An Embedded Approach to Fall Detection and Human Activity Recognition using Wi-Fi Channel State Information

(This project report has been submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering)

Submitted By

Exam Roll: 63969

Reg No: 2017614892

Session: 2017-18

Exam Roll: 63949

Reg No: 2017114842

Session: 2017-18

Dept. of Electrical and Electronic Engineering

University of Dhaka

June 27, 2022

Abstract

Fall detection is an essential part of any elderly or patient assistance system. Human activity recognition techniques are widely used to implement fall detection systems. Most solutions employ wearable devices or cameras to collect data, both of which possess various critical drawbacks. We propose a device-free fall detection system using WiFi channel state information(CSI) which lacks the drawbacks of mentioned solutions yet provides competitive performance. We used two embedded devices(ESP32) to collect CSI (Channel State Information) as an embedded approach provides more flexibility in case of deployment. We recorded data while 13 volunteers performed various tasks. After rigorous preprocessing involving resampling, phase calibration, amplitude denoising, filtering, feature extraction, feature selection, and classification, we were able to achieve an F1 score of 98.5% in 10-fold cross-validation for fall detection and 96.9% for human activity recognition.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Healthcare has always been a fundamental concern for human society, and it is especially needed for the sick and the elderly. The percentage of persons aged over 65 was 9% in 2020, with the rate being as high as 28% in a single country [4]. For an elderly person or a sick patient, falling can be devastating, and immediate attention is required in such a scenario. The development of fall detection systems is essential for this reason. Although many fall detection systems exist today, most of them are based on wearable devices or computer vision, which have some critical drawbacks, caused by the fundamental nature of the systems. This project is aimed to provide a non-contact fall detection system that is free of those drawbacks and is yet as effective as the existing solutions. Furthermore, we use embedded devices as this makes the system highly modifiable, deployable in many different environments, and comparatively affordable.

1.1 Motivation

Sensor-based activity detection requires the user to wear a device containing the sensors in many cases[5]. Some methods use computer vision for the task, which requires a camera to collect video or image data[6]. Although these methods can detect a fall incident quite accurately, some issues caused by these methods can make implementing them in real-world scenario a challenge. The issues faced while implementing solutions based on these methods are:

- A wearable device can be perceived as uncomfortable, causing unwillingness to use them.
- Remembering to wear a device every day can be an issue for elderly people.
- A wearable device is more prone to wear and tear than a stationary device.
- A camera-based solution can be both computationally and monetarily expensive.
- A camera-based solution cannot provide reliability in adverse lighting conditions.

Comfort is of utmost importance for the elderly and the sick. A user might be understandably unwilling to wear a device if it is uncomfortable for them. Even if a system is perfectly capable of performing its assigned task, implementing it is challenging, if not impossible when the users are not willing to cooperate. Also to have a widespread application of a system, cost and durability have to be considered, especially in developing countries. So, in this project, we aim to create a non-contact fall detection system using wifi channel state information that is more comfortable, more reliable, and less costly.

1.2 Objectives

We aim to build a non-contact fall detection system for monitoring the elderly and the sick. The objectives we aim to achieve are as follows:

- Implementing a system that can detect if a person in its area of operation has fallen down
- Ensuring comfort by eliminating the need for wearing any device
- Creating a dataset of channel state information recorded during different activities including fall
- Recognition of different human activities using Wi-Fi channel state information to facilitate future improvement opportunities

- Implementing a system to detect if the area of operation is empty
- Ensuring that the system is easy to deploy and affordable by using embedded devices

CHAPTER 2

RELATED WORKS

2.1 Device Free Human Activity Recognition using WiFi Channel State Information

In this study [7], the authors implemented an activity detection system using wifi channel state information. They were able to detect human activities like Walk, Stand, Sit, Run, etc. in a Line of Sight scenario (LOS) and a Non-Line of Sight (N-LOS) scenario within an indoor environment. They used two algorithms for classification, Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) recurrent neural network. To collect the data, they used Intel WiFi Link 5300 Network Interface Card (NIC). This card supports the 802.11n standard and hence makes it possible to record channel state information. There are 64 subcarriers in 20 MHz channel and 128 subcarriers in 40 MHz channel. Two Lenovo laptops were used which were equipped with Intel WiFi Link 5300 Network Interface Card (NIC). The operating system on said laptops was 64 Bit Ubuntu version 14.04 LTS. The kernel version was 4.2.0-42. They modified the hardware using instructions provided by Halperin et al. [8] who proposed the 'Linux 802.11n CSI Tool". For classification with SVM, they performed preprocessing and feature extraction using Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA), etc, and then used the classifier. For classification with LSTM, only CSI-extraction and denoising were done. They achieved results where precision, recall, and F1 score were 95, 98, and 96 respectively for their best model.

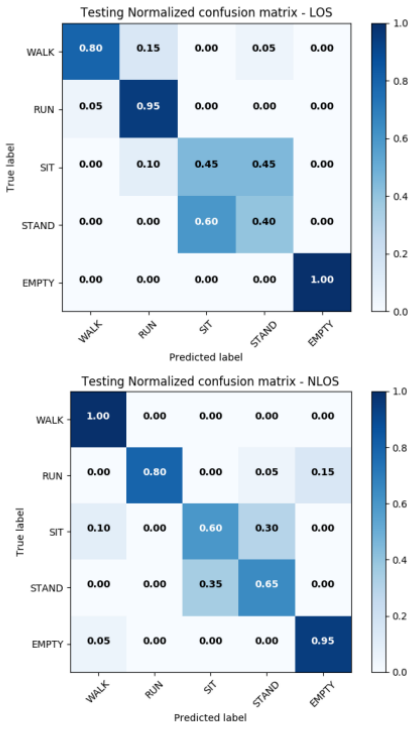


Fig. 4. Confusion Matrix SVM

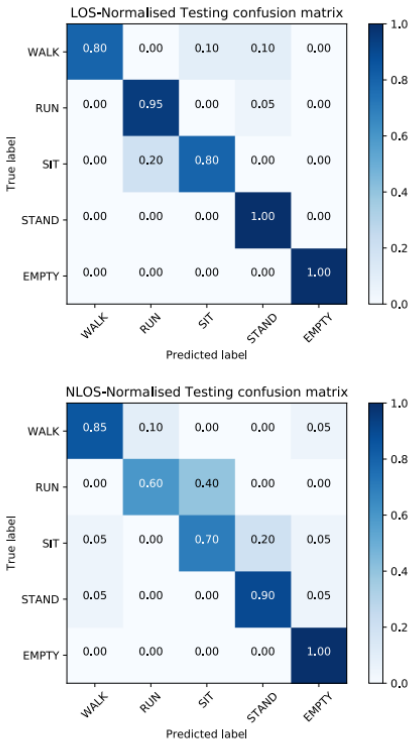


Fig. 5. Confusion Matrix LSTM

Figure 2.1: Confusion matrices in the paper

There are some limitations in this method:

- The hardware used for the experiment was two laptops, making this solution impractical for use in a real-world scenario.
- The system shows lower accuracy in detecting activities involving slow movement, like sitting or walking.
- The system does not generalize for different environments.

2.2 Wi-Motion: A Robust Human Activity Recognition Using WiFi Signals

This study [9]proposes a wifi-based human activity recognition system, Wi-Motion. The authors were able to classify five different pre-defined activities with impressive accuracy. The system showed a 96% accuracy in line of sight arrangement and 92% accuracy in non-line of sight arrangement. Furthermore, the authors evaluate the effect of the age of the experimental subjects and relatively complex environments. Wi-Motion jointly leverages the amplitude and phase information extracted from the CSI sequence. The authors first construct the classifiers using amplitude and phase, respectively. The output of classifiers is then combined by a posterior probability-based combination strategy. The authors used a commercial Tp-Link wireless router as the transmitter operating in the IEEE 802.11n AP mode at 2.4GHz. An Acer Aspire EC laptop running Ubuntu 14.04 was used as a receiver, which is equipped with an off-the-shelf Intel 5300 card (three antennas) and a modified firmware. During the process of receiving WiFi signals, the receiver pings the router 33 pkts/s and records the CSI of each packet. For each activity in different environments, every user provides 30 instances to evaluate the performance of their system. Two complex office environments were selected for data collection and 6 participants provided the data. They extracted amplitude features using DWT (Discrete Wavelet Transform). The phase feature extraction is done using WMA method and PCA. The authors use a support vector machine (SVM) algorithm for the classification of the five activities:

1. Bend
2. Halve squat
3. Step
4. Stretch leg
5. Jump

The authors also showed that the system provided accuracy higher than 80% even when there were multiple users present. Moreover, they showed with their analysis

that during the process of data collection, the physiological function of the person decays as the age increases, which makes the movement slower and difficult to control in a stable situation.

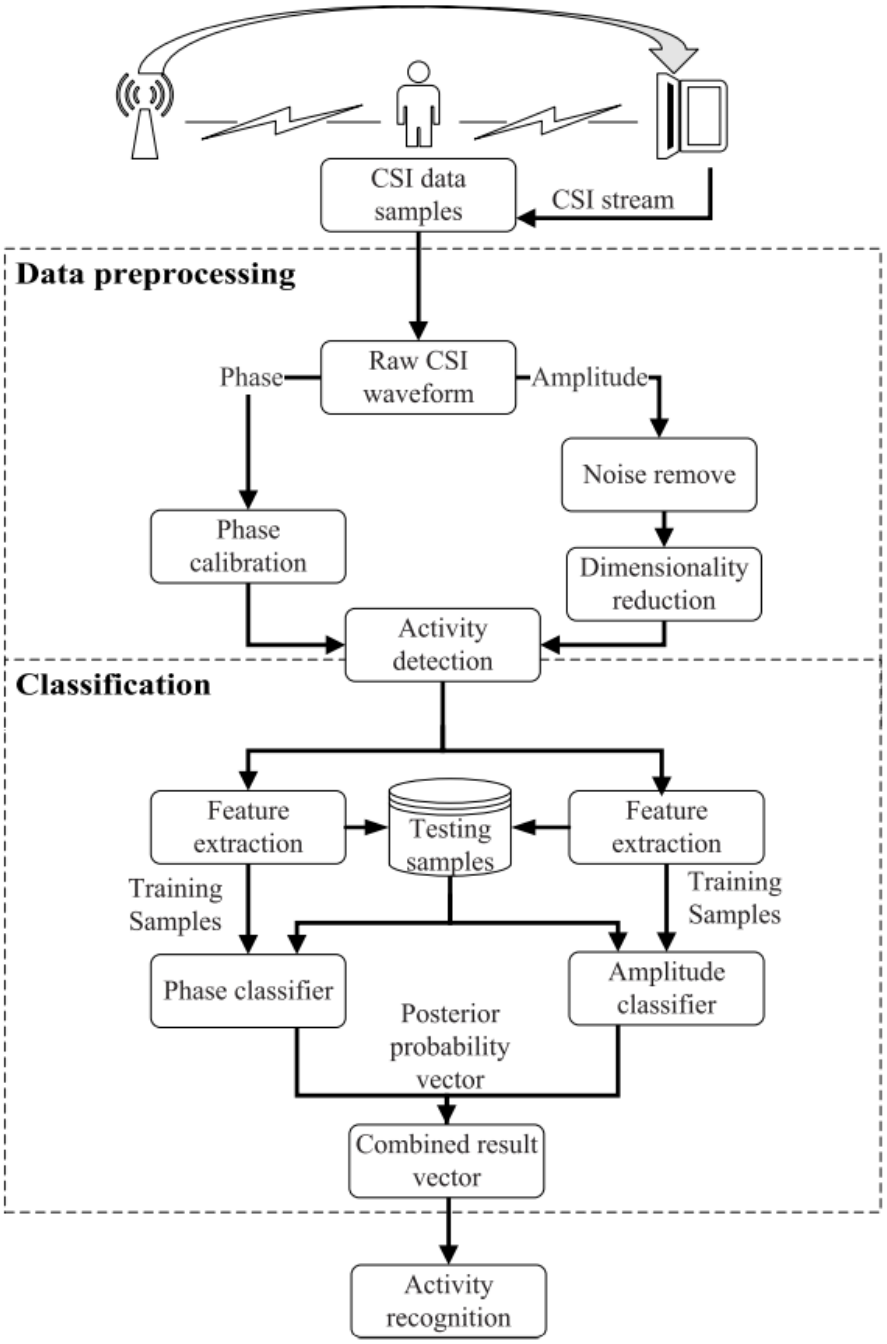
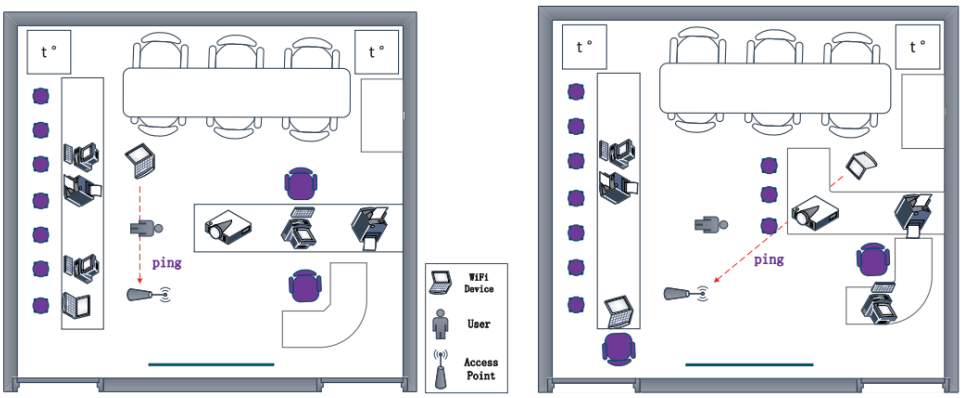


Figure 2.2: System structure for Wi-Motion



(a) Environment A of data collection (LOS). (b) Environment B of data collection (NLOS).

Figure 2.3: Environment set of data collection.

Limitations:

- The hardware used for the experiment were two laptops, making this solution impractical for use in a real-world scenerio.

2.3 MultiSense: Enabling Multi-person Respiration Sensing with Commodity WiFi

The study [10] proposes Multisense, a WiFi-based system that can continuously sense the detailed respiration patterns of multiple persons. It can provide a robust performance even if they have very similar respiration rates and are physically closely located. The main contributions of the paper are as follows:

- The authors offered a novel method for canceling out the time-varying phase offset of WiFi CSI without distorting the linear mixture.
- They showed that respiration sensing can be treated as a BSS problem that the ICA approach can efficiently address.
- They put MultiSense on common WiFi devices and ran comprehensive tests to see how well it works.

The authors collected CSI data using the CSI tool [11], which reports the complex-valued CSI samples for each received packet and can be used to collect CSI data from the receiver. For reporting CSI, the Intel 5300 WiFi card in the receiver is set to run at 5.24 GHz with a sample rate of 200 Hz and provides CSI information on 30 sub-carriers. The transmitter and receiver are both equipped with three antennas unless otherwise noted.

(a) (b)

Figure 2.4: The experimental setup in two scenarios: (a) all subjects sleep on a bed in the bedroom; (b) each subject sits on a couch or chair in the living room.

The authors used ICA (Independent Component Analysis) to separate the breathing patterns of different persons. The added time-varying phase offset (t) and background static signals affect CSI from commodity WiFi. As a result, using raw CSI retrieved from commodity WiFi equipment, the authors were unable to use ICA to distinguish multi-person respiration signals. To overcome this, they proposed a novel method where they canceled the time-variant phase offset and removed the background static signal. Even in the presence of four people and only a pair of Wi-Fi transceivers, MultiSense is quite accurate, with a mean absolute respiration rate inaccuracy of 0.73 bpm.

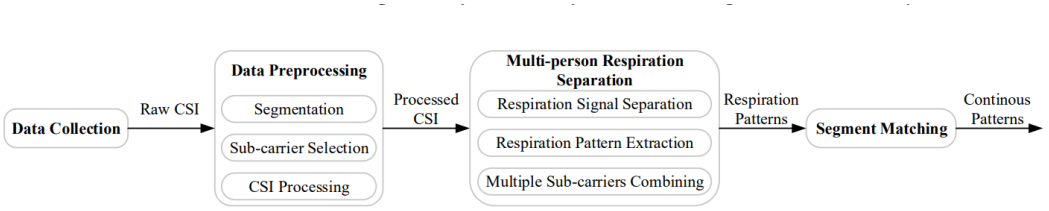


Figure 2.5: The MultiSense system overview.

Limitations:

- When performing blind source separation using ICA method, the number of persons is required as the input as an inherent characteristic of ICA is that it cannot identify the actual number of source signals in general.
- The system is not able to assign the breathing patterns to users in the case of multiple persons.

