.83
5	1	73.62	88.97	1.82 ± 2.39	2.26 ± 1.52	84.72	81.22	87.26	71.37	92.45
5	5	90.32	100.0	1.21 ± 1.56	1.91 ± 1.33	96.11	81.41	87.65	81.24	98.18
6	1	80.17	92.42	1.89 ± 1.74	1.69 ± 1.38	98.64	84.29	90.46	86.19	97.26
7	1	88.84	98.35	1.37 ± 1.28	1.45 ± 0.88	93.24	87.45	96.89	85.94	98.43
8	1	84.52	96.69	1.64 ± 1.21	1.58 ± 1.06	95.86	88.96	98.77	89.98	98.45
9	1	83.42	93.79	1.56 ± 0.57	1.29 ± 1.05	94.63	88.06	96.34	87.25	97.89
10	4	85.64	92.41	1.54 ± 1.28	1.66 ± 0.78	92.05	86.05	92.35	82.11	97.43
Average		**84.91**	**94.82**	1.53 ± 1.53	$\mathbf{1.72 \pm 1.19}$	93.60	85.51	93.62	83.89	**97.41**

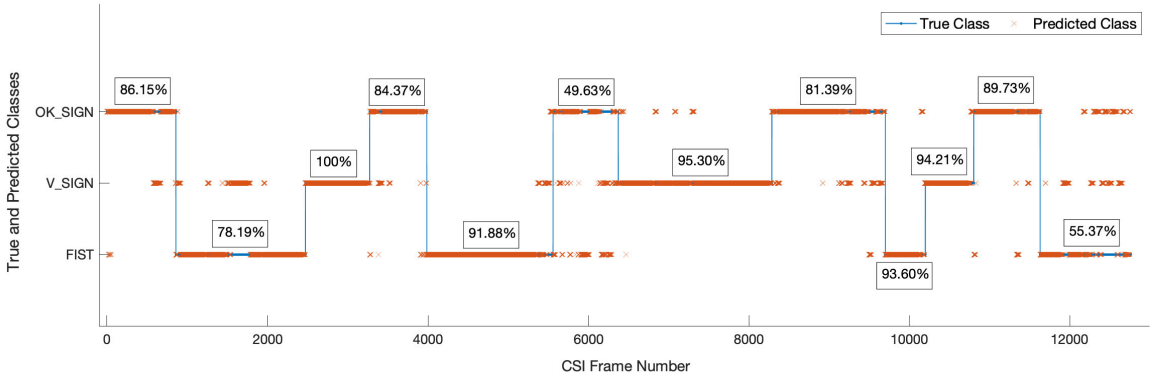


Fig. 19.: Per-segment classification results showing that with majority voting each segment can be classified to one class accurately (even in the dataset that gives the worst case CSI level accuracy i.e., volunteer 3, environment 2 in Table 8).

mis-segmented CSI for actions, we also consider the overall classification accuracy (C) for those parts. In the end, our segment-level classification accuracy can be calculated as:

$$C_s \leftarrow S_c + C \times (100 - S_c). \tag{3.3}$$

Such segment-level classification accuracy per data set is shown in the last column of the Table 7 and Table 8.

While in some cases we see slightly lower results, especially for the finger and wrist action classification, this can be considered acceptable as the finger and wrist movement varies for different people but the movement detection as well as segmentation for moving vs. static environment evaluation remains the same. The fluctuation of CSI amplitudes obtained in the proposed counting system also works almost the same for all volunteers. Per activity results for all volunteers are also inline with the action-wise results given in Table 6 for the primary data set. From all of these results, we can say that the proposed system, once optimized for the hyper-parameters with any volunteer, is generalizable for data sets collected in different environments with

different volunteers.

3.4.5.2 Special Scenarios

As for some special scenarios, we first consider some hand conditions for the users, like wearing a wrist thumb support, since this could be the case in practice for such patients. Similarly, while we assume that there is only one person[2] monitored in the physical therapy session in this study, it is possible that there could be some other people around while the user is performing physical therapy exercises. Thus, we also consider such scenarios and analyze the effect on results. The details of these special scenarios and the results with each of them are provided in Table 8. Overall, we see a slightly lower average performance compared to the results in Table 7. As expected, this is due to the effect of either the movements of other people in the environment or due to the hand conditions of volunteers. However, the system's performance can still be considered acceptable.

3.4.5.3 New Set of Movements

To evaluate the scalability of our system with different sets of movements, we collected a data set with three new movements. Complicating the movements further, we consider physical therapy exercises that include interaction with an object. These movements are (i) grip and pinch a hand therapy ball, (ii) finger spread using a hand band, and (iii) drop and grab a hand therapy ball. Note that the hierarchical design used in the classification part of the proposed system allows for developing a model for this new set of movements without changing the models for other movements. We only need to retrain the activity group classifier to recognize this new group.

[2]Monitoring more than one person through WiFi sensing [190] is more challenging and is out of the scope of this study.

Table 8.: Validation on special scenarios

Vol. No.	Env.	Speciality	Acc.(%) Per CSI	Segment Det. (%)	Duration Error	Counting Error	Classification (%)	
							Overall	By Segment
1	2	Multiple people	76.27	90.34	2.73 ± 2.18	1.58 ± 2.89	68.28	92.47
1	1	Wearing gloves	90.13	99.31	1.52 ± 1.84	1.83 ± 1.63	80.98	98.12
1	1	Wearing wrist-support	88.74	97.93	1.44 ± 1.73	1.44 ± 1.57	82.44	98.02
1	3	Wearing gloves	83.78	91.72	1.94 ± 1.66	1.89 ± 1.49	82.05	97.09
2	3	Wearing gloves	77.59	91.03	1.98 ± 1.55	1.84 ± 1.55	78.99	95.29
2	3	Wearing wrist-support	79.74	92.41	1.54 ± 1.78	1.93 ± 1.19	80.11	95.97
3	2	Multiple people	74.91	85.52	1.04 ± 0.94	1.52 ± 3.04	65.04	91.23
4	1	Wearing wrist-support	87.16	95.17	1.89 ± 1.09	1.29 ± 1.53	81.03	97.56
Average			**82.29**	**92.93**	1.76 ± 1.60	$\mathbf{1.67 \pm 1.86}$	77.37	**95.72**

However, we do not need to change anything for segmentation and counting. As a result, using the same optimized parameters for segmentation and counting, we still obtain similar segmentation accuracy (84.36% per CSI and 93.10% detection accuracy) and counting error (1.79±1.19). For the classification, the retrained activity group classifier for the three groups provides 96.12% accuracy, and the action classifier for this new group of three activities gives 75.57% accuracy. When we incorporate the previous two models, i.e., wrist and finger classifiers, we calculate the overall classification accuracy with all nine actions as 81.61%. With the aforementioned majority voting idea, we then obtain a segment level classification of 97.12%. This is quite close to the previous accuracy for six actions (i.e., 97.41% with two groups of activities), showing the generalizability of the proposed solution to include new movements.

3.4.6 Comparison to Existing Systems

Among all the existing systems, the work of Chen et. al. [180] is the closest work to our study in terms of its design. While it does not provide a counting component, its segmentation results look promising. On the other hand, similar to all other datasets for segmentation studies, the dataset of this study is also for large-scale human movements, like sitting down, standing up, walking, stooping, and waving. Thus, their approach is less likely to suit our use case of small-scale movements like finger movements. However, for the sake of comparison, we implement their approach with the Butterworth low-pass filter, PCA, and using their dynamic threshold calculation and sliding window-based segmentation methods on our small-scale activity datasets. The threshold weight parameter, α, is not properly introduced in their study, and the lengths of the two specific windows are not well specified. Thus, we run the algorithm for different possible floating values ranging from 0.1 to 0.9 for α and consider the

value of 0.5 giving the best result. Besides, the threshold is described to be updated every two minutes, which led us to consider the larger window size having two minutes of CSI data frames, i.e., consisting of $2 \times 60 \times 100 = 12000$ CSI frames of our dataset with 100Hz transmission frequency. The shorter window is calculated to be one-sixth part of the larger window, as depicted in their segmentation figure. A major downside of their segmentation algorithm is they calculate the start points and end points of the segments disjointly, which leads to discontinuous segmentation in our dataset. Our implementation of their system can identify at best 17 starting points out of the 118 starting points of the action segments, while only nine ending points were identified. Therefore, it can be deduced that their system for large-scale movement segmentation cannot be applied to our small-scale movement segmentation.

3.5 Clinically Evaluated Enhancement: Spatial Hand Movements and Real-Time Inference

Building upon the Wi-PT-Hand system presented in Section 3.3, this section describes the clinically validated next-generation system designed to handle spatially dynamic hand rehabilitation exercises involving multi-point trajectories and variable execution speeds. While the original system assumes spatially stationary hand movements (wrist flexion, finger extension, etc.), practical physiotherapy for neurological patients often involves moving the gripping hand between multiple spatial targets in sequence. This introduces new challenges including speed variability between healthy volunteers and patients with Parkinson's disease (PD) or stroke, environment-independent signal stability, and the need for real-time feedback during live therapy sessions. The system described here addresses all of these challenges through a redesigned Faraday-enclosed hardware setup with four WiFi links, a speed-invariant classification pipeline, a multi-link CNN ensemble with temporal majority

voting, and a pull-based real-time inference architecture running entirely on a single Raspberry Pi 4B edge device.

3.5.1 System Requirements

Designing an at-home rehabilitation tracking system for spatially dynamic movements introduces several technical challenges that remain insufficiently addressed in prior work. The system must simultaneously satisfy the following requirements:

1. **Real-time inference:** enabling immediate visual feedback during therapeutic sessions.
2. **Portability and at-home deployment:** ensuring the platform can be used outside of clinical environments without professional setup.
3. **Remote serviceability:** allowing maintenance, debugging, and software updates without requiring in-person intervention.
4. **Offline operation:** preserving user privacy by eliminating dependency on cloud connectivity during data collection and inference.
5. **Low-power consumption:** supporting continuous operation on resource-constrained IoT devices.
6. **Low-cost design:** reducing adoption barriers without compromising recognition accuracy.
7. **Speed invariance:** maintaining classification accuracy despite differences in movement execution speed between healthy volunteers and impaired patients.

Addressing these requirements in a unified solution constitutes the central problem targeted in this section.

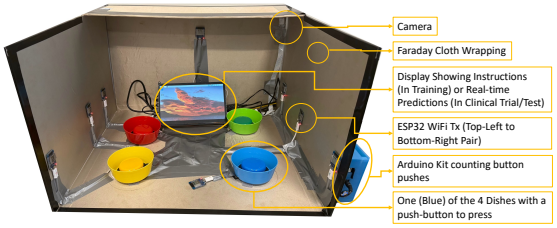
3.5.2 Faraday Enclosure with Multiple WiFi Links

To ensure reliable CSI data acquisition and eliminate environmental electromagnetic interference, the system employs a dedicated Faraday enclosure built from Faraday cloth [191] around a portable box. Within the enclosure, four ESP32 Tx-Rx pairs are deployed in horizontal (HH), vertical (VV), and two diagonal (LL and RR) alignments, forming criss-crossing WiFi links inside the box. This configuration was chosen after experimenting with several alternative alignments, as it provides the most consistent and spatially diverse CSI variations for fine-grained hand movement recognition.

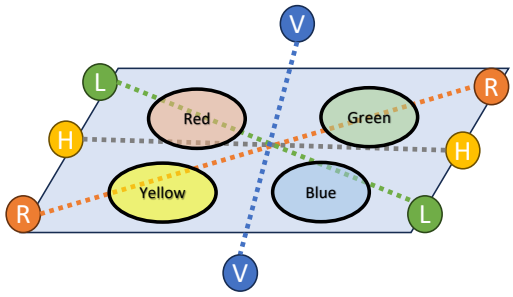
As shown in Fig. 20, four colored dishes are placed at the corners of a square on the floor of the box: red (top-left), green (top-right), yellow (bottom-right), and blue (bottom-left). The patient moves the affected hand between any two dishes in a round-robin fashion, covering all six possible directions: red–green (RG), red–yellow (RY), red–blue (RB), green–yellow (GY), green–blue (GB), and yellow–blue (YB). Each dish is equipped with a button that the participant presses upon arrival, triggering a connected Arduino kit that records ground truth for repetition counting. A monitor centered inside the box displays animated exercise instructions during data collection (Fig. 20c) and shows real-time predictions during live sessions (Fig. 20d).

3.5.3 Remote Serviceability and Ease of Use

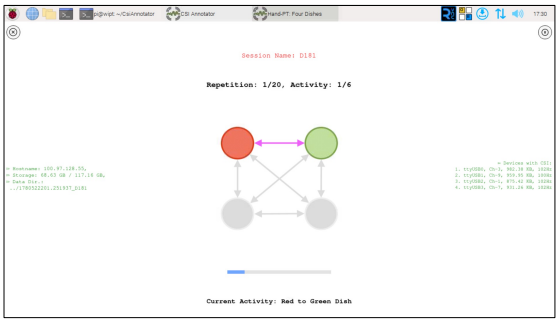
The Rx units of the four ESP32 pairs are connected to a Raspberry Pi 4B that handles local CSI data storage and real-time processing. A key enhancement over the original Wi-PT-Hand setup is support for *remote monitoring and serviceability* when the system is connected to a home WiFi network or LAN, substantially reducing deployment cost and setup complexity. Each deployed unit can be securely accessed



(a) Box with Faraday cloth, WiFi dishes, and display.



(b) Six movement directions among the four color-coded dishes.



(c) Display showing animated exercise instructions.

(d) Display showing real-time activity predictions.

Fig. 20.: Data collection and inference box: (a) hardware enclosure with Faraday cloth, (b) the six movement directions among color-coded dishes, (c) instruction display shown during data collection, and (d) real-time prediction display shown during live sessions.

through a VPN tunnel [192], allowing the research team to remotely inspect system status, debug errors, and deploy software updates without physical access. The system is further integrated with smart power control units, enabling remote restarts in the event of a malfunction. Automated health-check messages periodically summarize system status and data collection progress, allowing disruptions to be detected and addressed proactively.

From the user's perspective, the system requires minimal technical involvement: once powered on and connected to the home network, it operates autonomously. In

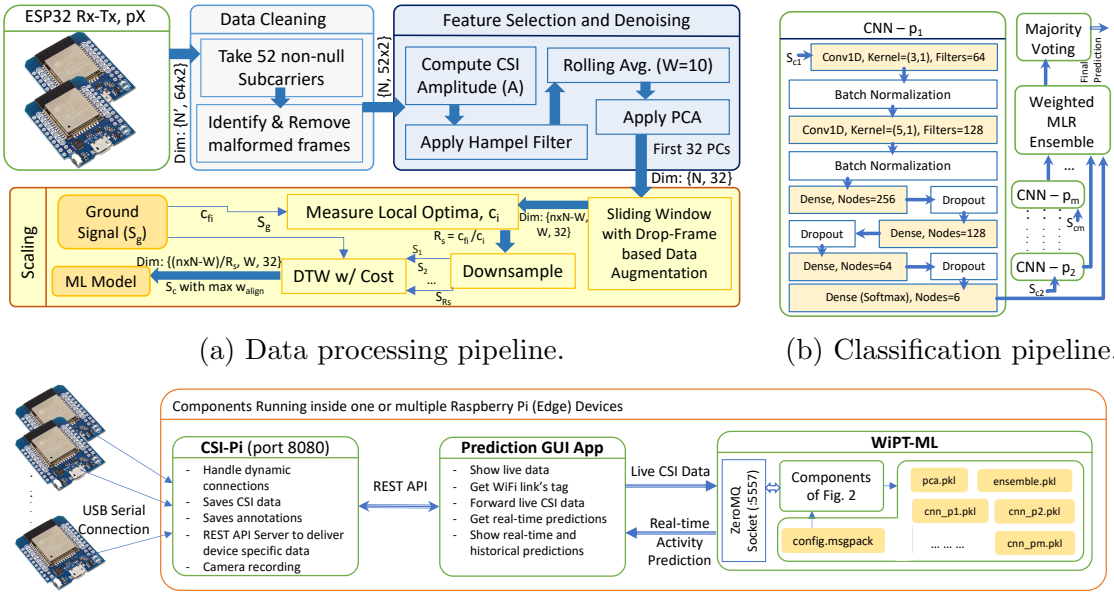
case of issues, participants can request assistance and the research team can remotely diagnose and resolve problems. This capability was critical during the extended deployment involving 51 volunteers and patients across multiple sessions, enabling reliable long-term CSI data collection in naturalistic environments. It is noteworthy that the system is fully functional with an air-gap network; an internet connection is required only for remote servicing and software updates.

3.5.4 Activity Classification Pipeline

The activity classification pipeline is designed to recognize six spatially distinct hand-based rehabilitation exercises from WiFi CSI measurements collected simultaneously from the four ESP32 transceiver pairs. The pipeline is engineered to remain robust under varying movement speeds, particularly addressing the mismatch between healthy volunteers performing exercises at regular or fast speeds and clinical patients who typically move more slowly due to motor impairments. An overview of the pipeline is shown in Fig. 21.

3.5.4.1 Preprocessing and Feature Extraction

Each CSI frame contains measurements from 64 subcarriers, of which 52 non-null subcarriers are retained. The CSI amplitudes are computed and denoised using a Hampel filter to suppress impulsive noise, followed by a rolling mean filter of size 10 to smooth short-term fluctuations. To reduce dimensionality while preserving maximum variance, PCA is applied, and the first 32 principal components are retained as features. These steps are applied independently per ESP32 transceiver pair.



(a) Data processing pipeline.

(b) Classification pipeline.

(c) Real-time prediction architecture on live CSI data.

Fig. 21.: WiFi CSI activity recognition pipeline: (a) raw data processing with speed-invariant downsampling and DTW alignment, (b) per-link CNN classifiers fused through a validation-weighted multinomial logistic regression ensemble, and (c) real-time prediction flow from the CSI-Pi server through the ZeroMQ socket to the GUI display.

3.5.4.2 Speed-Invariant Scaling

To account for variable movement speeds, a reference signal S_g is recorded from the fastest-moving participant (i.e., one or more full movement repetitions per second) and used as the temporal reference for all other sessions. For each incoming segment, a resampling factor R_s is computed as the ratio of the local-optima frequency in S_g to that in the current segment. Let c_{fi} denote the average number of local optima in the i-th segment of the fastest-moving reference dataset, and c_i the average number

of local optima in the current segment. The resampling factor is:

$$R_s = \frac{c_{fi}}{c_i}, \qquad R_s \geq 1. \tag{3.4}$$

For each segment, R_s offset candidates $\{S_1, S_2, \ldots, S_{R_s}\}$ are generated by taking every R_s-th sample starting at offset $k \in [1, R_s]$. Prior to downsampling, a fourth-order Butterworth low-pass filter is applied with cutoff frequency

$$f_c = \frac{1}{2} \cdot \frac{f_s}{R_s}, \tag{3.5}$$

where f_s is the original sampling rate, with zero-phase filtering employed to avoid phase distortion. The resampling factor R_s is recalculated on the fly over the last τ_s segments, where τ_s is a hyperparameter tuned during training, so that the system adapts dynamically if the participant's movement speed changes across repetitions within the same session. This design maintains speed invariance both intra-session and inter-session.

3.5.4.3 Dynamic Time Warping Alignment

Since different downsampling offsets may align differently with the ground-truth movement dynamics, each offset candidate S_k is projected to one dimension by computing the ℓ_2 norm of its 32 principal components. The candidate is then temporally aligned with the reference fast signal S_g using FastDTW [193] with a Sakoe–Chiba band radius of 50 frames (i.e., 500 ms at 100 Hz transmission frequency). The normalized alignment cost is defined as:

$$\text{cost}_{\text{norm}} = \frac{d_{\text{DTW}}}{L_{\text{path}}}, \tag{3.6}$$

where d_{DTW} is the DTW distance and L_{path} is the warping path length. The alignment weight for each candidate is:

$$w_{\mathrm{align}} = \frac{1}{1 + \mathrm{cost}_{\mathrm{norm}}}. \tag{3.7}$$

The candidate with the maximum w_{align} is selected and linearly interpolated to 150 frames to serve as the classifier input.

3.5.4.4 Base CNN Classifiers

Four base classifiers are trained independently, one per ESP32 transceiver pair spatially positioned inside the box. As depicted in Fig. 21b, each base model is a one-dimensional convolutional neural network (1D-CNN) accepting a CSI window through a Conv1D layer with kernel size 3 and 64 filters, followed by batch normalization. A second Conv1D layer with kernel size 5 and 128 filters (with batch normalization) captures longer-range temporal dependencies. Three fully connected dense layers with 256, 128, and 64 ReLU neurons follow, each with dropout regularization. The final classification output is produced by a six-neuron softmax layer. All models are trained using the Adam optimizer with sparse categorical cross-entropy loss. Validation F1-scores are recorded per link and used as weights in the ensemble stage.

To capture different stages of a movement within a single repetition, temporal windows of 180 CSI frames are extracted with a stride of one. Each window is augmented by randomly dropping 30 frames three times, producing three augmented samples of 150 frames each. This drop-frame augmentation effectively triples the training set size and enables the base models to learn from sub-temporal views of each movement, improving robustness to speed variations even within a single session.

3.5.4.5 Ensemble Learning

To leverage the complementary spatial sensing characteristics of the four WiFi links, the class-posterior output vectors of the four base CNNs are concatenated and used as input to a multinomial logistic regression ensemble model. For a windowed CSI segment $\mathbf{x}$ and link $j \in \{1, \ldots, P\}$ where P=4, the j-th CNN produces a probability vector:

$$\mathbf{p}_j(\mathbf{x}) = \left[p_j^{(1)}(\mathbf{x}), \ldots, p_j^{(C)}(\mathbf{x})\right]^\top, \quad \sum_{c=1}^{C} p_j^{(c)}(\mathbf{x}) = 1, \tag{3.8}$$

where C=6 is the number of activity classes. The ensemble feature vector is formed by concatenation:

$$\mathbf{z}(\mathbf{x}) = \left[\mathbf{p}_1(\mathbf{x})^\top, \ldots, \mathbf{p}_P(\mathbf{x})^\top\right]^\top \in \mathbb{R}^{PC}. \tag{3.9}$$

A multinomial logistic regression then produces the fused posterior:

$$\hat{\mathbf{p}}_e(\mathbf{x}) = \mathrm{softmax}\big(\mathbf{W}\mathbf{z}(\mathbf{x}) + \mathbf{b}\big), \qquad \hat{y}(\mathbf{x}) = \arg\max_c \ \hat{p}_e^{(c)}(\mathbf{x}), \tag{3.10}$$

where $\mathbf{W}$ and $\mathbf{b}$ are learned on a held-out validation set.

The validation F1-scores of the base learners serve as reliability weights to modulate their contribution to the ensemble. Let f_j denote the validation F1-score of base model j. The normalized weights are:

$$w_j = \frac{f_j}{\sum_{k=1}^{P} f_k}, \qquad \sum_{j=1}^{P} w_j = 1. \tag{3.11}$$

This weighting is motivated by the spatial specialization of individual WiFi links. Due to the Faraday enclosure increasing multipath signal density through reflections from the cloth surfaces, line-of-sight (LoS) transmissions become more prominent. Actions whose trajectories intercept a given link's LoS path achieve high accuracy on that link but lower accuracy on other links. By up-weighting base models with higher

validation F1-scores, the ensemble emphasizes the most reliable links per activity class and mitigates link-specific sensing biases, resulting in more robust fusion across all six movement directions.

3.5.4.6 Temporal Aggregation via Majority Voting

To suppress transient misclassifications, predictions are aggregated over the most recent K=100 sliding windows with a stride of one using majority voting. Since each sliding window spans 150 CSI frames (1.5 s) and windows overlap with stride one, this buffer covers approximately 2.8 seconds of activity. Given that each exercise direction is performed for at least 8 seconds, this strategy yields stable predictions without mixing adjacent activities. The impact of varying K is studied in Section 3.5.5.5.

3.5.4.7 Multi-Link Weighted Repetition Counting

The segmentation algorithm from the original Wi-PT-Hand system [194] is reused directly, using the horizontal (H) WiFi link pair for optimal segmentation performance. However, the single-link local-optima-based counting algorithm from [194] exhibits high mean absolute error (MAE) for slow-moving and clinical participants, because slower movements generate weaker and less distinct CSI perturbations. To address this, a normalized weighted average of local optima across all four WiFi links is used. For link j, the Euclidean distance between successive maxima and minima in the time-vs.-principal-component plot is computed, and the average α_j of all such optima-pair distances is recorded per segment. The normalized weight for link j out of P links is $w_j = \alpha_j / \sum_{k=1}^{P} \alpha_k$, and the predicted repetition count for the i-th segment is:

$$\hat{c}_i = \sum_{j=1}^{P} w_j \, c_{ij}, \quad w_j = \frac{\alpha_j}{\sum_{k=1}^{P} \alpha_k} \geq 0, \quad \sum_{j=1}^{P} w_j = 1, \tag{3.12}$$

where c_{ij} is the raw count estimate from link j for segment i.

3.5.4.8 Real-Time Prediction Implementation

To support real-time prediction on the resource-constrained Raspberry Pi 4B, the system uses a pull-based REST API architecture. The CSI-Pi data collection server [195] exposes raw CSI frames on port 8080. A GUI application periodically fetches the most recent 100 windows worth of CSI frames and forwards them to a ZeroMQ socket [195] hosting the complete data pipeline—preprocessing, speed-invariant scaling, DTW alignment, CNN inference, and ensemble fusion. All three applications (CSI-Pi server, ZeroMQ ML server, and GUI app) run on the same Raspberry Pi device, eliminating data-transfer latency. Trained CNN models and ensemble weights are stored as Pickle and MsgPack files for fast I/O. The system requires an initial accumulation period of approximately 2.8 seconds to fill the majority voting buffer, after which predictions are produced at approximately four predictions per second and displayed on the attached touchscreen in real time.

3.5.5 Evaluation

3.5.5.1 Data Collection and Demographics

The system is evaluated through a multi-phase data collection campaign involving 40 healthy volunteers and 11 clinical patients. Each volunteer performed the same set of six hand rehabilitation exercises (20 repetitions per exercise, 8-12 seconds per exercise) in a round-robin order, with a 10-second static interval between successive exercises. Clinical datasets are labeled `T-Y`, where Y is the patient index. Volunteer datasets are labeled `D-X-Y`, where X denotes the collection day, and Y denotes the participant index on that day. However, `D-0` is reserved for the fastest-moving reference dataset, while inserting a `0` in `D-X-0-Y` serves as a flag indicating that the session was intentionally slowed down for speed invariance training.

Table 9.: Participant Demographics and Characteristics

Category	**Volunteers (40)**	**Patients (11)**	**Overall (51)**
Gender (M/F)	15 / 25	7 / 4	22 / 29
Age (years)	23.6 ± 1.8	54.3 ± 13.5	29.98 ± 13.97
Speed Category	Regular: 28 Slow: 11 Fast: 1	Regular: 1 Slow: 10	Regular: 29 Slow: 21 Fast: 1
Clinical Diagnosis	Healthy	PD: 7 Stroke: 4	–

During clinical data collection, we observed that patients with PD or stroke performed exercises substantially more slowly than healthy volunteers, occasionally paused mid-movement, or dropped the affected hand between dishes. Patients with PD also exhibited tremors. To better represent speed variability in the training data, a subset of healthy volunteers was subsequently asked to perform the exercises at intentionally slower speeds. Participants were categorized as regular, slow, or fast based on their observed average exercise repetition frequency. Table 9 summarizes the demographic and clinical characteristics of all 51 participants.

3.5.5.2 Training and Validation

The model was developed using 16 volunteer datasets collected across different days and participants, providing temporal and inter-subject diversity. For each of the 16 training datasets, 60%, 20%, and 20% of the data were used for training, validation, and testing, respectively. The remaining 24 volunteer datasets and all 11 patient datasets were withheld entirely for external testing. Drop-frame data augmentation was applied to the training data, tripling its effective size. The impact of augmentation parameters is analyzed separately in Section 3.5.5.5.

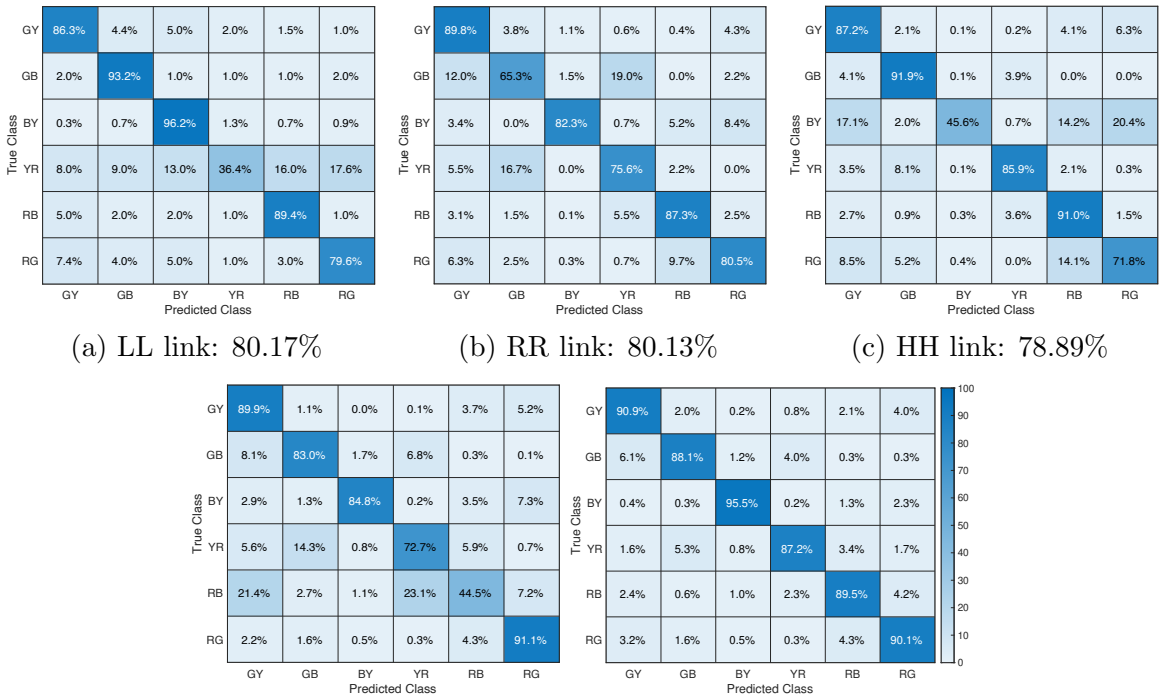
3.5.5.3 Segmentation and Counting Results

The segmentation algorithm [194] achieves a segment detection accuracy of 98.6%± 0.51% across all volunteer datasets and $92.7\% \pm 3.8\%$ on patient datasets, confirming its robustness in the new spatial exercise setting.

For repetition counting, the single-link local-optima algorithm achieves a MAE of 1.67 ± 0.9 for regular-paced participants but degrades to 4.3 ± 3.1 for slow-moving volunteers and patients, due to weaker and less distinct CSI perturbations at lower speeds. The multi-link weighted counting approach of Eq. (3.12) reduces the error to 1.43 ± 0.68 for regular-paced participants, 1.80 ± 0.95 for slow-moving volunteers, and 2.57 ± 1.58 for clinical patients. The higher patient error is attributable to PD tremors and intermittent hand drops between dishes, which introduce irregular CSI perturbations. Nevertheless, the multi-link approach consistently outperforms the single-link baseline across all speed categories.

3.5.5.4 Classification Results

Individual WiFi link classifiers exhibit strong spatial specialization due to the LoS sensitivity of the enclosed sensing environment. For example, in one representative dataset, the LL-direction link (top-left to bottom-right diagonal) achieves 96.2% recognition accuracy for the blue–yellow (BY) movement direction but only 36.4% for the yellow–red (YR) direction, while the horizontal (HH) link reverses this pattern with 45.6% on BY and 85.9% on YR. This behavior arises because actions traversing a link's LoS transmission path produce stronger CSI perturbations, whereas off-LoS actions produce weaker ones. By applying the validation F1-score weighting scheme of Eq. (3.11), the ensemble fuses the four link-specific classifiers to achieve high accuracy across all six movement directions. The resulting confusion matrices for the



(a) LL link: 80.17% (b) RR link: 80.13% (c) HH link: 78.89%

(d) VV link: 77.67% (e) Ensemble: 90.2%

Fig. 22.: Confusion matrices for the four individual WiFi link CNN classifiers and the final validation-weighted ensemble model on an example dataset. The per-link spatial specialization is clearly visible; the ensemble fusion recovers high accuracy across all six movement directions.

four individual link classifiers and the ensemble model on the same example dataset are shown in Fig. 22, with the ensemble achieving 90.2% overall accuracy compared to individual link accuracies ranging from 77.67% to 80.17%.

Fig. 23 shows dataset-specific classification accuracy across all 51 datasets in chronological order of collection, illustrating the temporal distribution of training, test, and clinical datasets. The full system achieves an average accuracy of 94.18% on internal validation splits, 91.37% on unseen volunteer test datasets, and 86.29% on the clinical patient cohort.

Fig. 24 shows the relationship between the number of training datasets and clas-

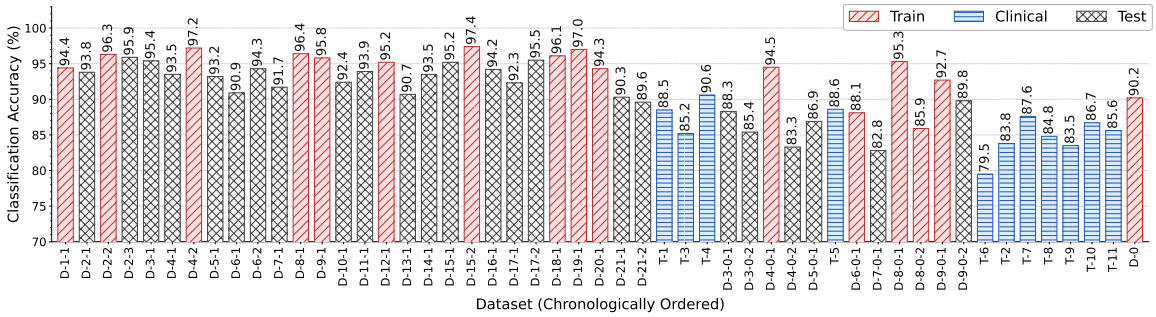


Fig. 23.: Dataset-specific classification accuracy for the full system across all 51 datasets (16 training, 24 external volunteer test, and 11 clinical), shown in chronological order of collection.

sification accuracy on the validation set. Using more datasets initially improves accuracy, but including too many datasets without sufficient variety causes overfitting. Training on 16 diverse datasets with drop-frame augmentation achieves the best generalization.

3.5.5.5 Ablation Studies

To assess the individual contribution of each pipeline component, nine ablation studies were conducted. Table 10 summarizes the mean overall classification accuracy and inference time per ablation configuration, while Fig. 25 visualizes per-group accuracy across internal validation, external test, and clinical cohort splits.

Hampel Filter: Without the Hampel filter, classification accuracy drops to (71.24 ± 7.86)%. The Hampel filter consumes the largest share of inference time in the pipeline; removing it reduces the overall inference time from 212.5 ± 51.1 ms to 78.24 ± 7.92 ms. This indicates a clear accuracy–latency trade-off: in time-critical deployments where sub-100 ms latency is essential, the Hampel step could be negotiated in exchange for reduced accuracy.

