CSI-based Gesture Recognition and Object Detection

Ruyin Zhao

Follow this and additional works at: https://lib.dr.iastate.edu/creativecomponents

Part of the Data Science Commons, and the Digital Communications and Networking Commons

Recommended Citation

This Creative Component is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Creative Components by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
CSI-based Gesture Recognition and Objection Detection

Ruyin Zhao

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
MASTER OF SCIENCE IN COMPUTER ENGINEERING

2021
# Table of Contents

**Abstract** .................................................. iii

**Lists of Figures** ........................................ vi

1 **Goals** ..................................................... 1

2 **Background and Related Works** ................. 2
   2.1 Overview ............................................. 2
   2.2 Introduction to CSI data ......................... 2
      2.2.1 Channel State Information .................. 2
      2.2.2 CSI Tools ....................................... 4
      2.2.3 CSI Data ........................................ 9
   2.3 Current Wi-Fi sensing applications and examples ........ 10
      2.3.1 Human Event Detection ...................... 11
      2.3.2 Activity recognition ......................... 14
      2.3.3 Device-Free Human Localization/Tracking ....... 16
   2.4 Data Processing of Wi-Fi Sensing ................. 18
      2.4.1 Experimental Settings ....................... 18
      2.4.2 Data preprocessing ............................. 18
      2.4.3 Feature extraction ............................. 23
      2.4.4 Training ......................................... 24

3 **Our Works** ............................................. 28
   3.1 Gesture Recognition ................................. 28
      3.1.1 Data Collection .................................. 28
      3.1.2 SVM Algorithm ................................... 45
      3.1.3 Random Forest Algorithm ..................... 49
      3.1.4 K-Nearest Neighbors Algorithm ............. 52
      3.1.5 Long Short-Term Memory ...................... 53
      3.1.6 Summary ......................................... 56
   3.2 Reflector Angle Recognition ....................... 58
      3.2.1 Data Collection .................................. 58
      3.2.2 Findings ......................................... 77

4 **Conclusions** ........................................... 87
<table>
<thead>
<tr>
<th>References</th>
<th>88</th>
</tr>
</thead>
</table>
Abstract

Nowadays, Wi-Fi is closely related to our life as we use Wi-Fi to communicate everyday. At the same time, people are constantly exploring other uses of Wi-Fi. With the release of Channel State Information (CSI) measurement tool, we can achieve more accurate indoor activity and object detection. In order to make a better used of the combination of Wi-Fi sensing tools and data analytic methods, we first have a deep look at CSI and methods of signal processing. Then, we use csi and SVM, KNN, Decision Tree, LSTM and Random Forest to perform gesture recognition and object detection.

**Keywords:** Wireless Sensing, Channel State Information (CSI), Machine Learning, Gesture Recognition, Object Detection.
List of Figures

2.1 MIMO System ......................................................... 3
2.2 MIMO system H ..................................................... 4
2.3 Atheros 802.11n chip - AR9580 .................................... 5
2.4 SAMSUNG BCM4339 Wi-Fi chips ................................... 6
2.5 ESP32 CSI Toolkit .................................................... 6
2.6 Intel 5300 NIC ....................................................... 7
2.7 The 30 measured subcarriers in 20MHz Channel ................. 8
2.8 The 30 measured subcarriers in 40MHz Channel ................. 8
2.9 CSI Data ............................................................. 9
2.10 CSI Matrix .......................................................... 10
2.11 Applications of Wi-Fi Sensing ....................................... 11
2.12 Floor plan of a store ................................................ 12
2.13 CLAC system ........................................................ 13
2.14 CSI data from 3 antennas .......................................... 14
2.15 EI system ............................................................ 15
2.16 Experimental setting ............................................... 15
2.17 Human activities .................................................... 16
2.18 C2IL system ........................................................ 17
2.19 The matching between fingerprint and floor plan ............... 21
2.20 An illustration of human activity sensing approaches ........... 24
2.21 Schema structure of CNNs .......................................... 26