2.4 A Wireless-Vision Dataset for Privacy Preserving Human Activity Recognition

This study [12] proposes a new WiFi-based and video-based neural network (WiNN) to improve the robustness of activity recognition where the synchronized video serves as the supplement for the wireless data. In three different visual circumstances, including scenes without occlusion, partial occlusion, and full occlusion, a wireless-vision benchmark (WiVi) is gathered for 9 class actions recognition. The accuracy of the data set is verified using both machine learning methods - support vector machine (SVM) and deep learning methods. The authors show that the WiVi data set meets the primary demand and that all three branches of the proposed pipeline maintain accuracy of more than 80% for multiple action segmentation from 1s to 3s.

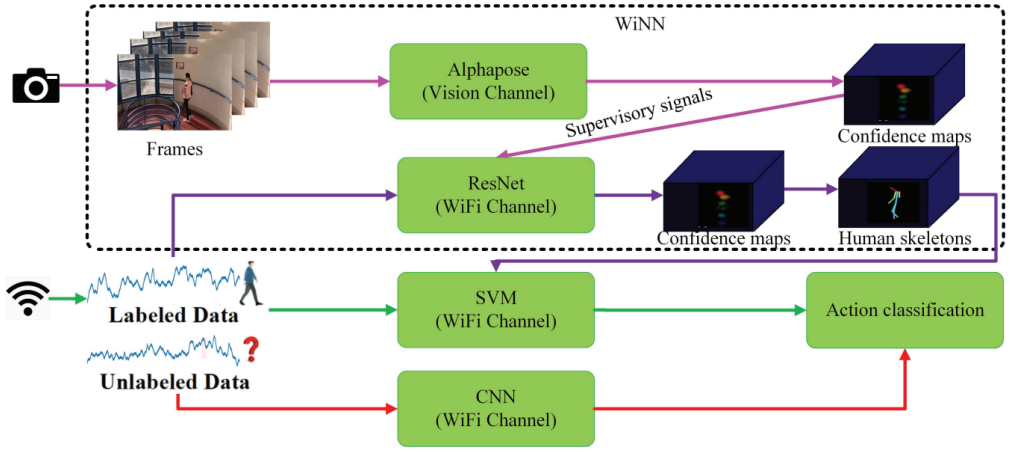


Figure 2.6: The flowchart of the network for WiVi dataset.

The contribution of the paper can be summarized by the following points:

- To test the effectiveness of existing activity identification systems, the authors first created WiVi, a wireless-vision activity data set. To verify the WiVi dataset's effectiveness, they used SVM, Convolutional Neural Networks (CNN), and WiNN.
- The authors proposed the WiNN, a WiFi-based and video-based neural network for activity recognition in partial and full occlusion scenarios, which improves the robustness of activity recognition using synchronous video as a supplement and complement to WiFi CSI signals.
- To verify the quality of the WiVi data set, the authors compared the machine learning method SVM with the deep learning methods CNN and WiNN. WiNN, in particular, delivered the most reliable results for multiple action segmentation from 1 to 3 seconds.

Figure 2.7: The visual skeleton result of the CSI in two scenarios, where A-G are without occlusion scene, and H-M are partial occlusion scene.

Limitations:

- The number of participants was very small.
- The baseline SVM model performed better than their proposed WiNN model.

CHAPTER 3

THEORETICAL OVERVIEW

This chapter provides a theoretical overview of the project as well as specifications for different hardware tools and technologies that are employed.

3.1 Hardware Specification

We use two esp32(ESP-WROOM32) microcontrollers one as a transmitter, and one as a receiver. The features of the device are given below [13]:

1 Processors

1a) CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS

1b) Ultra-low power (ULP) co-processor

2 Memory: 320 KiB RAM, 448 KiB ROM

3 Wireless connectivity:

3a) Wi-Fi: 802.11 b/g/n

3b) Bluetooth: v4.2 BR/EDR and BLE (shares the radio with Wi-Fi)

4 Peripheral interfaces:

4a) 34 × programmable GPIOs

4b) 12-bit SAR ADC up to 18 channels

4c) 2 × 8-bit DACs

4d) 10 × touch sensors (capacitive sensing GPIOs)

4e) 4 × SPI

4f) 2 × I^2S interfaces

4g) 2 × I^2C interfaces

4h) 3 × UART

4i) SD/SDIO/CE-ATA/MMC/eMMC host controller

4j) SDIO/SPI slave controller

4k) Ethernet MAC interface with dedicated DMA and planned IEEE 1588 Precision Time Protocol support[4]

4l) CAN bus 2.0

4m) Infrared remote controller (TX/RX, up to 8 channels)

4n) Motor PWM

4o) LED PWM (up to 16 channels)

4p) Hall effect sensor

4q) Ultra low power analog preamplifier Security: IEEE 802.11 standard security features all supported, including WPA, WPA2, WPA3 (depending on version)[5] and WLAN Authentication and Privacy Infrastructure (WAPI)

4q)0.0.1. Secure boot

4r) Flash encryption

4s) 1024-bit OTP, up to 768-bit for customers

4t) Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG) Power management:

4t)1. Internal low-dropout regulator

4u) Individual power domain for RTC

4v) 5 μA deep sleep current

4w) Wake up from GPIO interrupt, timer, ADC measurements, capacitive touch sensor interrupt

Figure 3.1: A esp32 microcontroller

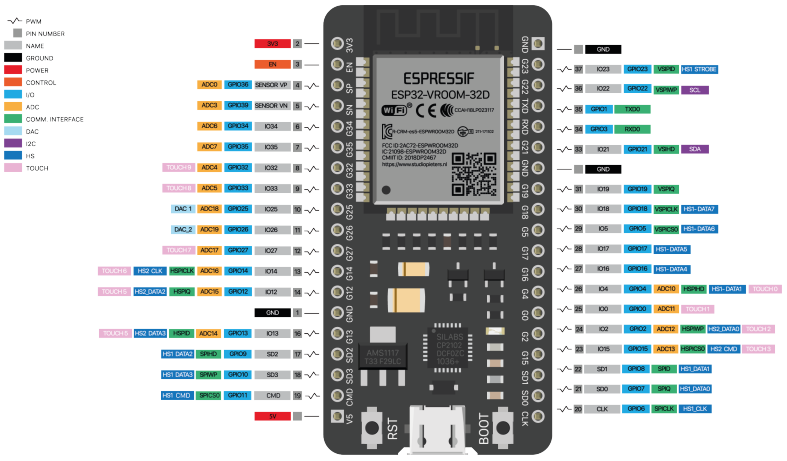


Figure 3.2: Pinout diagram of esp32 microcontroller

The signal specification for esp32 microconroller is given below:

- Bandwidth: 20 MHz
- Antenna: 1 RX and 1 TX
- Protocol: 802.11n
- Modulation: OFDM (16 QAM)
- Subcarrier Number: 64
- Sampling Rate: 3.9 Hz
- Average RSSI: -77 dBm
- Guard Interval: 800 ns (MCS Index: 4)
- Technologies: MIMO, Frame Aggregation

The advantages of using esp32 are:

- esp32 can function as a stand-alone system or as a slave device to a host MCU, eliminating communication stack overhead on the primary application CPU.

- Through its SPI / SDIO or I2C / UART interfaces, the esp32 may communicate with other systems to provide Wi-Fi and Bluetooth capability.
- esp32 has a low-power processor designed for mobile devices, wearable electronics, and IoT applications. It uses a combination of proprietary software to achieve ultra-low power consumption.
- esp32 is highly-integrated with in-built antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules.

In summary, esp32 is ideal for our project because is a power-efficient device that is capable of using wifi communication, is easily integrable with other systems, and has a fair amount of computing power.

3.2 Wi-Fi

In this project, we exploited the capability of Wi-Fi technology to implement fall detection. Wi-Fi, vastly used in high-speed internet access and wireless communication, is a set of protocols governed by IEEE 802.11 standards [14]. IEEE 802.11 is part of the local area network (LAN) technical standards and specifies the set of protocols for implementing wireless local area network (WLAN) computer communication. These standards are maintained by the Institute of Electrical and Electronics Engineers (IEEE). Though the first edition of these standards was released in 1997, continuous development is being made and new standards are coming with more capabilities to meet the ever-increasing demand for high-speed wireless communication. The most notable standards of IEEE 802.11 are 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, and 802.11ax.

3.2.1 802.11a

This was the first standard to use the 5 GHz band for Wi-Fi which might seem to be ahead of its time. But because of the higher frequency, its coverage area

was much lower than the traditional 2.4 GHz band and suffered much from interference problem. That is why 802.11a was not so popular compared to its 2.4 GHz counterpart even though it had a higher data rate and went obsolete quickly. But the main contribution of this standard was the introduction of Orthogonal Frequency-Division Multiplexing (OFDM) which improved the data transmission drastically. OFDM is based on the concept of orthogonal subcarriers with minimal interference that makes it possible to cope with severe channel conditions without complex equalization filters. OFDM is described in detail later in this section. 802.11a uses 52 subcarriers in OFDM, of which 48 subcarriers are used for data transmission and the rest 4 subcarriers are used as pilot subcarriers.

3.2.2 802.11b

802.11b was the first widely accepted standard of Wi-Fi. Both 802.11a and 802.11b were released in 1999, with a major difference between them. Unlike 802.11a, 802.11b uses 2.4 GHz band. The 2.4 GHz band was not as crowded as today and offered higher coverage and the capability to withstand interference. These advantages made 802.11b popular despite having a much lower data rate (up to 5.5 Mbit/s). This standard is still in use in some legacy devices.

3.2.3 802.11g

Introduced in 2003, 802.11g was a mixture of the previous two standards. It operated in the 2.4 GHz band like 802.11b and utilizes the same OFDM-based transmission scheme as 802.11a. This technical change gave a burst increase in the data rate which could go up to 54 Mbit/s. 802.11g also uses a total of 52 subcarriers with a carrier separation of 0.3125 MHz. There are 14 partially overlapping channels each of which has a separation of 20 or 25 MHz.

3.2.4 802.11n

This standard is also known as Wi-Fi Generation 4 (Wi-Fi 4). It includes several new technologies that increased the capability of Wi-Fi further. The most notable additions to this standard are:

- Multiple Input Multiple Output (MIMO)
- Frame aggregation
- WiFi Beamforming (Optional)
- 40 MHz channel bandwidth
- Security enhancement

MIMO technology is capable of conducting simultaneous data transmission over multiple antennas. Frame aggregation allows sending two or more frames in a single transmission. Beamforming improves the user experience by focusing the Wi-Fi beams in the user's direction. Thus these new features along with OFDM increased the data rate from 72 Mbit/s to 600 Mvit/s. 802.11n has support for the 2.4 GHz band and optionally for the 5 GHz band. Most Wi-Fi-enabled devices are still using this standard today.

3.2.5 Newer standards

After 802.11n, a few major standards have come out that have increased the data rate, reliability, and security further. 802.11ac (Wi-Fi 5) is currently spreading in the consumer community which uses only the 5 GHz band. It introduced a few new features, such as Multi-User MIMO (MU-MIMO), wider 80 MHz and 160 MHz channels, and Beamforming.

802.11ax (Wi-Fi 6E) is the most recently approved standard adopted in 202 which uses three bands: 2.4 GHz, 5 GHz, and 6 GHz. The data rate can vary from 600 Mbit/s to 9608 Mbit/s. Currently, the development is being made for 802.11be standard or Wi-Fi 7 which will provide even more data rate.

The hardware used in our proposed method, esp32 uses the popular IEEE 802.11n standard. It currently has the largest user base and can utilize several recent technologies including OFDM, MIMO, and frame aggregation.

3.2.6 OFDM

Orthogonal Frequency-Division Multiplexing (OFDM) is a sort of digital transmission and a way of encoding digital data on multiple carrier frequencies that is used in telecommunications. OFDM is a widely used wideband digital communication technique, with applications including digital television and audio broadcasting, DSL internet access, wireless networks, power line networks, and 4G/5G mobile communications.[15] The capacity of OFDM to cope with severe channel conditions without the use of sophisticated equalization filters is its fundamental benefit over single-carrier methods. Because OFDM uses numerous slowly modulated narrowband signals rather than a single rapidly modulated wideband signal, channel equalization is simpler. This mechanism also makes it easier to design single frequency networks (SFNs), in which multiple adjacent transmitters send the same signal at the same frequency at the same time, because the signals from multiple distant transmitters can be constructively recombined, avoiding the interference that a traditional single-carrier system would face.

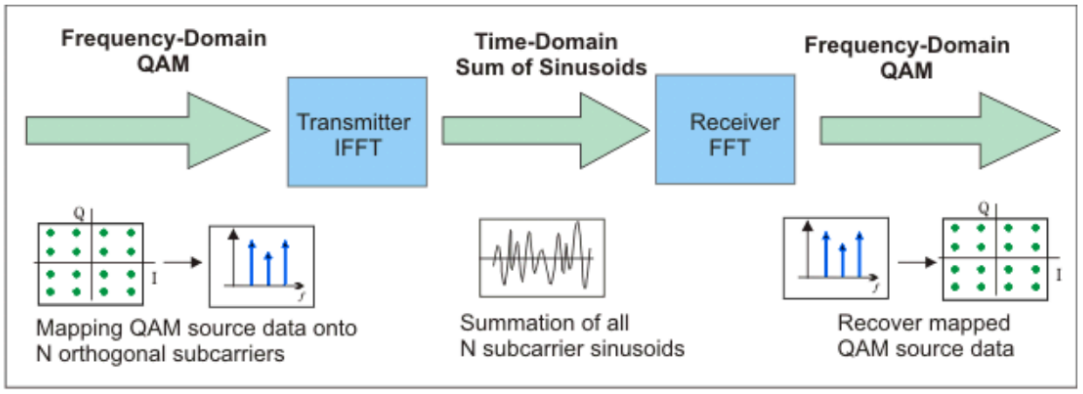


Figure 3.3: Block Diagram of Simplified OFDM System

3.2.7 MIMO

Multiple-Input MIMO (Many-Input, Multiple-Output) is a wireless technology that uses multiple transmitters and receivers to carry more data at once. MIMO is supported by all 802.11n wireless equipment. The technology enables 802.11n to achieve faster rates than goods that do not have 802.11n.[16] MIMO must be supported by the station (mobile device) or the access point (AP) to be implemented. Both the station and the access point must support MIMO for the best performance and range. Multipath, a natural radio-wave phenomenon, is used in MIMO technology. Multipath occurs when transmitted data bounces off walls, ceilings, and other obstacles, arriving at the receiving antenna numerous times at slightly varying angles and times. Multipath created interference and hindered wireless communications in the past. MIMO technology with multipath combines numerous, smart transmitters and receivers with an extra spatial dimension to improve performance and range. By allowing antennas to mix data streams arriving from diverse paths and at different times, MIMO boosts the signal-capturing power of receivers. Smart antennas make use of spatial diversity technology, which makes use of unused antennas. When the number of antennas outnumbers the number of spatial streams, the antennas can boost receiver variety and range.[17]

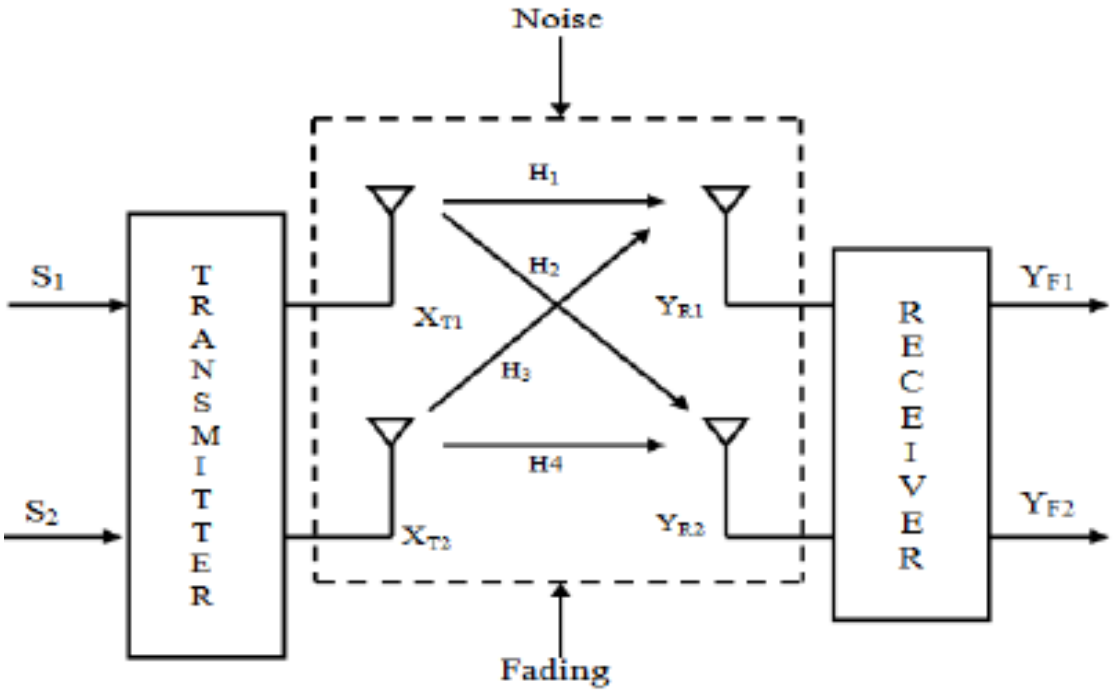


Figure 3.4: Block Diagram of MIMO System

3.3 Signals used for analysis

Activity recognition using Wi-Fi data is not a completely new domain. Older methods of human activity recognition used Received Signal Strength Indication (RSSI) signal. But recently with the development in WLAN fields, newer standards starting from 802.11g can provide Channel State Information (CSI) signal too. In our proposed method, we utilized both signals to classify human activity more accurately. Here is an overview of these signals.