PCA: Removing PCA causes the most severe accuracy drop in the entire study, to

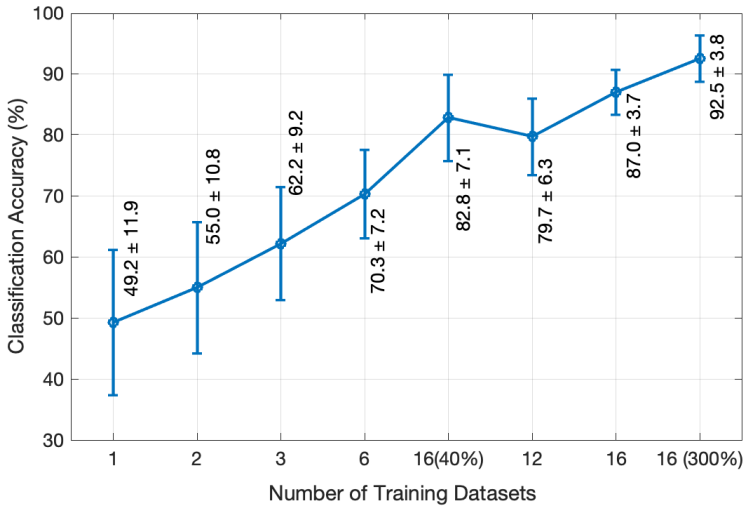
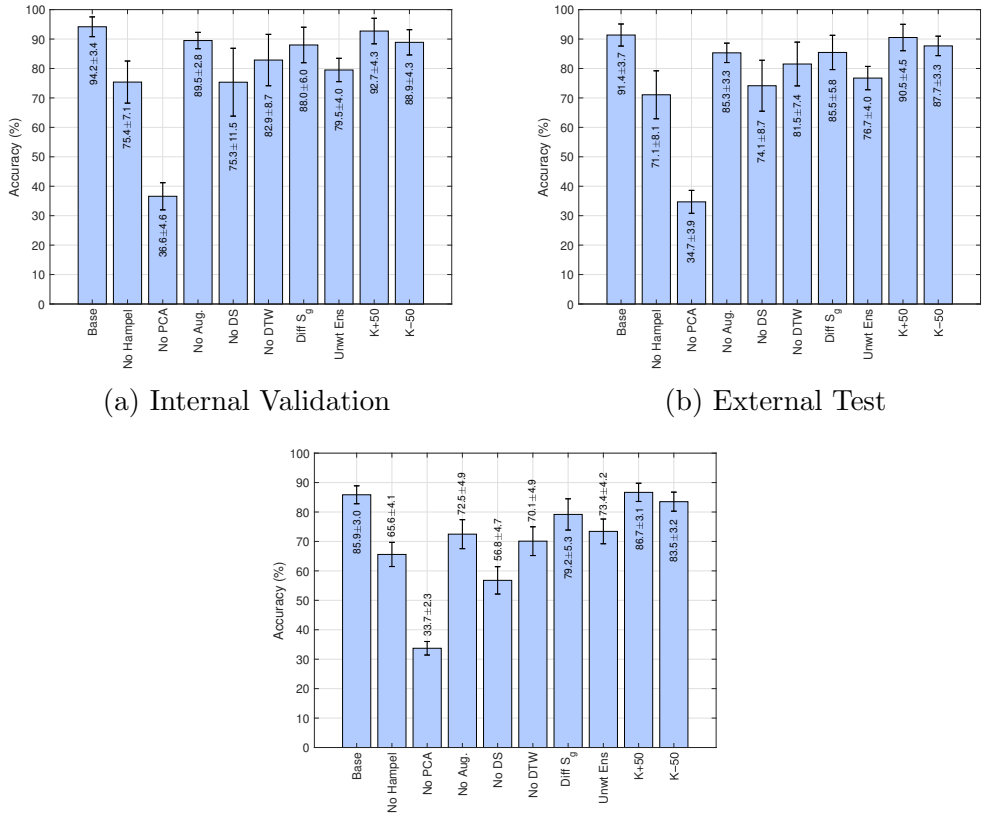


Fig. 24.: Classification accuracy as a function of training dataset count and configuration, showing the benefit of diverse training data with drop-frame augmentation.

Table 10.: Classification Accuracy and Inference Time (Mean ± Std. Dev.) for Each Ablation and Dataset Group

Ablation	**Acc. (%)**	**Inference (ms)**			
	Overall	**Overall**	**Clin.**	**Test**	**Val.**
Full System	91.06 ± 4.56	212.5 ± 51.1	296.0 ± 51.3	188.0 ± 20.0	191.8 ± 21.6
No Hampel	71.24 ± 7.86	78.24 ± 7.92	81.52 ± 7.81	76.38 ± 8.96	78.56 ± 6.38
No PCA	35.08 ± 3.95	171.8 ± 19.3	179.7 ± 19.2	172.5 ± 21.8	164.7 ± 14.3
No Aug.	83.86 ± 7.20	139.2 ± 15.7	153.9 ± 15.6	135.4 ± 14.6	133.8 ± 14.1
No DS	70.77 ± 11.56	122.8 ± 13.6	128.9 ± 13.1	120.0 ± 10.5	122.3 ± 16.4
No DTW	79.48 ± 8.84	131.7 ± 14.6	145.5 ± 14.1	129.8 ± 9.1	124.3 ± 13.0
Diff. S_g	84.90 ± 6.52	216.9 ± 57.2	307.7 ± 56.9	187.0 ± 26.4	193.6 ± 15.8
Unwt. Ens.	76.89 ± 4.50	204.8 ± 58.7	300.4 ± 58.4	175.5 ± 20.4	177.2 ± 22.6
K+50 (150)	90.38 ± 4.65	206.2 ± 39.1	268.8 ± 39.0	184.2 ± 11.0	192.3 ± 17.0
K−50 (50)	87.16 ± 4.08	242.8 ± 54.9	328.0 ± 54.5	219.0 ± 26.4	214.6 ± 24.8

$(35.08 \pm 3.95)\%$, confirming that the principal components of the 52 non-null subcarrier amplitudes constitute the primary discriminative features for all downstream



(a) Internal Validation

(b) External Test

(c) Clinical Cohort

Fig. 25.: Ablation study classification accuracy (mean ± std. dev.) across nine configurations for (a) internal validation, (b) external test, and (c) clinical cohort datasets.

classifiers. Inference time decreases only marginally because the raw 64-subcarrier data dimension is close to the 32 PCs retained.

Data Augmentation: Without drop-frame augmentation, accuracy drops to (83.86± 7.20)%. Inference time also decreases due to lower computational load during batch processing of the smaller dataset. The augmentation hyperparameter study described next reveals the optimal configuration.

Downsampling (DS): Removing the speed-invariant downsampling step causes a severe accuracy drop to $(70.77 \pm 11.56)\%$, with accuracy falling as low as 48.9% on

slow-moving sessions. This represents the second largest impact on both accuracy and inference time (122.8 ± 13.6 ms), confirming that speed normalization is critical for handling the clinical population.

Dynamic Time Warping: Removing the DTW alignment step while retaining downsampling reduces accuracy to $(79.48 \pm 8.84)\%$. DTW produces the third largest impact on inference time (131.7 ± 14.6 ms), suggesting that the combination of downsampling and DTW is essential for full speed invariance, with each component contributing complementarily.

Different Ground Signal (S_g): When the reference signal S_g is replaced with any non-fast-moving session, accuracy drops to $(84.90 \pm 6.52)\%$ with no significant change in inference time. This confirms that S_g must always be a fast-moving session to provide a meaningful temporal reference for downsampling.

Unweighted Ensemble: Replacing the validation F1-score-weighted multinomial logistic regression ensemble with an unweighted variant reduces accuracy to $(76.89 \pm 4.50)\%$, with the lowest observed accuracy being 67.4%. This confirms the importance of the performance-aware weighting scheme in counteracting link-specific sensing biases.

Majority Voting Window Size: Reducing the majority voting window from K=100 to K=50 lowers accuracy to $(87.16 \pm 4.08)\%$, while increasing it to K=150 yields $(90.38 \pm 4.65)\%$ – slightly lower than the optimal K=100. The marginal degradation at K=150 is attributed to boundary overlap with adjacent actions when the voting window spans more than the typical 8-second exercise duration. K=100 is therefore the optimal setting.

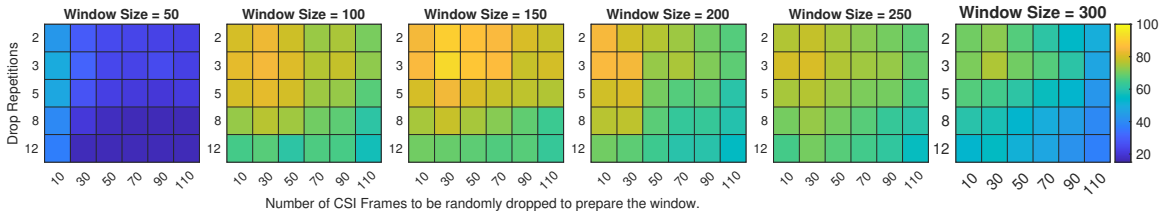


Fig. 26.: Classification accuracy as a function of CSI window size (50–300 frames), drop-frame count (10–110 frames per drop), and drop repetition count (2–12 times). The optimal configuration uses a window of 100–200 frames, 30 drop frames, and 3 drop repetitions, yielding 92.49% validation accuracy.

3.5.5.6 Data Augmentation Hyperparameter Study

Fig. 26 shows the classification accuracy on the validation set as a function of CSI window size, drop-frame count, and drop repetition count. Windows below 50 frames lose feature completeness as they may not cover the full duration of a single movement repetition. Conversely, windows larger than 200 frames unnecessarily include context from adjacent time steps. The zone of 100–200 CSI frames provides the best accuracy across all hyperparameter settings. Within a window, dropping 30 frames repeated 3 times yields the optimal classification accuracy of 92.49%, indicating that leniently sub-temporal views of the CSI window (rather than dense or highly sparse views) are most effective for speed-invariant generalization.

3.5.5.7 Clinical Trials

The system was evaluated on 11 clinical participants, comprising 7 patients with Parkinson's disease and 4 with post-stroke motor impairments, who were not involved in any stage of model development. As observed during data collection, patients with PD exhibited hand tremors and occasionally had difficulty locating the target dish, while stroke patients sometimes dropped the affected hand between dishes.

These behaviors introduce irregular CSI perturbations that are not present in healthy volunteer data. To bridge the speed gap, slow-moving healthy volunteer datasets were incorporated into the training set alongside the original training data.

The system achieved an average classification accuracy of $(85.9 \pm 3.0)\%$ across all 11 clinical datasets, with a minimum of 79.5% and a maximum of 90.6%. The higher inference time for clinical datasets (296.0 ± 51.3 ms vs. 188.0 ms for volunteers) reflects the additional computational cost of the speed-invariant downsampling and DTW alignment steps, which are invoked more frequently and with larger resampling factors for slow-moving patients.

3.5.5.8 Real-Time Performance

The proposed activity recognition pipeline has been implemented and evaluated under strict real-time constraints on a single Raspberry Pi 4B with 2 GB LPDDR4 RAM and a Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC running at 1.5 GHz, with Bullseye (Legacy) 32-bit Raspberry Pi OS. For a single sliding window of 150 CSI frames, each of the four trained CNN classifiers achieves an average inference latency of 2.1 ms. The end-to-end pipeline latency across all 51 datasets – including preprocessing, speed-invariant scaling, DTW alignment, CNN inference, ensemble fusion, and majority voting – averages 212.5 ± 51.1 ms, corresponding to approximately four predictions per second. Clinical datasets incur a higher average latency of 296.0 ± 51.3 ms due to more intensive speed-invariant processing. The system requires an initial 2.8 seconds of data collection before the first majority voting prediction is emitted; thereafter, predictions are produced continuously at the above rate. These results confirm that the system satisfies real-time inference requirements on commodity edge hardware for activities spanning more than four seconds.

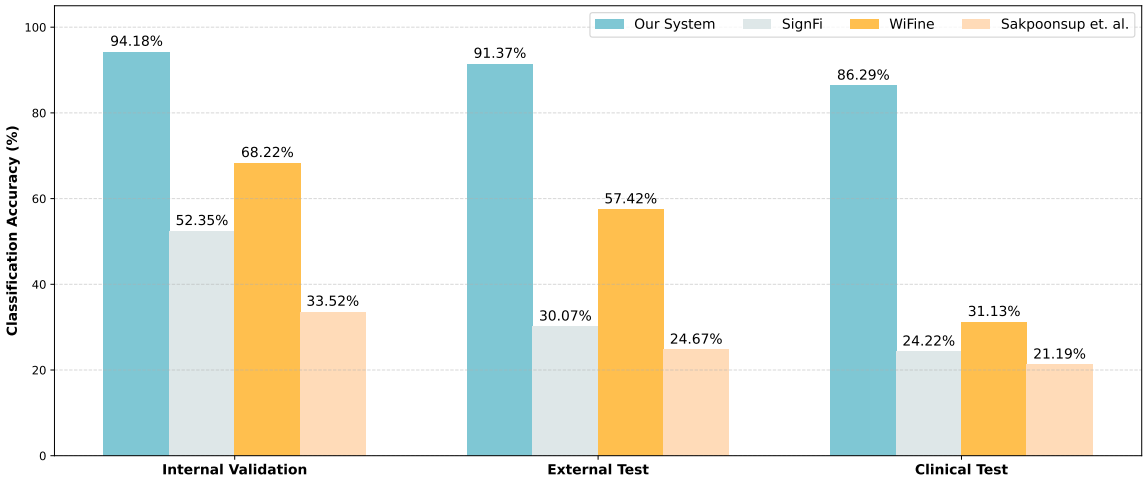


Fig. 27.: Comparative classification accuracy across internal validation, external test, and clinical patient cohort datasets. The proposed system substantially outperforms all three baselines in every evaluation split, with the largest gap observed in the clinical cohort.

3.5.5.9 Comparison with State-of-the-Art Baselines

To benchmark the proposed framework, replicated implementations of three state-of-the-art WiFi sensing architectures were evaluated: SignFi [196], WiFine [155], and Sakpoonsup et al. [197]. Since public code repositories for WiFine and Sakpoonsup et al. are unavailable, these baselines were reconstructed entirely from the structural and algorithmic descriptions in their respective publications. To ensure a fair comparison, all replicated baselines were trained using data from each of the four ESP32 links separately and then fused using the same validation-weighted multinomial logistic regression ensemble and identical 100-window majority voting buffer as the proposed system.

As shown in Fig. 27, the proposed system substantially outperforms all three baselines across every evaluated cohort. In the internal validation set, the pro-

posed system achieves 94.18% accuracy, compared to 68.22%, 52.35%, and 33.52% for WiFine, SignFi, and Sakpoonsup et al., respectively. On the unseen external test set, the gap widens to 91.37% vs. 57.42%, 30.07%, and 24.67%. The degradation of baseline methods is most severe on the clinical cohort: while the proposed system maintains 86.29% accuracy, WiFine drops to 31.13%, SignFi to 24.22%, and Sakpoonsup et al. to 21.19%. This dramatic degradation of baselines on clinical data directly demonstrates their vulnerability to highly variable execution speeds, irregular movement trajectories, and physiological tremors inherent to PD and stroke populations—precisely the challenges addressed by the speed-invariant scaling and DTW alignment pipeline of the proposed system.

In terms of computational efficiency, all four frameworks satisfy the real-time inference requirements for edge deployment on a Raspberry Pi. The replicated SignFi and WiFine implementations achieve mean end-to-end latencies of 228.37 ± 101.24 ms and 203.56 ± 86.25 ms, respectively, while Sakpoonsup et al. requires 307.67 ± 79.32 ms due to its raw tensor processing structure. The proposed system's latency of 212.5 ± 51.1 ms is highly competitive, with a notably lower standard deviation, indicating more stable and predictable real-time performance.

3.5.6 Discussion

The experimental evaluations confirm the clinical and practical viability of the device-free, Faraday-enclosed WiFi sensing framework for real-time hand rehabilitation monitoring. By decoupling sensing from open-room environmental variability, the Faraday enclosure provides signal stability that is essential for fine-grained movement recognition in the presence of the irregular and slow motions characteristic of neurological disorders. The multi-link architecture addresses the fundamental spatial specialization of individual WiFi links by fusing their complementary LoS sensitivities

through a validation-weighted ensemble, yielding consistently high accuracy across all six movement directions.

3.5.6.1 Clinical Implications and Deployment Advantages

Unlike open-room WiFi sensing systems, the proposed enclosure guarantees a predictable signal-to-noise ratio (SNR) independent of the surrounding home environment, which is a prerequisite for clinical reliability. From a practical deployment perspective, the system offers two key advantages:

1. *Offline Resource Efficiency:* The system operates entirely offline on commodity edge hardware (ESP32 microcontrollers and a Raspberry Pi 4B), with zero cloud dependency. This eliminates both ongoing subscription costs and internet latency bottlenecks, enabling long-term home deployment without infrastructure requirements beyond a home power outlet.

2. *Air-Gap Network Security:* Because the entire inference pipeline runs locally and the system is functional with an air-gap network [198], it provides absolute data privacy and robust security against external cyber threats. This places it in compliance with strict medical data privacy standards (e.g., HIPAA), even during extended unsupervised home use.

3.5.6.2 Limitations and Future Directions

Despite these strengths, several limitations remain. The Faraday enclosure constrains the system to fine-grained hand and wrist movements, limiting scalability to gross motor tasks or full-body rehabilitation exercises. Additionally, extreme pauses or highly non-stationary tremors can still distort temporal segmentation in the most severe cases.

To transition this framework toward a production-ready at-home medical device, the following future directions are identified:

1. *On-Edge Adaptive Personalization:* Implementing lightweight on-chip online learning algorithms, such as streaming linear discriminant analysis or tiny federated updates, directly on the Raspberry Pi to dynamically adapt base classifiers to a patient's progressive physical recovery without retraining from scratch.

2. *Fail-Safe Multimodal Fusion:* Integrating a low-cost complementary sensing modality, such as a single-chip mmWave radar, as a fail-safe backup for highly irregular motion trajectories that challenge the WiFi-only pipeline.

3. *Longitudinal Unsupervised Trials:* Conducting long-term, unsupervised clinical trials in patient homes to rigorously evaluate the system's structural durability, software robustness under continuous use, and sensitivity to long-term micro-movements indicative of functional rehabilitation progress.

CHAPTER 4

DAILY ACTIVITY TRACKING IN SENIOR HOUSING ENVIRONMENTS

4.1 Introduction

The ability to perform Activities of Daily Living (ADL) independently is one of the critical needs for healthy aging. Loss of independence in older ages represents the transition from health to disability. Thus, older adults who lose the ability to ambulate or care for themselves are at a high risk of adverse health outcomes (e.g., falls) and decreased quality of life. Aging in place is difficult in particular for low-income older adults with multiple chronic conditions and disabilities, lack of transportation, and limited social capital. That is why it is essential to routinely monitor daily activities and mobility to capture early decline before a clinical symptom arises.

Traditional activity and mobility assessment tools are primarily self-report, subjective, and episodic. These assessments are prone to recall bias, especially among older adults with memory issues. Additionally, they do not capture variability over time, making it challenging to track a patient's decline in function. Recently, digital health technologies have been proposed to obtain objective, high-frequency, and remote monitoring. However, significant user challenges (e.g., loss of device, incorrect use) threaten the reliability of the data collected. Motion detection sensors in the context of in-home unobtrusive physical performance assessment show limited results as they could not detect different types of human behaviors (e.g., sitting, walking, eating, and leaving home). There is a need to develop and test new sensing technology that can characterize and quantify different types of daily activities in a real-world

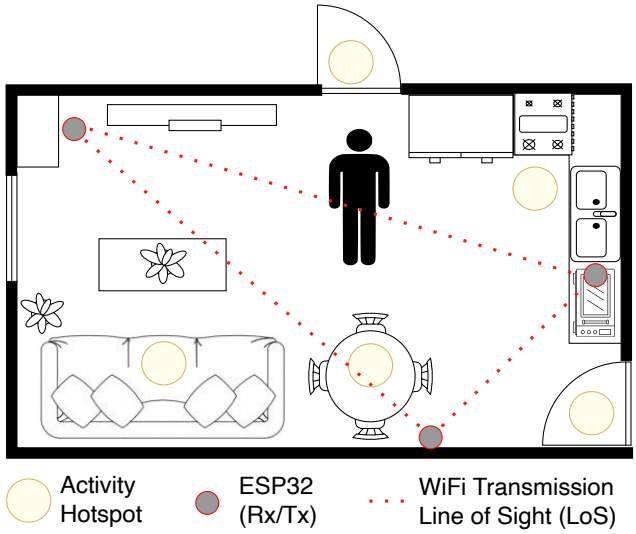


Fig. 28.: A typical deployment with potential activity hotspots in the living and kitchen area with sample WiFi RX/TX positions.

setting while also being discreet, affordable, and requiring minimal user engagement.

In this study, we propose to leverage the ubiquity of WiFi technology and low-cost devices that can transmit WiFi signals to monitor the daily activities of senior adults in a home environment (see Fig. 28 for a sample setup). This allows for an activity recognition opportunity without any wearables on the body of the person that is monitored, and even allows monitoring beyond visual line of sight (i.e., in multiple rooms of the house) as WiFi signals penetrate through the walls. This is achieved through the fine-grained analysis of WiFi signals (i.e., CSI on subcarriers), which reflect from individuals' bodies while propagating in the environment and thus carry information regarding their location and movements. Existing studies on WiFi sensing usually consider a limited and controlled/lab environment (e.g., a room) and focus on recognition of random activities happening in a non-natural environment (i.e., subjects are asked to perform the activities rather than doing them in their

daily routine). These studies also utilize specialized hardware (i.e., a laptop and an updated Network Interface Card [21]) to collect WiFi CSI data from the transmitter devices in the environment. Because these devices are costly and bulky, they are not amenable to scalable deployments. To address this, we recently developed an Internet of Things (IoT)-based standalone and lightweight solution to WiFi sensing which facilitates large-scale deployments [199].

Utilizing our IoT based solution that uses low-cost ESP32 microcontrollers, we deployed our WiFi sensing based system in the houses of eleven different older adults and evaluated its performance. In this study, we go through the steps we followed during deployment together with the challenges we faced. We also present our initial results from a subset of the data collected and discuss our further steps.

To the best of our knowledge, despite the variety of studies on WiFi sensing, there is no study that deploys a WiFi sensing-based system in a senior house setting for five to seven days and collects data in a natural setting, i.e., as the participants perform these activities in their daily routine, and provides its evaluation.

The rest of this chapter is organized as follows. Section 4.2 presents the details of our system setup, highlighting its features and functionalities. In Section 4.3, we go through the details of the deployment process together with the details of the participants involved. We then present the performance results of the WiFi sensing system in the deployed environments. Finally, we discuss our summary and potential future directions in Section 4.4.

4.2 Overview of Deployed WiFi Sensing System

We have developed a complete IoT infrastructure to deploy in several nursing home apartments for old people. As shown in Fig. 29, our system involves both offline data collection components and online (over the Internet) remote monitoring

components. The following subsections describe the functionalities of and challenges faced with these components.

4.2.1 Deployed System Components

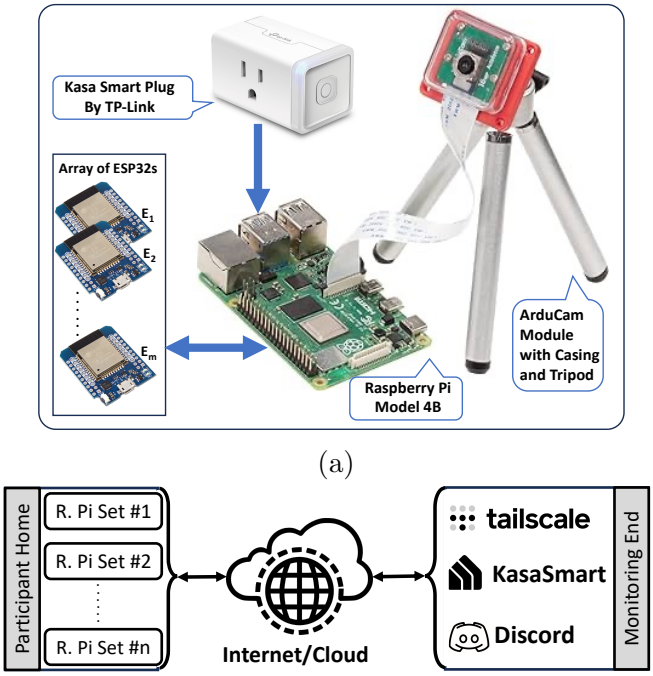
Depending on the floor maps of the participants, we deployed several Raspberry Pi 4B devices to collect WiFi CSI data from ESP32 WiFi receiver (Rx) microcontrollers and record activities using camera modules to prepare the ground truths.

4.2.1.1 Raspberry Pi

We used CanaKit package with Raspberry Pi 4B boards having 2GB RAM. The principal function of this mini-computer is to collect CSI data from all the connected ESP32 Rx devices using our developed CSI-Pi [195] server app. This server app is also equipped with capabilities like recording and post-processing videos.

4.2.1.2 Camera Module

We recorded the video of the participant performing several daily activities through a connected ArduCam module using the PiCamera [200] library. The camera was often supported by a tripod, while a few deployments required it to be taped on some higher ground to have a better view of the apartment. As per IRB protocols, we recorded the activities from 8 AM to 8 PM giving the participant a fully private night time. Consequently, we consider only these 12 hours of CSI data in our system.



(a)

(b)

Fig. 29.: (a) Single set of devices with Raspberry Pi (R. Pi) 4B, multiple ESP32 WiFi receivers, Kasa SmartPlug, and camera module, (b) Overall end-to-end monitoring system.

4.2.1.3 ESP32 Microcontrollers

We deployed multiple pairs of ESP32 WiFi receiver and transmitter. These were flashed with ESP32-CSI-Tool [199] firmware. We saved CSI data only from the receiver devices connected with Raspberry Pi. The transmitters were powered on at some far-reach corner of the apartment ensuring the line of sight of some observable activities.

4.2.2 Remote Live Monitoring and Health Check

We developed a fully connected ecosystem of multiple online components, i.e., a continuous monitoring system to let us know the status of the data collection process

and also to let us debug and even reset the system remotely.

4.2.2.1 Tailscale

Tailscale [192] is a partially open-source software-defined mesh virtual private network (VPN). Each of our Raspberry Pi devices was equipped with a Tailscale client app which made it accessible via SSH from our developer end in case of any discrepancy suspected in the data collection status. Tailscale ensures end-to-end encryption of any such remote connection using WireGuard [201] so that no man-in-middle attack can happen while we access any remote device.

4.2.2.2 Kasa Smart Plugs

Kasa Smart plug enables us to power on and off any connected IoT device remotely, even through the Internet. We powered almost all of our Raspberry Pi and ESP32 devices with these smart plugs. Each of these plugs was connected to the Internet and manageable through the Kasa smartphone app. Whenever we observed any device not working properly, we accessed it using Tailscale and tried to fix it first. If it did not work, we could restart the device by controlling its power through its connected smart plug.

4.2.2.3 Discord Webhook Messages

A summary of the collected CSI data and recorded videos were periodically sent to a webhook in the Discord messaging app to let all concerned personnel know the most recent status of the data collection process. We could analyze and proactively detect any failing ESP32, failing camera, power outage, etc. occurrences in the deployed home by looking into a few kilobytes of sent messages in the Discord app from time to time.

4.2.2.4 5G WiFi Hotspot

All of our Raspberry Pi and smart plug devices were connected to the Internet through one or more Verizon Orbic 5G WiFi hotspots. This was needed as there were no Internet available at the residences of participants.

4.2.3 Challenges and Issues Faced

We faced various practical issues in our live deployments which are not usual in a controlled lab environment. We gathered experiences through these issues as we solved those in consecutive deployments. However, some issues were found inevitable and costed us lose a significant amount of data.

4.2.3.1 Unexpected Person Attendance

WiFi sensing is susceptible to the presence of multiple people and we were to monitor the participant living alone in the apartment. However, we observed several occasions involving the presence of an outsider like a nurse, maintenance officials, visiting friends, our deployment team, and so on in the monitored environment. We needed to exclude those parts of the CSI data during data annotation looking into the recorded videos. Note that recent studies [202, 203] show that it can be possible to monitor multiple people's activities separately, however for the sake of our initial deployments we targeted single person based movements.

4.2.3.2 Device Heating

In one of our early deployments, we noticed our Raspberry Pi functioning extremely slow and barely was reachable. Investigation revealed that our used aluminum case of CanaKit was not good enough to release the device's heat and it was getting too hot to continue its operation at the usual speed. We then used only plastic cas-

ings with enough vents, thermal heat sinks, and a cooling fan installed inside for the consecutive deployments and never faced a similar issue.

4.2.3.3 Device Positioning

A few of the apartments were so challenging to cover with our camera and WiFi transmission LoS that we needed to tape our devices, i.e., camera and ESP32, onto the wall or high-standing types of furniture. While it might not be favorable to all the participants, it even left marks on the wall in a deployment. On several other occasions, either the taped camera or the camera with tripod or the Raspberry Pi was displaced due to activities around it. One positive takeaway of this type of issue is none of our ESP32 devices were displaced as those are comparatively lightweight and safely placed to maintain the same transmission line.

4.2.3.4 Pets

We attempted to recruit participants without pets; however, one participant owned two cats. She was asked to keep the cats in an unmonitored area of the home (i.e., the bedroom) during data collection. As this could not always be guaranteed, we carefully reviewed the corresponding recordings during annotation and excluded data segments in which the cats entered the monitored area.

4.2.3.5 Heaters and Fans in the Environment

Airflow from fans can cause random movements of lightweight materials in the environment. In addition, we observed a deteriorating data rate from one transmitter, which eventually stopped transmitting after a few days. We later discovered that the participant had placed a heater next to the transmitter. These issues caused us to lose a significant amount of data.

4.2.3.6 Maintenance Emergency

In one of our deployments, the participant had a water leak from her roof to the side wall. One of our smart plugs was soaked and fused in the leaked water, which made the attached Raspberry Pi power off. We noticed the device missing its periodic Discord messages while not reachable through Tailscale and found out about this issue after contacting the participant. We admit that this kind of maintenance emergency can cause havoc in this type of live deployment.

4.2.4 Post-processing

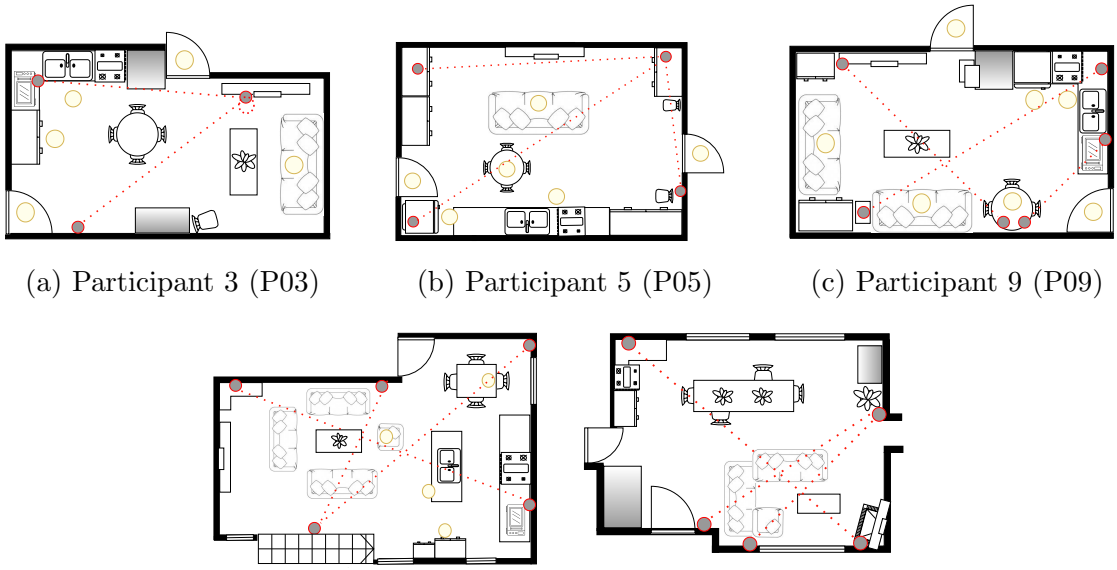
After collecting the data, a time-consuming step was to look into all the recorded videos and annotate the interesting activities to train our WiFi sensing ML model for activity classification. We recorded the videos with online synchronized timestamps embedded on top of each frame so that we could record the actual millisecond accurate timestamp of any activity. This manual post-processing step got particularly tougher with more camera angles recording movements simultaneously.

Additionally, the volume of the collected CSI data and recorded video files was over a hundred gigabytes per participant. With more participants deployed and more data accumulated, processing this volume of data became more challenging.

4.3 Experiments

4.3.1 Participants Information and IRB

We deployed our system and collected data from eleven participants after being approved by the VCU Institutional Review Board (IRB). Among them, there is one African-American man, two American white men, five African-American women, and three white women. Their ages range from 55 to 77, with an average of 67.3. We



(a) Participant 3 (P03) (b) Participant 5 (P05) (c) Participant 9 (P09)

(d) PD Participant 1 (PD1) (e) PD Participant 2 (PD2)

Fig. 30.: A few floor-maps of the deployed apartments (a, b, and c as Group-1) and home (d and e, as Group-2) environments.

anonymously name the participants as P03, P04, P05, P06, P07, P08, P09, P10, PD1, PD2, and PD3. The first seven participants, P03 to P09, have a one-bed and one-bath apartment layout, and we monitored only their living area, which also accommodates their kitchen and dining area. Participant P10 had the kitchen area well separated from the living area. The other three deployments, i.e., PD1, PD2, and PD3, were done in bigger houses having multiple floors and several entrances to the monitored living room area. For this reason, we group the ADLs into two groups: Group-1 with six ADLs for seven participants P03 to P09 and Group-2 with seven ADLs for the other four participants, i.e., P10, PD1, PD2, and PD3. A few of their floor maps are shown in Fig. 30 with sensor placements, which are done mainly based on camera angle visibility among the furniture positions, power outlets, and floor layout. Table 11 and 12 enlist the repetition counts per observable ADLs for Group-1 and Group-2,

Table 11.: The number of occurrences of the selected six ADLs of seven participants of Group-1 (apartment deployments).

Activity	Code	P03	P04	P05	P06	P07	P08	P09
Enter Apartment	enapt	16	27	33	18	20	15	11
Exit Apartment	exapt	27	65	36	18	21	16	13
Enter Private Area	enpri	48	28	71	43	15	81	59
Exit Private Area	expri	54	67	78	40	16	87	59
Kitchen Activity	kit	134	216	262	103	50	156	171
Fridge Open/Close	fri	66	12	63	74	19	164	108

Table 12.: The number of occurrences of the selected seven ADLs of four participants of Group-2 (home/isolated apartment deployments).

Activity	Code	P10	PD1	PD2	PD3
Enter Private Area 1	enpri1	26	7	30	50
Exit Private Area 1	expri1	27	5	29	51
Enter Private Area 2	enpri2	40	37	78	8
Exit Private Area 2	expri2	39	39	78	8
Sit on a Couch	sit	79	64	30	27
Stand up from a couch	stand	78	58	28	24
Walk around	walk	546	169	579	353

respectively.