3.1 Experimental Scenarios ............................................... 29
3.2 List of WLAN Channels .............................................. 30
3.3 Subcarrier Frequency in 802.11n - 56 Subcarriers ................ 31
3.4 High Arm Wave - CSI Amplitude normalized to an Intel’s internal reference level ........................................ 31
3.5 Hand Clapping - CSI Amplitude normalized to an Intel’s internal reference level ........................................ 32
3.6 High Arm Wave - RSSI ................................................ 33
3.7 Hand Clapping - RSSI ............................................... 34
3.8 Phase Calibration Algorithm ......................................... 36
3.9 Original Phase ....................................................... 37
3.10 Original Phase - one packet ........................................ 38
3.11 Unwrapped Phase ................................................... 39
3.12 Unwrapped Phase - one packet .................................................. 40
3.13 Phase After Linear Transformation ........................................... 41
3.14 Phase After Linear Transformation - one packet .......................... 42
3.15 high arm wave phase ............................................................... 43
3.16 high arm wave phase - one packet ............................................ 43
3.17 hand clapping phase ............................................................... 44
3.18 hand clapping phase - one packet ............................................ 45
3.19 CSI - Preprocessing ............................................................... 46
3.20 RSSI - Preprocessing ............................................................. 46
3.21 SVM Result ............................................................................ 48
3.22 SVM Model With and Without Phase ......................................... 49
3.23 Predictor Importance Estimates ............................................... 50
3.24 RF Result ............................................................................... 51
3.25 Predictor Importance Estimates - Phase Added ............................ 51
3.26 RF Model With and Without Phase ........................................... 52
3.27 K Values and Corresponding Accuracy ....................................... 53
3.28 KNN Result ............................................................................ 54
3.29 KNN-k values (phase added) ..................................................... 54
3.30 KNN Model With and Without Phase ......................................... 55
3.31 LSTM Result ........................................................................... 57
3.32 Gesture Recognition Result ...................................................... 57
3.33 Experiment Setting - A15 ......................................................... 59
3.34 Picture of the Reflector .............................................................. 60
3.35 A15 - CSI Amplitude in the First Data file .................................. 60
3.36 A15 - CSI Phase in one of the packets ...................................... 61
3.37 A15 - RSSI Array in the First Data File ..................................... 62
3.38 A30 - Experiment Setting ........................................................ 63
3.39 A30 - CSI Amplitude in the First Data file .................................. 64
3.40 A30 - RSSI Array in the First Data File ..................................... 65
3.41 A45 - Experiment Setting ........................................................ 66
3.42 A45 - CSI Amplitude in the First Data file .................................. 67
3.43 A45 - RSSI Array in the First Data File ..................................... 68
3.44 A60 - Experiment Setting ........................................................ 69
3.45 A60 - CSI Amplitude in the First Data file .................................. 70
3.46 A60 - CSI Phase in one of the packets ...................................... 71
3.47 A60 - RSSI Array in the First Data File ..................................... 72
3.48 A90 - Experiment Setting ........................................................ 73
3.49 A90 - CSI Amplitude in the First Data file .................................. 74
3.50 A90 - CSI Phase in one of the packets ...................................... 75
3.51 A90 - RSSI Array in the First Data File ..................................... 76
3.52 Binary Classification Between A15 and A90 ............................... 78
3.53 Binary Classification Between A15 and A60 ............................... 79
3.54 Binary Classification Between A15 and A45 ............................... 80
3.55 Binary Classification Between A15 and A30 ............................... 81
3.56 Multi Classification Between All Angles ................................... 83
LIST OF FIGURES

3.57 Multi Classification Between All Angles – Phase Added .................. 83
3.58 Multi Classification Between All Angles – Phase Added – Feature Order Adjusted .......................... 84
3.59 Feature Occurrence ........................................................................ 85
3.60 Multi Classification - Selected Features ........................................... 85
Chapter 1

Goals

The purpose of this project is to explore the use of CSI data. We hope to know the status of indoor objects from the information carried by CSI data, so as to monitor indoor objects. Therefore, we start by studying a popular topic, namely gesture recognition, and then we will try use the knowledge gained from this research to monitor indoor objects. When the reflector is at different angles indoors, we will collect data and try to use CSI data to identify the state of the reflector.
Chapter 2

Background and Related Works

2.1 Overview

Since we will use CSI data for Wi-Fi sensing, in this chapter, we will discuss CSI data, current Wi-Fi sensing applications, and data processing methods.

2.2 Introduction to CSI data

2.2.1 Channel State Information

In wireless communications, channel state information (CSI) refers to known channel properties of a communication link. This information describes how a signal propagates from the transmitter to the receiver and represents the combined effect of, for example, scattering, fading, and power decay with distance. Many IEEE 802.11 standards use orthogonal frequency-division multiplexing (OFDM) modulation technology to transmit signals through multiple orthogonal subcarriers, and the signals transmitted on each subcarrier have different signal strengths and phases. At the same time, the current spectrum resources are
becoming increasingly scarce, so multiple-input and multiple-output (MIMO) technology has been applied widely. MIMO technology uses multiple antennas to increase data throughput and transmission distance without increasing bandwidth and transmission power [1]. In the wireless communication process, RSS can only make a rough estimation of the wireless channel, without considering the number of antennas and the number of subcarriers, and CSI can provide detailed amplitude and phase information of different subcarriers, so CSI makes it possible to adapt transmissions to current channel conditions, which is crucial for achieving reliable communication with high data rates in multiple input multi output systems.

For the MIMO system shown in figure 1, there are t transmit antennas and r receive antennas, for each transmit-receive antenna pair, the channel could represent by a complex fading coefficient $h_{ij}$, in which i denotes the index of receive antenna, and j denotes the index of transmit antenna. So, the channel of this MIMO system could be represented by a CSI matrix $H$. 