3.3.1 RSSI

The RSSI is a measurement of the signal's power at the moment it reaches the receiver. Signal energy diminishes with distance, according to signal propagation models and experiments. As a result, RSSI is frequently used in conjunction with multilateration algorithms to estimate position. When impediments are present

in the region of interest, RSSI suffers from several drawbacks. RSSI values are distributed randomly and its correlation with the distance is not strong due to Multipath and shadowing fading effects. RSSI values are comparatively coarse information which is the result of averaging the amplitudes of all incoming signals to the receiver. These drawbacks lead to poor localization performance using RSSI [18]. This problem can be solved by using rich channel information from different subcarriers.

3.3.2 CSI

In IEEE 802.11 a/g/n/ac/ax networks, data transmission and reception is done using OFDM. As discussed earlier, OFDM uses a number of orthogonal subcarriers to transmit data in multiple spatial paths. While a transmitting packet is in the medium, it is subjected to different obstructions, such as, fading, scattering and power loss. As the subcarriers follow different spatial paths, these obstructions affect each subcarrier differently. Thus this physical layer information specific to each subcarrier is known as Channel State Information (CSI). CSI is an overall depiction of the channel state that includes scattering, fading, and multipath effects in the signal's propagation. In contrast to the received power strength provided by RSSI, CSI statistics provide more information about the channel degradation effects that the signal suffers due to its granularity of sub-carrier frequencies and vector representation. Data is sent using MIMO and OFDM systems.

In narrow-band flat fading channel, a MIMO system is represented by:

$$y_i = Hx_i + N_i \tag{3.1}$$

where y_i and x_i are the received and the transmitted signal vectors respectively, H denotes the channel matrix which contains the CSI information and N_i is the noise vector. To estimate the channel matrix H, a known training sequence or the pilot sequence is transmitted and channel response H is measured at the receiver side. If the pilot sequence is expressed by $x_1, x_2, x_3, ..., x_n$, the received can be represented by:

$$Y = [y_1 + y_2 + ... + y_n] = HX + N \tag{3.2}$$

Thus the channel matric can be determined by:

$$\hat{H} = \frac{Y}{X} \tag{3.3}$$

For any MIMO system of $n \times m$ dimension, H can be shown in matrix form:

$$H_i = \begin{bmatrix} h_{11} & h_{12} & h_{13} & ... & h_{1m} \\ h_{21} & h_{22} & h_{23} & ... & h_{2m} \\ ... & ... & ... & ... & ... \\ h_{n1} & h_{n2} & h_{n3} & ... & h_{nm} \end{bmatrix} \tag{3.4}$$

where i is the subcarrier index and h_nm is a complex number representing the amplitude and phase information of Channel State Information (CSI).

3.4 Machine Learning

In the 1950s, a branch of artificial intelligence known as machine learning was discovered and developed. The earliest machine learning techniques date back to the 1950s, however there have been very few notable studies and advancements in this field. However, this field of study underwent a resurgence in the 1990s and has continued to this day. Future advancements in this field of study are expected. The complexity of analyzing and interpreting the data, which is continually expanding, is what has led to this development. The foundation of machine learning is the idea that, with the help of this growing data, the best model for the new data may be found among the old data. As a result, research into machine learning will continue along with the growth in data.[19] The actions performed by computers, which are based on an algorithm and follow specific procedures, have no margin for error. In some circumstances, computers make judgments based on the current sample data, which is different from commands that are created to produce an outcome depending on an input. In some circumstances, computers may err in their decision-making just like people do. Putting it another way, machine learning is the process of giving computers the capacity to learn from data and experience just like a human brain.[20]The primary goal of machine learning is to develop

models that can learn from previous data to become better, recognize complicated patterns, and find answers to new problems.[21]

3.4.1 Machine Learning Categories

We can divide machine learning approaches in four categories.They are:

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforced Learning

Supervised learning is a technique where the currently available input data is used to arrive at the outcome set. Classification and regression supervised learning are the two categories of supervised learning.

1. Classification: Dividing the data into the categories specified in the data set in accordance with their unique characteristics.
2. Regression: Predicting or drawing conclusions about the other characteristics of the data from the known characteristics.

Unsupervised learning is the technique where the output is not provided while training the model.The algorithms following this technique can be classified into two categories:

1. Clustering: When intrinsic groupings in the data are unknown, finding groups of data that are comparable to one another.
2. Association: Figuring out the links and relationships between the data in the same data collection.

Semi-supervised learning is a method of machine learning that, during training, blends a sizable amount of unlabeled data with a small amount of labeled data. Between supervised learning (with labeled training data) and unsupervised learning is semi-supervised learning (with only labeled training data). It is a unique illustration of poor supervision. Either inductive learning or transductive learning may be referred to as semi-supervised learning.[22]

Reinforcement learning is the challenge that an agent faces when learning behavior through trial-and-error interactions with a dynamic environment. Reinforcement learning differs from supervised learning in that it does not need the presentation of labeled input/output pairings or the explicit correction of suboptimal behaviors. Instead, the emphasis is on striking a balance between exploitation and exploration (of undiscovered territory) (of current knowledge). The benefits of supervised and RL algorithms can be combined with partially supervised RL algorithms.[23]

3.5 Human Activity Recognition

Human activity recognition is important for interpersonal interactions and human-to-human communication. It is challenging to extract since it contains details about a person's identity, personality, and psychological condition. One of the key research topics in the fields of computer vision and machine learning is the human capacity for activity recognition. This research has led to the need for multiple activity detection systems in numerous applications, such as video surveillance systems, human-computer interaction, and robotics for characterizing human behavior. Most of the work in human activity recognition assumes a figure-centric scene of an uncluttered background, where the actor is free to perform an activity. The development of a fully automated human activity recognition system, capable of classifying a person's activities with low error, is a challenging task due to problems, such as background clutter, partial occlusion, changes in scale, viewpoint, lighting and appearance, and frame resolution.[24]

To solve these issues, a task is needed that combines three elements: (i) background subtraction, in which the system tries to distinguish between the foreground's changing or moving objects and the background's parts;[25][26] (ii) human tracking, in which the system tracks a person's motion over time;[27] and (iii) human action and object detection, in which the system can localize a person's activity.[28]

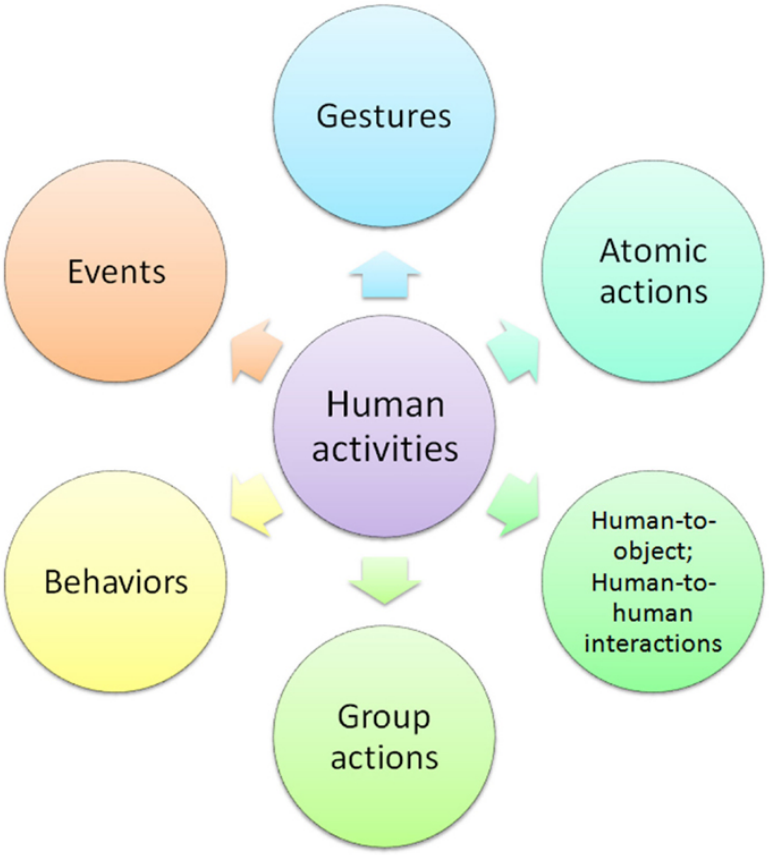


Figure 3.5: Decomposition of human activities.

3.6 Classification Algorithms

Classification algorithm is a Supervised Learning technique that is used to categorize new observations based on training data. In classification, a program makes use of the dataset or observations that are provided to learn how to categorize new observations into various classes or groups. Some algorithms are discussed here:

- SVM: Support vector machines (SVMs) are a group of supervised learning techniques for classifying data, performing regression analysis, and identifying outliers. Support vector machines have the following benefits: efficient in high-dimensional environments. Still useful in situations where the number of dimensions exceeds the number of samples.[29]

- Random Forest: A large number of decision trees are built during the training phase of the random forests or random decision forests ensemble learning approach, which is used for classification, regression, and other tasks. The class that the majority of the trees chose is the output of the random forest for classification problems. Decision trees tend to overfit their training set, and random decision forests correct for this. Although they frequently outperform decision trees, gradient boosted trees are more accurate than random forests.[30]

- Extra Trees Classifier: In essence, it involves dividing a tree node while severely randomizing the choice of attribute and cut-point. In the worst situation, it creates completely random trees, whose architectures are independent of the learning sample's output values. By selecting the right parameter, the strength of the randomization can be adjusted to the particulars of the problem. The algorithm's biggest advantage, aside from accuracy, is computational speed.[31]

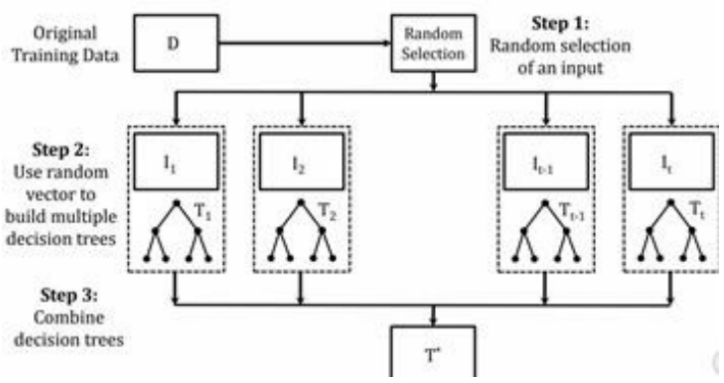


Figure 3.6: Visual Representation of Extra Trees Classifier.

- Artificial Neural Network(ANN): Artificial neural network is based on research into the brain and nervous system, as seen in Fig. 1. Although

they employ a condensed set of biological brain system ideas, these networks mimic biological neural networks. Particularly, ANN models mimic the electrical activity of the nerve system and brain. Connected to other processing elements are processing elements (sometimes called neurodes or perceptrons). The neurodes are typically organized in layers or vectors, with the output of one layer acting as the input for the following layer and maybe other layers.[32]

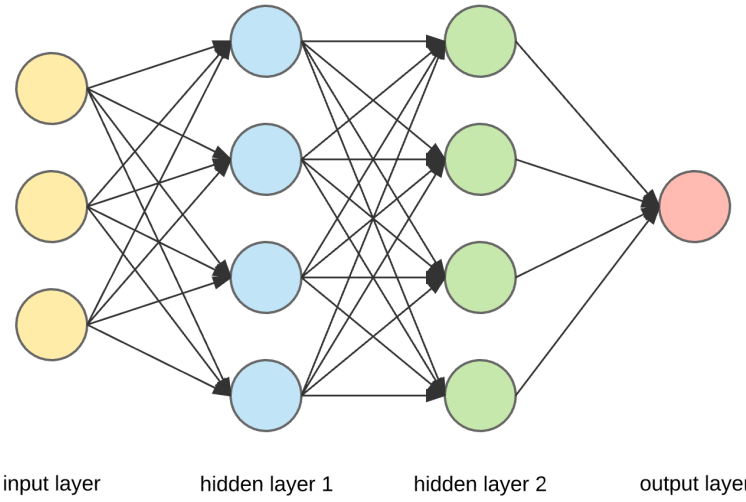


Figure 3.7: Visual Representation of an Artificial Neural Network.

CHAPTER 4

METHODOLOGY

In this chapter, we will discuss the implementation of our proposed system. At first, we will describe the hardware setup used for collecting user data. Different aspects of the dataset were mentioned in the next subsection. Later, we discussed how we extracted, processed, calibrated, and denoised the raw data and made them usable for our model. Finally, a brief overview of our proposed model and approach is given.

4.1 Hardware Setup

The main hardware we used for this system is a pair of ESP32 MCU manufactured by espressif. The specific model of the used hardware module is ESP32-WROOM-32E. This is an ESP32-D0WD-based module with Wi-Fi 802.11 b/g/n and Bluetooth LE 4.2 connectivity and a dual-core processor. Traditional research on Wi-Fi-based human activity recognition uses Intel 5300 or Atheros 9390 Network Interface Card (NIC) of a laptop computer [7, 9, 12, 33, 34] which is not a realistic choice for practical use because each node is a computer. The device we used is small, low-cost, programmable, and deployment-friendly and the whole system needs only one computer to process the data.

For our experiment, we need to send CSI data from one ESP32 device to another. But ESP32 does not transmit CSI data with the initially provided firmware. So, a

customized firmware by Espressif Systems [35] is flashed to the devices to enable the transmission of CSI data. CSI data can be received using three ways:

1. Get router CSI data: In this process, a router is used to send CSI data to the ESP32. Firstly, the ESP32 device sends a Ping request to the router with an empty ICMP packet. The router acknowledges the request by sending a Ping Replay back to the requesting device. The CSI information is transmitted with the Ping Replay. One disadvantage of this process is that we need an extra router device and set it up separately to send CSI data.

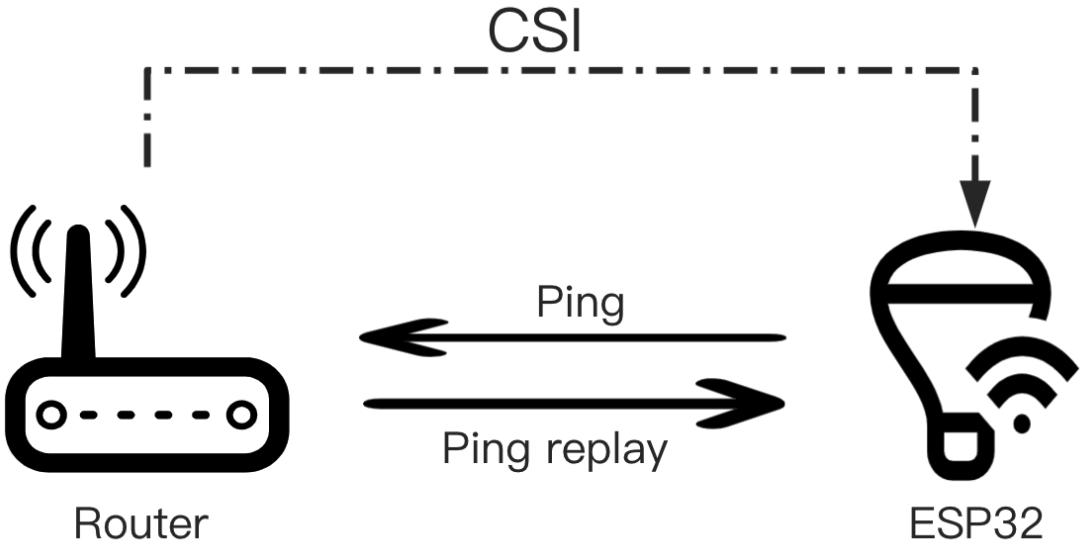


Figure 4.1: Get CSI data of the router

2. Get device CSI data using router: To implement this method, we need two ESP32 devices. ESP32 A and B both send Ping packets to the router, and ESP32 A receives the CSI information carried in the Ping Replay returned by ESP32 B. In this method, the CSI data is passed from ESP32 A to ESP32 B using an intermediate router. This intermediate connection may reduce the packet receiving rate of the system.