4.3.2 Data Collection

During the experiments, we collect CSI data using WiFi-enabled ESP32 microcontrollers and the ESP32-CSI Toolkit [199]. Unlike other data collection methods that require a host laptop with an updated Network Interface Card, these microcontrollers offer a compact, cost-effective, and independent solution. The portability and versatility of the ESP32s facilitate easy deployment. In order to run the proposed solution in resource limited edge devices efficiently, we integrate solutions like online sampling of collected CSI data [204]. In our system, each transmitter sends data frames at $100Hz$ to its respective receiver. Multiple pairs of these RX and TX devices

are deployed based on the floor map of the apartment. The received CSI data at the RX ESP32 are exported to a file stored inside a Raspberry Pi 4B. Due to the multiple days-long deployment period, all video and CSI data stored grows beyond a hundred gigabytes in size and thereby makes it resource-consuming to transfer, annotate, and process for machine learning.

4.3.3 Preprocessing and ML Model Development Per Participant

The CSI data is preprocessed before being fed into the machine learning model for training. First of all, the loaded CSI data is cleaned of stale timestamps and broken packets. Then we remove static background clutter and denoise by rolling window averaging of the Hampel filter per subcarrier. Next, we use PCA to further denoise and reduce the dimensionality of the collected data. The per-WiFi link data is then fed to individually trained CNN models, out of which we compute a multinomial logistic regression ensemble and then apply a weighted majority voting scheme to get the final prediction, as depicted in Fig. 31.

4.3.3.1 Per-Day Static Background Clutter Removal

Raw CSI amplitude at subcarrier k and time frame t can be modeled as $A_k(t) = S_k(t) + C_k^{(d)} + \eta_k(t)$, where $S_k(t)$ is the motion-induced component, $C_k^{(d)}$ is a day-specific static clutter term arising from multipath reflections off stationary objects, and $\eta_k(t)$ is zero-mean measurement noise. Because $C_k^{(d)}$ dominates raw amplitudes and drifts across days due to temperature-induced hardware variation and incremental environmental changes, we estimate it from CSI frames collected during periods when no participant is present in the monitored space, which is identifiable as the intervals between annotated `exapt` and `enapt` events on each deployment day d.

Denoting this set of empty-scene frames as $\mathcal{F}_{\text{empty}}^{(d)}$, we compute the per-subcarrier

day-specific mean,

$\mu_k^{(d)} = |\mathcal{F}_{\text{empty}}^{(d)}|^{-1} \sum_{t \in \mathcal{F}_{\text{empty}}^{(d)}} A_k(t)$

and standard deviation,

$\sigma_k^{(d)} = \big((|\mathcal{F}_{\text{empty}}^{(d)}|-1)^{-1} \sum_{t \in \mathcal{F}_{\text{empty}}^{(d)}} (A_k(t) - \mu_k^{(d)})^2\big)^{1/2}$,

and replace every activity-frame amplitude with the clutter-normalised value:

$$\tilde{A}_k^{(d)}(t) \;=\; \frac{A_k(t) \;-\; \mu_k^{(d)}}{\sigma_k^{(d)} + \varepsilon}, \tag{4.1}$$

where $\varepsilon = 10^{-6}$ guards against division by zero on near-silent subcarriers. The result is a z-score relative to that day's empty-room distribution: static clutter is zeroed by construction, each subcarrier is variance-normalised so that it contributes equally to the subsequent PCA step, and any residual drift $\Delta C_k = \mu_k^{(d_2)} - \mu_k^{(d_1)}$ that would otherwise bias cross-day inference is eliminated. Equation (4.1) is applied after amplitude extraction and before Hampel denoising; the vectors $\boldsymbol{\mu}^{(d)}$ and $\boldsymbol{\sigma}^{(d)}$ are persisted alongside the PCA model for consistent application at inference time.

4.3.3.2 Denoising and Dimensionality Reduction

After per-day background normalisation, the clutter-removed amplitude matrix $\tilde{\mathbf{A}}^{(d)} \in \mathbb{R}^{T \times K}$ is denoised in two sequential steps before feature extraction.

Null subcarrier removal. Of the ESP32's 64 subcarriers, 12 are DC or pilot tones that carry no motion information (indices $\{0\text{–}5, 32, 59\text{–}63\}$), yielding $K' = 52$ informative subcarriers. Columns whose amplitude variance remains identically zero after denoising are additionally discarded.

Hampel outlier filter. Impulsive spikes caused by packet collisions or hardware glitches are removed independently on each subcarrier time series. For subcarrier k and frame index t, a symmetric window of half-width $h = \lfloor w/2 \rfloor$ (with $w = 10$ frames) is centred at t. The local median $m_k(t)$ and median absolute deviation $\text{MAD}_k(t) =$

$\kappa \cdot \text{median}_{j \in \mathcal{W}(t)} |\tilde{A}_k(j) - m_k(t)|$ are computed over the window $\mathcal{W}(t) = \{t-h, \ldots, t+h\}$, where $\kappa = 1.4826$ is the consistency factor for a Gaussian distribution. A sample is flagged as an outlier and replaced by the local median if

$$|\tilde{A}_k(t) - m_k(t)| \; > \; n_\sigma \cdot \text{MAD}_k(t), \qquad n_\sigma = 3, \tag{4.2}$$

leaving impulsive noise below the 3σ threshold undisturbed.

Double rolling mean. The Hampel-filtered signal is then smoothed by applying a causal rolling mean of window width w twice in succession. Denoting the output of pass r as $\mathbf{X}^{(r)}$, each frame is updated as

$$X_k^{(r)}(t) \; = \; \frac{1}{w} \sum_{j=0}^{w-1} X_k^{(r-1)}(t-j), \qquad r = 1, 2, \tag{4.3}$$

with $\mathbf{X}^{(0)}$ being the Hampel output. Two passes provide a steeper effective low-pass response than a single wider window while preserving sharper activity-onset edges.

PCA-based further denoising and dimensionality reduction. PCA is fitted on the training split only and applied to all splits. Retaining the top $P = 32$ principal components from the $K' = 52$ non-null subcarriers preserves $\sum_{i=1}^{32} \lambda_i \,/\, \sum_{i=1}^{52} \lambda_i$ of the total variance (where λ_i are the eigenvalues of the sample covariance matrix) while discarding residual low-variance noise directions. The projected feature matrix $\mathbf{Z} \in \mathbb{R}^{T \times 32}$ serves as input to all subsequent sliding-window and classification stages.

MockiFi Generated Synthetic Data. During training, we expand the variation of data by adding synthetic data produced by our MockiFi study [205], which is also detailed in the following chapter in this dissertation. As input to MockiFi, we provide the PCA-projected windows of one activity on one deployment day as a base, and learn the activity-to-activity transformation in the PCA component space from another day of the same participant, treating each recorded deployment

day as a structurally distinct environment in MockiFi's original formulation. Concretely, let $d_1, d_2, \ldots, d_Z$ denote the ordered deployment days of a participant, and let $a_1, a_2, \ldots, a_K$ denote the target ADLs. For each consecutive day pair (d_x, d_{x+1}) indexed circularly (so that d_Z wraps back to d_1), d_x takes the role of the known person p_x in MockiFi's formulation providing PCA windows for both a source activity a_i and a target activity a_j, while d_{x+1} takes the role of the new person p_y, for which only the PCA windows of base activity a_i are supplied. The CNP model then synthesizes d_{x+1}'s PCA-space fingerprint of a_j by transferring the $a_i \rightarrow a_j$ transformation learned from d_x:

$$\widehat{\mathbf{Z}}_{d_{x+1},\,a_j} \;\approx\; \mathbf{Z}_{d_x,\,a_j} \;\diamond\; \mathbf{Z}_{d_x,\,a_i} \;\diamond'\; \mathbf{Z}_{d_{x+1},\,a_i}, \tag{4.4}$$

where $\mathbf{Z}_{d,\,a} \in \mathbb{R}^{w \times P}$ denotes the matrix of $P = 32$ principal component values over a window of $w = 150$ frames for activity a on day d, and $\diamond$, $\diamond'$ follow the transformation relation of [205]. This round-robin process is repeated for all K activity pairs, with a_i rotating across activities so that each ADL serves as the base at least once per participant per day. The synthesized PCA windows are appended directly to the training set of each link's CNN, downstream of PCA projection and upstream of the sliding-window stage, so that no additional preprocessing is required for the generated data. By treating each deployment day as a distinct environment rather than a distinct person, this approach serves two complementary purposes: (1) it normalises the temporal drift of the principal components that accumulates over the week-long deployment, which is a known challenge in long-duration real-world sensing [206], and (2) it implicitly trains each CNN to recognise the same activity across day-to-day variation in the PCA space, which structurally resembles the cross-person variation that the SupCon model addresses at the multi-participant level.

4.3.3.3 Base CNN Model Training Per WiFi Link

We train a separate CNN model M_m for each of the N WiFi links independently, using only that link's CSI amplitude windows as input. Formally, for the m-th link, M_m maps a CSI window $\mathbf{x} \in \mathbb{R}^{w \times P}$ (with $w = 150$ frames and $P = 32$ principal components) to a probability distribution over C activity classes:

$$\mathbf{p}^{(m)}(\mathbf{x}) = M_m(\mathbf{x}) \ \in \ \Delta^{C-1}, \tag{4.5}$$

where Δ^{C-1} denotes the $(C-1)$-simplex. Each M_m is trained end-to-end by minimising the categorical cross-entropy loss,

$$\mathcal{L}_m = -\frac{1}{|\mathcal{D}_m|} \sum_{(\mathbf{x},y)\in\mathcal{D}_m} \log p_y^{(m)}(\mathbf{x}), \tag{4.6}$$

where $p_y^{(m)}$ is the predicted probability for the true class y. The architecture processes the temporal axis of each window with a `TimeDistributed`-wrapped Conv1D block: the first layer applies 64 filters with a kernel of size 3, followed by batch normalisation and ReLU activation; the second layer applies 128 filters with a kernel of size 5, again followed by batch normalisation and ReLU. The resulting feature maps are flattened and passed through three fully connected layers of 256, 128, and 64 hidden units with ReLU activations and dropout layers interleaved between them, before the final softmax output.

4.3.3.4 F1-Weighted Stacking Ensemble

Ensemble learning [207] enhances predictive performance by combining the complementary strengths of multiple base learners [55]. Because each WiFi link covers a different spatial viewpoint of the monitored area, the N base models $\{M_m\}_{m=1}^{N}$ are individually strong at recognising different subsets of activities. We exploit this diversity

through a two-stage aggregation strategy that (i) weights each model's class-specific predictions by its validated competence and (ii) learns an optimal linear combination via multinomial logistic regression.

Stage 1: Per-class F1 weighting. After training, each base model M_m is evaluated on the held-out validation set $\mathcal{D}_{\text{val}}$. Let $f_m^{(c)}$ denote the per-class F1 score of M_m for class c. We form a weight vector $\mathbf{w}_m \in \mathbb{R}^C$ by applying softmax to these scores:

$$w_m^{(c)} = \frac{\exp\left(f_m^{(c)}\right)}{\sum_{c'=1}^{C} \exp\left(f_m^{(c')}\right)}, \qquad c = 1, \ldots, C. \tag{4.7}$$

Softmax is preferred over linear normalisation because it amplifies the contrast between well-performing and poorly-performing classes, causing the ensemble to rely more heavily on a link's prediction precisely in the classes where that link is most accurate. The weighted prediction of M_m for any input $\mathbf{x}$ is then

$$\tilde{\mathbf{p}}^{(m)}(\mathbf{x}) = \mathbf{p}^{(m)}(\mathbf{x}) \odot \mathbf{w}_m, \tag{4.8}$$

where $\odot$ denotes element-wise multiplication.

Stage 2: Logistic regression meta-learner. For every sample $\mathbf{x}$ in the validation set, the weighted predictions of all N links are concatenated into an aggregated feature vector:

$$\boldsymbol{\phi}(\mathbf{x}) = \left[\, \tilde{\mathbf{p}}^{(1)}(\mathbf{x}) \parallel \tilde{\mathbf{p}}^{(2)}(\mathbf{x}) \parallel \cdots \parallel \tilde{\mathbf{p}}^{(N)}(\mathbf{x}) \,\right] \in \mathbb{R}^{N \cdot C}. \tag{4.9}$$

A multinomial logistic regression model $\mathcal{LR}$ is then trained on the set $\{(\boldsymbol{\phi}(\mathbf{x}),\, y)\}_{(\mathbf{x},y)\in\mathcal{D}_{\text{val}}}$, learning the optimal linear combination of per-link, per-class confidence scores. For any input $\mathbf{x}$, we define the ensemble probability vector as the average of the N

weighted base predictions and the logistic regression output:

$$\mathbf{q}(\mathbf{x}) = \frac{1}{N+1}\left(\sum_{m=1}^{N} \tilde{\mathbf{p}}^{(m)}(\mathbf{x}) \;+\; \mathcal{LR}(\boldsymbol{\phi}(\mathbf{x}))\right) \;\in\; \Delta^{C-1}. \tag{4.10}$$

Weighted majority voting over a sliding prediction buffer. A single-window classification is susceptible to brief signal artifacts and activity-transition ambiguity at the boundary frames of a window. To smooth the final decision, we maintain a rolling buffer of the $L = 50$ most recent per-window ensemble outputs and perform a *confidence-weighted majority vote* (WMV). At sampling rate $f_s = 100\,\text{Hz}$ with window stride $s = 1$ frame, a new prediction is emitted every $s/f_s = 0.01\,\text{s}$. The L-window buffer therefore covers a combined temporal span of

$$T_{\text{buf}} = \frac{(L-1)\cdot s + w}{f_s} = \frac{49 \times 1 + 150}{100} = 1.99\,\text{s}, \tag{4.11}$$

which is well-matched to the typical duration of each ADL (1.5–3 s). Let $\tau \in \{0, s, 2s, \ldots, (L-1)s\}$ index the stride offsets of the buffered windows and let $\mathbf{x}_{t,\tau}$ denote the CSI window whose most recent frame is τ frames before the current frame t. Each buffered prediction carries a confidence weight equal to the maximum class probability produced by the ensemble:

$$v_{t,\tau} = \max_{c}\; q^{(c)}(\mathbf{x}_{t,\tau})\,. \tag{4.12}$$

The weighted vote accumulated for class c over the buffer is

$$V_t^{(c)} = \sum_{\tau=0}^{(L-1)s} v_{t,\tau} \;\cdot\; \mathbf{1}\Big[\arg\max_{c'} q^{(c')}(\mathbf{x}_{t,\tau}) = c\Big]\,, \tag{4.13}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Intuitively, windows whose ensemble output is sharply peaked (high $v_{t,\tau}$, indicating an unambiguous activity signature) contribute more strongly to the final decision than borderline windows near activity transitions.

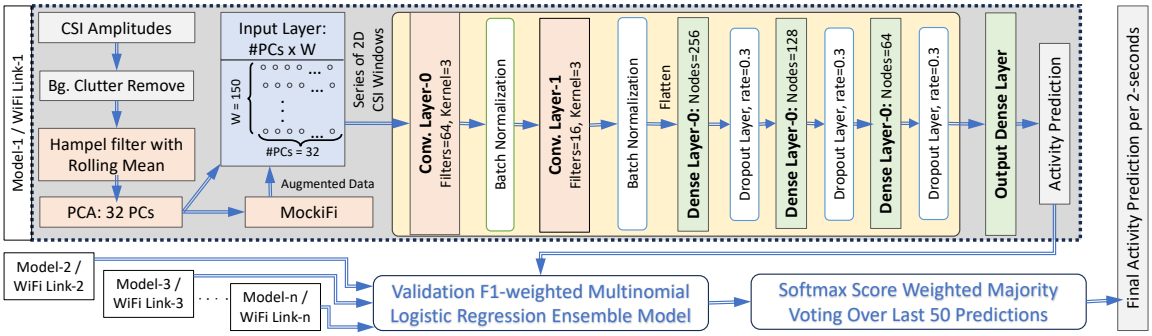


Fig. 31.: Data flow through preprocessing steps, multiple CNN models, the Multinomial Logistic Regression Ensemble model, and then the weighted Majority Voting scheme for activity prediction.

Inference. The final activity label at time t is the class that accumulates the greatest weighted support across the buffer:

$$\hat{y}_t^* = \arg\max_c \; V_t^{(c)}. \tag{4.14}$$

Because $V_t^{(c)}$ is updated by sliding the buffer forward by one stride step ($s = 1$ frame $= 0.01\,\text{s}$), a refreshed activity estimate is available at $100\,\text{Hz}$, while the effective decision horizon of $1.99\,\text{s}$ (Eq. (4.11)) provides sufficient temporal context to resolve activity identity robustly.

4.3.4 Cross-Person and Cross-Environment Generalization via Supervised Contrastive Learning

4.3.4.1 Motivation

The per-person ensemble model described above is trained and evaluated within a single participant's apartment. Deploying a separate model for every new resident, however, requires weeks of annotated data collection per deployment. The core obstacle is that raw CSI amplitudes reflect the geometry of the specific room as much as

the activity being performed, i.e., the same activity produces systematically different amplitude patterns in different homes. We address this by training a shared encoder across all available participants so that its output space is organized by activity class rather than by environment or individual. Adapting to an entirely new participant then requires only a small labeled sample, collected during a brief initial setup visit.

4.3.4.2 Encoder Architecture

Let $f_\theta : \mathbb{R}^{w \times P} \to \mathbb{R}^d$ be a convolutional encoder that maps a CSI window of $w = 150$ frames and $P = 32$ principal components to a d-dimensional embedding. The encoder consists of three Conv1D blocks with filter counts $\{64, 128, 256\}$, kernel sizes $\{7, 5, 3\}$, batch normalization, and ReLU activations, followed by global average pooling. During contrastive training, a two-layer projection head $g_\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$ (with $d' = 64$) maps the encoder output to the space in which the loss is computed. The projection head is discarded after training; all downstream operations use the raw encoder output $f_\theta(\mathbf{x})$. Both f_θ and g_ϕ are initialized randomly and trained end-to-end.

4.3.4.3 Training Pool Construction

Let $\mathcal{P}_{\text{train}} = \{P_1, P_2, \ldots, P_R\}$ denote the set of R training participants. Each participant P operates several WiFi links $\mathcal{W}_p = \{W_1, W_2, \ldots, W_{N_p}\}$, where N_p is typically three (for example, links placed in the kitchen, living room, and entrance area of the apartment). Every link carries a spatial tag $\text{tag}(W) \in \{\texttt{kitchen}, \texttt{living}, \texttt{entrance}, \ldots\}$ that was assigned at deployment time. For participant P, link W, and activity class c, let $\mathcal{D}_{p,W}^{(c)}$ be the complete collection of annotated CSI windows extracted across all recorded sessions. The full training pool uses every available window from every link

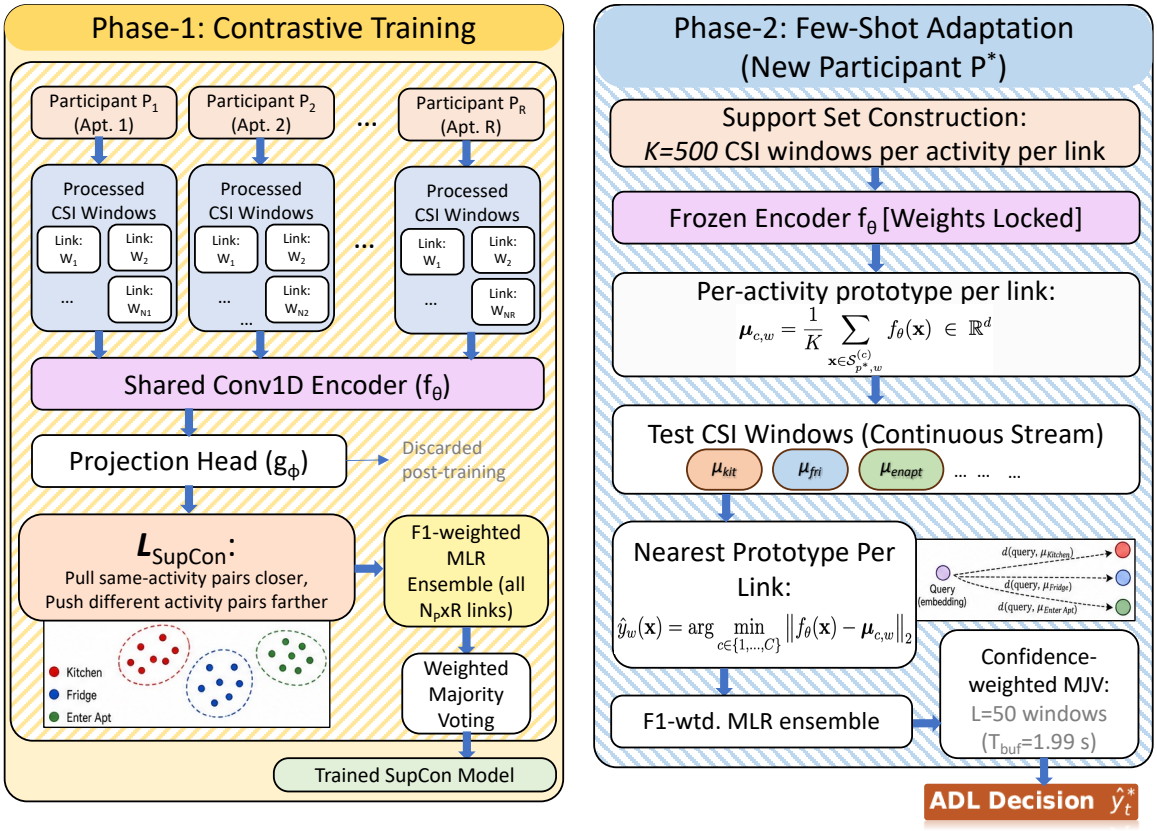


Fig. 32.: Training (left) and few-shot inference (right) of the proposed supervised contrastive learning pipeline. During training, all annotated windows from every WiFi link of every training participant feed a shared Conv1D encoder; the SupCon loss is computed using same-activity, same-tag positive pairs drawn across apartments, and training performance is measured through the F1-weighted multi-link ensemble followed by weighted majority voting. At inference, a new participant provides $K = 500$ support windows per activity per link, from which per-activity prototypes are computed; incoming test windows are classified by nearest-prototype assignment and smoothed by the confidence-weighted majority voting buffer of $L = 50$ CSI windows, i.e., $T_{buff} = 1.99$ *seconds* continuous activity duration.

of every training participant, without any down-sampling:

$$\mathcal{T} = \bigcup_{p \in \mathcal{P}_{\text{train}}} \bigcup_{W \in \mathcal{W}_p} \bigcup_{c=1}^{C} \{(\mathbf{x},\, c,\, W) \mid \mathbf{x} \in \mathcal{D}_{p,W}^{(c)}\}. \tag{4.15}$$

Each sample in $\mathcal{T}$ is tagged with its activity label c and its link identifier W (which carries the spatial tag). The participant identity is intentionally withheld as a label, so the encoder receives no signal that would cause it to cluster embeddings by environment rather than by activity.

4.3.4.4 Multi-Link Supervised Contrastive Training

Because each apartment contains N_p simultaneously operating WiFi links, a single batch window $\mathbf{x}$ from link W of participant p cannot stand alone as the unit of analysis. Instead, the links are used cooperatively in two ways: through spatial-tag matching for the SupCon loss, and through F1-weighted ensemble combination for the classification objectives.

Spatial-tag matching: Links across different apartments that share the same spatial tag, for example, two kitchen links from two different residents, capture broadly similar zones of domestic activity even though the exact floor-plan geometry differs. When sampling a mini-batch from $\mathcal{T}$, windows from same-tagged links across participants are eligible to serve as positive pairs in the contrastive loss (subject to also sharing the same activity class c). This choice encourages the encoder to learn spatial-context-invariant representations of each activity.

Contrastive loss: For a mini-batch of B windows $\{(\mathbf{x}_i, c_i, w_i)\}_{i=1}^{B}$ drawn from $\mathcal{T}$, let

$$\mathbf{z}_i = \frac{g_\phi(f_\theta(\mathbf{x}_i))}{\|g_\phi(f_\theta(\mathbf{x}_i))\|_2} \in \mathbb{S}^{d'-1} \tag{4.16}$$

be the normalized projection of window i. The positive index set for anchor i is

$$\mathcal{P}(i) = \big\{ j \neq i \;\big|\; c_j = c_i \;\text{ and }\; \text{tag}(W_j) = \text{tag}(W_i) \big\}, \tag{4.17}$$

i.e., all other windows in the batch that share the same activity class and the same spatial tag, regardless of which participant or apartment they originate from. The Supervised Contrastive (SupCon) loss [208] over this batch is then

$$\mathcal{L}_{\text{SupCon}} = \sum_{i=1}^{B} \frac{-1}{|\mathcal{P}(i)|} \sum_{j \in \mathcal{P}(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j \,/\, \tau)}{\sum_{k \neq i} \exp(\mathbf{z}_i \cdot \mathbf{z}_k \,/\, \tau)}, \tag{4.18}$$

where $\tau > 0$ is a temperature hyperparameter and the denominator treats every window from a different class (or a different spatial tag) as a negative. Minimizing $\mathcal{L}_{\text{SupCon}}$ pulls together embeddings of the same activity recorded in equivalent spatial zones of different apartments, while pushing apart embeddings of distinct activities, yielding a representation space organized by motion pattern rather than by room geometry.

F1-weighted multi-link ensemble and training metrics: After the encoder reaches a stable state, the per-link CNN classifier heads $\{h_w\}$ (each producing a softmax probability vector over C activities) are evaluated on a held-out validation fold. Let $F_w^{(c)}$ be the per-class F1 score achieved by the classifier of link W on the validation data. The F1-weighted multinomial logistic regression (MLR) ensemble described in Section 4.3.3.4 combines all $N_p \times R$ link classifiers from all training participants into a single probability estimate $\mathbf{q}(\mathbf{x})$. Confidence-weighted majority voting (Eq. 4.13) is then applied over a buffer of $L = 50$ consecutive windows to obtain the smoothed activity decision. This full pipeline constitutes the training classification evaluation; the reported accuracy, loss, and F1 scores during training are measured at the output of the WMV buffer, not at the raw encoder embeddings.

4.3.4.5 Prototype-Based Few-Shot Adaptation

Once contrastive training is complete, the encoder f_θ is frozen. Adapting to a new participant p^* whose apartment was not seen during training requires only a small labeled support set, collected by asking the participant to perform each of the C ADLs a small number of times during the initial system setup visit.

Support set construction. For each activity c and each link $W \in \mathcal{L}_{p^*}$, the support set is formed by drawing $K = 500$ windows uniformly at random without replacement from the full set of annotated windows $\mathcal{D}^{(c)}_{p^*,W}$ collected across all available deployment days (typically five to nine days):

$$\mathcal{S}^{(c)}_{p^*,w} \;\sim\; \text{Uniform}\Big(\mathcal{D}^{(c)}_{p^*,w},\; K\Big)\,, \qquad \big|\mathcal{S}^{(c)}_{p^*,w}\big| = K = 500. \tag{4.19}$$

Sampling from the full deployment span rather than from a single recording session ensures that the support set captures the natural intra-activity variability of the new participant across different times of day and different environmental conditions.

Per-link prototypes: For each link w and activity c, the class prototype is the mean encoder output over the support windows:

$$\boldsymbol{\mu}_{c,w} = \frac{1}{K} \sum_{\mathbf{x} \in \mathcal{S}^{(c)}_{p^*,w}} f_\theta(\mathbf{x}) \;\in\; \mathbb{R}^d. \tag{4.20}$$

This yields $C \times N_{p^*}$ prototype vectors, one per (activity, link) pair.

Classification: At inference time, each incoming test window $\mathbf{x}$ from link W is assigned a per-link label by nearest-prototype distance:

$$\hat{y}_w(\mathbf{x}) = \arg\min_{c \in \{1,\ldots,C\}} \big\| f_\theta(\mathbf{x}) - \boldsymbol{\mu}_{c,w} \big\|_2 \,. \tag{4.21}$$

The per-link labels $\{\hat{y}_w(\mathbf{x})\}_w$ are then combined by the F1-weighted MLR ensemble (weights derived from the validation F1 scores of the training participants' corre-

sponding spatial-tag-matched links), producing a fused probability vector $\mathbf{q}(\mathbf{x})$ over the C activity classes. The confidence weight for the WMV buffer is

$$v(\mathbf{x}) = 1 - \frac{\min_w \left\| f_\theta(\mathbf{x}) - \boldsymbol{\mu}_{\hat{y}_w,w} \right\|_2}{\max_w \max_c \left\| f_\theta(\mathbf{x}) - \boldsymbol{\mu}_{c,w} \right\|_2}, \tag{4.22}$$

which assigns higher weight to windows that land close to any prototype relative to the most distant class prototype across all links. The per-window label and weight ($\hat{y}(\mathbf{x})$, $v(\mathbf{x})$) then enter the weighted majority voting buffer of Eq. (4.13), preserving the temporal smoothing behavior described in Section 4.3.3.4.

4.3.5 Evaluation Results

We evaluate the system across eleven participants divided into two groups. Group 1 comprises seven participants (P03–P09) deployed in senior housing apartments, while Group 2 comprises four participants (P10, PD1, PD2, PD3) from deployments in different residential settings and have a different set of ADLs. Group 1 (G1) participants performed six ADLs and Group 2 (G2) participants performed seven different ADLs. Each deployment used two or three ESP32 RX-TX pairs (WiFi links), labeled W1, W2, and W3. We report individual per-link CNN accuracy, ensemble accuracy (Ens.), and confidence-weighted majority voting accuracy (MJV) using a buffer of $L = 50$ windows with stride $s = 1$ frame, corresponding to a 1.99 s decision horizon. A dash (–) in any cell indicates that the corresponding WiFi link was not needed or malfunctioned for that participant.

4.3.5.1 Per-Person ADL Classification

Tables 13 and 14 report per-person model results for Group 1 and Group 2, respectively. Fig. 33 compares the MJV accuracy across all eleven participants.

For Group 1, individual WiFi link accuracies range from 37.53% to 82.35%,

Table 13.: Per-person model results for Group 1 (apartment deployments, P03–P09).

Participant	**W1**	**W2**	**W3**	**Ens.**	**MJV**
P03	78.53	68.91	71.46	81.24	85.32
P04	77.42	43.56	71.34	80.88	84.44
P05	79.31	59.72	74.89	86.37	**90.17**
P06	70.74	37.53	69.91	79.56	82.53
P07	68.46	65.35	–	73.59	80.71
P08	82.35	79.25	73.47	87.89	**91.63**
P09	71.32	73.94	61.73	80.30	85.11
Mean					*85.70*

Table 14.: Per-person model results for Group 2 (home/isolated apartment deployment for P10, PD1, PD2, and PD3).

Participant	**W1**	**W2**	**W3**	**Ens.**	**MJV**
P10	41.14	56.22	59.71	67.35	74.63
PD1	57.11	–	63.56	76.77	82.31
PD2	60.32	42.44	65.44	78.42	84.77
PD3	59.63	62.47	67.88	79.29	**85.91**
Mean					*81.91*

confirming that no single link dominates across all participants or activities, and that ensemble combination is essential. The MJV step further raises accuracy by 4–8 percentage points above the raw ensemble in every case. The mean Group 1 MJV of 85.70% represents a marked improvement over the preliminary GB-ensemble results reported in our earlier work [206], where the best per-person accuracy among the three originally evaluated participants was 76.90% (P05). This gain is attributed to the updated Conv1D architecture, the F1-weighted logistic regression meta-learner, and the per-day background clutter normalisation introduced in this work. Group 2 participants achieve a mean MJV of 81.91%, with P10 being the lowest at 74.63%, which corresponds to a comparatively smaller annotation pool for that participant as presented in Table 12. The slightly lower accuracy of this group reflects the added complexity of seven ADLs relative to Group-1's six, including the spatially dynamic

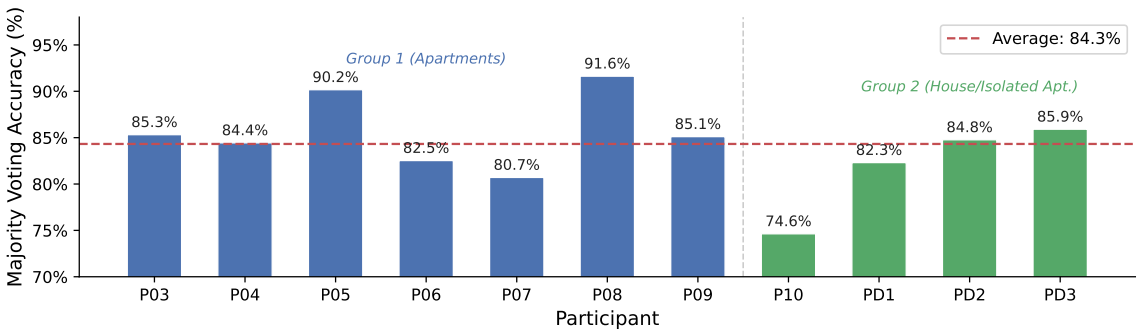
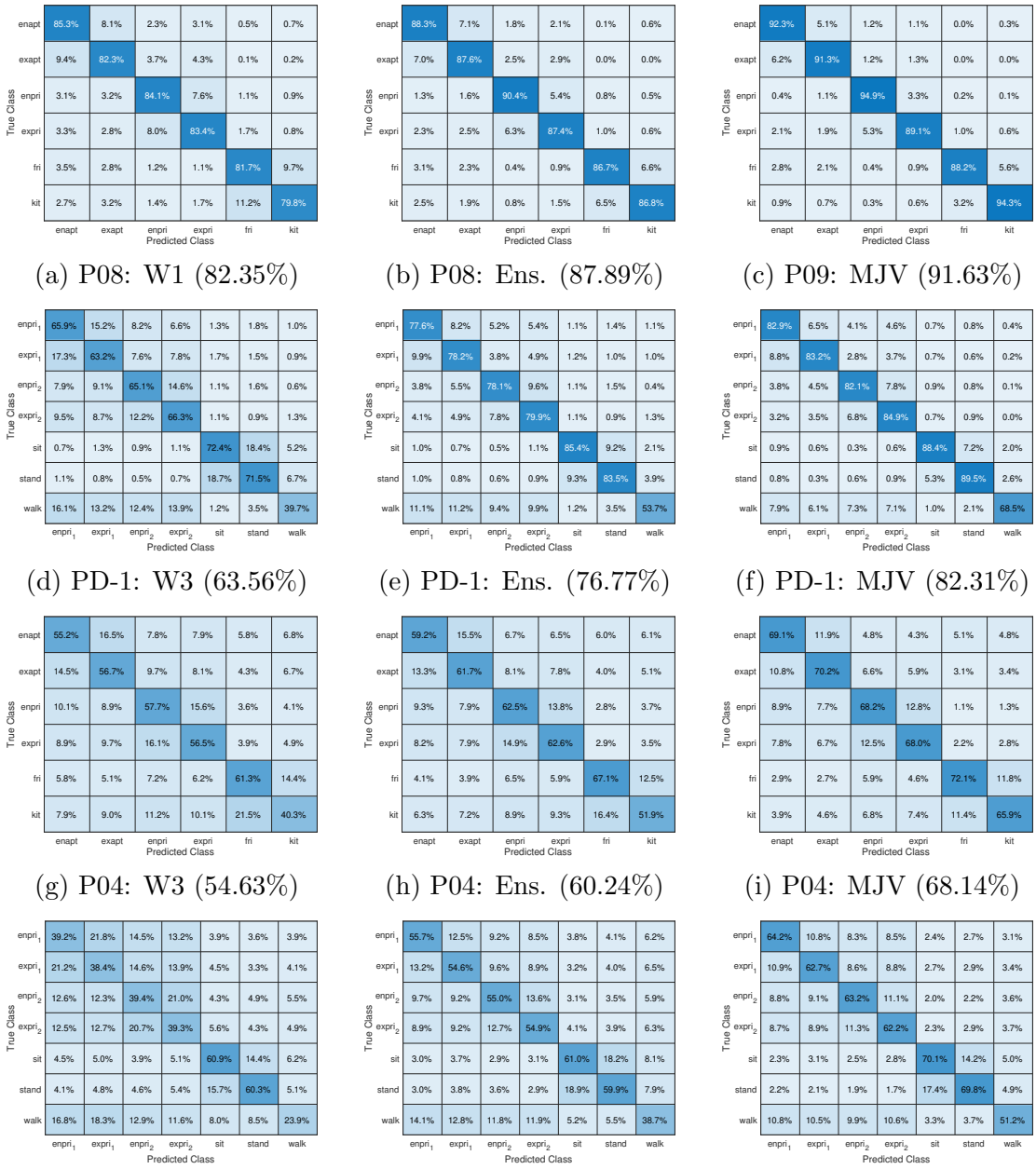


Fig. 33.: Majority voting accuracy for all eleven participants under the per-person trained model. Blue bars: Group 1 apartment deployments (P03–P09). Green bars: Group 2 pilot deployment participants (P10, PD1, PD2, and PD3). The dashed red line marks the overall mean across all participants.