![MIMO System](image.png)

Figure 2.1: MIMO System
According to the amount of transmit and receive antennas, $H$ would be a $r \times t$ matrix, where

$$H = \begin{bmatrix} h_{11} & h_{12} & \ldots & h_{1t} \\ h_{21} & h_{22} & \ldots & h_{2t} \\ \vdots & & & \vdots \\ h_{r1} & h_{r2} & \ldots & h_{rt} \end{bmatrix}$$

All in all, estimating $H$ is the problem of ‘Channel Estimation’.

### 2.2.2 CSI Tools

There are currently four popular tools to collect CSI information.

**Atheros CSI Tool**

Atheros-CS1-Tool is built on top of ath9k, which is an open source Linux kernel driver supporting Atheros 802.11n PCI/PCI-E chips, so theoretically this tool is supposed to be able to support all types of Atheros 802.11n Wi-Fi chipsets. [2]
With using Atheros CSI Tool, the CSI of a packet transmitted with $M$ transmitting antennas, $N$ receiving antennas, 20MHz channel bandwidth, is a matrix of size $M \times N \times 56$. If the bandwidth is 40MHz, then the size of CSI matrix becomes $M \times N \times 114$. In CSI Matrix, all the elements are complex numbers. Atheros Wi-Fi NIC uses 20 bits to give the value of the real part and another 10 bits to describe the imaginary part.

**SAMSUNG BCM4339 Wi-Fi chips + NEXMON extractor**

Nexmon is our C-based firmware patching framework for Broadcom/Cypress WiFi chips that enables you to write your own firmware patches, for example, to enable monitor mode with radiotap headers and frame injection [3]. By combining the SAMSUNG BCM4339 Wi-Fi chips, which is shown in figure 2.4 and the NEXMON extractor, it allows to extract CSI of up to 4x4 MIMO transmissions at 80 MHz bandwidth.
ESCAPE CSI Toolkit

This ESP32 tool shown in figure 2.5 can work as a complete device by collecting the CSI measurements as well as processing further for DFWS applications. It can work as standalone device, unlike other CSI tools, and can offer large-scale deployment to many DFWS applications [4]. This tool can provide all 64 subcarrier information in 20Mhz bandwidth. [5]
Linux 802.11n CSI Tool

Linux 802.11n CSI Tool is built on the Intel Wi-Fi Wireless Link 5300 802.11n MIMO radios, using a custom modified firmware and open source Linux wireless drivers [6].

![Figure 2.6: Intel 5300 NIC](image)

The IWL5300 provides 802.11n channel state information in a format that reports the channel matrices for 30 subcarrier groups, which is about one group for every 2 subcarriers at 20 MHz or one in 4 at 40 MHz [7]. So, With using Linux 802.11n CSI Tool, the CSI of a packet transmitted with M transmitting antennas, N receiving antennas is a matrix of size M×N×30. Each channel matrix entry is a complex number, with signed 8-bit resolution each for the real and imaginary parts. To process CSI data, we can use Matlab or Octave [8]. The set of subcarriers measured is specified by the IEEE 802.11n-2009 standard (in Table 7-25f on page 50). Intel’s implementation only supports the versions that measure 30 subcarriers each, corresponding to Ng = 2 for 20 MHz and Ng = 4 for 40 MHz. And the subcarrier frequency spacing is 312.5KHz (20MHz / 64 or 40MHz/128) [9].

In the 20 MHz HT format, the signal is transmitted on subcarriers –28 to –1 and 1 to 28, and the channel information of 30 of them are measured by this 5300 CSI tool. The 30 selected subcarriers are marked ’1’ in Figure 2.7.
CHAPTER 2. BACKGROUND AND RELATED WORKS

Figure 2.7: The 30 measured subcarriers in 20MHz Channel

Note: The frequency of each subcarrier will be listed in the next chapter.

In the case of the 40 MHz HT format, a 40 MHz channel is used. The channel is divided into 128 subcarriers (on page 267 of [9]). The signal is transmitted on subcarriers –58 to –2 and 2 to 58. The 30 selected subcarriers are marked ‘1’ in Figure 2.8

Figure 2.8: The 30 measured subcarriers in 40MHz Channel

As Atheros CSI Tool provides information for all 56 and 114 subcarriers at 20MHz and
40MHz respectively, whereas Linux 802.11n CSI Tool supports only 30 subcarriers, Atheros CSI Tool is a better choice for projects that require more information. Also, as Atheros CSI Tool describes entries of CSI matrix, Atheros can provide a more accurate channel information. However, after reading the comments online, I decided to go with the 5300 NIC because the intel card has better performance than the Atheros do. People said they had a hard time connecting devices using Atheros.

### 2.2.3 CSI Data

Linux 802.11n CSI Tool is used to show CSI data. Figure 5 is a CSI sample with 278 packets, every packet is a structure, which contains timestamp, number of receive antenna, number of transmit antenna, CSI matrix, and so on.