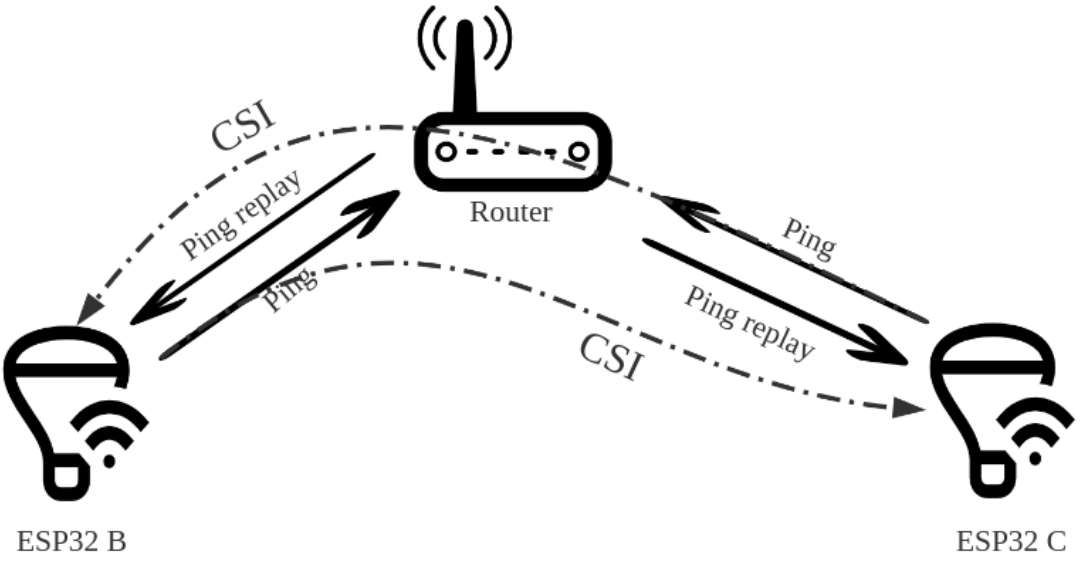


Figure 4.2: Get CSI data between devices using a router

3. Get device CSI data using broadcasting: In this method, one ESP32 device acts as a transmitting device and all other devices are receiving device. The transmitting ESP32 A sends CSI data using broadcasting. This method has the highest detection accuracy and reliability and does not require any router device.

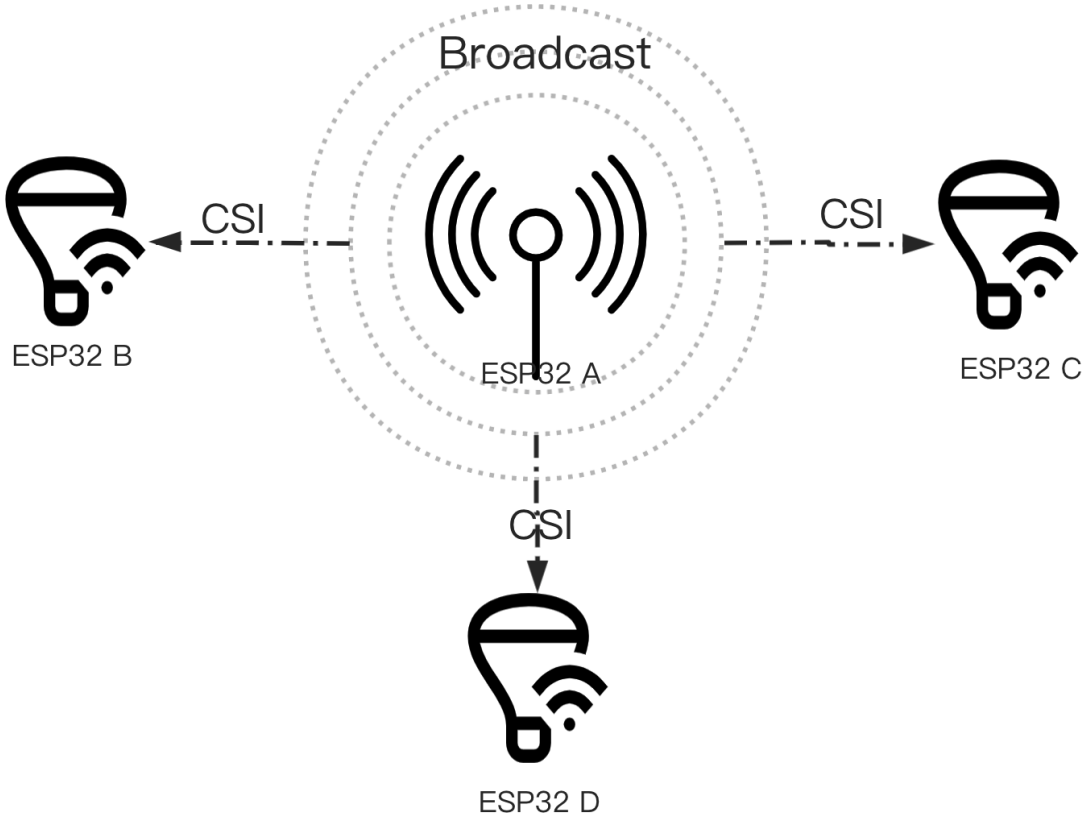


Figure 4.3: Get CSI data using a broadcasting ESP32

As we focus on reliability and accuracy, we choose the third method by making one ESP32 device a broadcaster and the other a receiver. We added an extra layer of security by specifying the Media Access Control (MAC) address of the receiving ESP32. As a result, if the receiving device is in the coverage area of the transmitting device, it gets CSI data automatically from the transmitting device. No overhead is required here.

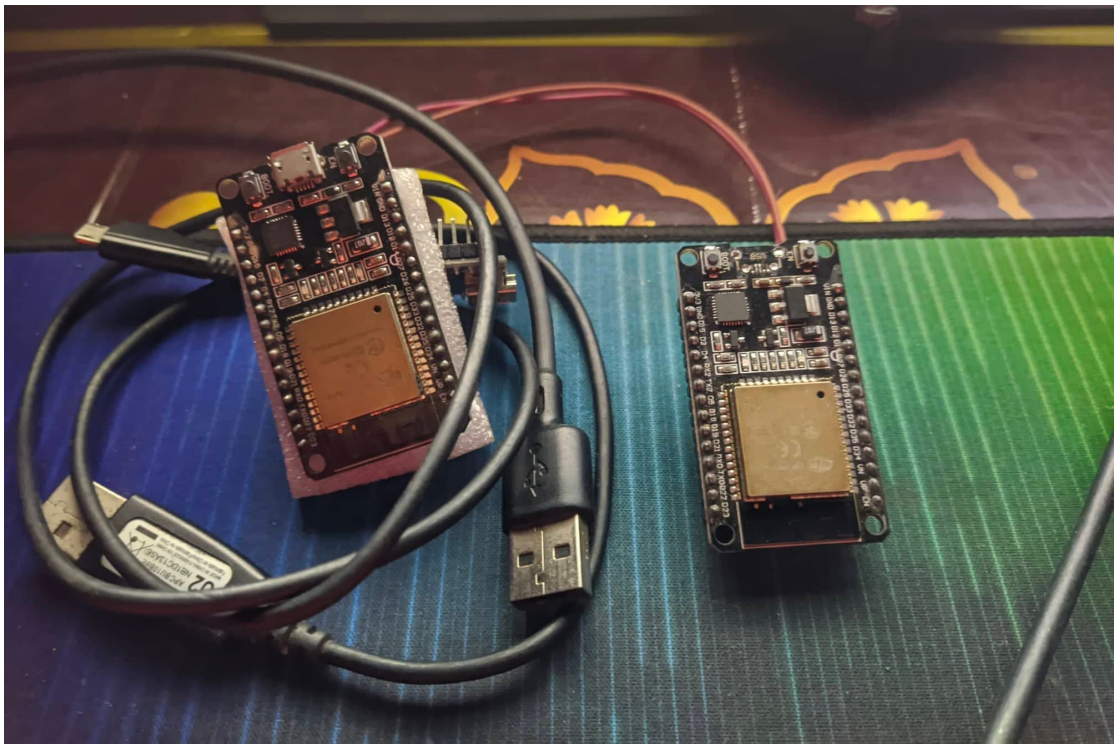


Figure 4.4: ESP32 devices used in the project

Each ESP32-WROOM-32E module has one built-in PCB antenna that can transmit or receive data. It is also possible to connect any external antenna with 50 Ω resistance. In this experiment, we restricted our study to the built-in PCB antenna. We used two ESP32 modules placed approximately 3.5 meters apart. One of them is used as a transmitting device connected to any power source, and the other as a receiving device connected to a computer to process the data and predict the activity. The space between the devices is kept empty for ensuring the Line-of-Sight (LoS). The subjects are instructed to do the activities in the 3.5 meters × 3.5 meter area between the transmitting and receiving devices. Because of the movement of the subject, the transmitting packets face multipath fading, scattering, reflection, and power loss. The Channel State Information (CSI) of each packet can be analyzed to find patterns between the transmitting packets using machine learning algorithms and thus recognize the activity performed by the subject.

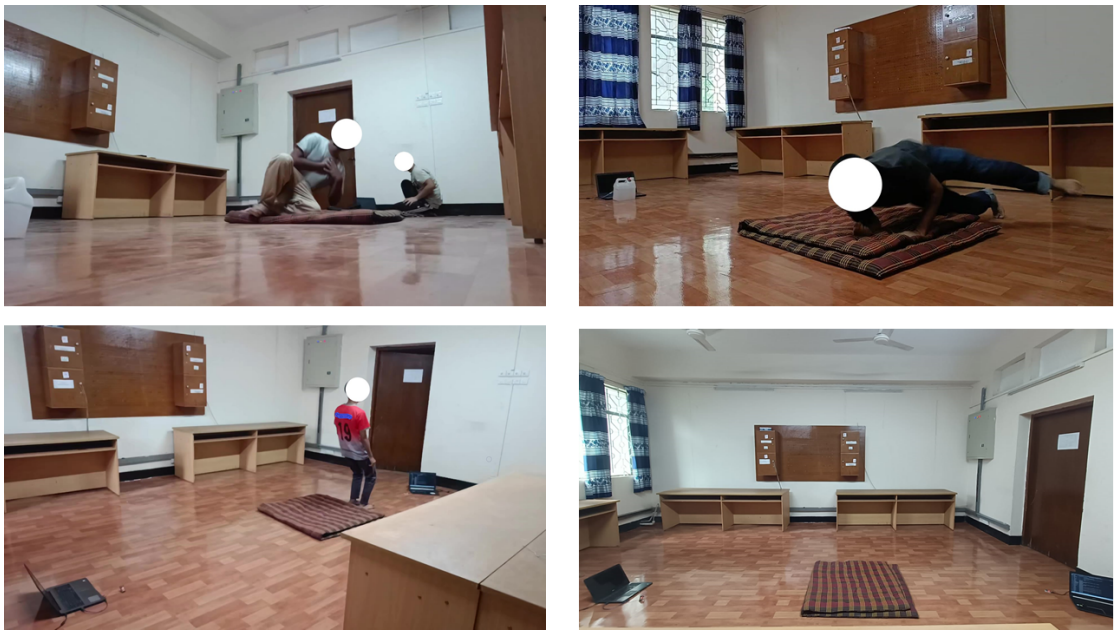


Figure 4.5: Data Collection Setup and Data Collection Example

ESP32-WROOM-32E supports Wi-Fi 802.11 b/g/n standards. For our experiment, we used 802.11n standard that provides support for OFDM, MIMO, frame aggregation, and higher data rate. But ESP32 does not support 5 GHz band and only works with 2.4 GHz band. The ESP32 devices we used in this experiment are configured to send data according to the following specifications:

Table 4.1: Customized signal specification of ESP32

Specification	Value
Standard	IEEE 802.11n
Band	2.4 GHz
Channel	20 MHz
MCS index	0
Guard interval	400 ns
Data rate	7.2 Mbit/s
Modulation	BPSK
Sampling rate	100 Hz
Coding rate	1/2
Spatial streams	1

4.2 Dataset Description

The accuracy and effectiveness of any data-driven study depend much on a well-prepared dataset. But there are only a few open datasets available for activity recognition using ESP32 CSI data. But these datasets do not have enough data or provide the activities we need for this system. Hence, we prepared our dataset using the hardware setup stated in the previous section. Wi-Fi CSI data is very sensitive to the outside environment which makes it very hard to collect data in the wild. Even rooms with different arrangements may affect the CSI data differently which may create a problem if a huge amount of data is not taken. For this project, we selected a neat and spacious room with minimum furniture and other things to collect the data. Two ESP32 devices are placed 3.5 meters apart and the activities are performed by different subjects in the area between the two devices. There are a total of 5 activities performed by 13 individual subjects. Each activity segment is recorded for a fixed time window of 4 seconds. This time window is chosen empirically by the type and complexity of the activities. A total of 966 samples of such segments are recorded on different calendar days.

4.2.1 Challenges

ESP32 sends Ping packets with CSI data at a rate of 100 Hz. But due to interference, unavailability of Line of Sight, and other issues, some packets are lost. As a result, though ideally each data segment of 4 seconds should have a total of $4 \times 100 = 400$ packets, most of the segments had packets between 300 to 340 due to packet loss.

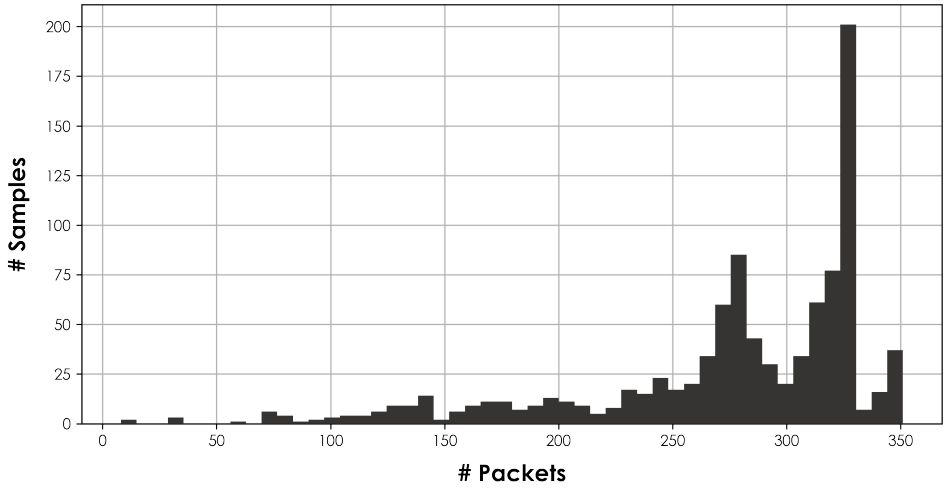


Figure 4.6: Distribution of number of packets in all the collected samples

The problem here is, some of the data samples had extremely lower number of packets that were not usable in the system. So, we discarded 68 data samples with less than 150 packets. Finally, the remaining 898 data samples were used in the next steps.

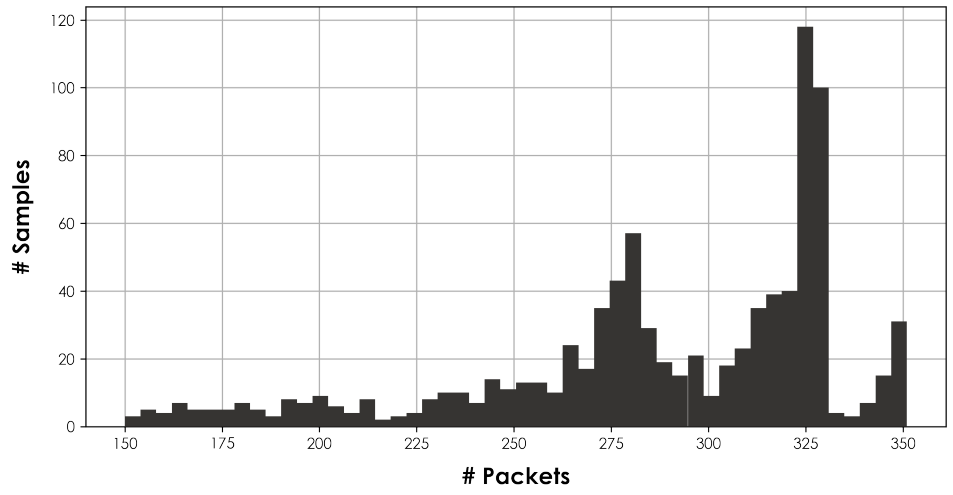


Figure 4.7: Distribution of number of packets in the selected samples

4.2.2 Activities

There are a total of 5 activities in this dataset: Fall, Stand, Walk, Empty Room and Presence. Fall, stand and walk activities are performed by one subject for each sample. For empty room activity, no subject was present in the room. For presence, a few subjects were present in the room doing various daily activities including gossiping, taking rest, writing, etc. Every activity excluding the empty room was done in a different direction and fashion to preserve generality. The number of samples for each of the activity are not the same and is shown below in 4.2.