`walk` activity and wider, multi-room floor layouts.

The confusion matrices presented in Fig. 34 show how each of the ensemble and weighted majority voting techniques improves the single-link-based models to recognize several activities in both of the groups. For Group-1, the `fri` and `kit` activities occur in a very close location, which makes them the most confusing ones, while the corresponding enter and exit actions are mostly confused among themselves in both groups. Similarly, in Group-2, `sit` and `stand` actions also happen in the same location and thereby make them confuse each other, even though they are the most recognizable activities out of the seven actions in Group-2. Another observation in confusion matrices for Group-2 (i.e., PD-1) is that `walk` is mostly confused with the enter and exit actions, which are similar walk actions happening in a specific location of the floor area in reality.



(a) P08: W1 (82.35%) (b) P08: Ens. (87.89%) (c) P09: MJV (91.63%)

(d) PD-1: W3 (63.56%) (e) PD-1: Ens. (76.77%) (f) PD-1: MJV (82.31%)

(g) P04: W3 (54.63%) (h) P04: Ens. (60.24%) (i) P04: MJV (68.14%)

(j) PD-1: W3 (43.03%) (k) Pd-1: Ens. (54.41%) (l) PD-1: MJV (63.56%)

Fig. 34.: Confusion matrices for one sampled participant of Group-1 (a, b, c, g, h, and i) and Group-2 (d, e, f, j, k, and l) participants out of per-participant trained model (a, b, c, d, e, and f) and SupCon-based cross-participant model's (trained on two participants) test results (g, h, i, j, k, and l). Classification accuracies are shown in % in subcaptions. W1 and W3 designate corresponding WiFi links as presented in Tables 13 to 20; "Ens." stands for "Ensemble model," and "MJV" stands for weighted majority voting scheme.

4.3.5.2 Cross-Person and Cross-Environment Generalization

We evaluate the SupCon generalization model by incrementally expanding the training set and measuring MJV accuracy on all remaining participants. In-training participants are marked with † in the tables; all other rows represent untrained participants in previously untrained apartments. The rows labeled *Avg. (untrained)* and *Std. (untrained)* summarize the cross-person performance excluding any in-training participants.

Group 1: Apartment Deployments. Tables 15–17 report results for the three Group 1 training configurations, and Fig. 35 shows the final classification accuracy out of the weighted MJV scheme across three configurations of one, two, and three training participant(s). It shows that while the two-participant training model gives better accuracy than the one-participant training model, the three-participant training model mostly confuses the model, as similar actions are trained on different layouts. Therefore, more training data cannot be more suitable for such a type of training scheme.

Table 15.: Group 1 cross-person results: SupCon model trained on P08 only.

Participant	**W1**	**W2**	**W3**	**Ens.**	**MJV**
P08†	82.35	79.25	73.47	87.89	91.63
P03	35.57	48.32	45.82	61.34	68.32
P04	40.36	25.11	38.53	52.45	59.74
P05	31.93	38.43	42.57	58.32	65.22
P06	37.23	–	34.62	49.63	56.25
P07	35.73	37.51	–	45.78	52.42
P09	34.66	39.97	31.56	47.25	56.39
Avg. (untrained)					*59.72*
Std. (untrained)					*6.01*

With a single training participant (P08), the SupCon model already achieves a mean cross-person MJV of 59.72% across six entirely untrained apartments, well

Table 16.: Group 1 cross-person results: SupCon model trained on P08 and P05.

Participant	**W1**	**W2**	**W3**	**Ens.**	**MJV**
P08†	80.64	76.43	68.83	83.56	88.45
P05†	78.46	54.24	72.45	82.35	87.12
P03	40.24	58.73	63.56	68.72	75.33
P04	47.32	39.62	54.63	60.24	68.14
P06	48.74	–	51.46	61.35	67.30
P07	43.67	53.92	–	59.73	69.25
P09	44.77	54.76	55.25	66.43	71.35
Avg. (untrained)					*70.27*
Std. (untrained)					*3.21*

Table 17.: Group 1 cross-person results: SupCon model trained on P08, P05, and P03.

Participant	**W1**	**W2**	**W3**	**Ens.**	**MJV**
P08†	77.43	70.11	61.35	78.43	85.73
P05†	62.46	48.22	65.73	74.11	82.35
P03†	76.35	74.63	75.73	77.32	81.62
P04	47.32	39.62	54.63	60.24	65.73
P06	48.74	–	51.46	61.35	60.21
P07	43.67	53.92	–	59.73	63.56
P09	44.77	54.76	55.25	66.43	65.84
Avg. (untrained)					*63.84*
Std. (untrained)					*2.63*

above random chance for a six-class problem (16.7%). Adding P05 to the training pool raises the untrained-participant mean to 70.27% (±3.21%), with every previously untrained participant improving by 7–15% points. The tighter standard deviation under the two-participant model compared to the one-participant model (3.21% vs. 6.01%) indicates that the two-environment encoder generalizes more uniformly across untrained homes. When a third participant (P03) is added, the untrained-only mean settles at 63.84% (std. 2.63%); this slight reduction from the two-participant peak occurs because P03's apartment is less geometrically diverse relative to the remaining four untrained apartments (P04, P06, P07, P09), whereas P08 and P05 collectively

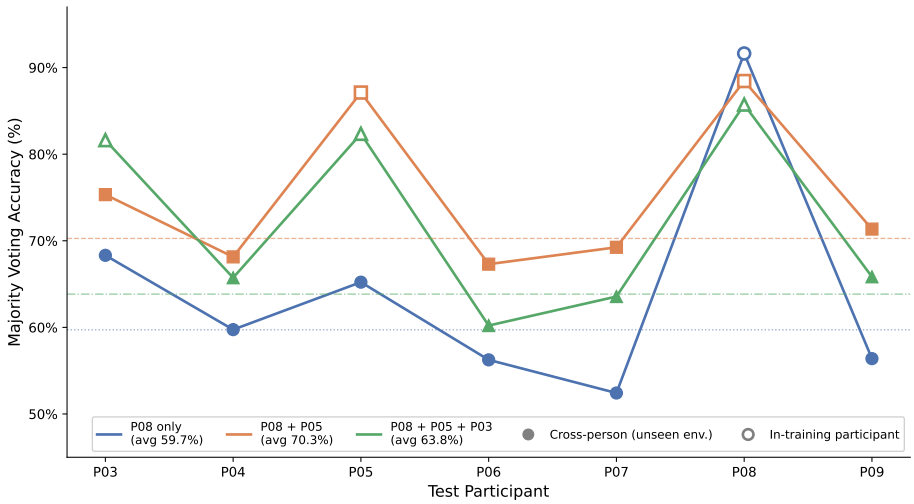


Fig. 35.: Group 1 SupCon cross-person generalization: MJV accuracy for each participant under three training configurations (P08 only, P08+P05, P08+P05+P03). Hollow markers denote in-training participants; filled markers denote untrained participants. Dashed horizontal lines mark the untrained-only average for each configuration.

provide a more complementary pair of spatial viewpoints. The standard deviation continues to narrow with each added participant, reflecting progressive stabilization of the learned embedding space.

Group 2: Home/Isolated Apartment Environments. Tables 18–20 report results for the three Group 2 training configurations, and Fig. 36 shows the corresponding MJV trends verifying that more training data from three participants can degrade the overall classification accuracy similar to Group-1's trend shown in Fig. 35.

Group 2 cross-person results follow a pattern consistent with Group 1. Starting from a single training participant (PD3), the untrained-participant mean is 59.32%, which is comparable to the Group 1 single-participant baseline (59.72%), suggesting that the difficulty of one-shot environment transfer is similar across both deploy-

Table 18.: Group 2 cross-person results: SupCon model trained on PD3 only.

Participant	**W1**	**W2**	**W3**	**Ens.**	**MJV**
PD3†	59.63	62.47	67.88	79.29	85.91
P10	23.89	29.45	27.55	45.61	56.28
PD1	33.52	–	37.54	49.05	60.13
PD2	35.25	21.45	38.52	53.81	61.55
Avg. (untrained)					*59.32*
Std. (untrained)					*2.73*
Overall avg.					*65.97*

Table 19.: Group 2 cross-person results: SupCon model trained on PD3 and PD2.

Participant	**W1**	**W2**	**W3**	**Ens.**	**MJV**
PD3†	57.91	60.21	64.86	77.39	84.93
PD2†	59.88	40.52	63.78	76.52	82.35
P10	27.42	38.63	35.93	49.66	59.72
PD1	41.37	–	43.03	54.41	63.56
Avg. (untrained)					*61.64*
Std. (untrained)					*2.72*
Overall avg.					*72.64*

ment settings. Adding PD2 to training marginally increases the untrained mean to 61.64%. With three training participants (PD2, PD3, P10), only PD1 remains untrained, achieving 52.41% MJV. The lower absolute cross-person accuracy in Group 2 compared to Group 1 is consistent with the lower per-person baselines in this group (mean 81.91% vs. 85.70%), indicating that Group 2 environments present inherently more variable CSI conditions.

4.3.5.3 Summary

Table 21 consolidates the key accuracy figures across both evaluation settings. The per-person models establish a strong upper bound, confirming that the full preprocessing and ensemble pipeline is effective when sufficient per-environment training data are available. The SupCon cross-person results demonstrate that mean-

Table 20.: Group 2 cross-person results: SupCon model trained on PD2, PD3, and P10.

Participant	**W1**	**W2**	**W3**	**Ens.**	**MJV**
PD3†	53.47	59.11	61.07	75.34	81.59
PD2†	57.07	39.53	62.72	70.12	77.11
P10†	35.25	50.24	53.56	65.24	69.09
PD1	40.02	–	39.44	47.77	52.41
Avg. (untrained)					*52.41*
Std. (untrained)					–
Overall avg.					*70.05*

Table 21.: Summary of per-person and cross-person MJV accuracies. Cross-person averages and std. dev. are over untrained persons only.

Setting	**Training config**	**Average (%)**	**Std. Dev. (%)**
Per-person (G1)	Each participant individually	85.70	3.92
Per-person (G2)	Each participant individually	81.91	5.08
Cross-person G1	P08	59.72	6.01
Cross-person G1	P08 + P05	70.27	3.21
Cross-person G1	P08 + P05 + P03	63.84	2.63
Cross-person G2	PD3	59.32	2.73
Cross-person G2	PD3 + PD2	61.64	2.72
Cross-person G2	PD2 + PD3 + P10	52.41	–

ingful zero-retraining generalization is achievable: with two training participants, the Group 1 model reaches 70.27% on four entirely untrained apartments, requiring only a brief support-set collection visit rather than a week-long annotated deployment. This represents the core practical contribution of the proposed generalization framework.

4.4 Discussion and Future Directions

This chapter presents a WiFi sensing-based system for monitoring the daily activities of older adults in real senior housing environments performed fully naturally, covering the full pipeline from IoT deployment and data collection through signal preprocessing, per-person classification, and cross-environment generalization. On

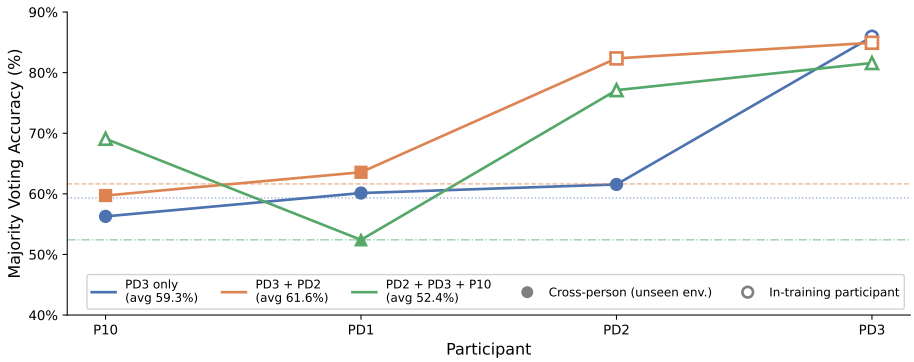


Fig. 36.: Group 2 SupCon cross-person generalization: MJV accuracy for each participant under three training configurations (PD3 only, PD3+PD2, PD2+PD3+P10). Hollow markers denote in-training participants; filled markers denote untrained participants. Dashed horizontal lines mark the untrained-only average for each configuration.

the deployment side, the study documents practical challenges encountered across eleven participant households over deployments lasting up to a week each, providing a resource for future researchers planning similar real-world sensing studies. The low-cost, device-free nature of the system, built entirely on ESP32 microcontrollers and Raspberry Pi, makes it viable for large-scale rollout without placing any instrumentation burden on the participant. On the machine learning side, the updated per-person pipeline combining per-day background clutter normalization, Hampel-based denoising, PCA dimensionality reduction, per-link Conv1D models, F1-weighted ensemble stacking, and confidence-weighted majority voting achieves a mean accuracy of 85.70% across seven Group-1 participants and 81.91% across four Group-2 participants. These results substantially exceed the 76.90% best-case accuracy reported in the initial deployment study [206], confirming that the improved preprocessing and ensemble design yield a meaningful accuracy gain without requiring any change to the physical sensing hardware. The SupCon generalization model further demonstrates

that a shared encoder trained on as few as two participants can classify ADLs in untrained apartments at 70.27% mean accuracy, requiring only a brief support-set collection visit at setup time rather than a week-long annotated deployment. This result is a step toward a deployable system that does not impose repeated, long-term data collection burdens on care providers or clinical researchers when expanding to new residents.

Several directions remain open for future work. First, expanding the SupCon training pool to include all eleven participants is expected to improve and stabilize cross-person generalization further, particularly for Group-2 environments whose wider, multi-room layouts currently yield lower transfer accuracy. Second, incorporating CSI phase information alongside amplitude, with proper per-device phase calibration, may sharpen activity boundaries in the learned embedding space. Third, automated or weakly supervised activity annotation—for example, by combining coarse motion-event detection with sparse user confirmation—would substantially reduce the manual labeling cost that currently limits how quickly new participants can be on-boarded. Finally, extending the framework to open-set detection, i.e., recognizing activities not observed during training, as explored in recent RF-based work [135], would bring the system closer to the unconstrained nature of genuine daily living, where residents perform activities that were not anticipated at model design time.

CHAPTER 5

MOCKIFI: CSI IMITATION FOR ZERO-SHOT LEARNING

5.1 Introduction

One of the key challenges in using CSI for activity recognition is addressing the variability introduced by different individuals and the temporal effects integrated into their activities when performed in different times [209]. The former refers to the differences in CSI patterns when different individuals perform the same activity, as the physical characteristics and movement style of each person can alter the signal in unique ways. On the other hand, temporal effects encompass the variations in CSI data when the same individual performs an activity multiple times, potentially under different environmental conditions or over various time intervals. These factors introduce significant complexity in developing robust and generalized models for accurate activity recognition.

To tackle these challenges and develop a generalized solution, several approaches have been studied recently. These approaches leverage principles from zero-shot [126] or few-shot learning [127, 210], transfer learning [211], federated learning [**wifederated**], generative adversarial networks (GANs) [212] and variational auto encoders (VAE) [213]. However, these systems either use one or a few samples from the unknown activities of people or use some additional information (e.g., text semantics) to relate the unknown activities to known activities (e.g., considering running as a faster version of walking [128]).

In this chapter, we explore a different approach and aim to generate CSI activity patterns of new individuals from transformations of CSI patterns across different

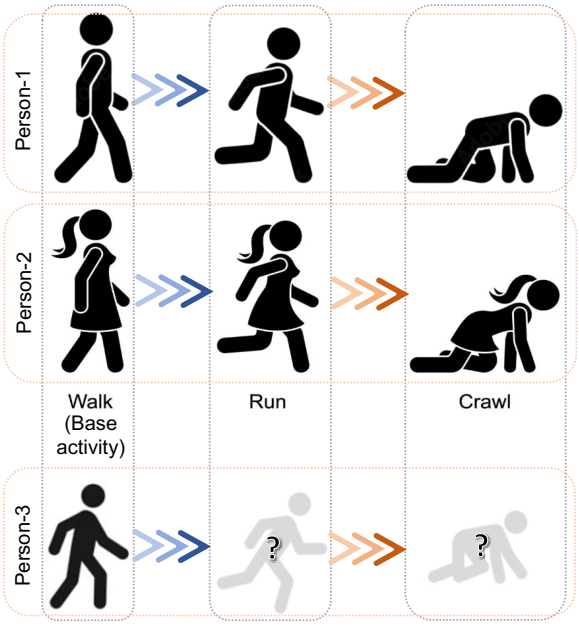


Fig. 37.: Learning CSI transformations across the activities of known users (i.e., Person A and B) can help obtain the CSI fingerprint of the activities of a new person (Person C) from a base activity (e.g., walk).

activities of known individuals. In this context, we propose *MockiFi*, a system which utilizes a context-aware conditional neural process to synthesize realistic WiFi CSI data for different activity patterns. Through this approach, we then aim to obtain realistic CSI data representing the activity fingerprints of unknown individuals, which can be used to develop recognition models without the hassle of collecting training data.

MockiFi is built upon understanding the context from known individuals (i.e., CSI transformations across their activities) and applying it to new individuals through a conditional neural process that uses only their base activity characteristics and a cosine similarity-based loss function. Through experimental evaluation on a dataset of five activities performed by seven different individuals, we demonstrate that the proposed MockiFi system is capable of accurately mimicking the unknown activity

patterns of individuals from their base activities as well as the learned transformations across activities of known people. This enhancement of WiFi CSI data generation for unknown activities of new individuals can help develop customized WiFi sensing solutions for different environments and people, without the need to collect a massive amount of training data from every new setting.

The rest of this chapter is organized as follows. Some relevant researches are presented in Section 5.2. In Section 5.3, we present our targeted problem and the details of the proposed MockiFi system. We evaluate the proposed system through experiments in Section 5.4. Finally, in Section 5.5, we summarize the contributions of this chapter.

5.2 Related Work

In this section, we provide an overview of the most related studies to our work explicitly presented in this chapter, in addition to the studies discussed in Chapter 2. **Conditional Neural Process:** Conditional Neural Process (CNP) [214] is a family of neural models structured as neural networks but inspired by the flexibility of stochastic processes, allowing them to make accurate predictions from a handful of training data points while scaling to complex functions and large datasets. In contrast to other generative models such as VAE and GAN, CNP can perform both as a classifier [215] and a generative [216] model with the proper given context. Such context-aware [217, 218] systems are proven to be less resource exhaustive in generating synthetic data for computer vision [216] as well as in natural language processing [218]. Contrastive [219] and evidential [220] variations of CNP show that we can design encoders and decoders to satisfy classification or data augmentation needs. Our work is inspired by the contextual awareness and the edge of CNP upon VAE and GAN with WiFi CSI data.

Latent Space: Latent space represents a compressed, lower-dimensional embedding of complex data, enabling generative models, like VAEs and GANs, to learn meaningful representations by mapping high-dimensional inputs to a continuous, semantically structured space [221]. While widely applied in computer vision [222] and natural language processing [223] for generating novel content, the potential of latent space exploration remains largely untapped for WiFi CSI data, presenting a promising frontier for signal processing and generative modeling [224].

Contributions: Different from previous studies, in this study, our goal is to extract the transformation between different activities of known individuals and use this transformation information to generate the CSI fingerprint of an utterly unknown activity of new individuals from their known base activities. To achieve this, we use a context-aware conditional neural process, which learns the context of transformations across activities of known individuals and apply this condition to the known/base activity of unknown individuals to obtain their other activities. As the target action fingerprint is completely unknown to the system, this approach can also be categorized as a zero-shot learning approach to generate the unknown CSI fingerprint. This makes the system a novel approach to generating synthetic data for the WiFi sensing applications of human action recognition.

5.3 Proposed Method

To synthesize the CSI data of any person's unknown set of actions, our proposed method utilizes a common known action of any other person along with the target action's CSI data of this person. As our classifiers indicate, any person's WiFi CSI dataset for all the actions is separable from another person's similar action set (need to refer to person-classifier from all actions). Based on this fact, we make our initial assumption that the context switch between any two actions can be identified

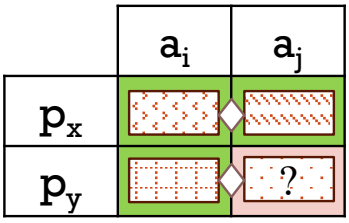


Fig. 38.: Targeted Contextual Relation to Learn

for different persons. This section describes the problem formulation based on this assumption, and the detailed implementation follows it.

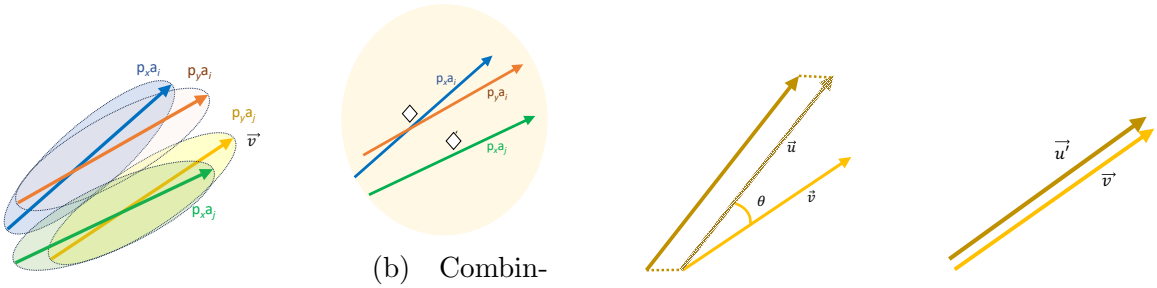
5.3.1 Problem Statement

To generate any action's CSI fingerprint for a person, we consider a set of persons P and a set of actions A. When a person, $p_x \in P$, performs two actions, $a_i \in A$ and $a_j \in A$, we are given the CSI fingerprint of another person $p_y \in P$ for the action a_i. This study aims to mock the CSI fingerprint of the action a_j performed by the person p_y, while the real CSI fingerprint of this person's action may not be known at all, making the system knowledgeable of this action with zero-shot.

Our proposed approach relies on the insight that a context switch between any two actions of a person has a definitive and similar pattern that can also be learned and applied to other people in their own environment. This insight can be stated below for the two mentioned persons, p_x and p_y, performing two actions a_i and a_j,

$$\begin{aligned} p_y a_j \diamondsuit p_y a_i \;&\sim\; p_x a_j \diamondsuit p_x a_i \\ or, p_y a_j \;&\sim\; p_x a_j \diamondsuit p_x a_i \diamondsuit' p_y a_i \end{aligned} \tag{5.1}$$

Leveraging this relationship, to mimic the second person's second action, e.g., $p_y a_j$, we can provide the first person's two actions, e.g., $p_x a_i$ and $p_x a_j$, as the context and input the second person's first action, e.g., $p_y a_i$, to a generative network so that



(a) Latent Space (Hypothetical Plane)

(b) Combining Known Latent Vectors into $\vec{u}$

(c) Match directions of $\vec{u}$ and $\vec{v}$ producing $\vec{u'}$

(d) Match magnitudes of $\vec{u'}$ and $\vec{v}$

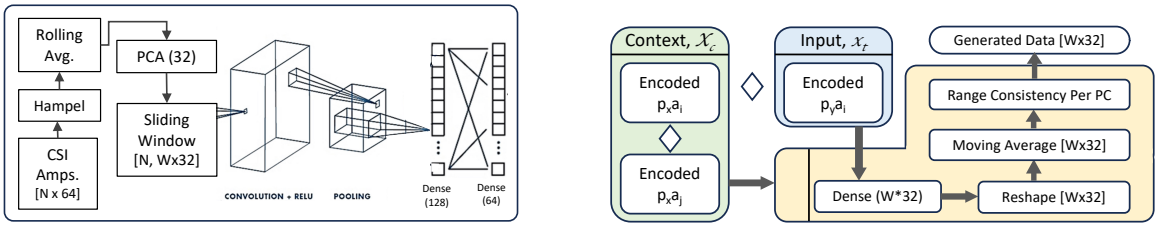
Fig. 39.: Development of CNP to find the latent vector ($\vec{v}$) by minimizing θ and matching magnitudes using loss functions.

it can learn the context switch (the $\Diamond$ operation) and incorporate the input (the $\Diamond'$ operation). The problem then simplifies into learning the $\Diamond$ and $\Diamond'$ operations.

5.3.2 Data Augmentation Approach

5.3.2.1 Preprocessing

We consider only CSI amplitudes in this study, as most WiFi sensing studies use it as the primary feature [152]. We then apply the Hampel filter [225] and take the moving average over all the non-null subcarriers of our single antenna ESP32 device to denoise the data. PCA is then applied to the dataset to allow us to take the first major principal components to reduce the dimension of the dataset. We take sliding windows of an optimized size of w over each principal component to expand the prediction duration. It expands the size of the dataset by w times and gives more meaning to the temporal effect of the actions. These preprocessing steps are applied all the CSI data whether they are used in generative or classifier model developments.



(a) Encoder for a single context along with pre-processing.

(b) Decoder Merging All Context Outputs with the Encoded Input

Fig. 40.: Encoder and decoder networks for the CNP model.

5.3.2.2 Normalization in Latent Space Using Loss Functions

We train a Conditional Neural Process (CNP) model, f_θ, with a custom loss function and the known three sets of actions mentioned in Eq. 5.1, so that it may learn the latent space representing those actions and optimize itself to align the output as a latent vector, $\vec{u}$, with the target latent vector $\vec{v}$ of $p_y a_j$. A hypothetical presentation of the latent spaces for the four mentioned CSI sets is shown in Fig. 39. The overlapping regions represent the confusion of the classifier between the two persons and between the two actions. The tentative latent vectors can be extrapolated as in Fig. 39b. Utilizing the relationship assumption given in Eq. 5.1, the three known latent vectors can be combined to generate some other vector, $\vec{u}$, which is to be matched with the second person's second action's latent vector, $\vec{v}$. If the angle between these vectors is θ, the cosine of the angle gives us the following relationship:

$$\cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\|\|\vec{v}\|}.$$

When the directions of the two vectors match, we get $\theta = 0$ and $cos(\theta) = 1$. Therefore, we can define a loss function, $\mathcal{L}_c$, based on maximization of $cos(\theta)$. During training, at any timestep t over n principal components, if the target data is y_t and the generated

data is $\hat{y}_t$, then we can define the cosine similarity loss function as

$$\mathcal{L}_{\mathrm{c}} = 1 - \frac{\sum_{i=1}^{n} \hat{y}_t^i \cdot y_t^i}{\|\hat{y}_t\| \|y_t\|}. \tag{5.2}$$

Data generation training with optimizing this cosine similarity loss function matches the directions of the latent vectors. However, the magnitudes of the vectors are also to be matched. To this end, we incorporate another loss function, $\mathcal{L}_{\mathrm{r}}$, to make sure the range of values generated are consistent with the actual data,

$$\begin{aligned} \mathcal{L}_{\mathrm{r}} = &\sum_{i=1}^{n} \left(\frac{\max(y_{t,:,i}) - \max(\hat{y}_{t,:,i})}{\max(y_{t,:,i})} \right)^2 \\ &+ \sum_{i=1}^{n} \left(\frac{\min(y_{t,:,i}) - \min(\hat{y}_{t,:,i})}{\max(y_{t,:,i})} \right)^2. \end{aligned} \tag{5.3}$$

The maximum and minimum in this formula are calculated over all rows of each principal component i among n for any time step or CSI window, t. We take the weighted sum with weights λ_c and λ_r for $\mathcal{L}_c$ and $\mathcal{L}_r$ respectively. These weights can be optimized as hyperparameters of the system. Therefore, the final loss function can be defined as

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_r \mathcal{L}_r. \tag{5.4}$$

It is noteworthy that this loss function is designed from the vector representations and is differentiable. Therefore, we can optimize our generative model by optimizing this loss function.

5.3.2.3 Context-Aware Conditional Neural Process

We formulate the proposed Context-Aware CNP as a mapping from a context set and a target input to the corresponding target prediction. We consider the first person's two actions as the context, $\mathcal{X}_c$, and the second person's known action as the input, x_t, to the model. Over these conditions, our context-aware CNP generates

predictions $\hat{y}_t$ for target inputs x_t using the context set $\mathcal{X}_\text{c}$ by learning the mapping:

$$f_\theta : (\mathcal{X}_\text{c}, x_t) \mapsto \hat{y}_t. \tag{5.5}$$

This mapping is performed in two steps: (i) encoding and (ii) decoding. For the encoder g_ϕ, we define a one-dimensional convolutional layer of kernel size three and use max-pooling with pool size two after it. Two dense layers of 128 and 64 hidden nodes follow the convolutional layer. The preprocessing steps followed by these neural processes are depicted in Fig. 40a. Hereby, the encoder g_ϕ maps the context set $\mathcal{X}_\text{c}$ into a latent representation u':

$$u' = g_\phi(\mathcal{X}_\text{c}). \tag{5.6}$$

The encoded outputs are then passed to a decoder, as shown in Fig. 40b. At the same time, the input, x_t, is also passed through similar encoding before feeding it to the decoder. The loss function, $\mathcal{L}$, is optimized at this point promising a well matching $\vec{u'}$ to the target $\vec{v}$ vector. The decoder h_ψ predicts the target output $\hat{y}_t$:

$$\hat{y}_t = h_\psi(u', x_t). \tag{5.7}$$

However, as the data points are discretely generated over very short time periods usually optimized below one second, we need to denoise the generated output. A sample of such noisy signals is shown in Fig. 44a. The range consistency loss function ($\mathcal{L}_r$) optimization promises that there may not be any outlier nor extremely noisy signal, so we apply a simple denoising method like a moving average for a few iterations per principal component to make the output signals smoother. This output may still get scaled and/or shifted in latent space due to temporal effects [226] during the data collection process. To avoid this kind of scaling and shifting, we apply our already available range consistency function, as used by $\mathcal{L}_r$, again over this output.

The ranges are observed to vary among principal components, so this post-processing step is applied per principal component over all CSI frames which also coincides with our already defined range consistency loss function. This denoising phase is finally packaged with our CNP model, which establishes the decoder. This flow is depicted in the Fig. 40b.

The CNP model achieves minimal θ^*, ϕ^*, ψ^*, through optimizing the loss function $\mathcal{L}_c$, the encoder g_ϕ, and the decoder h_ψ, respectively. This optimization requires training of CNP by minimizing the total loss over all target data, $\mathcal{X}_t$, as in

$$\theta^*, \phi^*, \psi^* = \underset{\theta,\phi,\psi}{\operatorname{argmin}} \mathbb{E}_{\mathcal{X}_\mathrm{c},\mathcal{X}_\mathrm{t}} \left[\mathcal{L}\right]. \tag{5.8}$$

Thereby, the proposed steps to design and train the context-aware conditional neural process can be summarized in the following steps.

1. Proprocess CSI amplitudes using Hampel filter, moving average, PCA and sliding window to prepare the context set $\mathcal{X}_\mathrm{c}$ and the input x_t.

2. Encode the context set $\mathcal{X}_\mathrm{c}$ using g_ϕ to produce a representation u'.

3. Decode the representation u' and encoded target inputs x_t and then denoise using h_ψ to predict $\hat{y}_t$.

4. Compute the total loss $\mathcal{L}$ using $\hat{y}_t$ and y_t.

5. Backpropagate and update the parameter sets ϕ and ψ to minimize $\mathcal{L}$.

5.4 Evaluation

To evaluate our data generation system, we have performed exhaustive experiments. We have collected a dataset of five activities performed by seven different volunteers and split that dataset into proper training and testing sets both for the

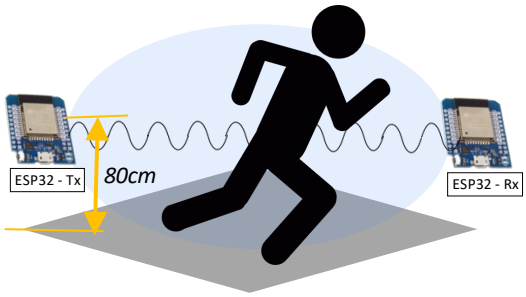


Fig. 41.: Experimental Setup for CSI Data Collection

generative model and the action as well as the person classifier models.

5.4.1 Data Collection

We have used the low-cost and portable edge device ESP32-mini in conjunction with an open-source firmware [227] to transmit and receive WiFi CSI data. One of the ESP32 devices is used as the receiver, and another ESP32 is paired with it as the transmitter. The transmitter transmits CSI frames at $100Hz$ rate so that the time difference between two subsequent CSI frames is 10 milliseconds on average. In other words, every 100 CSI frames represents one second of CSI action fingerprint.

As depicted in Fig. 41, the transmitter and receiver were placed across the left and right side of the volunteer at around the hip-level height of $80cm$ from the ground. The data are saved from the receiver in a connected Raspberry Pi 4B device using our open-source project CSI-Pi.

As for the performed actions, we consider walking back and forth (a_1), standing running (a_2), joycing with two hands throwing upward (a_3), waving the right hand (a_4), and repeating sitting down and standing up from a chair (a_5).

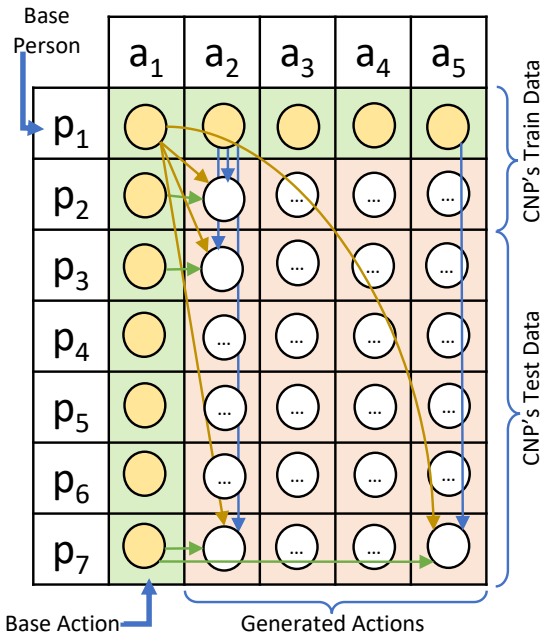


Fig. 42.: Data generation flow for context and input selection among the seven volunteers' (i.e., p_1, p_2, p_3, p_4, p_5, p_6, and p_7) five performed actions (a_1, a_2, a_3, a_4, and a_5) along with their train-test split for the generative model.