![Figure 2.9: CSI Data](image)

I did matrix transpose, so the dimension of this matrix is 1*3*30, 1 is the number of transmit antenna, 3 is the number of receive antenna, and 30 is the number of subcarriers. Inside the matrix, all the elements are complex numbers.
There are some surveys talk about how people use channel state information to do sensing. [7] presents a comprehensive survey of recent advances in the CSI-based sensing mechanism and illustrates that CSI outperforms RSSI in sensing accuracy due to its stability and rich information. [10] gives a comprehensive summary and comparison of the signal processing techniques and algorithms of a wide variety of Wi-Fi sensing applications. Based on this knowledge, we will introduce CSI in more detail from the perspective of applications.

### 2.3 Current Wi-Fi sensing applications and examples

Figure 7 shows the existing Wi-Fi sensing applications. They are divided into 3 categories: Detection, recognition and estimation. Detection is a binary classification problem, recognition is a multi-class classification problem, and estimation is to figure quantity values of size, length, angle, distance, duration, frequency, counts and so on [11].

The following are examples of each category.
### 2.3.1 Human Event Detection

Figure 8 is a floor plan of a store, and people may appear in these 20 locations. Location number 1 - 16 are inside the store, location number 18-20 are outside the store, and location number 17 is in between. The access point is placed at the counter. The owner of this store wants Wi-Fi to be accessed only by people inside the door. So [12] design a system named CLAC, which is a crowdsourced location aware Access Control scheme using physical layer information to restrict network accessing. The system is shown in figure 9.

<table>
<thead>
<tr>
<th>Type</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>Human Present Detection</td>
</tr>
<tr>
<td></td>
<td>Human Event Detection</td>
</tr>
<tr>
<td></td>
<td>Object Detection, LoS/Non-LoS Detection</td>
</tr>
<tr>
<td>Recognition</td>
<td>Activity Recognition</td>
</tr>
<tr>
<td></td>
<td>Gesture Recognition</td>
</tr>
<tr>
<td></td>
<td>Human/User Identification, Human/User Authentication</td>
</tr>
<tr>
<td></td>
<td>Object Recognition, Object Event Recognition</td>
</tr>
<tr>
<td>Estimation</td>
<td>Device-Free Human Localization/Tracking</td>
</tr>
<tr>
<td></td>
<td>Device-Based Human Localization/Tracking</td>
</tr>
<tr>
<td></td>
<td>Object Localization/Tracking, Humidity Estimation</td>
</tr>
<tr>
<td></td>
<td>Breathing/Respiration Rate Estimation, Heart Rate Estimation</td>
</tr>
<tr>
<td></td>
<td>Human Counting, WiFi Imaging</td>
</tr>
</tbody>
</table>
Figure 2.12: Floor plan of a store

The blue part is the core of the system, a white list, a CSI model, and a Machine Learning model. If a customer enters an access code, he would be added to a white list. If a customer in direct proximity to access point, he would gain a high score in CSI model. For the other customer, if he is stable and behave similar to the admitted users, he would gain a high score in machine learning model. Those who get high score could access to the network. During these procedures, as this system uses CSI and RSS to analyse the location of customers and then restrict network access, two CSI metrics is used: CSI Cross-Antenna Stability Metric and CSI Cross-Frame Stability Metric, which summarize well the spatial and temporal CSI characteristics respectively. CSI could help to figure out whether a person is closed to access point, and whether he is stable. As shown in figure 10, each color denotes one transmit-receive antenna pair. Location 3 is closed to access point, so the curves are clear and very similar to each other. Location 12 is not that close, the curves are still clear but not that similar. So CSI could be used to distinguish the location of customer. When a customer tries to connect to AP, the model receive CSI information and then calculate ‘Cross-Antenna CSI’ using pearson correlation coefficient. If the coefficient is bigger than a given threshold, then the customer would be recognized as ‘stable and close to AP’. Also, For the same position, for example position 3, if a person
is stable, the curves are clear, if a person is moving, the curves are messy. So we can figure the status of people by calculating ‘Cross-Frame CSI’.

Figure 2.13: CLAC system
2.3.2 Activity recognition

Figure 11 shows a system named EI, which is a deep-learning based device free activity recognition framework that can extract environment/subject-independent features shared by the data collected on different subjects under different environments. Figure 12 shows the experimental setting, and a person is doing some activity between transmitter and receiver.
It’s shown that, by using this EI model, it is very likely for us to figure out whether the person is (a) wiping the whiteboard, (b) walking, (c) moving a suitcase, (d) rotating the chair, (e) sitting or (f) standing up and sitting down.
2.3.3 Device-Free Human Localization/Tracking

Figure 14 shows a system named $C^2$IL, which is a passive crowdsourcing CSI-based indoor localization scheme. It is built to accurately estimate the moving distance purely based on 802.11n channel state information [14]. The model only requires the locating-device (e.g., a phone) to have an 802.11n wireless connection, and it does not rely on inertial sensors only existing in some smartphones [15]. As wireless signal propagation in indoor environments can be well modelled as Rician fading channel, the moving distance could be measured according to the moving velocity before and after the Rician fading [16]. In this model, only the first spatial stream and only the amplitude information were used. Because of the reception rate was unstable, the authors resampled the data to a stable reception rate. After that, convolution was applied to filter out high frequency noise. And another convolution was applied to enhanced fading. Later, expected frequency
of every subcarrier was extracted by using STFT with 50% overlapping window, which reveals the spectral density of subcarrier $i$ along with time. Then the moving speed was estimated by the expected frequency. What’s more, the correlation between consecutive CSI samples was used to detect start/stop.