Table 4.2: Number of samples by activity

Activity	Number of samples
Fall	224
Stand	100
Walk	133
Empty room	299
Presence	142

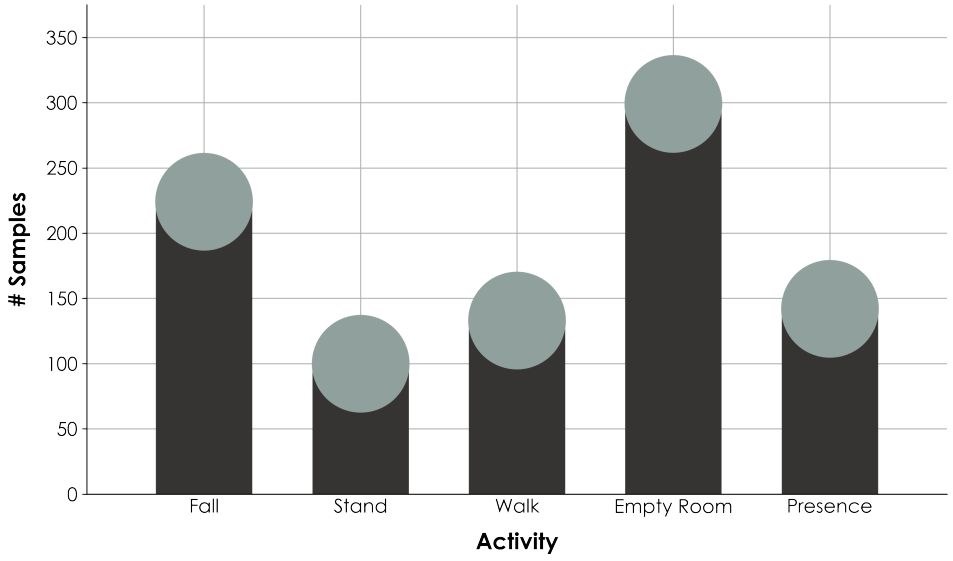


Figure 4.8: Number of samples by activity

Packet size distribution is also different for different types of activities. As sub-carriers experience a different level of scattering, reflection, or delay for different types and speeds of movement, packet loss will also be different. For example, stand activity involves a very small amount of movement, but there should be a much higher level of movement in fall activity. This difference in movement causes a difference in packet loss and segment size as illustrated in 4.9.

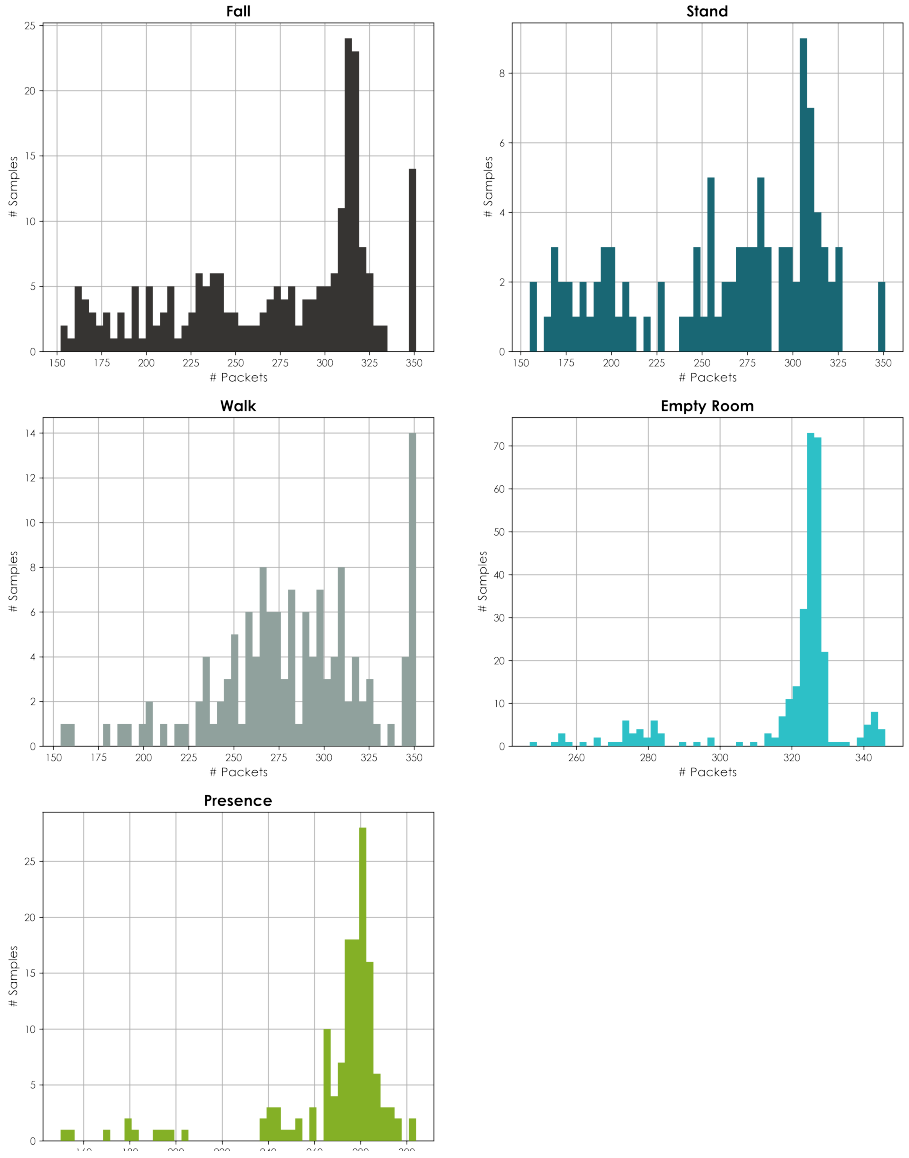


Figure 4.9: Packet counts of segments for different activities

4.2.3 Subjects

The number of subjects is an important aspect of any dataset. A higher number of subjects increase the generalized performance of any system. 13 subjects voluntarily performed different activities for this dataset. Most of the subjects have

performed three different activities and some of them performed two or four activities out of five. In total, the subjects performed 20 to 45 segments of different activities.

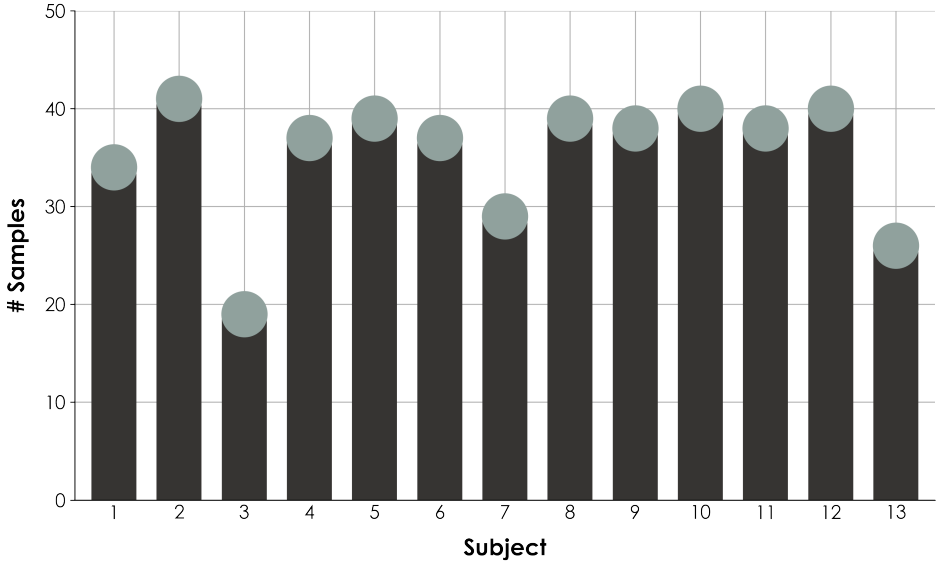


Figure 4.10: Number of samples by subject

4.2.4 Dataset Summary

Different aspects of the dataset is summarized in 4.3.

Table 4.3: Dataset summary

Specification	Value
Device used	ESP32-WROOM-32E
Signals used	CSI, RSSI
Activities	5
Subjects	13
Subject age range	18-25
Total samples	966
Selected samples	898
Sampling rate	100 Hz
Each sample time window	4 seconds
Mean packet count	287
Room temperature	25 ° C

4.3 Data Preprocessing Pipeline

The data we collected includes the raw signal information and packets. In this section, we propose a robust and complex preprocessing pipeline to preprocess the raw data and make the raw data used for the system. At first, we need to extract the CSI data from the raw signal and isolate the phase and amplitude information from it. As CSI data is inherently sensitive to different environmental parameters, some cleaning process needs to be applied such as calibration, denoising, and dimensionality reduction. There are various denoising algorithms to choose from. We compare their performances and choose the best one. After preconditioning the signals, we feed the data to the feature extraction module that extracts different features from the cleaned data. Not all the features are important for the outcome. So, we have to use different mathematical models to find the most relevant features. A normalization technique is employed before handing the data over to a competent machine learning model. Lastly, we tune different hyperparameters of the model to increase the performance of the model on the dataset.

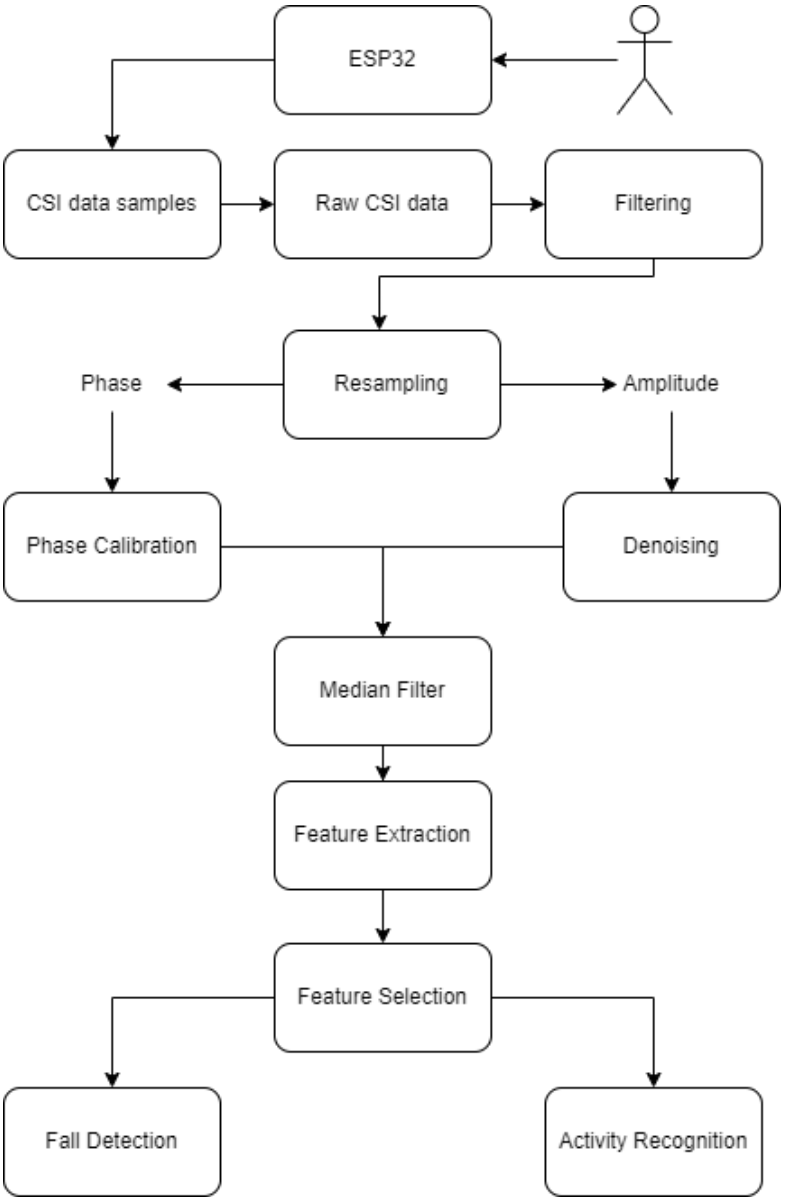


Figure 4.11: Complete Pipeline of The System

4.3.1 CSI Data Extraction

CSI essentially enables us to comprehend what transpires on the channel between the transmitter and receiver. CSI is calculated by examining how the preamble with known content is changed during transmission. Thus, we have a collection of complex numbers in the form $a_n \exp j\theta_n$, where a_n is the amplitude and θ_n is the phase. We need to extract this amplitude and phase from the raw CSI data. A raw CSI packet comprises the real and imaginary part of the channel state

and is transmitted separately for each subcarrier. But not all subcarriers carry useful information. Different subcarriers respond differently to human activity; for example, certain subcarriers are quite sensitive to motion and exhibit obvious fluctuations. It is preferable to use only the CSI information from these sensitive subcarriers. The computing complexity of the system is also increased by using all of the data from all of the subcarriers. So, we removed some of the irrelevant subcarriers by examining the acquired CSI sequence, and skillfully isolate the signal segments mostly corresponding to human activity. Then we transformed the real and imaginary parts of the selected subcarriers into polar form to get the required amplitude and phase information.

4.3.2 Time Series Representation

The extracted CSI data ideally should be a time series having a constant time difference between two samples. But in practice, the sampling rate was not constant and there was a considerable amount of packet loss which led to non-uniform data. To apply different time and frequency domain analysis, this data need to convert to a time series representation. That's why we resampled the data at a constant 100 Hz sampling rate. This process involves resulting in some missing values which are linearly interpolated to generate an equivalent waveform.

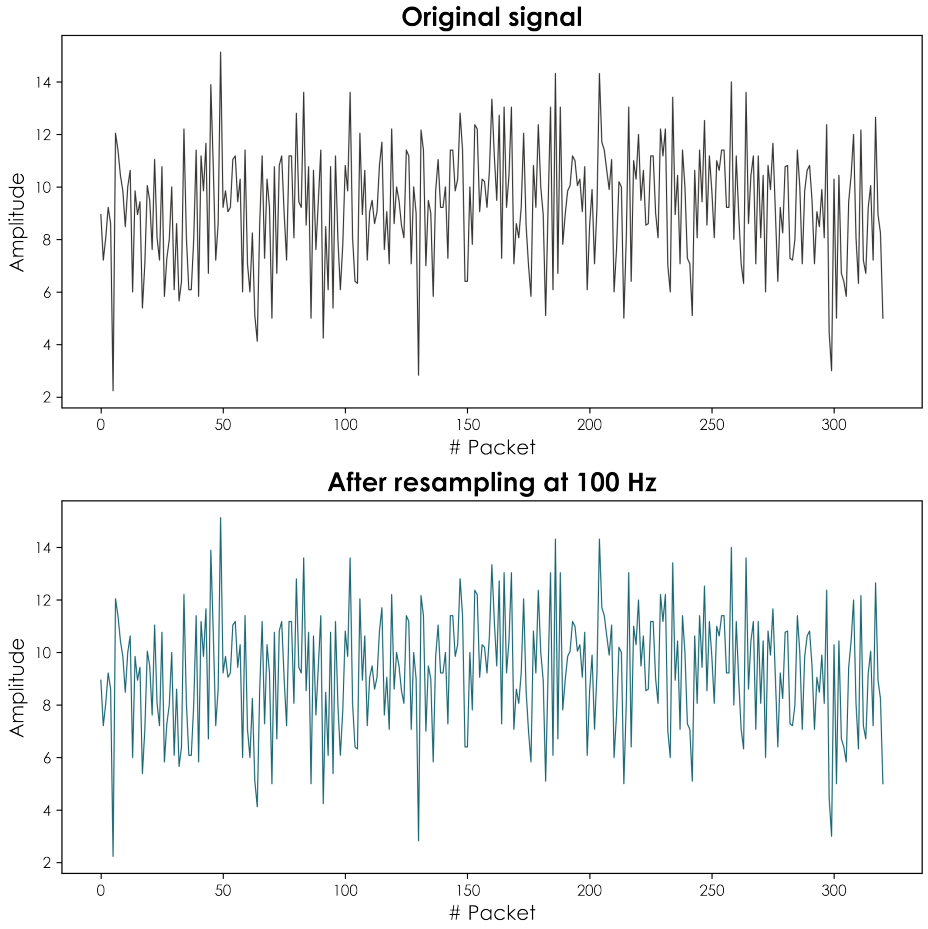


Figure 4.12: Resampling for time series representation

4.3.3 Phase Signal Analysis

The isolated phase information of i^{th} subcarrier can be expressed by the following equation:

$$\hat{\phi}_i = \phi_i - 2\pi \frac{s_i}{N} \tau + \beta + Z \tag{4.1}$$

Here, ϕ_i is the actual phase which is deteriorating by the time offset at receiver τ, unknown time offset β, and measurement error Z. s_i is the subcarrier index and N signifies the Fast Fourier Transform (FFT) size which is 64 for IEEE 802.11n. Due to this distortion, the phase must be calibrated the restore the actual phase

as much as possible. As shown by [36], the time offsets, τ, and β can be removed by considering the phase across the frequency band given by the equation:

$$\hat{\phi}_i = \hat{\phi}_i - as_i - b = \hat{\phi}_i - \frac{\phi_n - \phi_1}{s_n - s_1} s_i - \frac{1}{n} \sum_{j=1}^{n} \phi_j \tag{4.2}$$

Here, a and b are intermediate variables. But in this process, the true phase is folded due to the recurrence characteristic of the phase. This problem can be solved by compensating multiple 2π's by judging whether the measured phase change between the adjacent subcarriers is greater than the given thresholds [37].