5.4.2 Training and Testing Activity Sets

Each of the seven participants performed each activity ten times in a round-robin fashion. These repetitions are split into two sets: six repetitions are used for training, and four repetitions are used for testing per person. Data preprocessing steps are applied to each of these fourteen sets of data, where we train the PCA model with the training data and only transform the test dataset using the already trained PCA model.

Once the data is preprocessed, we consider the training and testing datasets similarly for the first two persons (p_1 and p_2) for both the generative and the classifier models. The rest five persons' data are fed to the classifiers as it is, but we consider these five sets (p_3 to p_7) to be entirely testing datasets for the generative CNP model.

At the same time, only the first action (a_1) or base action of the persons p_3 to p_7 is deemed to be known to the system. All the actions of the first person (p_1) are considered as the actions of the base person and, thereby, known to the system. The unknown activities of p_3 to p_4 (a_2, a_3, a_4, and a_5) are then generated from these known activities. This kind of split is explained in the Fig. 42. As the generative model's test results, we evaluate the cosine similarity scores for each of the 20 augmentation processes, where we generate data for the pairs between persons' set of p_3 to p_7 and actions' set of a_2 to a_5, e.g., p_3-a_2, p_3-a_3, p_7-a_5, etc. Regarding data splits for the CNP model, these pairs are considered the test data generated by the model. Note that these actions by these persons are totally unknown to the generative system. We only use the original CSI data of these actions to evaluate the quality and similarity of the generated data to the real data.

5.4.3 Training CNP by Optimizing Loss Functions

The CNP model is trained with the first two person's (i.e., p_1 and p_2) all five activities (a_1 to a_5), where it generates the activities a_2, a_3, a_4, and a_5 performed by the second person (p_2). Then the loss function $\mathcal{L}$ is calculated to optimize the model. Once the model is trained with a specific number of epochs and with optimal direction deviation (θ) for the generated and the target latent vectors of the real data, it is then used to generate the CSI fingerprints of the other activities of the rest five persons (p_3 to p_7). Fig. 43 shows the cosine similarity scores for each of the 20 generated unknown person-activity combinations during the testing phase. The cosine similarity scores for the 20 datasets range from 0.7576 to 0.9589, averaging 0.8502. These scores indicate that the angular difference (θ) between the target and the generated latent vector ranges from 40.8° to 16.5° and the average θ is 31.2°. While these show that the directions are not perfectly matched as described in Eq.

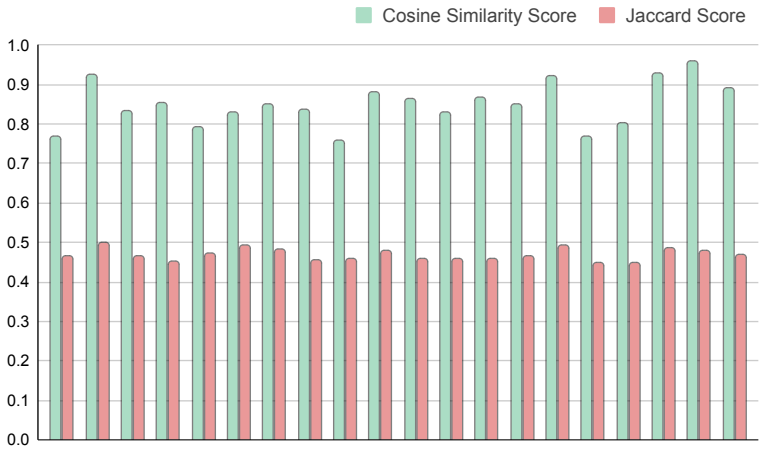


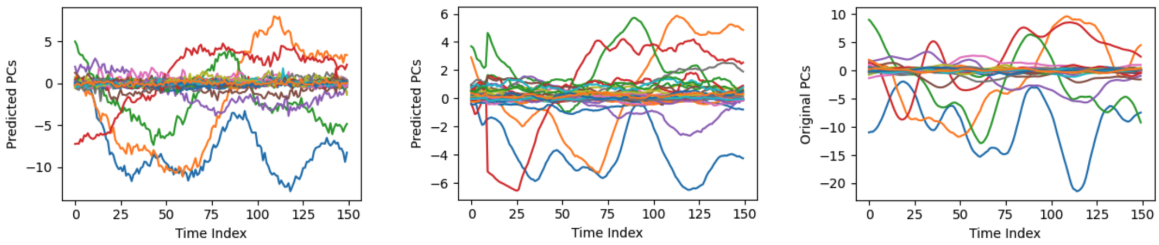
Fig. 43.: Cosine similarity (average 0.85) and Jaccard similarity (average 0.47) score comparison for the generated data.

5.1, they show evidence of targeted conversion to some extent.

In the Fig. 43, we also show Jaccard similarity scores for the generated data. While the cosine similarity score focuses on the directional deviation to be optimized by optimizing the loss function $\mathcal{L}_c$, the Jaccard similarity score represents the degree of scaling and shifting in the latent space in addition to the directional deviation. For this reason, the average Jaccard score is 0.47, whereas the best matching score is one.

5.4.4 Generated Data Identification by the Classifiers

We have trained separate CNN models for person and action classification tasks. The CNN model has a one-dimensional convolutional layer and three hidden dense layers with 256, 128, and 64 hidden nodes, respectively. It also has a dropout layer between the hidden layers to prevent overfitting. A single CSI window is fed into it as input to get predictions for the targeted persons or activity labels.

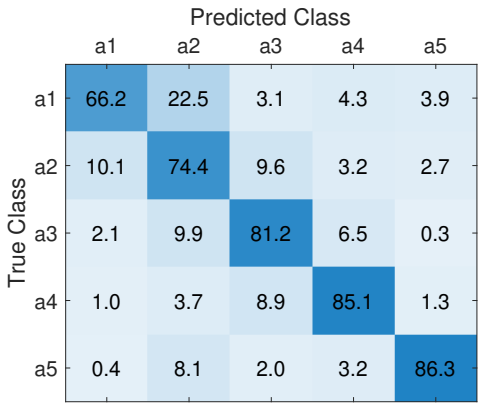


(a) CNP generated data (b) Post-processed data (c) Original data

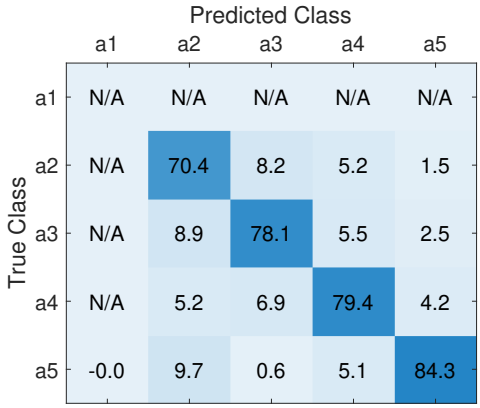
Fig. 44.: A sample CSI window comparison showing the effect of the denoising methods in the decoder as the final output in (b) gets closer to the original dataset shown in (c) (each different colored lines represents one of the 32 principal components).

5.4.4.1 Activity Classifier

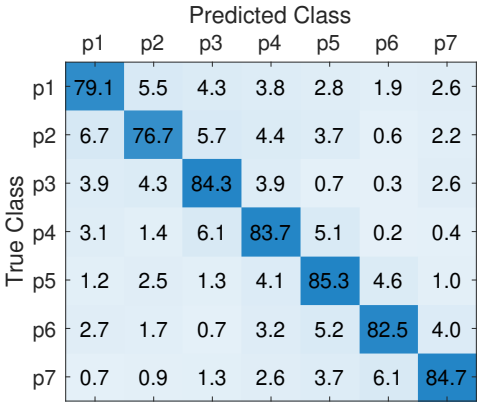
We train a single activity classifier model for all seven people. The testing accuracy for the five activities is 78.24% as shown in Fig. 45a. Within it, the targeted four activities have a testing accuracy of 81.75% with the original data. When our generated data are passed to the same classifier, their prediction accuracy comes up to be 78.79% for the four generated activities as shown in the confusion matrix in the Fig. 45b. This accuracy is very close to the classifier's performance with the original data. This indicates that the classifier model treats the generated and original data similarly, proving a good quality of the generated data.



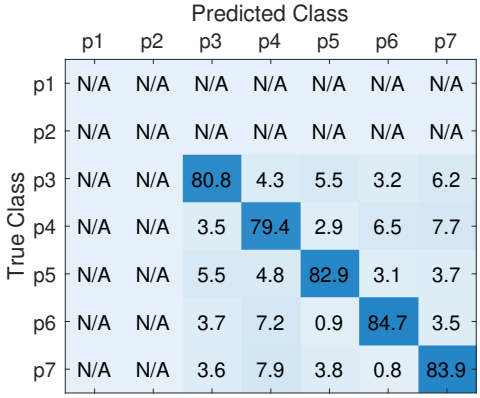
(a) Action Classification with Original Data (Overall accuracy: 78.24%, a_2-a_5 accuracy: 81.75%).



(b) Action Classification with Generated Data (78.79%). Predictions for non-generated data are shown with N/A.



(c) Person Classification with Original Data (Overall accuracy: 82.32%, p_3-p_7 accuracy: 84.11%).



(d) Person Classification with Generated Data (82.35%). Predictions for non-generated data are shown with N/A.

Fig. 45.: Confusion matrices for activity and person classifiers for original and generated data, with closely matching total accuracies (81.75% vs. 78.79% for action, 84.11% vs. 82.35% for person).

5.4.4.2 Person Classifier

To assess the quality of the generated data from another perspective, we also evaluate the results using a person classifier. This classifier is trained with training activity data of the seven persons. The accuracy of 82.32% for the seven persons' original data indicates that each individual is separable from another even if they perform the same activity in the same environment due to the unique CSI fingerprint generated by each person's performance. We train our generative model with the data from persons p_1 and p_2 while model testing is performed with persons p_3 to p_7. Therefore, we only have testing results of person classification on the generated data for the persons p_3 to p_7. These five persons have 84.11% testing accuracy with the original dataset, and their generated data give us 82.35% accuracy, which are illustrated in the confusion matrices in the Fig. 45c and Fig. 45d, respectively. These two close performances prove again that the generated data is as well structured as the original dataset for machine learning purposes.

Note that the accuracy of the classifier can potentially be improved with a more complex model architecture and with more training data, which should also reflect on the prediction accuracy of the generated data. Our goal was to be able to mimic the original data without knowing it from the activities of the other people and the transformations happening between them. In that context, classification results with similar accuracy show that this target is achieved.

5.4.5 Base Action and Person Switching

We also evaluate the robustness of the proposed system by changing the activities and base people considered during the training of the CNP model. That is, for example, when a_2 is used as the base activity, the transition learned from a_2 to a_3 for

Base Action				
a_1	a_2	a_3	a_4	a_5
78.79%	76.26%	80.14%	84.25%	79.24%

Table 22.: Classification of the four actions on the generated data with different base actions.

Base Person						
p_1	p_2	p_3	p_4	p_5	p_6	p_7
82.4%	77.1%	79.3%	82.1%	80.5%	78.3%	81.8%

Table 23.: Identifiability of the five persons having generated data with different base persons.

the base person (e.g., p_1) is leveraged to generate the activity CSI data for a_1, a_4 and a_5 of other people. In Table 22, we see that when the base activity considered switches to other activities, we see similar activity recognition accuracies for the generated data, showing the robustness of the proposed solution. Similarly, in Table 23, we show the person identification accuracies when the base person changes. These accuracies do not vary much and remain very close to the accuracy obtained with the original data. It is noteworthy that these accuracies may not be considered high enough, but our primary goal in this study is not to classify the actions and persons but to assess the quality of the generated data by examining how similarly it is classified compared to real data, which is reflected in these results.

5.4.6 Zero-shot Learning Performance

As we generate data for p_3 to p_7, we train two other classifier models to identify activities and persons with the training of the classifier models performed with the original CSI data for p_1 and p_2, and CNP based synthesized CSI data for p_3 to p_7, as if those five persons (i.e., p_3 to p_7) are not present during the model training phase.

Once the model is trained, we test it with the original data of p_3 to p_7 to identify their activities and with the data of p_1 and p_2 to identify the person. The activity classification results are shown in the Table 24, where we also present the variations of the results when the base person and activities are changed. We see that activity classification accuracy varies from 68.3% to 83.2% for the original data, where the model never sees that original data but only the synthetic data during training. On the other hand, the person classification result is shown in the Fig. 46, where we use the original data of p_1 and p_2 but synthetic data of the other five persons to train the model and then present the test classification result using the original data of all persons for the non-base actions, i.e., actions a_2 to a_5. The person classification accuracies for the target five persons vary from 72.9% to 82.1%, which is close to the original accuracies presented in the Fig. 45c. Since the original data of p_3 to p_7 are not present in the training set, we can call this kind of classification as zero-shot learning.

5.5 Discussion and Future Directions

In this chapter, we have explored the possibility of generating realistic CSI fingerprints of new and unknown activities of people using one of their activity data and the CSI transformations learned across different activities from other people. We have shown that this is possible through a context-aware conditional neural process. The generated data can be classified as well as the real data in terms of activity and person identification.

	Base Action				
	a_1	a_2	a_3	a_4	a_5
p_3	72.2%	74.5%	77.6%	76.2%	79.3%
p_4	73.4%	76.3%	78.4%	78.3%	81.2%
p_5	71.3%	77.4%	79.4%	81.2%	80.2%
p_6	68.3%	75.3%	80.2%	81.6%	83.2%
p_7	70.9%	76.3%	78.6%	80.5%	78.8%

Table 24.: Action classification results on real data of the four non-base actions performed by p_3 to p_7, while the model is trained with the real data of the base persons p_1 and p_2 but synthetic data for p_3 to p_7.

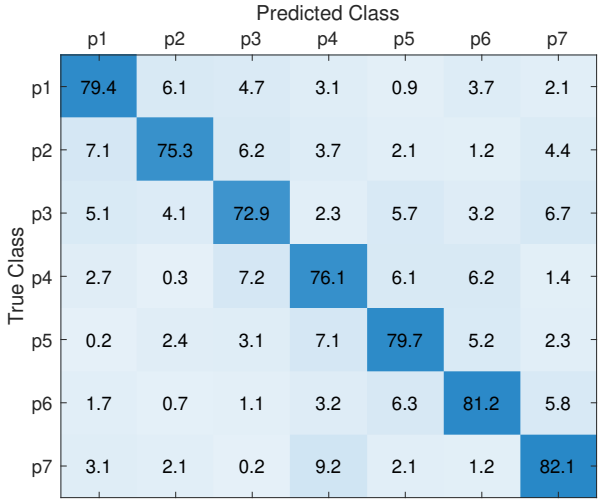


Fig. 46.: Confusion matrix of person classification on real data of p_3 to p_7, while the model is trained with the real data of p_1 and p_2 but synthetic data of the other five persons.

Leveraging this finding, WiFi sensing-based human activity recognition systems can be generalized to new environments and new people. Given a classifier model trained with some base person and with a set of activities, whenever a new user is added to the system, the newcomer needs to perform only one activity, which is considered the base activity, and MockiFi can generate the CSI fingerprints for all other required activities to let the model retrain to recognize those activities of the

newcomer in real-time.

We think that MockiFi presents a significant step towards generalized WiFi-based activity recognition systems. As a refinement and expansion of this approach, future research can increase the number of activities and people evaluated with more data and utilize more complicated models to increase the base classifier's accuracy with real data.

CHAPTER 6

AGNOGEST: ROBUST MULTIMODAL LOCOMOTION AGNOSTIC GESTURE SENSING USING MMWAVE POINT-CLOUD & WIFI CSI

6.1 Introduction

The preceding chapters established WiFi sensing as a capable modality for contactless activity recognition and gross gesture classification, progressively refining the signal-processing and learning pipeline from single-environment setups to subject-agnostic deployments. However, a practical limitation emerges the moment sensing is considered outside a controlled laboratory: in everyday environments, users do not perform gestures while standing still. A person waves to acknowledge a smart display while walking toward it; push-pull commands are issued mid-stride; fine-grained wrist rotations occur against a backdrop of concurrent torso movement. As illustrated in Fig. 47, this co-occurrence of subtle hand gestures and full-body locomotion is the norm rather than the exception in real IoT deployments. This co-occurrence creates what we term an *interference bottleneck*: the reflected signal signatures of delicate hand gestures are swamped by the far stronger Doppler and micro-Doppler returns from simultaneous body motion.

Prior work has addressed domain shift caused by changing environments or unseen subjects [228], but the orthogonal problem of disentangling *gesture intent* from *locomotion context* within the same receiver snapshot remains largely open. A framework that is *locomotion-agnostic*, i.e., one that preserves gesture discriminability regardless of what the body is doing concurrently, is therefore the logical next step in moving RF sensing from the lab to the field.

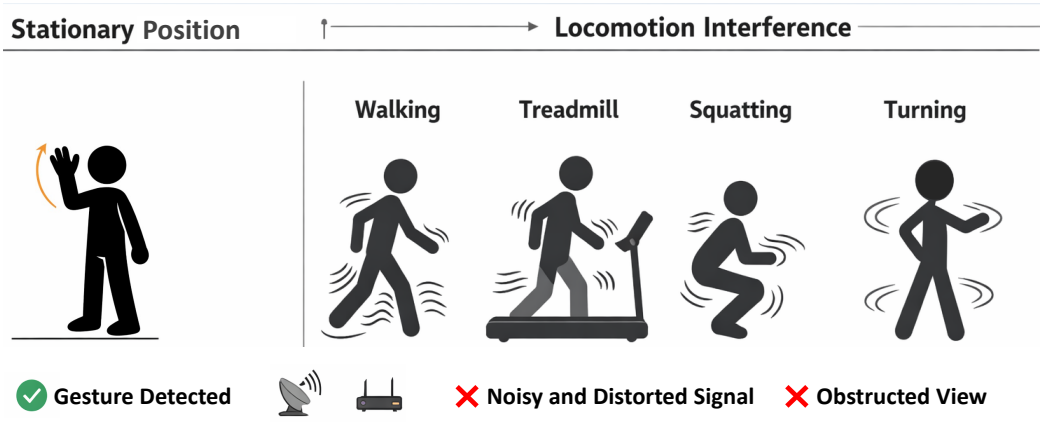


Fig. 47.: Challenges of gesture detection under locomotion.

To this end, this chapter presents **AgnoGest**, a multimodal approach that exploits the complementary physical properties of mmWave radar and WiFi to address this challenge. The 60 GHz mmWave radar provides high spatial and Doppler resolution, enabling precise point-cloud reconstruction of the fine-grained hand motion, while WiFi CSI captures wide-area multipath variations, providing resilience in NLoS scenarios where the user may be occluded [152, 34]. By synergizing these modalities, we can exploit the spatial precision of radar and the pervasive nature of WiFi to build a system that remains resilient even when the user is in motion, i.e., a robust multimodal system resilient to locomotion interference.

Our edge-optimized, locomotion-agnostic multimodal sensing platform, AgnoGest, is integrated on a Raspberry Pi 4B device. Moving beyond single-subject baselines, which often fail to generalize across diverse body biometrics, we propose a novel architecture that synchronizes 60 GHz mmWave point clouds and WiFi CSI through two attention-based gated fusion mechanisms dedicated to gesture and locomotion representations, respectively. By employing a triple-adversarial learning framework with Gradient Reversal Layers (GRL), our system aims to disentangle gesture-specific information from dominant locomotion interference, while reducing the influence of

person-specific biometric variations. This allows the model to maintain high recognition accuracy across unseen volunteers and locomotion domains while operating within the computational constraints of real-time edge inference.

The key contributions of this chapter are as follows:

- **Multimodal Gated Fusion:** We introduce two attention-based multimodal gated fusion mechanisms that dynamically weigh mmWave and WiFi CSI features for gesture and locomotion representations, respectively. This gating strategy prevents noisy auxiliary signals from degrading primary feature accuracy, ensuring the system remains resilient to the biometric variances of unseen subjects.

- **Locomotion-Agnostic Adversarial Learning:** We develop a triple-adversarial architecture using GRL to suppress locomotion-specific characteristics from learned representations across two WiFi links and one mmWave radar stream. This strategy mitigates the locomotion-interference bottleneck, improving generalization across unseen locomotion contexts.

- **Edge-Optimized Multimodal Platform:** We implement a lightweight framework on a Raspberry Pi 4B that synchronizes high-resolution 60 GHz mmWave point-clouds [229] and wide-area WiFi CSI [195, 152]. The optimized model achieves substantial memory reduction and efficient inference performance, enabling real-time operation on resource-constrained IoT gateways [230, 231].

- **Cross-Subject Validation:** We provide a comprehensive evaluation using a multi-volunteer dataset, demonstrating that while unimodal models often fail on unseen subjects, our multimodal adversarial gated fusion framework maintains stable performance, achieving an average accuracy of 87.64% across entirely

unseen volunteers.

The remainder of this chapter is organized as follows. Section 6.2 reviews related work and foundational concepts. Section 6.3 presents the AgnoGest system architecture and multimodal learning framework. Section 6.4 describes the experimental setup and evaluation results. Finally, Section 6.5 provides a summary of the chapter and outlines future directions.

6.2 Preliminaries

6.2.1 WiFi Sensing

As presented in prior chapters, we utilize OFDM-based WiFi sensing as a modality of this research work. For temporal modeling, CSI measurements across S subcarriers are organized over a sliding time window of length w into a matrix $\mathbf{H}[t] \in \mathbb{C}^{S \times w}$, as already described in Eq. 2.9. This matrix structure preserves the temporal evolution of the signal within each CSI window, serving as a snapshot of that moment through the lens of WiFi, which is critical for deep learning models to extract unique motion fingerprints from the sequence.

Prior works have successfully utilized these CSI dynamics for various IoT tasks. However, while highly accurate whole-body and even fine-grained hand-gesture sensing systems exist, robust gesture recognition remains challenging in dynamic environments where dominant full-body locomotion frequently overwhelms subtle hand-induced multipath variations.

6.2.2 mmWave Sensing

The AgnoGest platform uses a Texas Instruments IWR6843ISK mmWave radar operating at a 60 GHz carrier frequency to produce a 3D point cloud of detected

scatterers within its field of view at 10 frames per second. The radar employs Frequency Modulated Continuous Wave (FMCW) signals [87], whose general processing pipeline, including Range FFT, Doppler FFT, CFAR detection, and Angle of Arrival (AoA) estimation, is described in detail in Section 2.2 of the background chapter. This subsection describes the specific radar configuration used in AgnoGest, the derived features extracted from each processing stage, and how those features are adapted for locomotion-agnostic gesture recognition.

6.2.2.1 Radar Configuration

The IWR6843ISK is configured with a 50 MHz/μs chirp ramp slope, using 96 chirps per frame and dual transmit antennas (TX). This configuration yields a range resolution of 7.03 cm via Eq. (2.11) and a maximum unambiguous radial velocity of 20.16 m/s via Eq. (2.13), with a velocity resolution of 0.63 m/s. The maximum detection range is 6.4 m. Each chirp cycle spans 62 μs (55 μs ramp plus 7 μs idle), yielding a 5.95 ms active burst duration per frame. At a 100 ms frame periodicity (10 frames/s), the system operates at a 5.95% duty cycle, ensuring energy-efficient continuous monitoring suited to resource-constrained edge devices. The configuration details are tabulated in Table 25 and illustrated in the chirp diagram in Fig. 49.

6.2.2.2 Range and Doppler Processing

Following the Range FFT of Eq. (2.10), each ADC sample frame is converted into range bins corresponding to target distances. The Doppler FFT of Eq. (2.12) is then applied across $N_{chirp} = 64$ chirps to resolve radial velocities, producing a full 2D Range-Doppler map per frame. The CFAR detector of Eq. (2.14) [97] is applied to this map to identify statistically significant reflections above the local noise floor, and AoA estimation via Eq. (2.15) adds spatial angular information to each detected

point. The final per-frame output is a structured 3D point cloud (x, y, z, v_r) with associated radial velocity, as described in Section 2.2.3.

6.2.2.3 Velocity Feature Extraction

The raw per-point radial velocity v_r in the point cloud conflates bulk locomotion signals (i.e., the entire body moving together) with fine-grained gesture signals (i.e., local hand motion). To suppress locomotion contributions and emphasize hand-specific dynamics, we compute the mean differential velocity v_d per frame as defined in Eq. (2.16) [89]. By centering the velocity distribution around the global mean v_r, v_d captures localized high-frequency bursts of motion that characterize hand gestures independently of the user's overall body motion. This design choice adds negligible model weight while meaningfully enriching the feature space for the adversarial disentanglement described in Section 6.3.2.3.

6.2.2.4 Point-Cloud Voxelization

Because the number of detected points varies per frame, a fixed-size volumetric representation is required for processing by the convolutional neural network branch. Following the voxelization procedure of Eqs. (2.17) described in Section 2.2.5, the 3D coordinate space is discretized into a $10 \times 32 \times 32$ grid ($N_z \times N_y \times N_x$). The higher horizontal and vertical resolution (32×32) captures fine-grained hand trajectory details, while the lower depth resolution (10 bins) is sufficient to distinguish the hand from the torso and background at typical deployment distances. Each voxel is represented by two channels: (i) an occupancy flag (1 if at least one point falls in the cell, 0 otherwise) encoding the spatial structure of the gesture, and (ii) the mean differential velocity v_d of all points within the cell (Eq. (2.16)), providing a localized Doppler signature robust to locomotion. The resulting dual-channel tensor of shape

($N_{frames} \times N_z \times N_y \times N_x \times 2$) is passed as input to the Time-Distributed 3D CNN branch described in Section 6.3. This voxel-based approach ensures deterministic inference latency and leverages optimized 3D-convolution kernels, enabling real-time operation on the Raspberry Pi 4B without a dedicated GPU. Compared to point-cloud-based methods like PointNet++ [98], which rely on computationally expensive ball-query operations ill-suited to sparse mmWave data, voxelization provides a better fit for edge hardware constraints as demonstrated in RadHAR [95]. To improve cross-person generalization under locomotion, the point cloud coordinates are augmented during training by applying a random scaling factor within $[0.9, 1.1]$ and a translation offset within $[-0.05, 0.05]$ meters per axis, quadrupling the original training set size. This strategy supports the model's generalization across diverse body biometrics and variable gesture positions, consistent with findings in prior work on synthetic data augmentation for wireless sensing [96, 93, 94].

6.2.3 Multimodal and Adversarial Machine Learning

Multimodal Machine Learning (MMML) is defined by the integration of information from diverse heterogeneous sources [232] to enhance the robustness and completeness of a shared latent representation [233]. Mathematically, for a set of K modalities $\{M_1, M_2, \ldots, M_K\}$, we define a joint embedding function F that maps modality-specific features $\mathbf{h}_k = G_k(\mathbf{x}_k; \theta_k)$ into a unified representation space $\mathcal{Z}$, such that $\mathcal{Z} = F(\mathbf{h}_1, \ldots, \mathbf{h}_K)$. This paradigm has achieved transformative success in cross-modal alignment frameworks such as CLIP [234], which uses contrastive learning to bridge the vision and language manifolds. In the RF-sensing domain, recent benchmarks like MM-Fi [121] and mm-Multi [235] have demonstrated that synergizing distinct signal properties, e.g., the high spatial resolution of 60 GHz mmWave and the pervasive coverage of WiFi CSI, enables sensing resilience in scenarios where

single-modality systems encounter physical limitations like signal attenuation or NLoS occlusions.

To ensure these multimodal representations remain invariant to domain shifts (e.g., changes in environment), adversarial learning strategies are employed to disentangle intent-driven features from nuisance variables. Following the Domain-Adversarial Neural Network (DANN) [236] framework, we define an objective that seeks a saddle point [237, 228] for the parameters of a feature extractor G_m, a label predictor G_y, and a domain/locomotion discriminator G_d:

$$\min_{\theta_f,\theta_y} \max_{\theta_d} \mathcal{L}_y(G_y(G_m(\mathbf{x})), y) - \alpha \mathcal{L}_d(G_d(G_m(\mathbf{x})), d), \tag{6.1}$$

where α serves as a trade-off parameter between classification accuracy and domain invariance. Similar adversarial architectures have been adapted in systems like MH-Track [100] to isolate hand-tracking dynamics from concurrent gait interference. By reverse-optimizing the domain loss $\mathcal{L}_d$ with GRL [238], the system effectively "strips" dominant biometric or environmental signatures, thereby enabling the emergence of robust, subject-agnostic latent features.

6.2.4 Related Work

WiFi sensing has emerged as a cornerstone of ambient intelligence [239], leveraging CSI to capture fine-grained environmental variations induced by human motion. By analyzing these signal perturbations, existing literature has demonstrated the efficacy of CSI for diverse tasks, including tracking daily activities within complex floor maps [206]. Specialized frameworks like WiPT-Hand [194] have achieved hand-gesture detection using single-link configurations, while WiDar [240] and WiTraj [241] utilize velocity and phase-difference analysis for high-accuracy trajectory estimation.

To achieve higher spatial precision, mmWave MIMO radar has been employed for

sophisticated sensing tasks. Hand gesture detection [99] is well studied using mmWave radar. Systems such as mmRidge [242] and mmSnap [243] utilize high-resolution Doppler and spatial frequency measurements to identify dense crowd flow structures and provide instantaneous target position estimates. Single-radar configurations have also enabled privacy-preserving crowd analytics [244] and contactless monitoring of vital signs, such as Heart Rate Variability (HRV) through minute chest movement decomposition [245]. Despite these advancements, unimodal systems often encounter an "interference bottleneck" when subtle gestures are masked by dominant full-body motion. While specialized systems like MHTrack [100] employ transformer-based encoders to isolate hand micro-motions during walking, they remain limited by NLoS sensitivity and a lack of generalization across diverse locomotions such as squatting or turning. Multimodal benchmarks, such as MM-Fi [121], enhance robustness by synergizing diverse modalities including mmWave, WiFi CSI, LiDAR, and RGB-D camera. However, they do not provide locomotion-agnostic frameworks and leave real-time edge deployment largely unaddressed.

The proposed system addresses these limitations through targeted architectural choices. While PointNet++ [98] pioneered unordered point-set processing, its reliance on ball-query operations is computationally expensive and ill-suited for sparse mmWave data. Conversely, RadHAR [95] demonstrates the efficiency of voxelizing point clouds into occupancy grids. We extend this paradigm by introducing velocity-aware voxelization to stabilize recognition under locomotion. Our voxel-based approach ensures deterministic inference latency and leverages optimized 3D-convolution kernels for edge hardware acceleration. By integrating these features into a triple-adversarial framework, AgnoGest achieves superior locomotion-agnostic accuracy compared to existing unimodal and non-adversarial benchmarks.

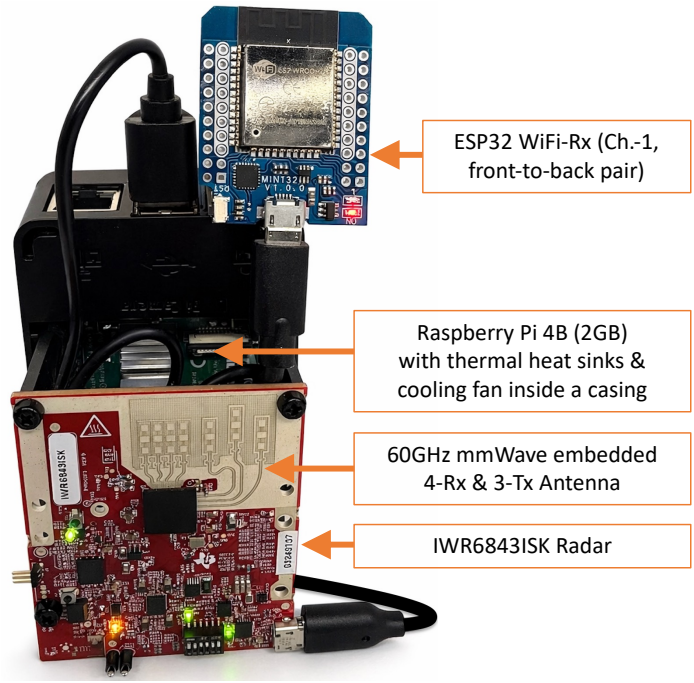


Fig. 48.: Proposed portable and multimodal edge-sensing platform featuring a 60GHz IWR6843ISK mmWave radar and ESP32 WiFi-CSI receiver integrated with a Raspberry Pi 4B.

6.3 Proposed System

We propose AgnoGest, a locomotion-agnostic hand gesture recognition platform comprising a Raspberry Pi 4B edge device with only 2GB RAM, two pairs of ESP32 microcontrollers for CSI-collection-enabled WiFi transmission, and a TI IWR6843ISK mmWave radar. This integrated platform, as shown in Fig. 48, collects CSI and mmWave point-cloud data and is capable of running a stripped-down optimized version of our proposed triple-adversarial model for real-time gesture classification while the person is in motion.

6.3.1 WiFi CSI Collection, Preprocessing and Machine Learning

6.3.1.1 CSI Data Collection

We deploy two pairs of ESP32 devices around the data collection region, as shown in Fig. 51:

Front-to-Back (FB) pair: The line of sight (LoS) of this pair is aligned with the radar's LoS and placed across the front and back of the standing-still volunteer. It is designed to capture gesture and locomotion dynamics in LoS.

Left-to-Right (LR) pair: This pair is aligned along the volunteer's walking axis and positioned across the left and right sides of the standing-still volunteer. It is designed to be sensitive to locomotion dynamics and is therefore primarily utilized for adversarial purposes.

The system utilizes an ESP32-based receiver [246] with CSI-Pi [195] to capture CSI packets transmitted at a rate of 100 Hz.

6.3.1.2 CSI Preprocessing Pipeline

Given the inherent noise in wireless environments, we preprocess the CSI amplitudes as shown in the previous chapters, i.e., by removing null-subcarriers, denoising with the Hampel filter [247] and rolling mean over 10 CSI frames, perform PCA to reduce the dimensionality by taking the first 32 subcarriers, and taking sliding window of 50 CSI frames with a stride of two (500 ms of activity snapshot with 96% overlapping frames) to capture sufficient temporal context for rapid hand gestures.

6.3.1.3 CSI-Based Deep Learning Architecture

To maintain edge efficiency, we employ two lightweight parallel branches designed to extract auxiliary temporal features from the dual-link WiFi streams. These

branches provide essential complementary information to the mmWave stream, ensuring recognition stability in NLoS scenarios within the triple-adversarial framework. Each link-specific network is structured as follows:

- *Temporal Convolution:* Two consecutive 1D convolutional layers, with kernel sizes of 3 and 5, process the 32 Principal Components (PCs). This stage captures local temporal correlations within the 500 ms CSI windows. Batch normalization is applied between layers to stabilize training and prevent overfitting. The resulting flattened feature map $\mathbf{f}_{csi}$ serves as the common input for both gesture and adversarial classification.

- *Gesture Classifier:* The map $\mathbf{f}_{csi}$ is passed through three ReLU-activated dense layers (256, 128, and 64 nodes). It yields the gesture-specific output $g_{csi_{FB}}$ and $g_{csi_{LR}}$, representing the link's contribution to the final classification. Dropout layers (rates 0.7 and 0.3) are utilized to enhance generalization.

- *Locomotion-Adversarial Classifier:* Operating in parallel, this branch seeks to neutralize locomotion-specific signatures, giving out $l_{csi_{FB}}$ and $l_{csi_{LR}}$ for the two WiFi links. To achieve this, a GRL node is inserted between the convolution output $\mathbf{f}_{csi}$ and the adversarial dense layers.