Relying on the moving distance estimation, an accurate mapping between RSS fingerprints and location was able to be constructed.

As shown in Figure 15, when a user enters the building, the device held by the user will connect to the APs inside the building. The APs will record the CSI and RSS values and send the data to a server. The server would quickly estimate the moving distance and build a fingerprint. And then, the fingerprint would be used to match with the locations, so we can achieve human tracking.
2.4 Data Processing of Wi-Fi Sensing

2.4.1 Experimental Settings

Among the papers talking about application of CSI, there are 5 papers illustrated their experimental settings. They all use 1 transmitter. And inorder to reduce the amount the computation, [17], [14], and [18] specify that they only take data which were from the first transmitting antenna. 1 transmitter (1 antenna) and at least 3 receivers (all of them have 3 antennas, which are placed in a line), 1000 packets per second. Devices are set to work in the monitor mode, on channel 165 at 5.825 GHz where there are few interfering radios as interference does pose severe impacts on the collected CSI measurements. [17] 1 client, 18 APs. Uses only the first spatial stream (1st transmitting antenna and 1st receiving antenna), so that the computation space is greatly reduced. [14] 1 transmitter, 1 receiver, 30 packets per second. Devices are set to work in the AP mode at 2.4 GHz. [19] 1 transmitter, 2 receivers, 200 packets per second. [13] Set the MIMO Tx-antenna Ntx = 1 and Rx-antenna Nrx = 3. [18]

2.4.2 Data preprocessing

Resample

Due to the wireless traffic control, the instantaneous reception rate of frames is unstable. To get a stable reception frequency, [14] and [14] did resample. In [14], the amplitude matrix $A = A_1, ..., A_n T$ is further defined, where $A_i$ is the i-th received column-wise amplitude. $A$ is resampled to a stable reception frequency $f$ with the even interval between each slot.
Interpolation

To deal with the CSI information loss caused by sampling jitter and outlier removal, the CSI measurements should be interpolated to obtain uniform sampling periods. In [20], [21] and [22], the authors used 1-D linear interpolation algorithm to process the raw CSI readings. I tried to find some other interpolation algorithms that were used for CSI processing but failed. In [13], the authors did interpolation but did not state which algorithm they used, so I would temporarily assume that they also used 1-D linear interpolation algorithm, and keep learning more about interpolation algorithm.

Dimensionality Reduction

In [18], the authors reduce the calculation by using only the amplitude information, and merging the adjacent amplitudes. Every 2 adjacent amplitudes are merged to their mean as \( \frac{Y_i + Y_{i+1}}{2} \).

Noise cancellation

Moving Average, Median Filter, Low-Pass Filter, Wavelet Filter, Hampel Filter, Local Outlier Factor and Signal Nulling are widely used methods to remove outliers and noises. Also, convolution method, experience-threshold, filters are sometimes been applied. In [14], the authors use convolution based noise cancellation, to filter out the high frequency noise.

In [23], the authors evaluate the possible experience-threshold of RSS values, and combine with variance of RSS to roughly detect outlier point. [23] uses low-pass filter to remove high frequency. [13] uses Hampel filter to remove outliers and do downsample.
CHAPTER 2. BACKGROUND AND RELATED WORKS

Smoothing

Weighted moving average was applied to smooth the sequence. [23]

Fading enhancement

To emphasize the fading. There are at least two methods. a) first-class derivation of A, quite sensitive to high frequency noise rather than low frequency ripples. b) convolution with sobel-style calculator. [14]

Time–frequency transforms

Signal transform methods are used for time-frequency analysis of a time series of CSI measurements. The signal transform output in this scope represents the frequency of CSI change patterns rather than the carrier frequency. Signal transform methods includes Fast Fourier Transform, Short Time Fourier Transform, Discrete Hilbert Transform, Discrete Wavelet Transform, and so on.

In order to obtain moving speed, [17] and [14] do Short-term Fourier transform, and [13] combines segments with their Fast Fourier transform as the input to the deep learning model.