4.3.4 Denoising Amplitude Signal

The amplitude of the CSI data is very sensitive to internal and external noises. Possible noise sources are scattering, reflection, delay, fading, and power line losses. The signal can be denoised by using various techniques including different filters or transformations.

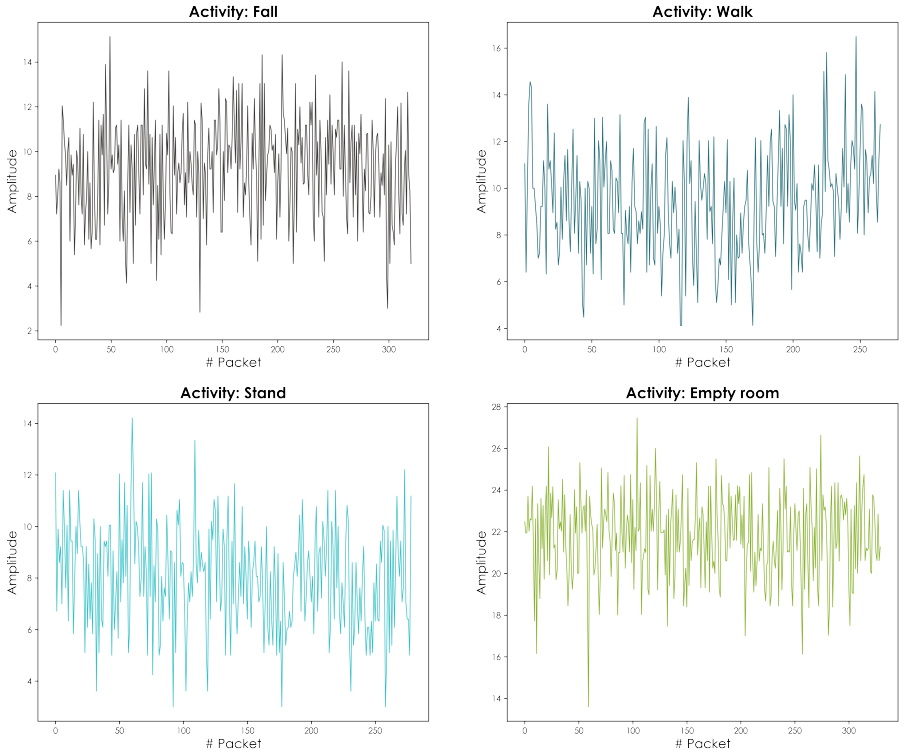


Figure 4.13: Raw CSI amplitude signal for different activities

4.3.4.1 Low Pass Filter (LPF)

A low-pass filter is a filter that attenuates the higher frequency portion of a signal than a chosen cutoff frequency. The filter's precise frequency response is determined by the filter's design. There are many types of low pass filters, but in this project, we will focus on a specific type named Butterworth filter [38]. Butterworth filter is a signal processing filter made to have a frequency response that is as flat as possible in the passband. Butterworth filter has a slower roll-off and thus will require a higher order to implement a particular stopband specification, but it has a more linear phase response in the passband than most others. Low pass butterworth filters are greatly used for noise removal from various signals. Generally, a noisy wave contains two different portions: high-frequency noise and low-frequency true signal. This also fits for CSI amplitude signal. Thus the high-frequency noise mixed in the signal can be removed by using a Butterworth low

pass filter. But a problem with this method arises when the signal and noise frequency are not separated by a large margin. As butterworth filter has a slower roll-off, it tends to attenuate some of the original signals too.

4.3.4.2 Fast Fourier Transform (FFT)

A fast Fourier transform (FFT) algorithm calculates a sequence's discrete Fourier transform (DFT) or its inverse (IDFT). Through the process of Fourier analysis, a signal is transformed from its original domain, which is frequently time or space, to a representation in the frequency domain, and vice versa. The DFT is produced by breaking down a series of numbers into components of various frequencies. [1] Although computing it straight from the specification is generally too time-consuming to be helpful, this technique has many applications. Such changes are quickly computed by an FFT by factorizing the DFT matrix into a product of sparse (mostly zero) elements.[39] This successfully reduces the complexity from $O(N^2)$ to $O(Nlog(N))$ where N is the number of samples.

4.3.4.3 Short-Time Fourier Transform (STFP)

A Fourier-related transform known as the Short-time Fourier transform (STFT) is used to ascertain the sinusoidal frequency and phase content of local parts of a signal as they change over time. To compute STFTs, it is necessary to split a longer temporal signal into equal-length shorter segments. Each shorter segment is then subjected to a separate Fourier transform computation. This makes each shorter segment's Fourier spectrum visible. The shifting spectra are then typically plotted as a function of time using a technique called a spectrogram or waterfall plot, which is frequently applied in Software Defined Radio (SDR)-based spectrum displays. On desktop PCs, Fast Fourier Transforms (FFTs) with 2^{24} points are frequently used for full bandwidth displays that span the whole SDR range.[40]

4.3.4.4 Wavelet Transform (WT)

In Fourier transform (FT), we represent a function using a series of sine and cosine waves, which is an excellent approach to understanding the frequencies present in the signal. However, it has a significant drawback. FT only contains the frequency information, but the spatial or temporal information is completely lost. It will not be possible to tell where the frequency is low or high in the space or time domain, or where frequency shifts are taking place. To overcome this problem, we do Wavelet Transform (WT). In WT, we represent a function using a certain orthonormal series produced by a wavelet. A wavelet is a waveform of effectively limited duration that has an average value of zero and nonzero norm. If a function ϕ can provide a Hilbert basis or a full orthonormal system, for the Hilbert space of square-integrable functions, then ϕ is referred to as an orthonormal wavelet. The fundamental principle of wavelet transform is that they should only be able to modify the length of time, not shape. This is impacted by selecting appropriate base functions that support this. Changes to the time extension should match up with the analysis frequency of the basis function.

4.3.4.5 Discrete Wavelet Transform (DWT)

A discrete wavelet transform (DWT) [41] decomposes an input signal into several sets, each set consisting of a time series of coefficients that describe the signal's temporal evolution in the associated frequency band. In this process, the wavelets are sampled in discrete steps. A key advantage it has over Fourier transforms is temporal resolution: it can show both frequency and location information (location in time). The DWT refers not just to a single transform, but rather to a set of transforms, each with a different set of wavelet basis functions. Two of the most common are the Haar wavelets and the Daubechies set of wavelets. Unlike the Continuous Wavelet Transform (CWT), it uses a finite set of wavelets, i.e., defined at a particular set of scales and locations. But the main idea remains the same; we multiply the signal with a particular wavelet with a particular scaling and then shift it over the whole signal to integrate. In DWT, we repeat the procedure after changing the scale. In these situations, scaling adjustments are made discretely.

The DWT can be defined by the following equation:

$$T_{m,n} = \int_{-\infty}^{\infty} x(t)\psi_{m,n}(t)dt \tag{4.3}$$

where ψ is the wavelet function. The components of the signal can be assembled back into the original signal without loss of information using a reconstruction algorithm known as the Inverse Discrete Wavelet Transform (IDWT). In IDWT, the DWT coefficients are first upsampled by inserting zeros between each coefficient, essentially doubling the length of each (the approximation and the detail coefficients are handled separately). The reconstruction scaling filter for approximation coefficients and the reconstruction wavelet filter for detail coefficients are then convolved with these. To get the original signal, these results are then put together.

4.3.4.6 Comparison of DWT and LPF

For our dataset, we employed two different types of denoising methods. One is a 4^{th} order Butterworth low pass filter with a cut-off frequency of 10 Hz. The other is a second-level Discrete Wavelet Transform using symlet10 wavelet.

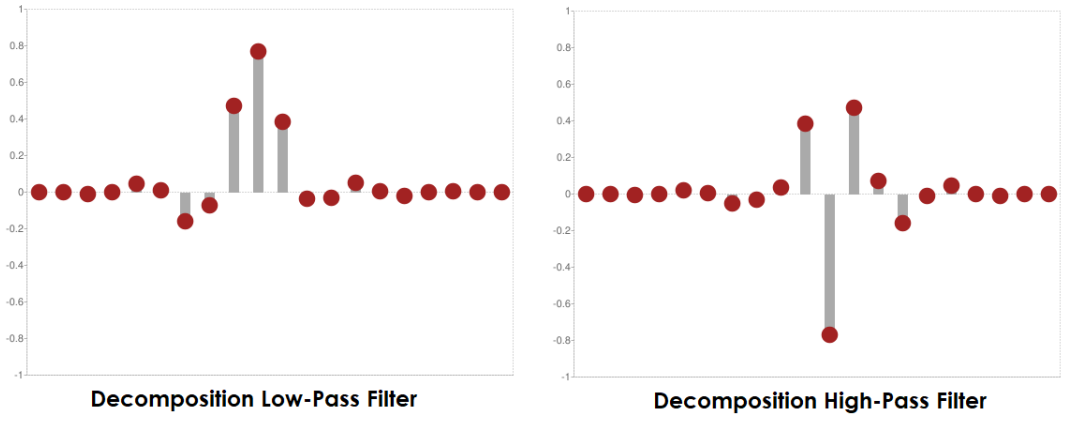


Figure 4.14: Symlet10 wavelet

We used both denoisers and compared how they work on different CSI amplitude waves. By careful comparison, we found the DWT denoising method performed better than low pass filter. So, we chose DWT denoising for this system.

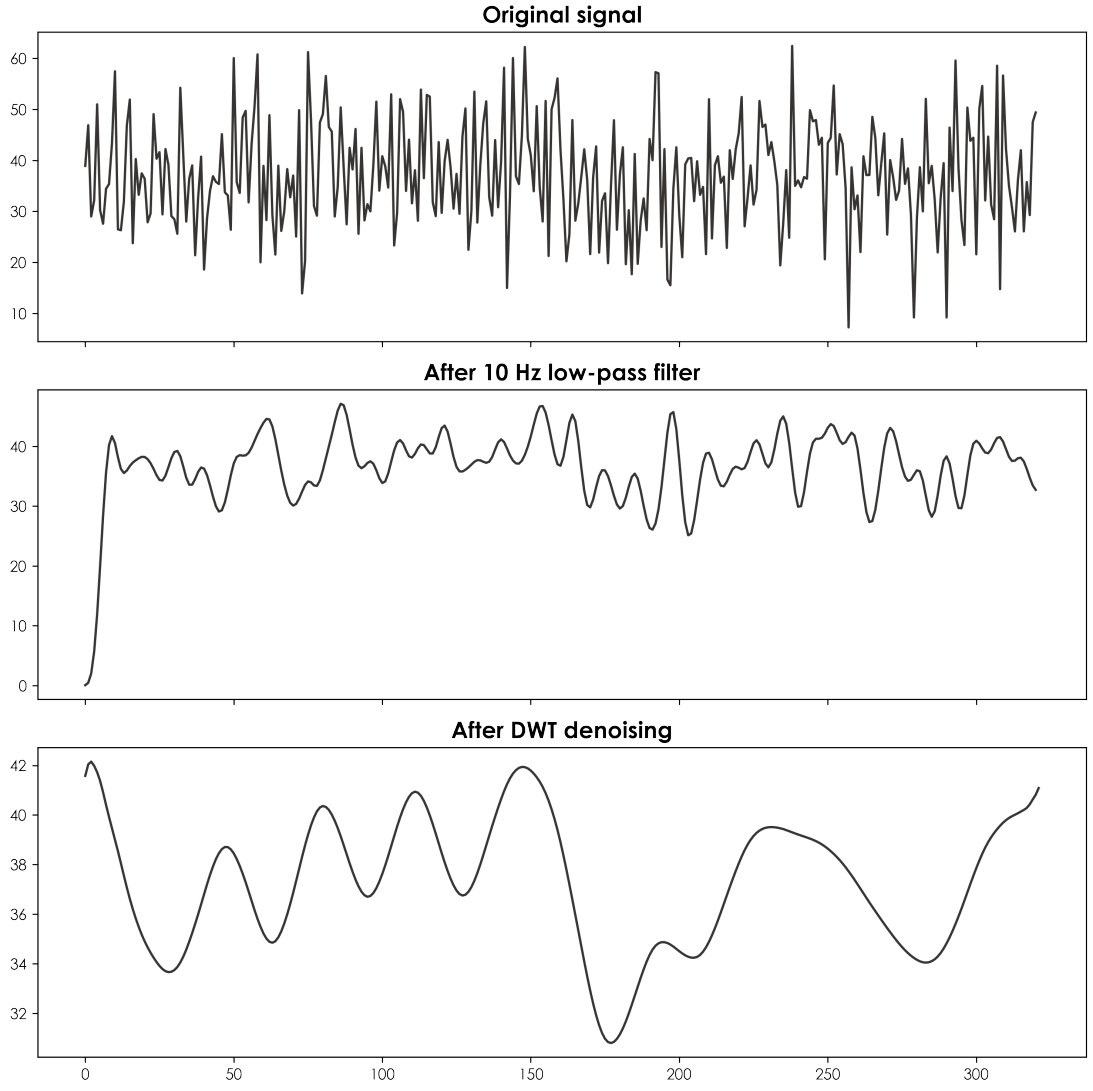


Figure 4.15: Comparison between low pass filter and DWT

4.4 Feature Selection

After conditioning and denoising the signals, we extracted some statistical features including min, max, median, and standard deviation from the phase and amplitude CSI and RSSI signals. But the initial feature size was 716 which is very large

considering the sample size which is only 898. So, we needed to prune the feature set to remove non-correlated features. There are different ways to measure the correlation or importance of the features with the output variable.

4.4.1 Chi-Square Test [1]

One technique to demonstrate a connection between two categorical variables is via a chi-square statistic. The chi-squared statistic is a single figure that indicates the degree to which the counts you saw deviate from the counts you would anticipate if there were no association at all in the population.

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{4.4}$$

4.4.2 Pearson's Correlation Coefficient (PCC) [2]

Pearson's correlation coefficient is the test statistics that assess the statistical association, or relationship, between two continuous variables. Because it is based on the method of covariance, it is regarded as the best method for determining the relationship between variables of interest. It provides details on the size of the association or correlation as well as the relationship direction. One problem with PCC is that it is not able to tell the difference between dependent variables and independent variables.

4.4.3 Decision Tree (DT) based feature selection [3]

Decision tree building algorithm selects the splits locally, i.e. concerning the splits selected in earlier stages, so that the features occurring in the decision tree, are complementary. Thus, Decision Tree based models provide feature importance metrics that can be utilized to select the most important features.

We used a combination of PCC and decision tree-based selection methods to select the most important 243 features for our proposed method.

4.5 Training Description

Every supervised machine learning system has three phases:

1. Training phase: We train the data for the known labels.
2. Testing phase: We evaluate the performance of the trained model keeping the labels away from the model.
3. Application phase: We apply our model for real-life unknown data.

Our proposed preprocessing pipeline is independent of the training phase. So, it gives us the advantage to use any machine learning model for our preprocessed data even for different learning tasks. In this project, we have two different learning objectives. One is fall detection and the other one is a generalized human activity recognition.

4.5.1 Fall Detection

For fall detection, we classified the walk and stand activities as non-fall activity, kept the fall activity as is and did not use the empty room and presence activities. Then, we trained binary classification algorithms on this data.

4.5.2 Human Activity Recognition

In this objective, we utilized the whole dataset with all the activities, i.e., fall, stand, walk, empty room and presence. So, we have five labels in this task and used this data in different multi-class classification algorithms.

In both cases, we compared the behavior and performance of these algorithms and tuned them to get the best performance on the test set. The performance statistics of these models are depicted in the next chapter.