Specifically, for each WiFi link $j \in \{FB, LR\}$, we define a feature extractor $G_{f,j}$ with parameters $\theta_{f,j}$ that produces the representation $\mathbf{f}_{csi,j}$. The adversarial training for the CSI branches is governed by a min-max optimization objective. We seek to minimize the gesture classification loss $\mathcal{L}_y$ while simultaneously maximizing the locomotion classification loss $\mathcal{L}_d$ to ensure $\mathbf{f}_{csi,j}$ is locomotion-invariant. Following the principles of domain-adversarial training [238], the total loss for a CSI branch is

defined as:

$$E(\theta_{f,j}, \theta_{y,j}, \theta_{d,j}) = \mathcal{L}_y(G_{y,j}(\mathbf{f}_{csi,j}), y) - \alpha\mathcal{L}_d(G_{d,j}(\mathbf{f}_{csi,j}), d), \tag{6.2}$$

where $\theta_{y,j}$ and $\theta_{d,j}$ are the parameters of the gesture and locomotion classifiers, and α is the adaptation factor.

To implement this via standard backpropagation, the GRL is defined as a pseudo-function $R_\alpha(\mathbf{x})$ with the following characteristics:

$$R_\alpha(\mathbf{x}) = \mathbf{x}, \quad \frac{dR_\alpha}{d\mathbf{x}} = -\alpha\mathbf{I}, \tag{6.3}$$

where $\mathbf{I}$ is the identity matrix. During the forward pass, the GRL acts as an identity transform, allowing the locomotion-adversarial head $G_{d,j}$ to receive $\mathbf{f}_{csi,j}$ for classification. During the backward pass, however, the GRL multiplies the gradient from the locomotion loss by $-\alpha$ before it reaches the feature extractor $G_{f,j}$. This inversion ensures that the emerging CSI features become indistinguishable across different locomotion states, successfully stripping away the dominant motion interference from the WiFi temporal stream before it is passed to the gated fusion module.

6.3.2 mmWave Point-Cloud-based Machine Learning

We use TI IWR6843ISK mmWave radar to generate a point-cloud formation of the performed actions and then train our model with a temporal deep learning approach. We design the system to consume a small amount of energy, RAM, and processing power while maintaining real-time inference on a Raspberry Pi device.

6.3.2.1 Radar Configuration and Point-Cloud Data

Operating at a 60 GHz carrier frequency with a 50 MHz/μs ramp slope, the radar system utilizes 96 chirps per frame and dual TX antennas to achieve a range

Table 25.: Radar Configuration and Performance Specifications

Parameter	**Value**
Transmitter (TX) Configuration	
Carrier Frequency (f_c)	60 GHz
Ramp Slope (S)	$50MHz/\mu s$
Sweep Bandwidth (Useful)	2133.33 MHz
Total Bandwidth (BW_{total})	3984 MHz
Chirp Duration (T_{ramp})	$55\mu s$
Idle Time	$7\mu s$
Chirp Cycle Time (T_c)	$62\mu s$
Receiver (RX) Configuration	
ADC Sampling Rate	6 Msps
ADC Samples per Chirp	256
Number of RX Channels	4
Number of Chirps per Frame	96
Range FFT Size	256
Doppler FFT Size	32
System Performance	
Range Resolution (d_{res})	0.0703 m
Maximum Unambiguous Range (d_{max})	6.4 m
Velocity Resolution (v_{res})	0.63 m/s
Maximum Velocity (v_{max})	20.16 m/s
Frame Periodicity	100 ms (10 FPS)
Duty Cycle	$\approx 5.95\%$

resolution of 7.03 cm and a maximum unambiguous velocity of 20.16 m/s, providing the high-fidelity spatial-temporal data required for robust gesture capture. However, although the radar can be configured for finer range resolution, we choose a higher range resolution of 7.03 cm in exchange for finer velocity resolution of 0.63 m/s with a higher detection range of 6.4 m. The configuration details are tabulated in Table 25 and depicted in the chirp diagram in Fig. 49.

Each $62\mu s$ chirp cycle (including $55\mu s$ ramp and $7\mu s$ idle time) contributes to a compact 5.95 ms active burst duration per frame. By maintaining a 100 ms frame periodicity (10 frames/second), the system effectively captures gestures while operating at a low 5.95% duty cycle, ensuring energy-efficient performance suitable for

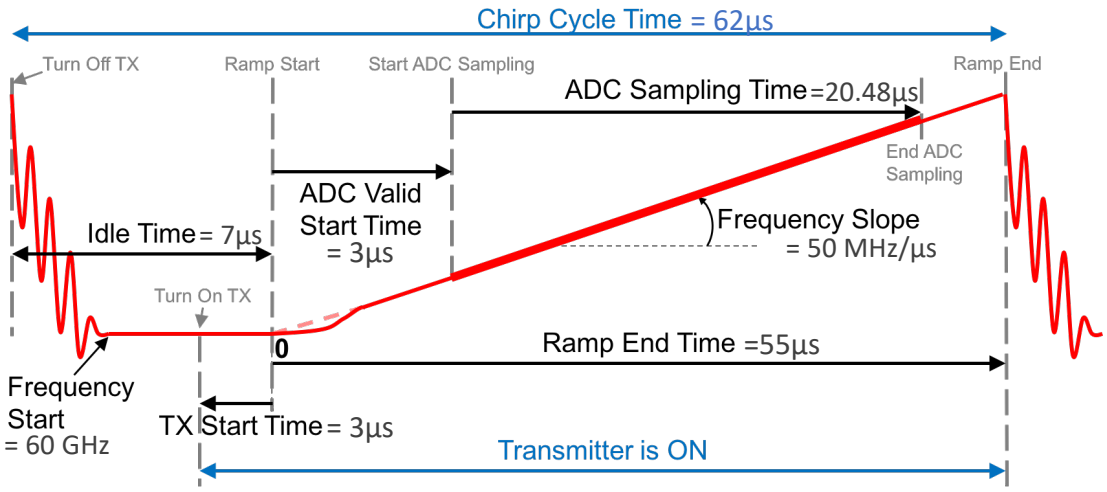


Fig. 49.: Chirp configuration for the used TI-IWR6843ISK

continuous monitoring even using resource-constrained edge devices.

Range FFT converts time-domain samples into spatial bins, while Doppler FFT estimates radial velocity per point. The resulting point cloud includes the metadata (x, y, z, v, SNR).

Our open-source platform [229] supports firmware flashing via UniFlash [248] and flexible configuration for both the IWR1443 and IWR6843 series radars through custom configuration files.

6.3.2.2 Point-cloud Voxelization and Grid Mapping

To accommodate the sparse point-cloud data to the spatiotemporal nature of the problem, the 3D point cloud is transformed into a fixed-size volumetric representation through voxelization. This process discretizes the continuous coordinate space into a structured $10 \times 32 \times 32$ grid ($N_z \times N_y \times N_x$), providing a consistent input format for convolutional layers while preserving spatial relationships.

The voxelization process maps each point $p_i(x_i, y_i, z_i)$ from the frame's point cloud into a specific grid cell (i_x, i_y, i_z) based on a defined physical bounding box.

Given a physical region defined by minimum bounds $(x_{min}, y_{min}, z_{min})$ and maximum bounds $(x_{max}, y_{max}, z_{max})$, the integer indices for each point are calculated by Eq. 2.17, where the cell resolutions are defined as $\Delta_x = \frac{x_{max}-x_{min}}{32}$, $\Delta_y = \frac{y_{max}-y_{min}}{32}$, and $\Delta_z = \frac{z_{max}-z_{min}}{10}$. Points falling outside these spatial bounds are discarded as noise or out-of-range artifacts.

Because multiple radar points can fall into a single voxel, we aggregate features to represent the cell's content using two separate input channels:

- **Channel 1 (Occupancy):** A binary value (1 if the voxel contains at least one point, 0 otherwise), representing the spatial structure of the gesture.
- **Channel 2 (Velocity):** The Mean Differential Velocity (v_d) of all points within the cell, as defined by Eq. 2.16, providing a localized Doppler signature that is robust to locomotion.

The choice of a $10 \times 32 \times 32$ grid is optimized for human gesture recognition in our evaluated scenarios. The higher horizontal (x) and vertical (y) resolution captures the fine-grained trajectory of hand movements, while the lower depth (z) resolution is sufficient to distinguish the hand from the torso and background. As already discussed in Section 2.2.5, this specific discretization of voxelization significantly reduces the memory footprint and computational complexity, enabling real-time inference on resource-constrained edge devices in contrast to other complex point-cloud-based systems.

To improve cross-person gesture detection under locomotion, we scale the points by a random factor within $[0.9, 1.1]$ and apply a translation offset within $[-0.05, 0.05]$. This quadruples the original training dataset, allowing the model to generalize across diverse physical attributes and variable gesture positions. This strategy is supported by studies [249], which demonstrate that synthetic data augmentation significantly

enhances system performance.

6.3.2.3 CNN-BiLSTM Based Adversarial Learning Architecture

The mmWave radar branch utilizes a hybrid CNN-BiLSTM architecture to process dual-channel point-cloud data, incorporating both occupancy and differential velocity information. To maintain temporal consistency with the WiFi sensing windows, the model processes successive voxel sequences over a 500 ms inference window.

Spatial Feature Extraction (CNN) from Point-Cloud: The raw point-cloud data is voxelized and reshaped into a sequence of volumetric tensors. The architecture employs Time-Distributed (TD) layers to apply spatial convolutions independently across the temporal dimension:

- *Time-Distributed 3D Convolution:* A TD-wrapped 3D convolutional layer with 32 filters of size $3 \times 3 \times 3$ and ReLU activation extracts local spatial-temporal features. This ensures that the significance of gesture dynamics is captured across successive voxel windows.

- *Volumetric Downsampling:* A TD-wrapped 3D MaxPooling layer follows the convolution to reduce the dimensionality of the feature maps while retaining dominant gesture signatures. The resulting spatial features are flattened into a sequence of vectors for temporal modeling.

Temporal Modeling (BiLSTM): To capture the trajectory and sequential nature of gestures, a Bidirectional Long Short-Term Memory (BiLSTM) network processes the flattened spatial features. By analyzing the input in both forward and backward directions, the BiLSTM provides a full context of the gesture's trajectory:

$$\overrightarrow{\mathbf{h}}_t = \text{LSTM}_{fwd}(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}), \quad \overleftarrow{\mathbf{h}}_t = \text{LSTM}_{bwd}(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}) \tag{6.4}$$

The final hidden state $\mathbf{h}_{mm} = [\overrightarrow{\mathbf{h}}_T; \overleftarrow{\mathbf{h}}_0]$ represents a robust temporal encoding of the point-cloud data.

Adversarial Disentanglement: Similar to the CSI branches, this encoding is divided into two parallel streams:

- *Gesture Branch (g_{mm}):* The encoding $\mathbf{h}_{mm}$ is processed through three ReLU-activated dense layers (256, 128, and 64 nodes) with dropouts to yield the gesture-specific representation.

- *Locomotion-Adversarial Branch (l_{mm}):* To ensure the radar features remain locomotion-invariant, a third GRL node is inserted before the adversarial classification head.

Specifically, let $G_{f,m}$ be the mmWave feature extractor with parameters $\theta_{f,m}$. The adversarial optimization follows the pseudo-function $R_\alpha(\mathbf{x})$ defined in Eq. 6.3. During backpropagation, the GRL inverts the locomotion classification gradient, forcing the spatial-temporal feature extractor to suppress biometric gait patterns and dominant locomotion signatures within the point-cloud latent space. This completes the triple-adversarial setup, ensuring that all three sensor encoders, i.e., dual-link WiFi and mmWave radar, output features that are highly discriminative for gestures but indistinguishable across different locomotion states.

6.3.3 Dual-Attention Gated Fusion Framework

Our triple-adversarial architecture is structured around three parallel input branches (i.e., one mmWave radar stream and two WiFi CSI streams (FB and LR)) that expand into a six-branch processing pipeline. In this framework, three branches are dedicated to independently penalizing locomotion signatures via GRL, while the remaining three branches extract features for gesture recognition. This decentralized

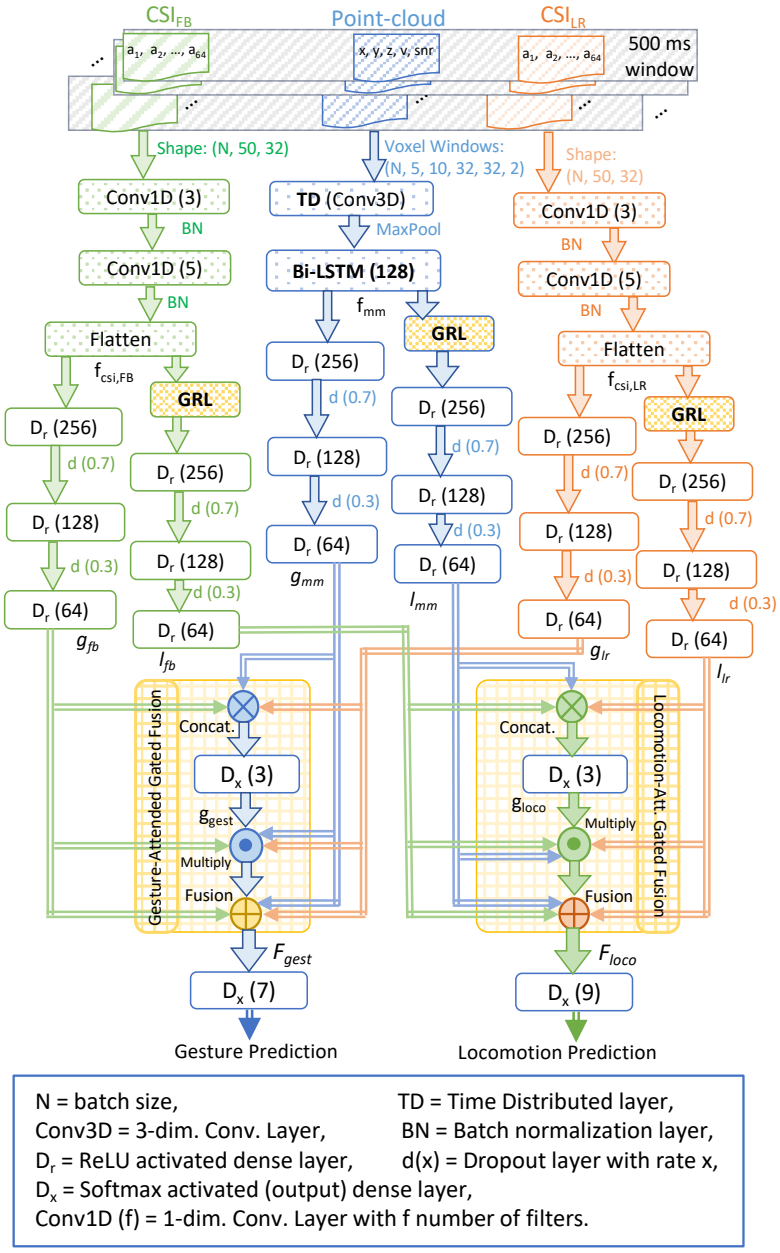


Fig. 50.: Multimodal attention-based dual-gated triple-GRL adversarial model with two WiFi links yielding CSI and one mmWave radar yielding point-cloud voxel data.

adversarial approach ensures that the global gesture classifier learns an intent-driven representation, allowing the system to maintain high accuracy even under unseen locomotion domains such as treadmill walking and squatting. Thereby, in the context of *AgnoGest*, our framework addresses three fundamental multimodal machine learning challenges [232]: representation, translation, and fusion.

6.3.3.1 Joint Representation

We seek a joint latent space where features from the mmWave radar ($\mathbf{x}_m$) and the dual-link WiFi CSI ($\mathbf{x}_{w_{FB}}$ and $\mathbf{x}_{w_{LR}}$) are projected. Unlike simple joint embeddings, our framework utilizes three distinct modality-specific encoders to produce separated feature sets for gestures ($\mathbf{g}$) and locomotion ($\mathbf{l}$):

$$\begin{aligned} \{\mathbf{g}_{mm}, \mathbf{l}_{mm}\} &= G_m(\mathbf{x}_m; \theta_m), \\ \{\mathbf{g}_{fb}, \mathbf{l}_{fb}\} &= G_{fb}(\mathbf{x}_{w_{FB}}; \theta_{fb}), \\ \{\mathbf{g}_{lr}, \mathbf{l}_{lr}\} &= G_{lr}(\mathbf{x}_{w_{LR}}; \theta_{lr}), \end{aligned} \tag{6.5}$$

where G_m, G_{fb}, and G_{lr} represent the CNN-BiLSTM and Conv1D architectures for radar and CSI, respectively. All inputs are time-synchronized into 500 ms windows to ensure temporal alignment across the multimodal manifold.

6.3.3.2 Multimodal Translation

Translation in *AgnoGest* involves mapping modality-specific latent features into a common space where cross-modal dependencies can be modeled. For each category $k \in \{\text{gesture, locomotion}\}$, we translate the three independent expert vectors into a joint context vector $\mathbf{C}_k$:

$$\mathbf{C}_{gest} = [\mathbf{g}_{mm}; \mathbf{g}_{fb}; \mathbf{g}_{lr}], \quad \mathbf{C}_{loco} = [\mathbf{l}_{mm}; \mathbf{l}_{fb}; \mathbf{l}_{lr}]. \tag{6.6}$$

This translation step allows the subsequent attention mechanism to perceive the global state of the environment, identifying which sensor provides the most reliable signal-to-noise ratio (SNR) for the current subject and locomotion state.

6.3.3.3 Dual-Attention Gated Fusion

To enable locomotion-agnostic gesture recognition, we employ a dual-attention gated fusion mechanism that independently weighs the contributions of the radar and dual-link CSI modalities for both gesture and locomotion heads. As depicted in Fig. 50, this approach acts as a learnable filter to prioritize reliable sensors in real-time. For each context $\mathbf{C}_k$, a gating weight vector $\mathbf{w}_k = [w_{mm}, w_{fb}, w_{lr}]$ is derived via a Softmax-activated dense layer:

$$\mathbf{w}_k = \text{Softmax}(\mathbf{W}_k \mathbf{C}_k + \mathbf{b}_k), \tag{6.7}$$

where $\sum w_i = 1$. This attention mechanism enforces a competitive relationship between the three branches. The final fused representations, $\mathbf{F}_{gest}$ and $\mathbf{F}_{loco}$, are computed as a weighted sum of the expert features:

$$\begin{aligned} \mathbf{F}_{gest} &= \sum_{i \in \{mm, fb, lr\}} w_{i,gest} \cdot \mathbf{g}_i, \\ \mathbf{F}_{loco} &= \sum_{i \in \{mm, fb, lr\}} w_{i,loco} \cdot \mathbf{l}_i. \end{aligned} \tag{6.8}$$

This dual-gated formulation ensures that the gesture classification head receives a refined feature set where noisy signatures, such as multi-path interference in CSI or occlusion in radar, are suppressed by the higher-performing modalities. Simultaneously, the locomotion gate ensures the GRL-backpropagation is driven by the most discriminative locomotion features, effectively stabilizing the performance floor across diverse volunteers.

6.3.4 Raspberry Pi Integration

Our proposed system is capable of running on a Raspberry Pi 4B with only 2GB RAM as an edge device. It acts as the central hub for the system, performing three critical roles:

1. *Data Acquisition*: It receives 3D point-cloud data from the radar via a UART serial interface using our developed open-source platform mmWaveSimplePC [229].

2. *CSI Aggregation*: It collects and synchronizes WiFi signal data through CSI-Pi [195] platform from the connected ESP32 receiver modules having ESP32-CSI-Tool [246].

3. *Edge Computing*: It runs a stripped-down version of the trained multimodal machine learning model for real-time inference.

Fig. 48 shows our setup of the Raspberry Pi kit, where the Raspberry Pi is put inside a plastic casing including heat-sinks on the heating chips and a cooling fan on the top of it. The mmWave radar is attached to the kit so that its antenna can be focused upward while the device can be made to stand firm. The two ESP32 chips are also connected to the USB UART serial. All of these devices are powered on through a stable 5V-3.5A power supply to the Raspberry Pi. Through the USB ports, the ESP32 modules draw 3.3V-5V at 100mA-300mA (0.35W to 1.35W) with a peak draw of 2W (over 600mA) while transmitting WiFi packets. The radar draws 1.15W to 1.5W of power with a 5.95% duty cycle, as already explained in the subsection 6.2.2.1. Overall, the enclosed system is portable, plug-and-play when moved to another location, and it is remotely serviceable through a reverse tunneling virtual private network [192], if it is connected to the internet, given the scenario of a remote home monitoring

system deployment.

Lightweight Edge Optimization: The AgnoGest model, as presented in Fig. 50, consists of approximately 2.87 million parameters with a disk footprint of 10.94 MB. To optimize for resource-constrained environments, we utilize 32-bit floating-point precision for model parameters while maintaining 8-bit integer quantization for input voxel intensities. Given the ARM-based architecture of the Raspberry Pi and the absence of a dedicated GPU, the system's SWAP memory is upgraded to 2 GB to ensure sufficient headroom for the concurrent loading of the model architecture and high-dimensional multimodal test tensors. These optimization strategies facilitate uninterrupted, real-time inference on synchronized CSI and point-cloud sequences, ensuring that the ensemble can leverage the complementary strengths of spatially distributed links even under strict hardware limitations.

6.4 Evaluation

6.4.1 Dataset

To evaluate the proposed system, extensive CSI and point-cloud data are collected from four volunteers in a controlled lab environment. We place the Raspberry Pi with the mmWave radar and the receiver end of the FB WiFi link on a 3 ft high table. The transmitter end of the FB pair is placed 2 m far from the receiver end, while the volunteer primarily stands at the middle of this link, having a cross-section with the other. The LR WiFi link is placed 5 m away on either perpendicular ends of the volunteer, as shown in the Fig. 51. The walkway for the locomotion walk and walk-back is 3 m long across the LR WiFi link. The FoV of the radar is $120 Doctor of Philosophy$, which renders the farther part of the walkway out of the FoV scope of the radar, emphasizing the need for a multimodal system incorporating

more ubiquitous WiFi signals. In reality, this kind of dynamic walk can encompass a larger space, requiring a more robust alliance of the WiFi signal, alongside the finer-scoped recognition of the mmWave radar.

We let the volunteers perform seven gestures in combination with nine different locomotions as depicted in Fig. 52. After removing a few non-practical locomotion-gesture pairs, we get 55 different pairs of actions as tabulated in Fig. 53. One of the volunteers, V1, performs all of these 55 pairs five times each for five seconds repeatedly in a round-robin fashion around the action pairs with a three-second pause in between successive action pairs. The total session duration is about 55 minutes. We use the first three repetitions of action pairs involving only stand, walk-back, turn, and treadmill-back locomotions for training, while the rest of the dataset is used for testing. The other volunteers perform separate training and testing sessions, where the training session only has the aforementioned four locomotions and the testing session has all of the nine locomotions, giving us an opportunity to assess the trained model on both seen and unseen locomotions to recognize the same set of gestures. Further challenging our system, we design the locomotions with counter-NLoS situations, like stand and stand-back, walk and walk-back, squat and squat-back, and treadmill and treadmill-back, while the turn movement keeps switching the LoS and NLoS scenario constantly within the locomotion.

6.4.2 CSI based Learning

When evaluated using the proposed Conv1D-based architecture with batch normalization and dense layers separated by dropout layers, CSI data from individual WiFi links gives an average gesture classification accuracy of 33.5% with the FB pair and 29.5% with the LR pair. Temporal variations across data collection sessions are observed in both WiFi links, as the volunteer data were collected on different days.

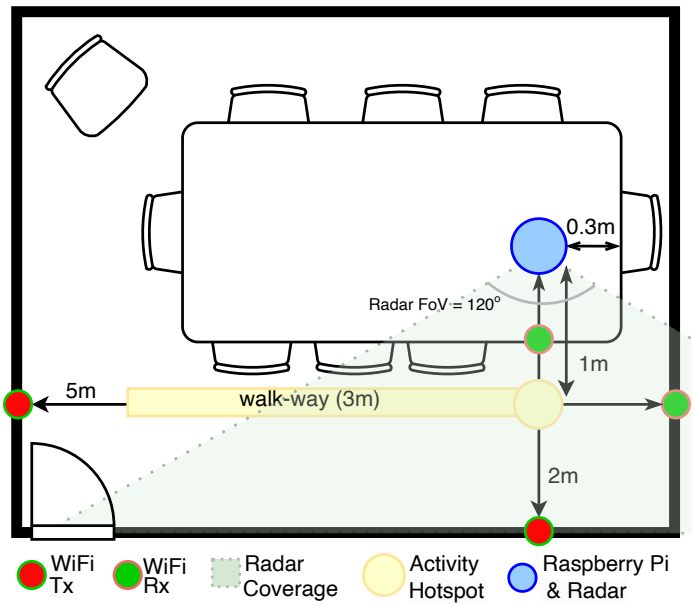
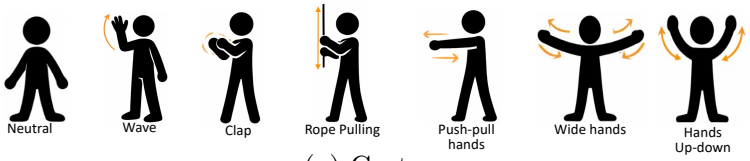


Fig. 51.: Floor map of the data collection and test-bed area.



(a) Gestures

(b) Locomotions

Fig. 52.: Symbolic representation of all seven gestures and nine locomotions.

Locomotions \ Gestures	None	Wave	Clap	Rope Pulling	Push-pull Hands	Wide Hands	Hands Up-down
Stand	None-Stand	Wave-Stand	Clap-Stand	Rope-Stand	Pushpull-Stand	Widehands-Stand	Handsupdown-Stand
Walk-back	None-Walkback	Wave-Walkback	Clap-Walkback	Rope-Walkback	Pushpull-Walkback	Widehands-Walkback	Handsupdown-Walkback
Turn	None-Turn	Wave-Turn	Clap-Turn	-	Pushpull-Turn	-	Handsupdown-Turn
Treadmill-back	None-Treadmillback	Wave-Treadmillback	Clap-Treadmillback	Rope-Treadmillback	Pushpull-Treadmillback	Widehands-Treadmillback	Handsupdown-Treadmillback
Stand-back	None-Standback	Wave-Standback	Clap-Standback	Rope-Standback	Pushpull-Standback	Widehands-Standback	Handsupdown-Standback
Walk	None-Walk	Wave-Walk	Clap-Walk	Rope-Walk	Pushpull-Walk	Widehands-Walk	Handsupdown-Walk
Treadmill	None-Treadmill	Wave-Treadmill	Clap-Treadmill	Rope-Treadmill	Pushpull-Treadmill	Widehands-Treadmill	Handsupdown-Treadmill
Squat	None-Squat	-	-	-	Pushpull-Squat	Widehands-Squat	Handsupdown-Squat
Squat-back	None-Squatback	-	-	-	Pushpull-Squatback	Widehands-Squatback	Handsupdown-Squatback

Training Locomotions

Fig. 53.: Gesture and Locomotion Pairs in the Dataset

As a result, we see 61.43% and 56.68% average gesture detection accuracy for seen volunteers with the FB and LR WiFi links, respectively, while their unseen volunteer accuracy lags behind to 24.2% and 20.45%. On the other hand, WiFi links perform better for locomotion detection, yielding 65.09% and 68.3% locomotion detection accuracy with the FB and LR links. This suggests that CSI contains useful locomotion-related information that can complement the adversarial framework.

Note that while a better accuracy can be achieved with a more complicated and larger model, we stick with a simplistic model with only two Conv1D and three dense layers to keep it optimized for edge inference with resource constraints. Moreover, CSI is evaluated to be useful in NLoS scenarios [250], like most of the locomotions depicted in our experiment sessions. In the end, the performance of the system improves with the adversarial fusion of the fine-grained gesture detection capability of mmWave radar point-cloud data, as explained in the following subsections.

6.4.3 Point-cloud based Learning

We use point-cloud-based voxel formation of the entire field of view of the radar as per Eq. 2.17 along with differential velocity as explained in Eq. 2.16.

For the initial approach, we start with a simplified training process. We utilize a deep learning model similar to the point-cloud branch illustrated in Fig. 50. This

model is trained exclusively on point-cloud-based voxels from the first volunteer's "stand-still" gestures. While this setup achieves an in-domain accuracy of 93.64% for the same volunteer and same locomotion, i.e., stand, performance decreases when NLoS (e.g., stand-back) and locomotion (e.g., walk, walkback, turn, squat, squat-back, treadmill, and treadmill-back) conditions are introduced. We observe gesture detection accuracies varying from 12.61% (turn locomotion) to 40.64% (stand-back NLoS condition) for the adverse situations, making the overall average go down to 31.29%.

We then train our model with four different locomotion/NLoS situations, i.e., stand, walkback, turn, and treadmillback. It increases the accuracy of the inference model over the other five unseen locomotions, i.e., stand-back, walk, squat, squat-back, and treadmill, varying from 22.91% (walk locomotion) to 50.32% (stand-back NLoS situation), while the in-domain accuracy averages down to 86.51% due to greater variance in the training data in comparison to the previous single-locomotion training data. Notably, training the model with all 55 locomotion-gesture pairs of the first volunteer results in 87.95% test accuracy, demonstrating that the voxel-based features alone for the gestures and locomotions are highly learnable.

To this end, we introduce differential velocity as a second channel to our input three-dimensional convolutional layer, which improves the gesture detection accuracy with the single-locomotion (stand) trained model to 91.95% in-domain and 16.76% (turn) to 39.67% (standback) out-of-domain, while the four-locomotion training model gets 87.8% in-domain and 23.5% (walk) to 46.94% out-of-domain accuracy. The direct accuracy gain from the velocity channel is modest when evaluated in the point-cloud branch in isolation; however, its contribution becomes substantially more impactful within the adversarial framework, where the richer Doppler-derived features provide the locomotion discriminator with a more separable signal, ultimately

enabling stronger gesture-locomotion disentanglement. This design choice adds negligible model weight while meaningfully enriching the feature space available to the adversarial heads, consistent with findings in prior works [251, 252, 253].

However, when we introduce unseen volunteers' data, the point-cloud-only model yields even poorer accuracy, averaging around 20.5% across all nine locomotions for the unseen volunteers. With the aforementioned augmentation through scaling and offsetting of the original data, the model achieves improved robustness across volunteers. As Table 26 shows, the PC only model achieves above 50% gesture detection accuracy for the same person used in training, while only 28.05% for unseen persons, averaging 34.35% across all volunteers. With the introduction of an adversarial approach to the locomotion features, we then aim to mitigate the effect of different locomotions.

6.4.4 Point-Cloud based Adversarial Model

Using dual-channel point-cloud input for voxels and differential velocity, we evaluate the GRL-based adversarial framework using only the mmWave point-cloud branch. When evaluated on held-out data from a seen volunteer, the model achieves 90.3% gesture classification accuracy. In contrast, performance on unseen volunteers still lacks high accuracy, varying from 61.8% to 81% across different locomotion conditions, with an average gesture recognition accuracy of 79% over nine locomotions.

6.4.5 Multimodal Adversarial Model

We provide data from both WiFi links and mmWave point-cloud voxels and differential velocity to the model described in Fig. 50 to improve gesture recognition performance under unseen locomotion and unseen volunteer settings. With a single-volunteer training dataset, we achieve over 93% gesture classification accuracy for the

Table 26.: Cross-Subject Performances of Different Sub-Modules of AgnoGest. Bold values indicate seen-subject (in-domain) testing; all others are cross-subject generalization. Vi denotes the i^{th} volunteer.

(a) Non-adversarial baselines: CSI_{FB} (avg. 33.51%), CSI_{LR} (avg. 29.51%), Point-Cloud Only (avg. 34.35%) models.

Train Set ↪		**CSI_{FB}-Only ML**				**CSI_{LR}-Only ML**				**Point-Cloud (PC)**			
		V1	V2	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4
Test Set	**V1**	**77.1**	27.6	28.6	25.8	**74.2**	20.8	20.7	21.2	**57.0**	28.1	24.8	27.4
	V2	24.2	**53.3**	23.7	22.9	21.2	**48.4**	19.9	20.2	24.7	**50.2**	31.3	28.4
	V3	20.1	25.8	**56.9**	24.7	19.6	18.7	**51.3**	23.1	24.2	31.0	**52.1**	29.7
	V4	21.2	23.6	22.1	**58.4**	18.7	19.6	21.7	**52.8**	26.8	29.7	30.6	**53.7**
Average		35.7	32.6	32.8	32.9	33.4	26.9	28.4	29.3	33.2	34.7	34.7	34.8

(b) Adversarial models: PC-Adversarial (avg. 79%), PC+CSI Adversarial (avg. 87.49%).

Train Set ↪		**PC-Adversarial**				**PC+CSI Adversarial**			
		V1	V2	V3	V4	V1	V2	V3	V4
Test Set	**V1**	**95.3**	68.2	79.5	80.1	**96.7**	85.7	88.5	86.9
	V2	74.6	**92.1**	78.2	76.4	82.5	**95.3**	84.6	84.6
	V3	61.8	68.4	**90.3**	76.3	83.7	82.8	**94.6**	85.4
	V4	68.5	79.5	81.0	**93.9**	81.6	86.7	86.8	**93.3**
Average		75.0	77.1	82.3	81.7	86.1	87.6	88.6	87.6

same volunteer's test dataset, while the unseen volunteers' data yield accuracy from 81.57% to 88.5% with an average of 84.99%.

To further mitigate the generalization gap between seen and unseen volunteers, even after applying scaling and offset-based augmentation, we train the model using data from multiple volunteers. Fig. 54 shows continuous improvement of the system as more volunteer datasets are added. While a model trained on a single volunteer achieves an overall average accuracy of 87.49%, models trained on two, three, and four volunteers achieve average accuracies of 90.02%, 94.29%, and 97.91%, respectively, across all volunteers and all nine locomotion conditions. As more volunteers are incorporated, the seen-volunteer dataset initially exhibits a slight reduction in accuracy (94.7% compared to 95% for the single-volunteer model) but subsequently improves, while unseen-volunteer performance continues to increase, as shown by the trending points in Fig. 54. The slight reduction in seen-volunteer accuracy likely stems from the increased variability introduced by incorporating additional volunteer data, while unseen-volunteer performance continues to improve as more diverse training examples become available. For the remaining unseen volunteer in the three-volunteer training configuration, the model achieves an average accuracy of 87.64%, ranging from 86.73% to 89.53%.