Short-term Fourier transform: yields power distribution over the time and frequency domains. [17]

Short-term Fourier transform: STFT with 50% overlapping window is applied to obtain the Power Spectral Density (PSD) of i-th envelope of A. It reveals the spectral density of subcarrier i along with time, and moving speed could be estimated with estimated frequency. [14]

Correlation and FFT: for each segment from the two receivers, calculate the correlation
between the segment and the segments lagged by no more than 128 time units. Then combine them with the FFT of each segment as the input to the deep learning model. [13]

Figure 2.19: The matching between fingerprint and floor plan

Normalization

In order to remove irrelevant variations caused by instances, persons and hardware settings, [17] and [13] did normalization. [17] adjusted the sum of all the elements in a profile to 1, and [13] normalized to have a mean of zero and standard deviation of one.

Start/Stop detection

Correlation between consecutive CSI samples was used to detect the start and end of an activity. [14]
CHAPTER 2. BACKGROUND AND RELATED WORKS

Cross-antenna correlation

When a user is stationary and close enough to an AP without obstacles between them, the corresponding CSI values on different antennae will be similar and stable, so, a CSI cross-antenna stability metric could be used to measure a user’s proximity to the AP. [12]

Cross-frame correlation

In order to determine whether a user is stable, they calculate the correlation coefficient between successive CSI samples on the same antenna. [12]

Subcarrier selection

By using K means algorithm, 30 subcarriers are divided into 3 clusters according to the sensitivity on activity. [23]

Activity segmentation

In [23], the authors use two methods. a) remove the first second and last second data sequence. b) calculate moving variance for the sequence and then detect the sharp points of each activity, and roughly detect the start and end of each activity. In [13], the authors segment the CSI measurements every 128 samples with 32 samples overlap, which corresponds to the human activity of about 5.12 seconds.
CHAPTER 2. BACKGROUND AND RELATED WORKS

Distance calculation

Earth moving distance: calculate the distance between two distributions (similar to loss function). [17] Degree of following: to characterize how closely a newly received MF M follows the trend defined by frames in the sliding window W. This DOF is determined by two factors: the distance to its k nearest points in the window W and the time difference between M’s arrival time \( t_M \) and the arrival times of its k nearest points. [8]

2.4.3 Feature extraction

The value of BVP (Body-coordinate velocity Profile), i.e. distribution of signal power over velocity components in the body coordinate system. [17] In [14], after estimation the moving distance of the client by analysing CSI data, the RSS value would be used to build a matching with the fingerprint database to achieve the localization. In [23], the authors built two feature sets. Which are:

RSS feature set: local variance, location of peak value, number of peaks;
CSI feature set: variance, envelop of CSI, signal entropy, velocity of signal change, median absolute deviation, period of motion, normalized standard deviation of CSI. In [13], after applying some simple data preprocessing procedures, the authors use three-layer stacked CNNs to extract features. [12] first determine if a user is valid by analysing CSI data, then add the RSS data of valid users into the training set. [18] use the value of DoF (Degree of following) as feature.
Figure 2.20: An illustration of human activity sensing approaches

### 2.4.4 Training

#### Statistical Model

Statistical models rely on empirical measurements or probability functions to characterize wireless channels. Rician fading is a stochastic model used by some Wi-Fi sensing applications. It is a simple model for multi-path channels with a dominant path that is stronger than others. CSI similarity is a widely used metric for motion-related Wi-Fi sensing applications. It is calculated by the cross correlation of two CSI matrices. Empirical measurements show that CSI similarity is a good indicator of whether the Wi-Fi device and surrounding objects are static or moving. For the papers I read, some of them got results without using learning algorithm. [18] uses a threshold to filter out invalid packets based on their experiences.
CHAPTER 2. BACKGROUND AND RELATED WORKS

Shallow Learning Algorithms

Instance-based learning algorithms, such as k Nearest Neighbor (kNN), Support Vector Machine (SVM), and self-Organizing Map (SOM), are widely used for detection and recognition applications. These algorithms compute the distance between each testing sample and every training samples. The input for shallow learning algorithms could be raw CSIs, pre-processed CSIs, or feature vectors. Feature engineering could help extract meaningful and compressed information from raw or preprocessed CSIs. Usually, feature extraction and selection have a great impact on the performance of shallow learning algorithms. For the papers I read, Some papers use machine learning method to do classification. Both \[23\] and \[12\] use SVM.

Deep Learning algorithms

For shallow learning algorithms, it is hard to extract and select the right features effectively and efficiently. Deep Learning Networks address the problem by learning features automatically. Also, DNN requires very little or none signal processing, feature engineering, and parameter tuning \[24\].

Deep Neural Network (DNN): usually serves as the dense layer of other deep models. For example, in a convolution neural network, several dense layers are often added after the convolution layers. when the HAR data is multi-dimensional and activities are more complex, more hidden layers can help the model train well since their representation capability is stronger \[25\].

Convolutional neural network (CNN): as shown in figure 17, CNNs are hierarchical structures that combine convolutional operations using learnable filters and non-linear activation functions, downsampling operations and classifiers \[26\]. They map their input into a more compact representation, or they classify their input into classes, depending on their objective function. Convolutional layers extract particular features at different locations from
CHAPTER 2. BACKGROUND AND RELATED WORKS

their inputs. By stacking them, and downsampling their outputs, CNNs extract more complex and abstract features, being at the same time invariant to distortions and translations.