CHAPTER 5

RESULT AND ANALYSIS

In this chapter, we aim to discuss the results of our implementation and analyze its performance in various scenarios and test setups.

5.1 Evaluation Metrics

Classification Accuracy: Classification Accuracy is what we usually mean by the term accuracy. It is the highest used metric for classification tasks which works best when the number of samples belonging to each class is nearly equal. But, it does not convey any useful information when the dataset is imbalanced. For example, if any dataset has two classes: class A, and B, and 98% of the dataset belongs to class A, blindly predicting each of the samples as class A will give an accuracy of 98% which is misleading. Hence, this metric is only used in balanced datasets.

$$Accuracy = \frac{\#\ correct\ predictions}{\#\ all\ samples} \tag{5.1}$$

Precision:

precision is a measure of result relevancy, it calculates how many positive predictions are correct. It focuses on the correctness of positive detection which can be a good metric for cases where a false positive can be more troublesome than a false negative.

$$Precision = \frac{TP}{FP + TP}$$

Recall: Unlike precision, Recall focuses on correctly identifying True Positives out of all the positive samples. It can be referred to it as Sensitivity or True Positive Rate. It is a good metric where false negatives are more troublesome, such as in disease detection.

$$Recall = \frac{TP}{FP + FN}$$

F1 Score: It is a metric combination of both the Precision and Recall scores. It lets us do a tradeoff between Precision and Recall. A good F1 score is the indication of both good Recall and good Precision. That is why F1 score is considered one of the most important metrics to evaluate the performance of a classification system.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Area Under Curve (AUC): It is also known as AUC-ROC which signifies the area under the Receiver Operating Characteristic (ROC) curve. The ROC curve is an evaluation metric initially proposed for binary classification problems. In essence, it separates the "signal" from the "noise" by plotting the TPR against the FPR at different threshold values. The higher the AUC, the model is considered to perform better. In general, the ROC is for many different levels of thresholds and thus it has many F score values. F1 score is applicable for any particular point on the ROC curve.

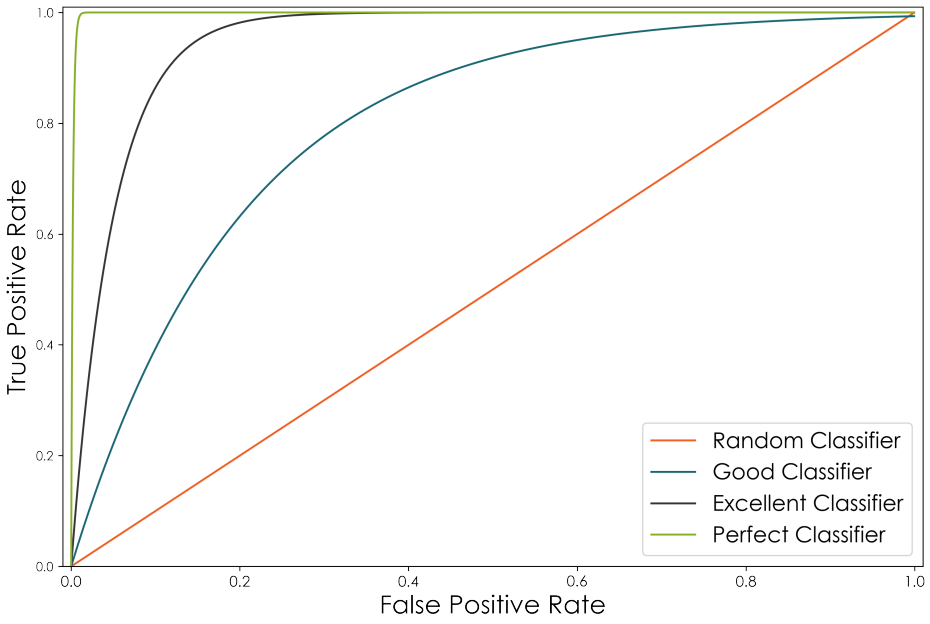


Figure 5.1: Area Under Curve of Receiver Operating Characteristic (AUC-ROC)

5.2 Testing Setups

The prerequisite for building a good machine learning model is to validate its performance of it against unknown data. If not, the model may overfit on the given dataset and perform worse in the actual application where we test the model on real-life unknown data. To do so, we generally split the dataset into some sets which are typically known as training set, testing set and so on. The model should not only work well on the training data but also give an accurate prediction on an unknown dataset. To evaluate how well the model is performing on unknown dataset, we employ different splitting techniques for better generalization.

5.2.1 Train-test Split

This is the most common type of the data splitting methods. In this method, we generally divide the dataset into two mutually exclusive sets. One is training set which is used to train the model and includes all the known labels. Another is testing set in which we hide the labels from the model and evaluate how does the model perform on unknown data. The split ratio is a term that defines how much data are in the training set and testing set. A split ratio of 75% means 75% of the total data are kept in training set and the rest 25% data are in testing set. The split ratio is typically chosen between 60% to 80%, but a split ratio outside this range may be picked depending on the dataset. Generally, the split ratio keeps increasing with the dataset size.

Figure 5.2: Train-test split

5.2.2 Train-validation-test Split

Sometimes the dataset is divided into three sets instead of two, adding another set known as validation set. This splitting technique is called train-validation-test split. In such a case, the data is first trained on training set and validated on the validation set to evaluate the performance. The model that has the best performance on the validation set is chosen and is tested on the testing set to obtain the actual performance of the model.

Figure 5.3: Train-validation-test split

5.2.3 K-fold Cross Validation

Cross-validation is a statistical splitting method used to estimate the skill of machine learning models. The procedure has a single parameter called k which signifies how many splits will be made. When a particular number for k is selected, it may be substituted for k in the model's reference, such as when k=10 is used to refer to 10-fold cross-validation. The general procedure of this method is given below:

1. Randomly shuffle the dataset.
2. Create k groups from the dataset.
3. For every distinct group:
   (a) The group should be used as a holdout or test data set.
   (b) Use the remaining groupings as the training data set.
   (c) Fit the model to the training data, then assess it against the test data.
4. Repeat the procedure for k times.
5. Using a model evaluation metric, summarize the model's skill.

By following this procedure k-fold cross validation removes the splitting bias which is present in the previous techniques.

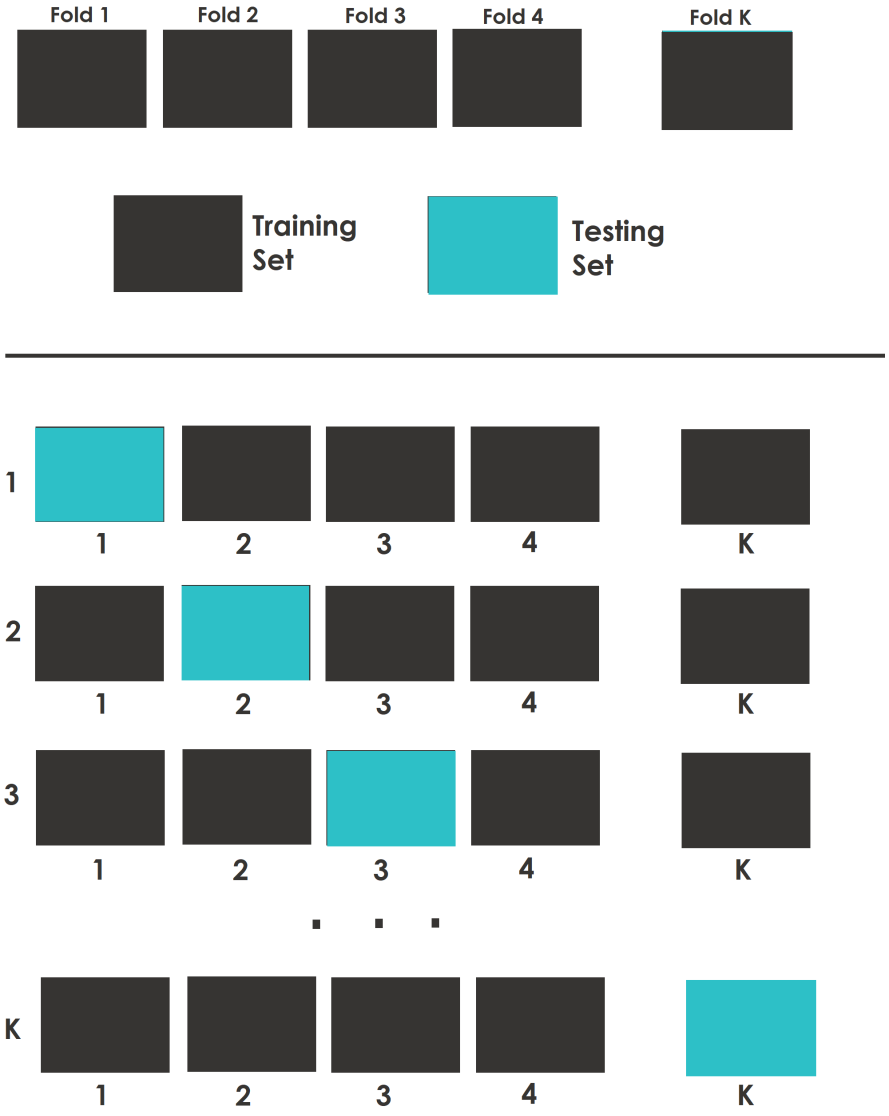


Figure 5.4: K-fold cross validation

5.2.4 Leave-One-Out Cross Validation (LOOCV)

LOOCV is specially used in datasets where data are related to human subjects like in this dataset of human activity recognition. Different subjects carry out different activities in unique ways. Because of this variation, it is harder to predict a new subject's activity. To address this problem, we split the data according to each subject. That means each split contains only one subject's data. Then we perform the cross validation to assess the performance of the model on a new subject.

5.3 Results

5.3.1 Fall Detection

For the fall detection task, we had two classes: fall and non-fall. We experimented with different models and conducted a thorough hyperparameter tuning. Then we evaluated the performance of the models using a 75% train-test split and a 10-fold cross validation.

Table 5.1: Result of fall detection

Model	75% Train-test split		10-fold CV	
	Accuracy	F1 score	Accuracy	F1 score
Logistic Regression	0.657	0.632	0.762	0.772
Support Vector Classifier	0.938	0.929	0.921	0.922
K Nearest Neighbors	0.923	0.909	0.921	0.924
Random Forest	0.966	0.962	0.967	0.967
Extra Trees	0.980	0.978	0.985	0.985
XGBoost	0.953	0.947	0.954	0.959

Using the 5.1, we can find the best model is the Extra Trees Classifier which obtained 98.5% accuracy and F1 score in 10-fold CV and 98% and 97.8% accuracy and F1 score in 75% train-test split. We plotted the confusion matrix of this model for 75% train-test split.

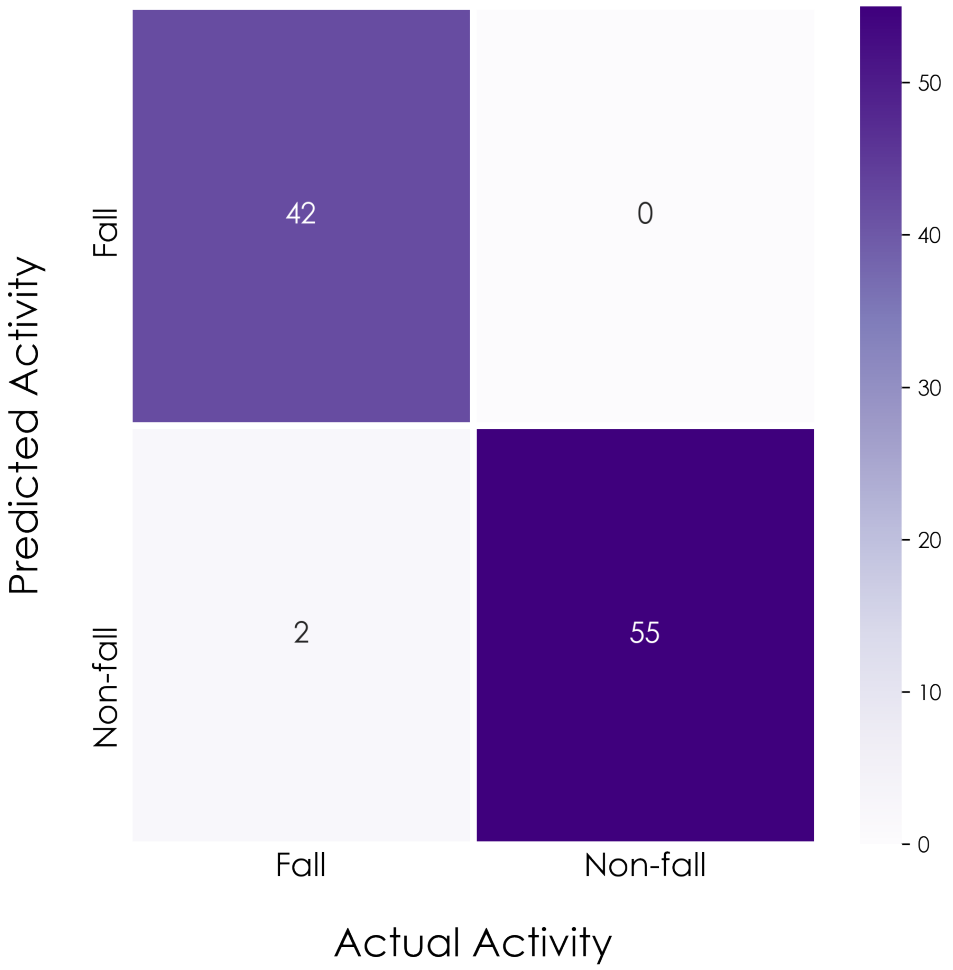


Figure 5.5: Confusion matrix for fall detection

The results of fall detection is summarized in 5.6

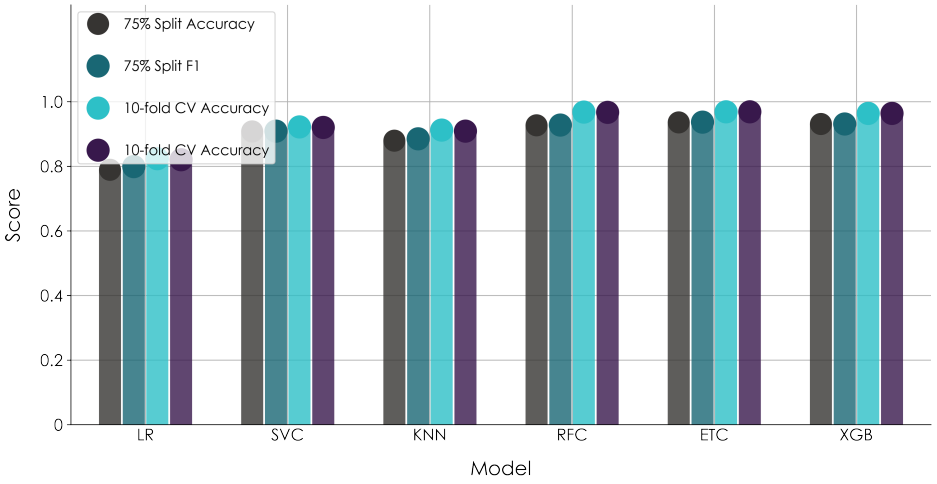


Figure 5.6: Fall detection results

5.3.2 Human Activity Recognition

We have five classes of human activity: fall, walk, stand, empty room and presence. We took a similar approach for human activity recognition as fall detection mentioned in the previous sub-section. The results are depicted in table 5.2.

Table 5.2: Result of human activity recognition

Model	75% Train-test split		10-fold CV	
	Accuracy	F1 score	Accuracy	F1 score
Logistic Regression	0.789	0.799	0.824	0.82
Support Vector Classifier	0.908	0.909	0.922	0.92
K Nearest Neighbors	0.879	0.885	0.912	0.909
Random Forest	0.927	0.928	0.968	0.967
Extra Trees	0.936	0.937	0.969	0.969
XGBoost	0.931	0.931	0.964	0.964

Like fall detection, the Extra Trees Classifier gives better results in this task too. The obtained confusion matrix is shown in 5.7.