6.4.6 Evaluation of Edge Inference

The AgnoGest model architecture comprises approximately 2.87 million parameters, occupying a base disk footprint of 10.94 MB. When trained using a quantized configuration and a three-volunteer dataset, the resulting saved model occupies 35.4 MB of disk space. Through further optimization and compression, the model size is reduced to 24.7 MB, facilitating performance evaluation on a resource-constrained edge device such as a Raspberry Pi 4B with 2 GB of RAM. To establish performance

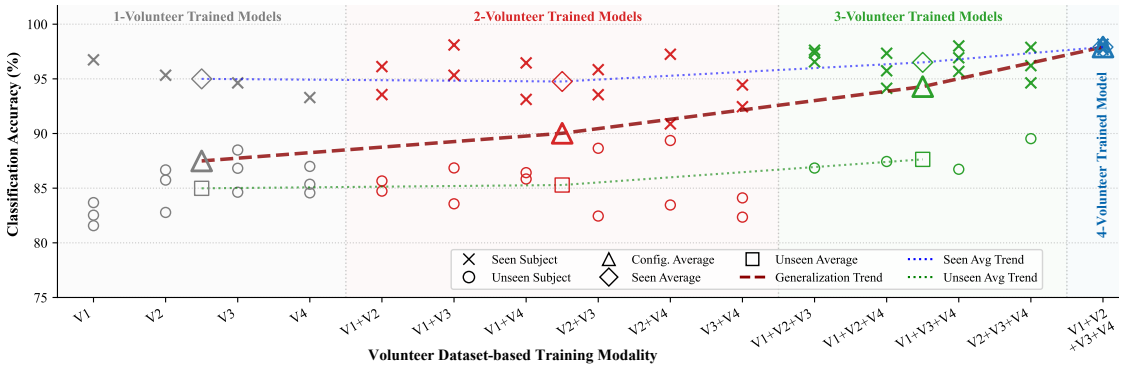


Fig. 54.: Improving gesture recognition accuracy generalization trend with multi-volunteer training sets.

metrics, we executed 100 prediction iterations across four volunteer test datasets. On the Raspberry Pi, the average RAM usage during inference was 432.6 MB, with a corresponding CPU utilization increase of 7.8%. The processing time for synchronized 500 ms CSI and voxel windows was 428.9 ms, resulting in an inference rate of 2.33 fps. For comparative analysis, the model was also evaluated on a 6-core Intel Core i7 2.6GHz MacBook, which yielded an average inference time of 222.09 ms (4.5 fps), 452.1 MB RAM occupancy, and a 2.9% increase in CPU utilization. Given the achieved inference rate of 2.33 fps, the system can continuously process synchronized CSI and voxel streams on resource-constrained hardware. For sustained gestures, majority voting across consecutive predictions may further reduce transient prediction fluctuations. These results demonstrate that AgnoGest enables robust real-time inference on edge hardware.

6.4.7 Comparative Analysis with State-of-the-Art Baselines

To compare the performance of *AgnoGest*, we implemented two representative state-of-the-art frameworks: MHTrack [100] and MM-Fi [121]. As the original source code for these systems was not publicly available at the time of this study, we metic-

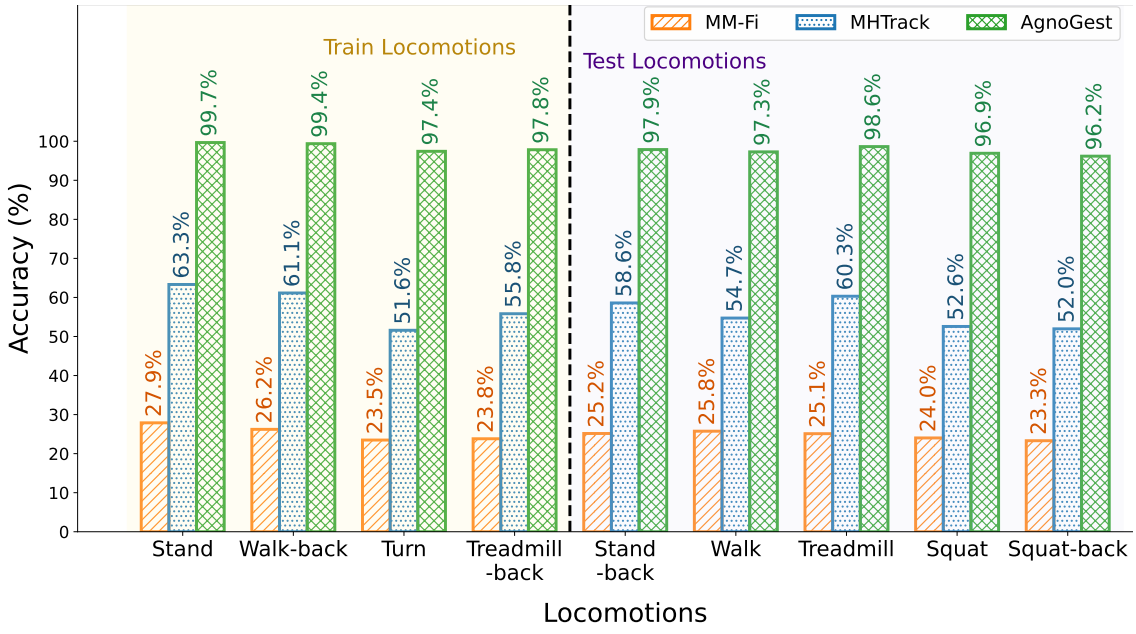


Fig. 55.: Test accuracy on four seen and five unseen locomotions: MM-Fi vs MHTrack vs AgnoGest.

ulously replicated their architectures based on the descriptions provided in their respective publications to ensure a consistent comparison within our experimental environment. The benchmarks are based on models trained on four volunteers data (i.e., no unseen volunteer), while evaluation is performed on five unseen locomotion conditions. The resulting performance comparisons are shown in Fig. 55.

1) MM-Fi (Multimodal Activity Recognition): The MM-Fi [121] framework was adapted to our setup by synergizing mmWave point clouds with WiFi CSI. While the original MM-Fi benchmark utilizes four modalities (i.e., RGB-D, LiDAR, mmWave, and WiFi), we restricted our implementation to the two RF-based modalities relevant to our edge-sensing scope. Under this dual-modal configuration, the MM-Fi baseline yields an average accuracy of only 24.97%, ranging from 23.31% to 27.89% across nine different locomotions. The reduced performance may partially stem from adapting the original quad-modality framework into an RF-only dual-

modality setting, where the original three-antenna WiFi configuration is replaced by our single-antenna ESP32-based FB link. Nevertheless, these findings suggest limitations of non-adversarial fusion approaches under high locomotion-induced signal variability and inter-subject biometric differences in multi-volunteer settings.

2) MHTrack (Locomotion-Resilient Tracking): We implemented the MHTrack pipeline by first performing standard point target tracking to capture the user's body trajectory. Concurrently, we isolated the absolute hand trajectory using a credibility-based wake-up mechanism to distinguish hand points from body noise. To achieve locomotion-agnostic sensing, we applied anchor correction by defining a reference hand anchor based on individual arm lengths to remove body displacement, resulting in a refined relative hand trajectory. For recognition, these trajectories were projected into 2D gesture patterns and processed by ESNet, a two-branch Siamese network [254] using an EfficientNet [255] backbone to compute similarity scores against a one-shot gesture database.

In our multi-locomotion tests, MHTrack achieved accuracies ranging from 51.59% to 63.32%, with an overall average of 56.67% across nine locomotion domains. While MHTrack demonstrated resilience during standard "stand" and "walk-back" states, yielding over 61% accuracy, its performance declined in complex scenarios like squatting or turning.

In contrast, as shown in Table 26 and in Fig. 55, *AgnoGest* substantially outperforms these baselines by achieving an average locomotion-agnostic accuracy of 97.91%. This superior performance is attributed to our triple-adversarial framework, which suppresses locomotion-related interference that otherwise overwhelms the fixed-feature extractors used in MHTrack and MM-Fi.

6.5 Discussion and Future Directions

We presented *AgnoGest*, a locomotion-agnostic and subject-independent multimodal gesture recognition system integrating triple-adversarial learning and dual-attention gated fusion on a resource-constrained edge platform. Through the synergy of velocity-enhanced voxel representations, independent adversarial heads for locomotion suppression, and specialized attention gates for feature fusion, the system effectively reduces the influence of dominant locomotion interference on gesture recognition. Experimental results across a four-volunteer cohort demonstrate robust performance, achieving a final cross-subject accuracy of 97.91% when trained on all subjects. This lightweight framework provides a foundation for privacy-preserving ambient intelligence for connected healthcare and smart-home applications on the edge. Future work can focus on larger and more diverse subject cohorts, multi-user interference scenarios, and deployment across varying room geometries and environmental conditions to further improve generalization and real-world robustness.

CHAPTER 7

CONCLUDING REMARKS

This dissertation investigates the application of wireless sensing technologies for real-life health monitoring, focusing on edge-deployable, privacy-preserving, and low-cost solutions that can be used in homes, clinics, and assisted living environments. The works presented span three sensing modalities, i.e., WiFi CSI, mmWave radar point-cloud data, and their multimodal combination, to address a range of problems including physical rehabilitation tracking, daily activity monitoring of older adults, synthetic data generation for zero-shot learning, and locomotion-agnostic gesture recognition. Throughout this dissertation, we demonstrated how ambient wireless signals, when paired with robust machine learning models and carefully designed preprocessing pipelines, can enable meaningful sensing systems that do not require wearable sensors or camera-based setups.

We began by introducing the foundational principles of wireless sensing, including the signal-level properties of OFDM, RSSI, and CSI for WiFi-based sensing, as well as the physical processing pipeline for mmWave FMCW radar, which includes Range FFT, Doppler FFT, CFAR detection, AoA estimation, and voxelized point-cloud formation. The literature review identified key trends and open challenges in applications such as human activity recognition, gesture detection, health monitoring, localization, and data augmentation. These insights guided the core contributions of this dissertation and informed the design choices in each system.

In Chapter 3, we proposed and implemented *Wi-PT-Hand*, the first device-free, end-to-end system for tracking small-scale hand and finger movements in physical

therapy exercises using low-cost ESP32 microcontrollers. We presented a complete pipeline with components for activity segmentation, classification using a hierarchical Dense Neural Network (DNN) model, and repetition counting via local optima detection. A Bayesian optimizer was employed to tune preprocessing and segmentation hyperparameters for edge deployment. Experimental results across multiple users, environments, and hand conditions confirmed the system's robustness and generalizability.

Building on this foundation, the chapter extended the Wi-PT-Hand system into a clinically validated, spatial-movement-aware platform targeting multi-direction hand rehabilitation exercises involving movement between multiple spatial targets. The enhanced system employs a Faraday-enclosed box with four ESP32 Tx-Rx pairs arranged in horizontal, vertical, and diagonal alignments to ensure environment-independent WiFi signal capture. A speed-invariant classification pipeline was introduced, combining Butterworth low-pass filtering and downsampling with DTW alignment to normalize signals from participants who move at varying speeds due to motor impairments. Four independent 1D-CNN classifiers, one per WiFi link, were fused through a validation-weighted multinomial logistic regression ensemble with temporal majority voting. The system was evaluated on 40 healthy volunteers and 11 clinical patients diagnosed with Parkinson's disease or stroke, achieving classification accuracies of 94.2% on internal validation splits, 91.4% on unseen volunteer test datasets, and 86.3% on the clinical patient cohort, with sub-300 ms inference latency on a Raspberry Pi 4B edge device.

Chapter 4 extended our efforts to a real-life deployment setting by presenting a daily activity monitoring system for senior housing environments. We deployed a scalable IoT infrastructure comprising multiple ESP32 Rx-Tx pairs, Raspberry Pi units with ArduCam camera modules, and cloud-connected services including Dis-

cord webhooks, Tailscale VPN, and Kasa smart plugs to monitor the routine activities of 11 senior participants over five to seven days per participant across two groups. Group 1 included seven participants in apartment settings with six target daily activities, while Group 2 included four participants in home settings with seven target activities, with several of those participants also having Parkinson's disease diagnoses. Despite substantial real-world challenges such as unexpected visitors, device thermal throttling, pets, heating and ventilation interference, and maintenance emergencies, we successfully collected and annotated long-duration naturalistic CSI datasets from all participants. We evaluated CNN-based per-link classifiers, a validation F1-score weighted multinomial logistic regression stacking ensemble model along with a softmax weighted majority voting scheme, and a supervised contrastive learning approach for cross-person and cross-environment generalization, demonstrating that activity recognition can be performed effectively even in uncontrolled, inhabited home environments.

In Chapter 5, we addressed the data scarcity problem in WiFi sensing by introducing *MockiFi*, a generative model for CSI data synthesis using a context-aware CNP. The key insight is that CSI transformations across activity types follow learnable latent patterns that can be transferred from known users to unknown ones, given only a single reference activity as context. By training the CNP on a small set of known participants, MockiFi learned to generate realistic CSI fingerprints for unseen person-action pairs using only a base activity from a new individual. The model optimized a custom loss function combining cosine similarity for directional alignment and range consistency for magnitude matching in the latent space. Evaluation showed that classifiers trained on generated data achieved accuracies of 78.79% on activity recognition and 82.35% on person identification, closely matching 78.24% and 84.11% achieved with real data, respectively. This result opens avenues for zero-shot learning

in WiFi sensing, substantially reducing the cost and effort of data collection for new environments and individuals.

In Chapter 6, we addressed a critical limitation faced by existing single-modality wireless sensing systems: the inability to recognize subtle hand gestures reliably when the user is simultaneously performing full-body locomotion such as walking, turning, or squatting. We proposed *AgnoGest*, an edge-optimized, locomotion-agnostic multimodal gesture recognition platform that synergizes 60 GHz mmWave radar point-cloud data and WiFi CSI from two spatially distributed ESP32 link pairs, all running on a single Raspberry Pi 4B with 2 GB RAM. The mmWave branch processes Range FFT, Doppler FFT, CFAR, and AoA outputs into voxelized $10\times32\times32$ grids with occupancy and mean differential velocity channels, then extracts spatial-temporal features through a Time-Distributed 3D CNN followed by a Bidirectional LSTM (BiLSTM). The two WiFi CSI branches each use lightweight 1D-CNN encoders. All three branches share a triple-adversarial architecture using GRL to suppress locomotion-specific signatures from the learned latent features, while a dual-attention gated fusion mechanism independently weighs the three modalities for both the gesture recognition and locomotion adversarial objectives. Evaluations on seven gestures and nine locomotion types, including unseen locomotion conditions, showed that our multimodal system achieves an average gesture recognition accuracy of 87.64% across entirely unseen volunteers when trained on three-volunteer datasets, substantially outperforming unimodal WiFi-only and radar-only baselines. The model operates within a 35.4 MB compressed footprint, enabling real-time edge inference without a GPU.

Together, these chapters illustrate the practicality, scalability, and feasibility of wireless sensing in real-world health and activity monitoring scenarios. By combining edge-friendly hardware, robust machine learning architectures, creative data generation approaches, and multimodal sensing strategies, this work contributes to

advancing ambient sensing technologies for healthcare and smart living environments.

7.1 Contributions

The key contributions of this dissertation can be summarized as follows:

1. Developed *Wi-PT-Hand*, the first end-to-end device-free rehabilitation tracking system for small-scale hand and finger movements using ESP32 devices, including a complete segmentation, classification, and repetition counting pipeline.

2. Introduced a *hierarchical DNN-based classification pipeline* for fine-grained activity recognition with minimal resource consumption, suitable for deployment on Raspberry Pi and ESP32 edge devices.

3. Designed a *local-average-based segmentation algorithm* and a dynamic thresholding approach for robust activity boundary detection in low-resource CSI data streams.

4. Proposed a *counting mechanism based on local optima detection* tailored for small-scale repetitive movements, extended with a multi-link weighted averaging scheme to improve robustness for slow-moving and clinical participants.

5. Developed a *clinically validated, speed-invariant enhancement* of Wi-PT-Hand for spatially dynamic multi-direction hand movements, incorporating Butterworth-based downsampling, DTW alignment, drop-frame data augmentation, a validation-weighted multi-link CNN ensemble, and temporal majority voting, evaluated on 40 healthy volunteers and 11 Parkinson's disease or stroke patients.

6. Deployed a *scalable WiFi sensing system in real senior housing environments*, collecting naturalistic long-duration datasets from 11 senior participants across

two groups, including participants with Parkinson's disease diagnoses, over five to seven days per participant.

7. Addressed and resolved numerous real-world deployment challenges, including device thermal management, remote serviceability via VPN, activity annotation under uncontrolled conditions, and cross-environment model generalization.

8. Designed a *supervised contrastive learning and validation F1-score weighted multinomial logistic regression ensemble model* pipeline to improve activity recognition performance using multiple Rx-Tx WiFi links in complex, naturalistic environments.

9. Introduced *MockiFi*, a novel CSI data generation system using Conditional Neural Processes for unseen person-action combinations, enabling zero-shot learning without re-collecting training data in new environments.

10. Defined and optimized a *custom loss function combining cosine similarity and range consistency* in latent space, enabling MockiFi to generate structurally realistic CSI samples that match real data accuracy within less than one percentage point.

11. Proposed *AgnoGest*, a multimodal locomotion-agnostic gesture recognition system that fuses 60 GHz mmWave point-cloud data and dual-link WiFi CSI through a triple-adversarial Gradient Reversal Layer architecture and dual-attention gated fusion, achieving 87.64% gesture recognition accuracy on unseen volunteers across nine locomotion conditions on a Raspberry Pi 4B edge device.

12. Demonstrated that *multimodal wireless fusion of mmWave and WiFi* overcomes the fundamental interference bottleneck that limits single-modality systems in

dynamic real-world environments, offering a blueprint for future pervasive ambient intelligence deployments.

7.2 Future Work

While this dissertation has demonstrated promising and clinically validated applications of wireless sensing for healthcare and daily activity monitoring, several future directions remain to be explored.

First, the Wi-PT-Hand clinical system currently supports only fine-grained hand and wrist movements within a Faraday-enclosed box. Extending its reach to gross motor rehabilitation tasks, such as arm lifting, shoulder rotation, or whole-body exercises, would require larger sensing enclosures or open-room multi-link configurations, combined with environment-invariant signal preprocessing techniques. On-edge adaptive personalization through lightweight online learning algorithms, such as streaming linear discriminant analysis or tiny federated updates on the Raspberry Pi itself, would allow the system to adapt to a patient's progressive recovery trajectory without full retraining.

Second, the nursing deployment showed that long-duration, real-life data collection in heterogeneous home environments introduces temporal drift, occupant variability, and interference from non-target individuals or pets. Future work should develop automatic recalibration mechanisms that detect and compensate for these environmental shifts without requiring manual annotation, leveraging self-supervised learning or change-point detection on the continuous CSI stream.

Third, MockiFi demonstrated zero-shot generation of CSI data for unseen users. Future research could extend this to unseen environments by learning environment-level transformations in addition to person-level ones. Combining MockiFi-style generation with federated learning frameworks would further address both data scarcity

and privacy constraints simultaneously, by allowing decentralized on-device learning without sharing raw CSI data.

Fourth, AgnoGest showed the power of multimodal wireless fusion for locomotion-agnostic gesture recognition. Future work should validate this system in clinical and smart home deployments involving real patients, elderly users, and larger naturalistic spaces where the user may move beyond the radar field of view. Integrating additional sensing modalities, such as UWB for precise ranging or barometric pressure sensors for floor-level detection, could extend the system's spatial awareness. Long-term, the architecture can be adapted for continuous vital sign monitoring and fall detection alongside gesture recognition, moving toward a single platform capable of comprehensive ambient health sensing.

Finally, across all works in this dissertation, the training pipelines rely on labeled data collected in specific contexts. Future research should explore self-supervised pre-training of wireless sensing encoders on large unlabeled CSI and point-cloud corpora, enabling the fine-tuning of models on very small annotated datasets for new applications, much in the way large language models have transformed natural language processing. With the increasing computational capacity of embedded edge devices and the growing ecosystem of lightweight neural architectures, there is a strong opportunity to scale up real-time sensing capabilities for multi-person tracking, emotional state estimation, and continuous health and wellness forecasting directly on-device.

REFERENCES

[1] Juha-Matti Torkkel. "Suitability of Rokoko Motion Capture Products for Content Creation in Simulation Training". In: (2022).

[2] Daniel Roetenberg, Henk Luinge, Per Slycke, et al. "Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors". In: *Xsens Motion Technologies BV, Tech. Rep* 1.2009 (2009), pp. 1–7.

[3] Imran Ghani et al. "3D motion capture based physical fitness using full body tracking suit". In: *Journal of Internet Computing and Services* 24.4 (2023), pp. 47–56.

[4] Alfred M Franz et al. "Polhemus EM tracked Micro Sensor for CT-guided interventions". In: *Medical physics* 46.1 (2019), pp. 15–24.

[5] Michelle Lui, Rhonda McEwen, and Martha Mullally. "Immersive virtual reality for supporting complex scientific knowledge: Augmenting our understanding with physiological monitoring". In: *British Journal of Educational Technology* 51.6 (2020), pp. 2181–2199.

[6] Google LLC. *Technology: Soli*. `https://www.norgram.co/work/google-soli-website`.

[7] Linksys. *Linksys Aware Homepage*. `https://www.linksys.com/us/linksysaware/`.

[8] Emerald Innovations. *Emerald Innovations Homepage*. `https://emeraldinno.com/`.

[9] Origin Wireless. *Origin Wireless Homepage*. `https://originwirelessai.com/`.

[10] Google LLC. *Google Cloud Edge TPU*. https://cloud.google.com/edge-tpu.

[11] Nvidia Corporation. *Nvidia Jetson Nano*. https://developer.nvidia.com/embedded/jetson-nano.

[12] Canaan Inc. *Kendryte K210*. https://canaan.io/product/kendryteai.

[13] Feng Zhang et al. "WiSpeed: A statistical electromagnetic approach for device-free indoor speed estimation". In: *IEEE Internet of Things Journal* 5.3 (2018), pp. 2163–2177.

[14] Ramjee Prasad. *OFDM for wireless communications systems*. Vol. 2. Artech House, 2004.

[15] Yushi Shen and Ed Martinez. "Channel estimation in OFDM systems". In: *Freescale semiconductor application note* (2006), pp. 1–15.

[16] Kamran Ali et al. "On goodness of WiFi based monitoring of vital signs in the wild". In: *arXiv preprint arXiv:2003.09386* (2020).

[17] Valentina Bianchi et al. "IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment". In: *IEEE Internet of Things Journal* 6.5 (2019), pp. 8553–8562.

[18] Min SH Aung et al. "The automatic detection of chronic pain-related expression: requirements, challenges and the multimodal EmoPain dataset". In: *IEEE Transactions on Affective Computing* 7.4 (2015), pp. 435–451.

[19] Yan Wang et al. "E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures". In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. 2014, pp. 617–628.

[20] Lei Wang et al. "WiTrace: Centimeter-level passive gesture tracking using WiFi signals". In: *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2018, pp. 1–9.

[21] Daniel Halperin et al. "Tool Release: Gathering 802.11n Traces with Channel State Information". In: *ACM SIGCOMM CCR* 41.1 (Jan. 2011), p. 53.

[22] Wenjun Jiang et al. "Towards environment independent device free human activity recognition". In: *Proceedings of the 24th annual international conference on mobile computing and networking*. 2018, pp. 289–304.

[23] Xiaofeng Liu et al. "Deep unsupervised domain adaptation: A review of recent advances and perspectives". In: *APSIPA Transactions on Signal and Information Processing* 11.1 (2022).

[24] Rasika S. Ransing and Manita Rajput. "Smart home for elderly care, based on Wireless Sensor Network". In: *2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE)*. 2015, pp. 1–5. DOI: 10.1109/ICNTE.2015.7029932.

[25] Martijn Bennebroek et al. "Deployment of wireless sensors for remote elderly monitoring". In: *The 12th IEEE International Conference on e-Health Networking, Applications and Services*. 2010, pp. 1–5. DOI: 10.1109/HEALTH.2010.5556586.

[26] Athanasios Dasios et al. "Wireless sensor network deployment for remote elderly care monitoring". In: *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. PETRA '15. Corfu, Greece: Association for Computing Machinery, 2015. ISBN: 9781450334525. DOI: 10.1145/2769493.2769539. URL: https://doi.org/10.1145/2769493.2769539.

[27] Parisa Rashidi and Alex Mihailidis. “A survey on ambient-assisted living tools for older adults”. In: *IEEE journal of biomedical and health informatics* 17.3 (2012), pp. 579–590.

[28] Tao Wang et al. “Wi-Alarm: Low-cost passive intrusion detection using WiFi”. In: *Sensors* 19.10 (2019), p. 2335.

[29] Wei Wang et al. “Device-free human activity recognition using commercial WiFi devices”. In: *IEEE Journal on Selected Areas in Communications* 35.5 (2017), pp. 1118–1131.

[30] Liangyi Gong et al. “WiFi-based real-time calibration-free passive human motion detection”. In: *Sensors* 15.12 (2015), pp. 32213–32229.

[31] Chen Wang et al. “Towards in-baggage suspicious object detection using commodity wifi”. In: *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2018, pp. 1–9.

[32] Fabio Gaiba et al. “Wi-Fi Sensing for Human Identification Through ESP32 Devices: An Experimental Study”. In: *IEEE 21st Consumer Communications & Networking Conference (CCNC)*. 2024, pp. 206–209.

[33] Md Touhiduzzaman et al. “Wi-PT-Hand: Wireless Sensing based Low-cost Physical Rehabilitation Tracking for Hand Movements”. In: *ACM Transactions on Computing for Healthcare* (2024).

[34] Steven M Hernandez and Eyuphan Bulut. “Adversarial occupancy monitoring using one-sided through-wall WiFi sensing”. In: *ICC 2021-IEEE International Conference on Communications*. IEEE. 2021, pp. 1–6.

[35] Raghav H Venkatnarayan, Shakir Mahmood, and Muhammad Shahzad. "WiFi based multi-user gesture recognition". In: *IEEE Transactions on Mobile Computing* 20.3 (2019), pp. 1242–1256.

[36] Zhiling Tang et al. "WiFi CSI gesture recognition based on parallel LSTM-FCN deep space-time neural network". In: *China Communications* 18.3 (2021), pp. 205–215.

[37] Chunjing Xiao et al. "DeepSeg: Deep-Learning-Based Activity Segmentation Framework for Activity Recognition Using WiFi". In: *IEEE Internet of Things Journal* 8.7 (2021), pp. 5669–5681. DOI: 10.1109/JIOT.2020.3033173.

[38] Jiang Xiao, Huichuwu Li, and Hai Jin. "Transtrack: Online meta-transfer learning and Otsu segmentation enabled wireless gesture tracking". In: *Pattern Recognition* 121 (2022), p. 108157.

[39] Lei Zhang et al. "Wi-Gym: Gymnastics Activity Assessment Using Commodity Wi-Fi". In: *IEEE Internet of Things Journal* 9.23 (2022), pp. 24051–24064. DOI: 10.1109/JIOT.2022.3188916.

[40] Ubaid Mehmood et al. "Occupancy estimation using WiFi: A case study for counting passengers on busses". In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE. 2019, pp. 165–170.

[41] Sameera Palipana et al. "FallDeFi: Ubiquitous fall detection using commodity Wi-Fi devices". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.4 (2018), pp. 1–25.

[42] Daqing Zhang et al. "Anti-fall: A non-intrusive and real-time fall detector leveraging CSI from commodity WiFi devices". In: *Inclusive Smart Cities and e-Health: 13th International Conference on Smart Homes and Health Telem-*

atics, ICOST 2015, Geneva, Switzerland, June 10-12, 2015, Proceedings 13. Springer. 2015, pp. 181–193.

[43] Hao Wang et al. "RT-Fall: A real-time and contactless fall detection system with commodity WiFi devices". In: *IEEE Transactions on Mobile Computing* 16.2 (2016), pp. 511–526.

[44] Steven M. Hernandez, Deniz Erdag, and Eyuphan Bulut. "Towards Dense and Scalable Soil Sensing Through Low-Cost WiFi Sensing Networks". In: *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. 2021, pp. 549–556. DOI: 10.1109/LCN52139.2021.9525003.

[45] Youwei Zeng et al. "FarSense: Pushing the Range Limit of WiFi-based Respiration Sensing with CSI Ratio of Two Antennas". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3.3 (Sept. 2019). DOI: 10.1145/3351279. URL: https://doi.org/10.1145/3351279.

[46] Youwei Zeng et al. "MultiSense: Enabling Multi-person Respiration Sensing with Commodity WiFi". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4.3 (Sept. 2020). DOI: 10.1145/3411816. URL: https://doi.org/10.1145/3411816.

[47] Heba Abdelnasser, Khaled A. Harras, and Moustafa Youssef. "UbiBreathe: A Ubiquitous non-Invasive WiFi-based Breathing Estimator". In: MobiHoc '15. Hangzhou, China: Association for Computing Machinery, 2015, pp. 277–286. ISBN: 9781450334891. DOI: 10.1145/2746285.2755969. URL: https://doi.org/10.1145/2746285.2755969.

[48] Dou Fan et al. "A Contactless Breathing Pattern Recognition System Using Deep Learning and WiFi Signal". In: *IEEE Internet of Things Journal* 11.13 (2024), pp. 23820–23834. DOI: 10.1109/JIOT.2024.3386645.

[49] Yu Gu et al. “WiFi-Based Real-Time Breathing and Heart Rate Monitoring during Sleep”. In: *2019 IEEE Global Communications Conference (GLOBECOM)*. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9014297.

[50] Xiaoqun Yu and Shuping Xiong. “A dynamic time warping based algorithm to evaluate kinect-enabled home-based physical rehabilitation exercises for older people”. In: *Sensors* 19.13 (2019), p. 2882.

[51] Daniel Leightley et al. “Human activity recognition for physical rehabilitation”. In: *IEEE International Conference on Systems, Man, and Cybernetics*. 2013, pp. 261–266.

[52] Sizhe An and Umit Y Ogras. “MARS: mmWave-based Assistive Rehabilitation System for Smart Healthcare”. In: *ACM Transactions on Embedded Computing Systems (TECS)* 20.5s (2021), pp. 1–22.

[53] Fei Wang et al. “U-Shape Networks are Unified Backbones for Human Action Understanding from Wi-Fi Signals”. In: *IEEE Internet of Things Journal* (2023).

[54] Jiaxing Liu. “StackFi: An Ensemble Learning-Based Model for WiFi Sensing Classification Tasks”. In: *2024 4th International Conference on Consumer Electronics and Computer Engineering (ICCECE)*. 2024, pp. 265–269. DOI: 10.1109/ICCECE61317.2024.10504185.

[55] Nafeez Fahad, Md Touhiduzzaman, and Eyuphan Bulut. “Ensemble Learning based WiFi Sensing using Spatially Distributed TX-RX Links”. In: *20th IEEE International Conference on Computing, Networking and Communications (ICNC)*. Honolulu, Hawaii, USA, Feb. 2025.

[56] José Ramón Merino Bernaola et al. "Ensemble Learning for Seated People Counting using WiFi Signals: Performance Study and Transferability Assessment". In: *2021 IEEE Globecom Workshops (GC Wkshps)*. 2021, pp. 1–6. DOI: 10.1109/GCWkshps52748.2021.9682014.

[57] Stefania Monica and Federico Bergenti. "A comparison of accurate indoor localization of static targets via WiFi and UWB ranging". In: *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer. 2016, pp. 111–123.

[58] Steven M Hernandez and Eyuphan Bulut. "TrinaryMC: Monte Carlo based anchorless relative positioning for indoor positioning". In: *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2020, pp. 1–6.

[59] Steven M Hernandez and Eyuphan Bulut. "Using perceived direction information for anchorless relative indoor localization". In: *Journal of Network and Computer Applications* 165 (2020), p. 102714.

[60] Ninad Jadhav et al. "A wireless signal-based sensing framework for robotics". In: *The International Journal of Robotics Research* 41.11-12 (2022), pp. 955–992.

[61] Bohong Xiang et al. "UAV assisted localization scheme of WSNs using RSSI and CSI information". In: *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. IEEE. 2020, pp. 718–722.

[62] Jian-guo Jiang et al. "CS-DICT: Accurate indoor localization with CSI selective amplitude and phase based regularized dictionary learning". In: *Algorithms and Architectures for Parallel Processing: 20th International Confer-*

ence, ICA3PP 2020, New York City, NY, USA, October 2–4, 2020, Proceedings, Part II 20. Springer. 2020, pp. 677–689.

[63] Rui Zhou et al. "Device-free localization based on CSI fingerprints and deep neural networks". In: *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2018, pp. 1–9.

[64] Rui Zhou et al. "Adaptive device-free localization in dynamic environments through adaptive neural networks". In: *IEEE Sensors Journal* 21.1 (2020), pp. 548–559.

[65] Liuyi Yang, Tomio Kamada, and Chikara Ohta. "Indoor localization based on CSI in dynamic environments through domain adaptation". In: *IEICE Communications Express* 10.8 (2021), pp. 564–569.

[66] Yong Zhang, Chengbin Wu, and Yang Chen. "A low-overhead indoor positioning system using CSI fingerprint based on transfer learning". In: *IEEE Sensors Journal* 21.16 (2021), pp. 18156–18165.

[67] Jialai Liu et al. "An efficient csi-based pedestrian monitoring approach via single pair of wifi transceivers". In: *International conference on neural computing for advanced applications*. Springer. 2021, pp. 685–700.

[68] Xingang Wang, Yufei Wang, and Dong Wang. "A real-time CSI-based passive intrusion detection method". In: *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. IEEE. 2020, pp. 1091–1098.

[69] Daniel Konings and Fakhrul Alam. "LifeCount: A device-free CSI-based human counting solution for emergency building evacuations". In: *2020 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2020, pp. 1–5.

[70] Xi Chen et al. "PHCount: Passive Human Number Counting Using WiFi". In: *Communications, Signal Processing, and Systems*. Ed. by Qilian Liang et al. Singapore: Springer Singapore, 2021, pp. 1214–1223. ISBN: 978-981-15-8411-4.

[71] Belal Korany and Yasamin Mostofi. "Counting a stationary crowd using off-the-shelf wifi". In: *Proceedings of the 19th annual international conference on mobile systems, applications, and services*. 2021, pp. 202–214.

[72] Yu Gu et al. "EmoSense: Data-Driven Emotion Sensing via Off-the-Shelf WiFi Devices". In: *2018 IEEE International Conference on Communications (ICC)*. 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422330.

[73] Zhenzhe Lin et al. "WiEat: Fine-grained Device-free Eating Monitoring Leveraging Wi-Fi Signals". In: *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. 2020, pp. 1–9. DOI: 10.1109/ICCCN49398.2020.9209628.

[74] Sheng Tan and Jie Yang. "Object sensing for fruit ripeness detection using WiFi signals". In: *arXiv preprint arXiv:2106.00860* (2021).

[75] Weidong Yang et al. "Wi-Wheat: Contact-free wheat moisture detection with commodity WiFi". In: *2018 IEEE International Conference on Communications (ICC)*. IEEE. 2018, pp. 1–6.

[76] Nafeez Fahad and Eyuphan Bulut. "WiFi CSI based Liquid Temperature Prediction: A Physics-Guided Machine Learning Approach". In: *26th Inter-*

national Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM). Fort Worth, Texas, USA, May 2025.

[77] Yili Ren et al. "Liquid level sensing using commodity wifi in a smart home environment". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.1 (2020), pp. 1–30.

[78] Hang Song et al. "WiEps: Measurement of dielectric property with commodity WiFi device—An application to ethanol/water mixture". In: *IEEE Internet of Things Journal* 7.12 (2020), pp. 11667–11677.

[79] Sirui Jian, Shigemi Ishida, and Yutaka Arakawa. "Initial attempt on Wi-Fi CSI based vibration sensing for factory equipment fault detection". In: *Adjunct proceedings of the 2021 international conference on distributed computing and networking*. 2021, pp. 163–168.

[80] Yu Gu et al. "WiFE: WiFi and vision based unobtrusive emotion recognition via gesture and facial expression". In: *IEEE Transactions on Affective Computing* 14.4 (2023), pp. 2567–2581.

[81] Vimal Kakaraparthi et al. "FaceSense: sensing face touch with an ear-worn system". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.3 (2021), pp. 1–27.

[82] Tobias Röddiger et al. "Sensing with earables: A systematic literature review and taxonomy of phenomena". In: *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 6.3 (2022), pp. 1–57.