![Figure 2.21: Schema structure of CNNs](image)

Autoencoder (AE): learns a latent representation of the input values through the hidden layers, which can be considered as an encoding-decoding procedure. The purpose of autoencoder is to learn more advanced feature representation via an unsupervised learning schema, which makes AE a powerful tool for feature extraction. But AE depends too much on its layers and activation functions which may be hard to search the optimal solutions.

Restricted Boltzmann Machine (RBM): a bipartite, fully connected, undirected graph consisting of a visible layer and a hidden layer. The stacked RBM is called deep belief network (DBN) by treating every two consecutive layers as an RBM. DBN/RBM is often followed by fully-connected layers [27]. Similar to autoencoder, RBM/DBN can also perform unsupervised feature learning for HAR.

Recurrent Neural Network (RNN): widely used in speech recognition and natural language processing by utilizing the temporal correlations between neurons. Few works used RNN for the HAR tasks, where the learning speed and resource consumption are the main concerns for HAR.

Hybrid Model: the combination of some deep models. One emerging hybrid model is the combination of CNN and RNN. [28] provided good examples for how to combine CNN and RNN. Based on the papers I read, CNN is more popular in activities recognition problems. [17] use a hybrid deep learning model, which from bottom to top consists of
a convolutional neural network (CNN) for spatial feature extraction and a recurrent neural network (RNN) for temporal modeling. [14] maps between RSS fingerprint and location using unsupervised learning. In [13], after using CNN to extract the features, the authors use cross entropy function to calculate the loss between the predictions and the ground truths.
Chapter 3

Our Works

3.1 Gesture Recognition

3.1.1 Data Collection

We used a dataset that contains 2 gestures, one of which is a high arm wave, and the other is a hand clapping. Each gesture was performed by ten volunteers, and each volunteer performed 20 to 30 times. Therefore, each gesture has 260 data files. Each gesture lasts about 6 seconds, and the sample rate is 30 packets per second. Therefore, each data file has about 200 packets. The data set was downloaded from GitPrime [29].

In the experiment, a commercial TP-Link wireless router was used as the transmitter operating in IEEE 802.11n AP mode at 2.4GHz. A ThinkPad 400 laptop with three antennae running Ubuntu 10.04 was used as a receiver, which was equipped with Intel 5300 card. The data was collected in three environments listed in figure 3.1. The distance between receiver and transmitter is 6m, and volunteer stands the middle point of transmitter receiver without obstacles. [23]
According to the [30], fourteen channels are designated in the 2.4 GHz range, spaced 5 MHz apart from each other except for a 12 MHz space before channel 14. And the list of WLAN channels are listed in figure 3.2. To guarantee no interference in any circumstances the Wi-Fi protocol requires 16.25 to 22 MHz of channel separation. The remaining 2 MHz gap is used as a guard band to allow sufficient attenuation along the edge of the band.

Figure 3.1: Experimental Scenarios
The channel can be selected in the web management interface of the wireless router. For example, if you selected channel 6, which frequency range is 2426-2448MHz and the center frequency is 2437, then after leaving the 2 MHz gap, the frequency you use will be from 2427 to 2447. In the 20 MHz HT format, the channel is divided into 64 subcarriers, and the signal is transmitted on subcarriers –28 to –1 and 1 to 28 (56 in total), with 0 being the center (dc) carrier [9]. The subcarrier frequency spacing is 312.5kHz (20 MHz/64). So, as we know the center frequency and the frequency spacing, we can calculate the frequency of each subcarrier:

![Figure 3.2: List of WLAN Channels](image_url)

The channel can be selected in the web management interface of the wireless router. For example, if you selected channel 6, which frequency range is 2426-2448MHz and the center frequency is 2437, then after leaving the 2 MHz gap, the frequency you use will be from 2427 to 2447. In the 20 MHz HT format, the channel is divided into 64 subcarriers, and the signal is transmitted on subcarriers –28 to –1 and 1 to 28 (56 in total), with 0 being the center (dc) carrier [9]. The subcarrier frequency spacing is 312.5kHz (20 MHz/64). So, as we know the center frequency and the frequency spacing, we can calculate the frequency of each subcarrier:
After collecting the data, from each data file, we extracted the CSI array and RSSI array for later analysis.