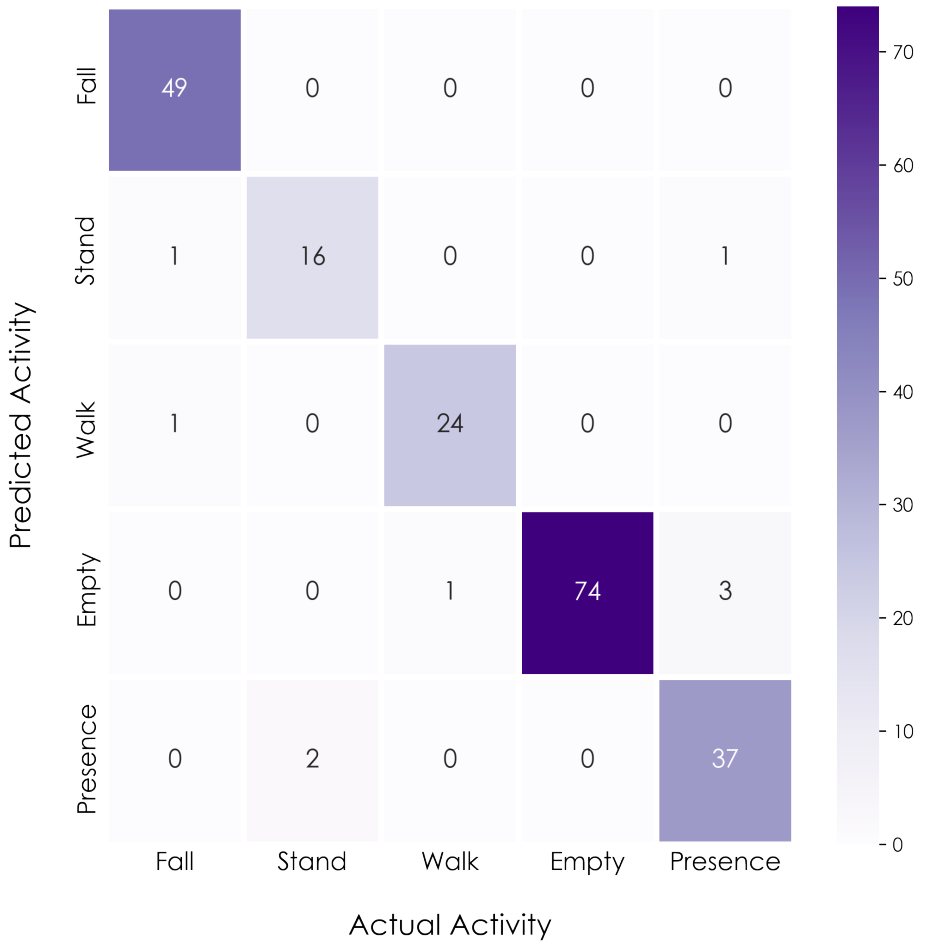


Figure 5.7: Confusion matrix for human activity recognition

The results of human activity recognition is summarized in 5.8

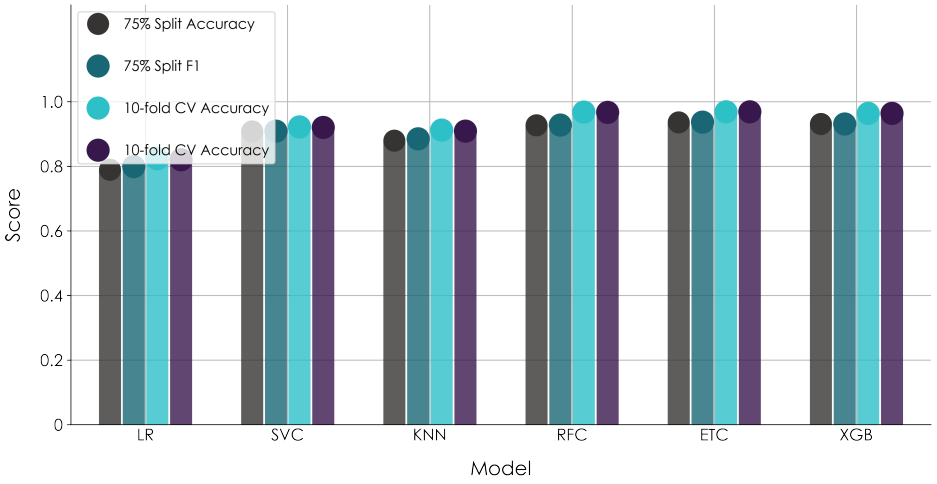


Figure 5.8: Human activity recognition results

5.4 Analysis of in Terms of Speed

We also conducted an execution time analysis to show that our proposed system is able to run in real-time. The analysis was done on a total of 400 data samples where 300 of them were in the training set and the remaining 100 data samples were in the test set. So, the training data collection time was 1200 seconds and the test data collection time was 400 seconds. We used a lower mid-level laptop with core i5 7200U processor clocked at 2.5 GHz and 8GB RAM to run the time analysis. By looking at 5.3 we can see that, to process the inference data of 400 seconds, the best model Extra trees took only 5 seconds to predict in the multiclass classification problem and 4 seconds to predict in the binary classification problem. So, this system is capable of processing data from up to 80 devices simultaneously in real-time. Also, the fastest model tested was K Nearest Neighbors which took only 0.1 seconds to predict a data of 400 seconds. So, it can easily be concluded that our proposed system can run and infer in real-time for a number of devices simultaneously.

Table 5.3: Execution time of the models

Model	Binary Classification		Multi-class Classification	
	Training time (s)	Inference time (s)	Training time (s)	Inference time (s)
Logistic Regression	1.4	0.2	1.6	0.2
Support Vector Classifier	2	0.2	2	0.2
K Nearest Neighbors	0.3	0.1	0.4	0.1
Random Forest	15	3	30	6
Extra Trees	15	4	17	5
XGBoost	18	4	26	6

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

In this project, we have proposed a non-contact fall detection and human activity recognition system using embedded devices. The objective of our project was to overcome the drawbacks of the now popular systems that use wearable devices and/or computer vision. Using esp32 and its Wi-Fi capabilities, we have implemented a robust fall detection system that is also able to recognize different activities, and is able to detect if a room is empty. We have analyzed CSI data collected from two esp32 microcontrollers and gone through an elaborate process of cleaning, preprocessing, feature extraction, feature selection and classification using various machine learning models. Through this process, we were able to perform accurate human activity recognition and fall detection. The use of embedded devices made the system open to heavy modifications and lowered its deployment cost, making it highly flexible and affordable.

6.1 Discussion

In this project, we conducted two different tasks and achieved great accuracy. For the first task, fall detection, the dataset is quite balanced. But for the human activity recognition task, the dataset was moderately unbalanced which led to lower accuracy. The confusion matrices in 5.5 and 5.7 also depict the same fact.

Although our system is able to recognize different human activities including falls as shown in Chapter 5, our primary goal for this project is fall detection. The reason behind this is that no CSI-based system is able to localize the activities, so the opportunity for real world application is limited for most activities. We can also see from the works of Li et el.[9] that accuracy of Wi-Fi CSI-based activity detection systems drops considerably when multiple users perform different activities at the same time. But in case of fall activity, these limitations are irrelevant as in an event of a fall, immediate attention is required regardless.

6.2 Future Scopes

Our proposed system is effective, yet has much room for improvement. The limitations we aim to overcome and the improvements we want to implement in the future are as follows:

1. The devices we have used to implement the system use PCB antennas which are not very effective. Using external omnidirectional antennas would increase the effectiveness of the antenna even more.

2. We were able to send and receive packets from one device to another only at relatively small distances and in LoS condition. The reason behind this is that the devices are low power consuming devices. Designing a device that is able to send more powerful signals will solve these problems.

3. Designing an enclosure that houses a power system would allow deploying the system easily in various locations.

4. We have only collected data in a controlled environment. Collecting and analyzing data from uncontrolled environments where a lot more moving objects are present would make the system more robust.

5. Implementing some sort of GUI or interfacing the system with other devices such as smart phones would make this system more user friendly.

6. Although we have been able to successfully recognize different activities, we are unable to localize the activity. This can be done by using multi-dimensional information such as ToF(Time of Flight), AoA(Angle of arrival) and Doppler shift[42].

BIBLIOGRAPHY

[1] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class svm," Journal of King Saud University - Computer and Information Sciences, vol. 29, no. 4, pp. 462–472, 10 2017.

[2] Y. Liu, Y. Mu, K. Chen, Y. Li, and J. Guo, "Daily activity feature selection in smart homes based on pearson correlation coefficient," Neural Processing Letters, vol. 51, pp. 1771–1787, 4 2020.

[3] V. Sugumaran, V. Muralidharan, and K. Ramachandran, "Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing," Mechanical Systems and Signal Processing, vol. 21, no. 2, pp. 930–942, 2 2007.

[4] "Life expectancy," 2020. [Online]. Available: https://data.worldbank.org/indicator/SP.POP.65UP.TO.ZS

[5] A. Harris, H. True, Z. Hu, J. Cho, N. Fell, and M. Sartipi, "Fall recognition using wearable technologies and machine learning algorithms," in 2016 IEEE International Conference on Big Data (Big Data), 2016, pp. 3974–3976.

[6] R. V. . M. Gutiérrez, J., "Comprehensive review of vision-based fall detection systems," 2021.

[7] N. Damodaran and J. Schäfer, "Device free human activity recognition using wifi channel state information," in 2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart

City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2019, pp. 1069–1074.

[8] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," SIGCOMM Comput. Commun. Rev., vol. 41, no. 1, p. 53, jan 2011. [Online]. Available: https://doi.org/10.1145/1925861.1925870

[9] H. Li, X. He, X. Chen, Y. Fang, and Q. Fang, "Wi-motion: A robust human activity recognition using wifi signals," IEEE Access, vol. 7, pp. 153 287–153 299, 2019.

[10] Y. Zeng, D. Wu, J. Xiong, J. Liu, Z. Liu, and D. Zhang, "Multisense: Enabling multi-person respiration sensing with commodity wifi," Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 4, no. 3, sep 2020. [Online]. Available: https://doi.org/10.1145/3411816

[11] X. Liu, J. Cao, S. Tang, and J. Wen, "Wi-sleep: Contactless sleep monitoring via wifi signals," in 2014 IEEE Real-Time Systems Symposium, 2014, pp. 346–355.

[12] Y. Hao, Z. Shi, and Y. Liu, "A wireless-vision dataset for privacy preserving human activity recognition," in 2020 Fourth International Conference on Multimedia Computing, Networking and Applications (MCNA), 2020, pp. 97–105.

[13] "Documentation for esp32," 2022. [Online]. Available: https://www.espressif.com/en/support/documents/technical-documents

[14] "Ieee standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012), pp. 1–3534, 2016.

[15] W. Kabir, "Orthogonal frequency division multiplexing (ofdm)," in 2008 China-Japan Joint Microwave Conference, 2008, pp. 178–184.

[16] "Learn about multiple-input multiple-output," 2021. [Online]. Available: https://www.intel.com/content/www/us/en/support/articles/000005714/wireless/legacy-intel-wireless-products.html

[17] R. Corvaja and A. García Armada, "Effect of multipath and antenna diversity in mimo-ofdm systems with imperfect channel estimation and phase noise compensation," Physical Communication, vol. 1, no. 4, pp. 288–297, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874490709000020

[18] X. Wang, L. Gao, and S. Mao, "Csi phase fingerprinting for indoor localization with a deep learning approach," IEEE Internet of Things Journal, vol. 3, pp. 1113–1123, 12 2016.

[19] Çelik, "A research on machine learning methods and its applications," 09 2018.

[20] ——, "A research on machine learning methods and its applications," Journal of Educational Technology and Online Learning, vol. 1, no. 3, pp. 25 – 40, 2018.

[21] C. Türkmenoğlu and A. Tantuğ, "Sentiment analysis in turkish media," 06 2014.

[22] X. Zhu, "1 contents," 2007.

[23] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," CoRR, vol. cs.AI/9605103, 1996. [Online]. Available: https://arxiv.org/abs/cs/9605103

[24] M. Vrigkas, C. Nikou, and I. A. Kakadiaris, "A review of human activity recognition methods," Frontiers in Robotics and AI, vol. 2, 2015. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2015.00028

[25] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," Proceedings of the IEEE, vol. 90, no. 7, pp. 1151–1163, 2002.

[26] G. Zhang, Z. Yuan, Q. Tong, and Q. Wang, "A novel and practical scheme for resolving the quality of samples in background modeling," Sensors, vol. 19, no. 6, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/6/1352

[27] X. Yan, I. A. Kakadiaris, and S. K. Shah, "Modeling local behavior for predicting social interactions towards human tracking," Pattern Recogn., vol. 47, no. 4, p. 1626–1641, apr 2014. [Online]. Available: https://doi.org/10.1016/j.patcog.2013.10.019

[28] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. Hauptmann, "Devnet: A deep event network for multimedia event detection and evidence recounting," 06 2015, pp. 2568–2577.

[29] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18–28, 1998.

[30] T. K. Ho, "The random subspace method for constructing decision forests," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832–844, 1998.

[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine Learning, vol. 63, no. 1, pp. 3–42, Apr 2006. [Online]. Available: https://doi.org/10.1007/s10994-006-6226-1

[32] S. Walczak and N. Cerpa, "Artificial neural networks," in Encyclopedia of Physical Science and Technology (Third Edition), third edition ed., R. A. Meyers, Ed. New York: Academic Press, 2003, pp. 631–645. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B0122274105008371

[33] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," ACM Comput. Surv., vol. 52, no. 3, jun 2019. [Online]. Available: https://doi.org/10.1145/3310194

[34] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in

Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, ser. MobiCom '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 65–76. [Online]. Available: https://doi.org/10.1145/2789168.2790093

[35] espressif, "Esp-csi," https://github.com/espressif/esp-csi, 2021.

[36] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the mona lisa: Spot localization using phy layer information," in Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, ser. MobiSys '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 183–196. [Online]. Available: https://doi.org/10.1145/2307636.2307654

[37] X. Wang, L. Gao, and S. Mao, "Csi phase fingerprinting for indoor localization with a deep learning approach," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 1113–1123, 2016.

[38] S. Butterworth et al., "On the theory of filter amplifiers," Wireless Engineer, vol. 7, no. 6, pp. 536–541, 1930.

[39] M. Heideman and et al., "Gauss and the history of the fast fourier transform."

[40] E. Sejdić, I. Djurović, and J. Jiang, "Time–frequency feature representation using energy concentration: An overview of recent advances," Digital Signal Processing, vol. 19, no. 1, pp. 153–183, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S105120040800002X

[41] M. J. Shensa et al., "The discrete wavelet transform: wedding the a trous and mallat algorithms," IEEE Transactions on signal processing, vol. 40, no. 10, pp. 2464–2482, 1992.

[42] Y. Xie, J. Xiong, and K. Jamieson, "md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking," 05 2019, pp. 1–16.

APPENDIX A: LIST OF ACRONYMS

CSI	Channel State Information
LOS	Line of Sight scenario
SVM	Support Vector Machine
LSTM	Long Short-Term Memory
NIC	Network Interface Card
DWT	Discrete Wavelet Transform
PCA	Principal Component Analysis
IEEE	Institute of Electrical and Electronics Engineers
AP	Access Point
WMA	Weighted Moving Average
ICA	Independent Component Analysis
CNN	Convolutional Neural Networks
ULP	Ultra-Low Power
RAM	Random Access Memory
ROM	Read Only Memory
EDR	Enhanced Data Rate
BLE	Bluetooth Low Energy
SPI	Serial Peripheral Interface
MAC	Media Access Control
DMA	Direct Memory Access
PWM	Pulse Width Modulation
WPA	Wi-Fi Protected Access

WLAN	Wireless Local Area Network
WAPI	WLAN Authentication and Privacy Infrastructure
ECC	Elliptic Curve Cryptography
RNG	Random Number Generator
GPIO	General-Purpose Input/Output
RTC	Real Time Clock
OFDM	Orthogonal Frequency Division Multiplexing
QAM	Quadrature Amplitude Modulation
RSSI	Received Signal Strength Indication
MCS	Modulation Coding Scheme
MIMO	Multiple-Input Multiple-Output
UART	Universal Asynchronous Receiver Transmitter
MCU	Micro Controller Unit
CPU	Central Processing Unit
RF	Radio Frequency
LAN	Local Area Network
MU-MIMO	Multi-User MIMO
DSL	Digital Subscriber Line
RL	Reinforcement Learning
ICMP	Internet Control Message Protocol
PCB	Printed Circuit Board
BPSK	Binary Phase-Shift keying
FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
IDFT	Inverse Discrete Fourier Transform
STFP	Short-Time Fourier Transform
SDR	Software Defined Radio
CWT	Continuous Wavelet Transform
IDWT	Inverse Discrete Wavelet Transform

PCC	Pearson's Correlation Coefficient
DT	Decision Tree
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
TPR	True Positive Rate
FPR	False Positive Rate
FNR	False Negative Rate
AUC	Area Under Curve
ROC	Receiver Operating Characteristic
CV	Cross Validation
LOOCV	Leave-One-Out Cross Validation
LR	Logistic Regression
SVC	Support Vector Classifier
KNN	K Nearest Neighbors
RFC	Randon Forest Classifier
ETC	Extra Trees Classifier
XGB	XGBoost
GUI	Graphical User Interface