[83] Abu Zafar Abbasi, Noman Islam, Zubair Ahmed Shaikh, et al. "A review of wireless sensors and networks' applications in agriculture". In: *Computer Standards & Interfaces* 36.2 (2014), pp. 263–270.

[84] Steven M Hernandez and Eyuphan Bulut. "Scheduled Spatial Sensing against Adversarial WiFi Sensing". In: *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2023, pp. 91–100.

[85] Siwang Zhou et al. "Adversarial WiFi sensing for privacy preservation of human behaviors". In: *IEEE Communications Letters* 24.2 (2019), pp. 259–263.

[86] Jean-François Determe et al. "Monitoring large crowds with WiFi: A privacy-preserving approach". In: *IEEE Systems Journal* 16.2 (2022), pp. 2148–2159.

[87] Merrill Ivan Skolnik et al. *Introduction to radar systems*. Vol. 3. McGraw-hill New York, 1980.

[88] Cesar Iovescu and Sandeep Rao. "The fundamentals of millimeter wave sensors". In: *Texas Instruments* (2017), pp. 1–8.

[89] Jia Zhang et al. "A survey of mmWave-based human sensing: Technology, platforms and applications". In: *IEEE Communications Surveys & Tutorials* 25.4 (2023), pp. 2052–2087.

[90] Jaime Lien et al. "Soli: Ubiquitous gesture sensing with millimeter wave radar". In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), pp. 1–19.

[91] Muhammed Zahid Ozturk et al. "Radiomic: Sound sensing via mmwave signals". In: *arXiv preprint arXiv:2108.03164* (2021).

[92] Jie Wang et al. "Device-free human gesture recognition with generative adversarial networks". In: *IEEE Internet of Things Journal* 7.8 (2020), pp. 7678–7688.

[93] Shahzad Ahmed et al. "Hand gestures recognition using radar sensors for human-computer-interaction: A review". In: *Remote Sensing* 13.3 (2021), p. 527.

[94] Yong Wang et al. “A novel detection and recognition method for continuous hand gesture using FMCW radar”. In: *IEEE Access* 8 (2020), pp. 167264–167275.

[95] Akash Deep Singh et al. “Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar”. In: *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*. 2019, pp. 51–56.

[96] Zhanjun Hao et al. “Millimeter wave gesture recognition using multi-feature fusion models in complex scenes”. In: *Scientific Reports* 14.1 (2024), p. 13758.

[97] Prashant P Gandhi and Saleem A Kassam. “Analysis of CFAR processors in nonhomogeneous background”. In: *IEEE Transactions on Aerospace and Electronic systems* 24.4 (1988), pp. 427–445.

[98] Charles Ruizhongtai Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: *Advances in neural information processing systems* 30 (2017).

[99] Sameera Palipana et al. “Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds”. In: *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 5.1 (2021), pp. 1–27.

[100] Xiulong Liu et al. “MHTrack: mmWave-Based Mobile Hand Tracking”. In: *IEEE Trans. Mob. Comput.* 24.9 (2025), pp. 9168–9183.

[101] Anurag Pallaprolu, Winston Hurst, and Yasamin Mostofi. “mmRidge: Revealing Emergent Crowd Flow Structures with mmWave MIMO Radar”. In: *Proceedings of the 27th International Workshop on Mobile Computing Systems and Applications*. 2026, pp. 31–36.

[102] Anirban Banik et al. “mmSnap: Bayesian One-Shot Fusion in a Self-Calibrated mmWave Radar Network”. In: *2025 IEEE Radar Conference (RadarConf25)*. IEEE. 2025, pp. 1116–1121.

[103] Anurag Pallaprolu et al. “Crowd size estimation for non-uniform spatial distributions with mmWave radar”. In: *2025 59th Asilomar Conference on Signals, Systems, and Computers*. IEEE. 2025, pp. 445–450.

[104] Fengyu Wang et al. “mmHRV: Contactless heart rate variability monitoring using millimeter-wave radio”. In: *IEEE Internet of Things Journal* 8.22 (2021), pp. 16623–16636.

[105] Shuqin Dong et al. “A review on recent advancements of biomedical radar for clinical applications”. In: *IEEE Open Journal of Engineering in Medicine and Biology* (2024).

[106] Damien Vivet, Paul Checchin, and Roland Chapuis. “Localization and mapping using only a rotating FMCW radar sensor”. In: *Sensors* 13.4 (2013), pp. 4527–4552.

[107] Shahzad Ahmed, Junbyung Park, and Sung Ho Cho. “FMCW radar sensor based human activity recognition using deep learning”. In: *2022 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE. 2022, pp. 1–5.

[108] Hira Hameed et al. “RF sensing enabled tracking of human facial expressions using machine learning algorithms”. In: *Scientific Reports* 14.1 (2024), p. 27800.

[109] Khalid Youssef et al. “Machine learning approach to RF transmitter identification”. In: *IEEE Journal of Radio Frequency Identification* 2.4 (2018), pp. 197–205.

[110] Quang DM Nguyen et al. “Deep Learning-Empowered RF Sensing in Outdoor Environments: Recent Advances, Challenges, and Future Directions”. In: *Electronics* 14.1 (2024), p. 125.

[111] Feng Jin et al. “MmWave Radar Point Cloud Segmentation using GMM in Multimodal Traffic Monitoring”. In: *2020 IEEE International Radar Conference (RADAR)*. 2020, pp. 732–737. DOI: 10.1109/RADAR42522.2020.9114662.

[112] John C Howell et al. “Super interferometric range resolution”. In: *Physical Review Letters* 131.5 (2023), p. 053803.

[113] Lang Deng et al. “GaitFi: Robust device-free human identification via WiFi and vision multimodal learning”. In: *IEEE Internet of Things Journal* 10.1 (2022), pp. 625–636.

[114] Han Zou et al. “WiFi and vision multimodal learning for accurate and robust device-free human activity recognition”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2019, pp. 0–0.

[115] Jiajing Chen et al. “Wimix: A lightweight multimodal human activity recognition system based on wifi and vision”. In: *2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE. 2023, pp. 406–414.

[116] Kian Behzad et al. “Robomnist: A multimodal dataset for multi-robot activity recognition using wifi sensing, video, and audio”. In: *Scientific Data* 12.1 (2025), p. 326.

[117] Xiangguo Li et al. “RFID-WiFi-Radar Fusion for Health Monitoring: A Cross-Modal Supervision Framework”. In: *IEEE Sensors Journal* (2025).

[118] Jiale Wang et al. “Multimodal Fusion-Based Human Action Recognition Using Wi-Fi CSI and Smartwatch Sensors”. In: *IEEE Internet of Things Journal* (2026).

[119] Muhammad Muaaz et al. “WiWeHAR: Multimodal human activity recognition using Wi-Fi and wearable sensing modalities”. In: *IEEE access* 8 (2020), pp. 164453–164470.

[120] Zhiyu Chen, Yanpeng Sun, and Lele Qu. “Research on cross-scene human activity recognition based on radar and wi-fi multimodal fusion”. In: *Electronics* 14.8 (2025), p. 1518.

[121] Jianfei Yang et al. “Mm-fi: Multi-modal non-intrusive 4d human dataset for versatile wireless sensing”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 18756–18768.

[122] Jin Zhang et al. “Data augmentation and dense-LSTM for human activity recognition using WiFi signal”. In: *IEEE Internet of Things Journal* 8.6 (2020), pp. 4628–4641.

[123] Hyunggeun Mo and Seungku Kim. “A deep learning-based human identification system with wi-fi csi data augmentation”. In: *IEEE Access* 9 (2021), pp. 91913–91920.

[124] Weiying Hou and Chenshu Wu. "RFBoost: Understanding and Boosting Deep WiFi Sensing via Physical Data Augmentation". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8.2 (2024), pp. 1–26.

[125] Julian Strohmayer and Martin Kampel. "Data augmentation techniques for cross-domain wifi csi-based human activity recognition". In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer. 2024, pp. 42–56.

[126] Yue Zheng et al. "Zero-effort cross-domain gesture recognition with Wi-Fi". In: *Proceedings of the 17th annual international conference on mobile systems, applications, and services*. 2019, pp. 313–325.

[127] Leqi Zhao et al. "One is Enough: Enabling One-shot Device-free Gesture Recognition with COTS WiFi". In: *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE. 2024, pp. 1231–1240.

[128] Md Tamzeed Islam and Shahriar Nirjon. "Wi-Fringe: Leveraging Text Semantics in WiFi CSI-Based Device-Free Named Gesture Recognition". In: *16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2020, pp. 35–42. DOI: 10.1109/DCOSS49796.2020.00019.

[129] Yimin Mao et al. "Wi-Cro: WiFi-based Cross Domain Activity Recognition via Modified GAN". In: *IEEE Transactions on Vehicular Technology* (2024).

[130] Nabeel Nisar Bhat, Rafael Berkvens, and Jeroen Famaey. "CSI4Free: GAN-Augmented mmWave CSI for Improved Pose Classification". In: *2024 IEEE 4th International Symposium on Joint Communications & Sensing (JC&S)*. IEEE. 2024, pp. 1–6.

[131] Hong Cai et al. “Teaching rf to sense without rf training measurements”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.4 (2020), pp. 1–22.

[132] Zhe Chen et al. “RF-based human activity recognition using signal adapted convolutional neural network”. In: *IEEE Transactions on Mobile Computing* 22.1 (2021), pp. 487–499.

[133] Marwan Yusuf et al. “Human sensing in reverberant environments: RF-based occupancy and fall detection in ships”. In: *IEEE Transactions on Vehicular Technology* 70.5 (2021), pp. 4512–4522.

[134] Mohammud J Bocus, Kevin Chetty, and Robert J Piechocki. “UWB and WiFi systems as passive opportunistic activity sensing radars”. In: *2021 IEEE Radar Conference (RadarConf21)*. IEEE. 2021, pp. 1–6.

[135] Mengjing Liu, Zongxing Xie, and Fan Ye. “ADL-CLIP: Text-Aligned RF Representation for Continuous ADL Detection in Home Environments”. In: *Proceedings of the 24th Annual International Conference on Mobile Systems, Applications and Services*. 2026, pp. 127–143.

[136] Fusang Zhang et al. “Exploring LoRa for Long-range Through-wall Sensing”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4.2 (June 2020). DOI: 10.1145/3397326. URL: https://doi.org/10.1145/3397326.

[137] Binbin Xie, Yuqing Yin, and Jie Xiong. “Pushing the limits of long range wireless sensing with lora”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.3 (2021), pp. 1–21.

[138] Mingxing Nie et al. “Enhancing human activity recognition with LoRa wireless RF signal preprocessing and deep learning”. In: *Electronics* 13.2 (2024), p. 264.

[139] Tianxing Li et al. “Human sensing using visible light communication”. In: *Proceedings of the 21st annual international conference on mobile computing and networking*. 2015, pp. 331–344.

[140] Binbin Xie et al. “Exploring commodity rfid for contactless sub-millimeter vibration sensing”. In: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 2020, pp. 15–27.

[141] Yuxiao Hou, Yanwen Wang, and Yuanqing Zheng. “TagBreathe: Monitor breathing with commodity RFID systems”. In: *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*. IEEE. 2017, pp. 404–413.

[142] Chao Feng et al. “RF-identity: Non-intrusive person identification based on commodity RFID devices”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.1 (2021), pp. 1–23.

[143] Ju Wang et al. “D-watch: Embracing” bad” multipaths for device-free localization with COTS RFID devices”. In: *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*. 2016, pp. 253–266.

[144] Ritochit Chakraborty, Sumit Roy, and Vikram Jandhyala. “Revisiting RFID Link Budgets for Technology Scaling: Range Maximization of RFID Tags”. In: *IEEE Transactions on Microwave Theory and Techniques* 59.2 (2011), pp. 496–503. DOI: 10.1109/TMTT.2010.2097711.

[145] Jaap C Haartsen. "The Bluetooth radio system". In: *IEEE personal communications* 7.1 (2000), pp. 28–36.

[146] Giancarlo Iannizzotto et al. "A perspective on passive human sensing with bluetooth". In: *Sensors* 22.9 (2022), p. 3523.

[147] Carlos De Lima et al. "Convergent Communication, Sensing and Localization in 6G Systems: An Overview of Technologies, Opportunities and Challenges". In: *IEEE Access* 9 (2021), pp. 26902–26925. DOI: 10.1109/ACCESS.2021.3053486.

[148] Fan Liu et al. "Integrated Sensing and Communications: Toward Dual-Functional Wireless Networks for 6G and Beyond". In: *IEEE Journal on Selected Areas in Communications* 40.6 (2022), pp. 1728–1767. DOI: 10.1109/JSAC.2022.3156632.

[149] Yijie Li et al. "MuDiS: An Audio-independent, Wide-angle, and Leak-free Multi-directional Speaker". In: *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. ACM MobiCom '24. Washington D.C., DC, USA: Association for Computing Machinery, 2024, pp. 263–278. ISBN: 9798400704895. DOI: 10.1145/3636534.3649360. URL: https://doi.org/10.1145/3636534.3649360.

[150] Jihun Lee et al. "An asynchronous wireless network for capturing event-driven data from large populations of autonomous sensors". In: *Nature Electronics* 7.4 (2024), pp. 313–324.

[151] Pixi Kang et al. "On-Device Training Empowered Transfer Learning For Human Activity Recognition". In: *arXiv preprint arXiv:2407.03644* (2024).

[152] Steven M. Hernandez and Eyuphan Bulut. "WiFi Sensing on the Edge: Signal Processing Techniques and Challenges for Real-World Systems". In: *IEEE Commun. Surv. Tutorials* 25.1 (2023), pp. 46–76.

[153] Muhammad Haseeb Arshad, Muhammad Bilal, and Abdullah Gani. "Human activity recognition: Review, taxonomy and open challenges". In: *Sensors* 22.17 (2022), p. 6463.

[154] Nazanin sedaghati, Sondos ardebili, and Ali Ghaffari. "Application of human activity/action recognition: a review". In: *Multimedia Tools and Applications* (2025), pp. 1–30.

[155] Tianzhang Xing et al. "WiFine: Real-Time Gesture Recognition Using Wi-Fi with Edge Intelligence". In: *ACM Transactions on Sensor Networks* 19.1 (2022).

[156] Rong Li et al. "Wecar: An end-edge collaborative inference and training framework for wifi-based continuous human activity recognition". In: *ACM Transactions on Sensor Networks* 22.3 (2026), pp. 1–24.

[157] En Li et al. "Edge AI: On-demand accelerating deep neural network inference via edge computing". In: *IEEE transactions on wireless communications* 19.1 (2019), pp. 447–457.

[158] Steven M Hernandez and Eyuphan Bulut. "WiFederated: Scalable WiFi sensing using edge-based federated learning". In: *IEEE Internet of Things Journal* 9.14 (2021), pp. 12628–12640.

[159] Zhiyuan Wu et al. "Survey of knowledge distillation in federated edge learning". In: *arXiv preprint arXiv:2301.05849* (2023).

[160] Ching-Hu Lu and Xiao-Zong Lin. "Toward direct edge-to-edge transfer learning for IoT-enabled edge cameras". In: *IEEE Internet of Things Journal* 8.6 (2020), pp. 4931–4943.

[161] Angelo Trotta et al. "Edge human activity recognition using federated learning on constrained devices". In: *Pervasive and Mobile Computing* 104 (2024), p. 101972.

[162] Han Cai et al. "Tinytl: Reduce memory, not parameters for efficient on-device learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 11285–11297.

[163] Dawei Li et al. "RILOD: Near real-time incremental learning for object detection at the edge". In: *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. 2019, pp. 113–126.

[164] Yuqing Du, Sheng Yang, and Kaibin Huang. "High-dimensional stochastic gradient quantization for communication-efficient edge learning". In: *IEEE transactions on signal processing* 68 (2020), pp. 2128–2142.

[165] Giovanni Diraco et al. "Review on human action recognition in smart living: Sensing technology, multimodality, real-time processing, interoperability, and resource-constrained processing". In: *Sensors* 23.11 (2023), p. 5281.

[166] Connie W Tsao et al. "Heart disease and stroke statistics—2022 update: a report from the American Heart Association". In: *Circulation* 145.8 (2022), e153–e639.

[167] Jin Park and Tae-Ho Kim. "The effects of balance and gait function on quality of life of stroke patients". In: *NeuroRehabilitation* 44.1 (2019), pp. 37–41.

[168] Kyoung Bo Lee et al. "Six-month functional recovery of stroke patients: a multi-time-point study". In: *International journal of rehabilitation research.* 38.2 (2015), p. 173.

[169] Giovanni Abbruzzese et al. "Rehabilitation for Parkinson's disease: current outlook and future challenges". In: *Parkinsonism & related disorders* 22 (2016), S60–S64.

[170] Md Atiqur Rahman Ahad, Anindya Das Antar, and Omar Shahid. "Vision-based Action Understanding for Assistive Healthcare: A Short Review." In: *CVPR Workshops*. IEEE Xplore, 2019, pp. 1–11.

[171] Qi Wang et al. "Interactive wearable systems for upper body rehabilitation: a systematic review". In: *Journal of neuroengineering and rehabilitation* 14.1 (2017), pp. 1–21.

[172] Octavian Postolache et al. "Remote monitoring of physical rehabilitation of stroke patients using IoT and virtual reality". In: *IEEE Journal on Selected Areas in Communications* 39.2 (2020), pp. 562–573.

[173] Jun Qi et al. "Examining sensor-based physical activity recognition and monitoring for healthcare using Internet of Things: A systematic review". In: *Journal of biomedical informatics* 87 (2018), pp. 138–153.

[174] Ganapati Bhat et al. "w-HAR: An activity recognition dataset and framework using low-power wearable devices". In: *Sensors* 20.18 (2020), p. 5356.

[175] Fadel Adib et al. "Capturing the human figure through a wall". In: *ACM Transactions on Graphics (TOG)* 34.6 (2015), pp. 1–13.

[176] Arindam Sengupta et al. “mm-Pose: Real-time human skeletal posture estimation using mmWave radars and CNNs”. In: *IEEE Sensors Journal* 20.17 (2020), pp. 10032–10044.

[177] Yongsen Ma, Gang Zhou, and Shuangquan Wang. “WiFi sensing with channel state information: A survey”. In: *ACM Computing Surveys (CSUR)* 52.3 (2019), pp. 1–36.

[178] Xiaonan Guo et al. “Device-free personalized fitness assistant using WiFi”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.4 (2018), pp. 1–23.

[179] Fei Wang et al. “Temporal Unet: Sample level human action recognition using wifi”. In: *arXiv preprint arXiv:1904.11953* (2019).

[180] Shiming Chen et al. “A Real-time Activity Recognition System based on Dynamic Adaptive Windows using WiFi Signals”. In: *2021 5th International Conference on Computer Science and Artificial Intelligence*. 2021, pp. 122–127.

[181] Steven M Hernandez and Eyuphan 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)*. Cork, Ireland, June 2020.

[182] Atheros CSI Tool. 2019. URL: `https://wands.sg/research/wifi/AtherosCSI/`.

[183] Nexmon: The c-based firmware patching framework. 2019. URL: `https://nexmon.org`..

[184] Md Touhiduzzaman. *CSI Annotator App*. `https://github.com/touhiDroid/CsiAnnotator`. 2024.

[185] Steven M Hernandez and Md Touhiduzzaman. *CSI-Pi*. https://github.com/MoWiNG-Lab/CSI-Pi. 2024.

[186] Adam D. Bull. *Convergence rates of efficient global optimization algorithms*. 2011. arXiv: 1101.3501 [stat.ML].

[187] Michael A. Gelbart, Jasper Snoek, and Ryan P. Adams. *Bayesian Optimization with Unknown Constraints*. 2014. arXiv: 1403.5607 [stat.ML].

[188] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. *Practical Bayesian Optimization of Machine Learning Algorithms*. 2012. arXiv: 1206.2944 [stat.ML].

[189] François Chollet et al. *Keras*. https://keras.io. 2015.

[190] Jingzhi Hu et al. "Muse-fi: Contactless multi-person sensing exploiting near-field wi-fi channel variation". In: *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 2023, pp. 1–15.

[191] Naoki Ohmura, Satoshi Ogino, and Yoshinobu Okano. "Optimized shielding pattern of RF faraday cage". In: *2014 International Symposium on Electromagnetic Compatibility, Tokyo*. 2014, pp. 765–768.

[192] Tailscale Inc. *Tailscale: The easiest way to create a secure network*. Accessed on July 29, 2024. 2024. URL: https://tailscale.com/.

[193] Stan Salvador and Philip Chan. "Toward Accurate Dynamic Time Warping in Linear Time and Space". In: *Intelligent Data Analysis* 11.5 (2007), pp. 561–580.

[194] Md Touhiduzzaman et al. "Wi-PT-Hand: Wireless Sensing based Low-cost Physical Rehabilitation Tracking for Hand Movements". In: *ACM Transactions on Computing for Healthcare* 6.1 (2025), pp. 1–25.

[195] MoWiNG-Lab. *CSI-Pi*. https://github.com/MoWiNG-Lab/CSI-Pi. Accessed on July 29, 2024. 2024.

[196] Yongsen Ma et al. "SignFi: Sign Language Recognition Using WiFi". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1 (2018), pp. 1–21.

[197] Tanund Sakpoonsup and Attaphongse Taparugssanagorn. "Real-Time Gesture Recognition Using Wi-Fi CSI: A CNN Approach for Multi-Environment Deployment". In: *Applied Soft Computing* (2025), p. 114407.

[198] Mohan Rajkumar Na and K. B. Sundharakumar. "A Study on Air-Gap Networks". In: *2024 5th International Conference on Innovative Trends in Information Technology (ICITIIT)*. IEEE. 2024, pp. 1–6.

[199] Steven M Hernandez and Eyuphan Bulut. "Lightweight and standalone IoT based WiFi sensing for active repositioning and mobility". In: *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE. 2020, pp. 277–286.

[200] Dave Jones. *picamera*. https://github.com/waveform80/picamera/tree/release-1.13. Release 1.13, Accessed on July 29, 2024. 2024.

[201] WireGuard. *WireGuard: Fast, Modern, Secure VPN Tunnel*. Accessed on July 29, 2024. 2024. URL: https://www.wireguard.com/.

[202] Rami Alazrai et al. "A dataset for Wi-Fi-based human-to-human interaction recognition". In: *Data in Brief* 31 (2020), p. 105668. ISSN: 2352-3409. DOI: https://doi.org/10.1016/j.dib.2020.105668. URL: https://www.sciencedirect.com/science/article/pii/S235234092030562X.

[203] Majid Ghosian Moghaddam, Ali Asghar Nazari Shirehjini, and Shervin Shirmohammadi. "A WiFi-based method for recognizing fine-grained multiple-subject human activities". In: *IEEE Transactions on Instrumentation and Measurement* 72 (2023), pp. 1–13.

[204] Steven M Hernandez and Eyuphan Bulut. "Online stream sampling for low-memory on-device edge training for WiFi sensing". In: *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*. 2022, pp. 9–14.

[205] Md Touhiduzzaman and Eyuphan Bulut. "MockiFi: CSI Imitation using Context-Aware Conditional Neural Process for Zero-shot Learning". In: *2025 34th International Conference on Computer Communications and Networks (ICCCN)*. IEEE. 2025, pp. 1–6.

[206] Md Touhiduzzaman et al. "Wireless Sensing-based Daily Activity Tracking System Deployment in Low-Income Senior Housing Environments". In: *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 2024, pp. 2260–2267.

[207] Thomas G Dietterich et al. "Ensemble learning". In: *The handbook of brain theory and neural networks* 2.1 (2002), pp. 110–125.

[208] Prannay Khosla et al. "Supervised contrastive learning". In: *Advances in neural information processing systems* 33 (2020), pp. 18661–18673.

[209] Chen Chen, Gang Zhou, and Youfang Lin. "Cross-domain wifi sensing with channel state information: A survey". In: *ACM Computing Surveys* 55.11 (2023), pp. 1–37.

[210] Zhenguo Shi et al. "Environment-robust device-free human activity recognition with channel-state-information enhancement and one-shot learning". In: *IEEE Transactions on Mobile Computing* 21.2 (2020), pp. 540–554.

[211] Yuanrun Fang et al. "WiTransfer: A cross-scene transfer activity recognition system using WiFi". In: *Proceedings of the ACM turing celebration conference-China*. 2020, pp. 59–63.

[212] Jin Zhang et al. "MetaGanFi: Cross-domain unseen individual identification using WiFi signals". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6.3 (2022), pp. 1–21.

[213] Xi Chen et al. "Fidora: Robust WiFi-based indoor localization via unsupervised domain adaptation". In: *IEEE IoT Journal* 9.12 (2022), pp. 9872–9888.

[214] Marta Garnelo et al. "Conditional neural processes". In: *International conference on machine learning*. PMLR. 2018, pp. 1704–1713.

[215] Biyun Sheng et al. "Context-Aware Faster RCNN for CSI-Based Human Action Perception". In: *IEEE Transactions on Human-Machine Systems* 53.2 (2023), pp. 438–448. DOI: 10.1109/THMS.2022.3225828.

[216] Han Cai et al. "Condition-Aware Neural Network for Controlled Image Generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 7194–7203.

[217] David Vander Mijnsbrugge, Femke Ongenae, and Sofie Van Hoecke. "Context-aware deep learning with dynamically assembled weight matrices". In: (2023).

[218] Xuanli He. "Context-aware Natural Language Understanding and Generation". PhD thesis. Monash University, 2022.

[219] Zesheng Ye and Lina Yao. "Contrastive conditional neural processes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9687–9696.

[220] Deep Shankar Pandey and Qi Yu. "Evidential conditional neural processes". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 8. 2023, pp. 9389–9397.

[221] Diederik P Kingma. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[222] Xiaofeng Liu et al. "Data augmentation via latent space interpolation for image classification". In: *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE. 2018, pp. 728–733.

[223] Ying Zhou et al. "DiffLM: Controllable Synthetic Data Generation via Diffusion Language Models". In: *arXiv preprint 2411.03250* (2024).

[224] Joao Fonseca and Fernando Bacao. "Tabular and latent space synthetic data generation: a literature review". In: *Journal of Big Data* 10.1 (2023), p. 115.

[225] Ronald K Pearson et al. "Generalized hampel filters". In: *EURASIP Journal on Advances in Signal Processing* 2016 (2016), pp. 1–18.

[226] Hoonyong Lee, Changbum Ryan Ahn, and Nakjung Choi. "Toward Single Occupant Activity Recognition for Long-Term Periods via Channel State Information". In: *IEEE Internet of Things Journal* 11.2 (2024), pp. 2796–2807. DOI: 10.1109/JIOT.2023.3296472.

[227] Steven M Hernandez and Eyuphan Bulut. "Lightweight and Standalone IoT based WiFi Sensing for Active Repositioning and Mobility". In: *21st Inter-*

national Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM). Cork, Ireland, June 2020.

[228] Bin-Bin Zhang et al. "Unsupervised domain adaptation for rf-based gesture recognition". In: *IEEE Internet of Things Journal* 10.23 (2023), pp. 21026–21038.

[229] Md Touhiduzzaman. *mmWaveSimplePC: A Lightweight 3D Point-Cloud Extraction Tool for TI IWR1443Boost and IWR6843ISK radar devices*. `https://github.com/touhiDroid/mmWaveSimplePC`. Accessed: April 30, 2026.

[230] Delia Velasco-Montero et al. "Performance analysis of real-time DNN inference on Raspberry Pi". In: *Real-Time Image and Video Processing 2018*. Vol. 10670. SPIE. 2018, pp. 115–123.

[231] Anthony Berthelier et al. "Deep model compression and architecture optimization for embedded systems: A survey". In: *Journal of Signal Processing Systems* 93.8 (2021), pp. 863–878.

[232] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal machine learning: A survey and taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (2018), pp. 423–443.

[233] Yongshuo Zong, Oisin Mac Aodha, and Timothy M Hospedales. "Self-supervised multimodal learning: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47.7 (2024), pp. 5299–5318.

[234] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PmLR. 2021, pp. 8748–8763.

[235] Rui Zhou et al. “MmMulti: Multi-person Action Recognition Based on Multi-task Learning Using Millimeter Waves”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 9.2 (2025), pp. 1–25.

[236] Hana Ajakan et al. “Domain-adversarial neural networks”. In: *arXiv preprint arXiv:1412.4446* (2014).

[237] Kuniaki Saito et al. “Semi-supervised domain adaptation via minimax entropy”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 8050–8058.

[238] Yaroslav Ganin and Victor Lempitsky. “Unsupervised domain adaptation by backpropagation”. In: *International conference on machine learning*. PMLR. 2015, pp. 1180–1189.

[239] Guozhen Zhu et al. *AM-FM: A Foundation Model for Ambient Intelligence Through WiFi*. 2026. arXiv: 2602.11200 [cs.LG]. URL: https://arxiv.org/abs/2602.11200.

[240] Yue Zheng et al. “Zero-effort cross-domain gesture recognition with Wi-Fi”. In: *Proceedings of the 17th annual international conference on mobile systems, applications, and services*. 2019, pp. 313–325.

[241] Dan Wu et al. “WiTraj: Robust indoor motion tracking with WiFi signals”. In: *IEEE Transactions on Mobile Computing* 22.5 (2021), pp. 3062–3078.

[242] Anurag Pallaprolu, Winston Hurst, and Yasamin Mostofi. “mmRidge: Revealing Emergent Crowd Flow Structures with mmWave MIMO Radar”. In: *Proceedings of the 27th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. 2026, pp. 31–36.

[243] Anirban Banik et al. “mmSnap: Bayesian One-Shot Fusion in a Self-Calibrated mmWave Radar Network”. In: *IEEE Radar Conference (RadarConf25)*. 2025, pp. 1116–1121.

[244] Anurag Pallaprolu et al. “Crowd Analytics with a Single mmWave Radar”. In: *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (MobiCom)*. Washington D.C., DC, USA, 2024, pp. 830–844.

[245] Fengyu Wang et al. “mmHRV: Contactless Heart Rate Variability Monitoring Using Millimeter-Wave Radio”. In: *IEEE Internet of Things Journal* 8.22 (2021), pp. 16623–16636.

[246] Steven M Hernandez and Eyuphan 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)*. Cork, Ireland, June 2020.

[247] Ronald K Pearson et al. “Generalized hampel filters”. In: *EURASIP Journal on Advances in Signal Processing* 2016.1 (2016), p. 87.

[248] Texas Instruments. *UniFlash Flash Programming Tool*. `https://www.ti.com/tool/UNIFLASH`. Accessed: 2026-04-24. 2026.

[249] Zheng Yang et al. “Generative AI Meets Wireless Sensing: Towards Wireless Foundation Model”. In: *arXiv preprint arXiv:2509.15258* (2025).

[250] Nafeez Fahad and Eyuphan Bulut. “Channel Matters: Exploring LoS/NLoS Channel Effects on WiFi Sensing Performance”. In: *IEEE INFOCOM 2025-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2025, pp. 1–6.

[251] Zhanjun Hao et al. “Millimeter wave gesture recognition using multi-feature fusion models in complex scenes”. In: *Scientific Reports* 14.1 (2024), p. 13758.

[252] Jih-Tsun Yu, Yen-Hsiang Tseng, and Po-Hsuan Tseng. “A mmWave MIMO radar-based gesture recognition using fusion of range, velocity, and angular information”. In: *IEEE Sensors Journal* 24.6 (2024), pp. 9124–9134.

[253] Zhaoyang Xia et al. “Multidimensional feature representation and learning for robust hand-gesture recognition on commercial millimeter-wave radar”. In: *IEEE Transactions on Geoscience and Remote Sensing* 59.6 (2020), pp. 4749–4764.

[254] Yikai Li, CL Philip Chen, and Tong Zhang. “A survey on siamese network: Methodologies, applications, and opportunities”. In: *IEEE Transactions on artificial intelligence* 3.6 (2022), pp. 994–1014.

[255] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

VITA

Md Touhiduzzaman completed his Bachelor of Science in Computer Science and Engineering (CSE) degree from Bangladesh University of Engineering and Technology (BUET) in 2016. Then he continued to build up a startup on education technology, expanding to telecom and health-tech sectors and, eventually, made it acquired by an international corporate group. While his portfolio exhibits vast enterprise-grade software development and entrepreneurship, he intends to excel in Machine Learning and wireless technologies. Therefore, he is currently pursuing a Ph.D. degree in the Computer Science Department of Virginia Commonwealth University under the supervision of Dr. Eyuphan Bulut, and working on WiFi and wireless sensing applications.

Publications:

1. **M. Touhiduzzaman**, P. E. Pidcoe and E. Bulut, "Clinically Evaluated Practical WiFi Sensing for Real-Time Hand Rehabilitation Monitoring". (*to be submitted*) IEEE Journal of Selected Areas in Sensors, Special Section on Wireless Sensing AI and Technologies (II), July-2026.

2. **M. Touhiduzzaman** and E. Bulut, "AgnoGest: Robust Multimodal Locomotion Agnostic Gesture Sensing Using mmWave Point-Cloud and WiFi CSI on Edge Devices". (*under review*) IEEE Journal of Selected Areas in Sensors, Special Section on Wireless Sensing AI and Technologies (II), May-2026.

3. **M. Touhiduzzaman** and E. Bulut, "MockiFi: CSI Imitation using Context-Aware Conditional Neural Process for Zero-shot Learning". [Under Review] In Proc. of the 34th International Conference on Computer Communication and Networks (ICCCN 2025), August 4 - 7, 2025, Tokyo, Japan.

4. **M. Touhiduzzaman**, S. M. Hernandez, P. E. Pidcoe and E. Bulut, "Wi-PT-Hand: Wireless Sensing based Low-cost Physical Rehabilitation Tracking for Hand Movements with Segmentation and Counting". ACM Transactions on Computing for Healthcare Special Issue (ACM Health, ID: HEALTH-2023-0090), August-2024.

5. **M. Touhiduzzaman**, J. Chung, I. Pretzer-Abof and E. Bulut, "Wireless Sensing-based Daily Activity Tracking System Deployment in Low-Income Senior Housing Environments". In Proc. of IEEE the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'24) - 1st ACM Workshop on Wireless and Mobile Computing for AgeTech (MobiCom4AgeTech), Washington, D.C., USA, Nov. 18-22, 2024.

6. J. Chung, **M. Touhiduzzaman**, I. Pretzer-Abof, J. Karlsen, M. Vain, and E. Bulut, "Developing a Wi-Fi Sensing-Based Deep Learning Solution for Recognizing Daily Activities in Older Adults". In the journal of Innovation in Aging, volume 9, No.S2, pages 518–519, Oxford University Press, 2025.

7. S. M. Hernandez*, **M. Touhiduzzaman***, P. E. Pidcoe and E. Bulut, "Wi-PT: Wireless Sensing based Low-cost Physical Rehabilitation Tracking". In Proc. of the IEEE International Conference on E-health Networking, Application and Services (IEEE Healthcom), 17-19 October 2022, Genoa, Italy.

8. N. Fahad, **M. Touhiduzzaman** and E. Bulut, "Ensemble Learning based WiFi

Sensing using Spatially Distributed TX-RX Links". In Proc. of IEEE International Conference on Computing, Networking and Communications (ICNC'25), February 17-20, 2025, Honolulu, Hawaii, USA.

9. M. McDonough, T. Moomaw, **M. Touhiduzzaman** and E. Bulut, "Wi-Alert: WiFi Sensing for Real-time Package Theft Alerts at Residential Doorsteps". In Proc. of the 24th International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, 2023.