**Figure 3.3: Subbarrier Frequency in 802.11n - 56 Subbarriers**

![Image of the table showing subbarrier frequency data]

**Figure 3.4: High Arm Wave - CSI Amplitude normalized to an Intel’s internal reference level**

![Image of the graph showing CSI amplitude]
Figure 3.5: Hand Clapping - CSI Amplitude normalized to an Intel’s internal reference level
Figure 3.6: High Arm Wave - RSSI
Figure 3.7: Hand Clapping - RSSI
CSI array is an n * 90 matrix, normalized to an Intel’s internal reference level. n is the number of packages in this data file. 90 is the number of subcarriers, that is, 1 transmitting antenna * 3 receiving antennas * 30 subcarriers = 90 subcarriers. The data in the CSI array data are all complex numbers. Our analysis only use the amplitude part, and ignore the phase part. RSSI array is an n * 3 matrix. The number 3 means 3 antennas. The RSSI values are in dBm [31].

Although the CSI phase is available from the IWL 5300 NIC, it has not yet been widely used. The problem is mainly due to hardware defects [32], leading to the measured phase error. One of the reasons is that the center frequency between the receiver and the transmitter cannot be completely synchronized [33]. The other is the sampling frequency offset (SFO) generated by the ADC due to nonsynchronized clocks. Minor related works proposed some specific algorithms to calibrate the phase information, and these algorithms are at the forefront. For example, this is a patent issued by HuaWei [34] for phase processing, and this is a paper [35] published by a research group that has been specializing in this topic for years.

We use the cai phase calibration algorithm from [36], and [32] to calibrate the phase. In the topic of indoor localization that uses CSI phase information, this algorithm is used by several research groups, such as [37] and [38].

The measured phase of a data file is shown in figure 3.9 In this data file, there are 203 packets.

Also, we selected a data packet and plotted the original phase of the data packet, as shown in figure 3.10.

The measured phase is first compensated for multiple 2’s by judging whether the measured phase change between the adjacent subcarriers is greater than the given threshold (we used as the threshold). The phase after unwrapping is shown in figure 3.11.
Algorithm 1: Phase Calibration

1. **Input:** measured phase values $M_P$ of 30 subcarriers;
2. **Output:** calibrated phase values $C_P$ of 30 subcarriers;
3. Set $T_P$ as a vector as the same size of $M_P$;
4. Set $m$ as a vector from -28 to 28;
5. Set $\text{diff} = 0$;
6. Set $\eta = \pi$;
7. Set $T_P(1)=M_P(1)$;
8. for $i = 2 : 30$ do
   9.     if $M_P(i) - M_P(i - 1) > \eta$ then
      10.        $\text{diff} = \text{diff} + 1$;
      11.    end
   12.   $T_P(i) = M_P(i) - \text{diff} \times 2 \times \pi$;
13. end
14. Compute $k = \frac{T_P(30) - T_P(1)}{m(30) - m(1)}$;
15. Compute $b = \text{sum}[T_P]/30$;
16. for $i = 1 : 30$ do
17.     $C_P(i) = T_P(i) - k \times m(i) - b$;
18. end

Figure 3.8: Phase Calibration Algorithm
Figure 3.9: Original Phase
Figure 3.10: Original Phase - one packet
Figure 3.11: Unwrapped Phase
Also, we selected a data packet and plotted the phase after unwrapping of the data packet, as shown in figure 3.12.

![Unwrapped Phase - one packet](image)

**Figure 3.12: Unwrapped Phase - one packet**

Then, a linear transformation was implemented to remove the time lag due to SFO, and the unknown phase offset due to CFO. The phase after linear transformation is shown in figure 3.13.

Also, we selected a data packet and plotted the phase after linear transformation of the data packet, as shown in figure 3.14.

The phase of high arm wave is shown in figure 3.15 and figure 3.16.

The phase of hand clapping is shown in figure 3.17 and figure 3.18.
Figure 3.13: Phase After Linear Transformation
Figure 3.14: Phase After Linear Transformation - one packet
CHAPTER 3. OUR WORKS

Figure 3.15: high arm wave phase

Figure 3.16: high arm wave phase - one packet
Figure 3.17: hand clapping phase
3.1.2 SVM Algorithm

We first tried SVM algorithm [39]. In this experiment, we removed outliers based on the RSSI values. For a package, if its RSSI values are outliers in all three antenna, then we treat this package as an outlier and therefore deleted this package. Then, we applied a low pass filter to remove noise on the CSI values. We also smoothed the CSI values using the weight moving average [40]. After the above processing, we performed segmentation to delete the time periods without gesture action. We found that at the beginning and end of the gesture, there will be obviously change points in the CSI array. Therefore, we first deleted all packages received in the first and last second, then found the first and last change points, and only kept the packages between the two points.

Also, in order to reduce the dimension of CSI array, we performed k-means to divide the 30 subcarriers into 3 groups, and only kept the group with the highest variance.
CHAPTER 3. OUR WORKS

Figure 3.19: CSI - Preprocessing

Figure 3.20: RSSI - Preprocessing
After the above data processing, seven features were extracted from CSI array: variance of each subcarrier, signal upper envelope of each subcarrier, signal lower envelope of each subcarrier, signal entropy of each subcarrier, median absolute deviation of each subcarrier, duration of motion and standard deviation of each subcarrier. Three features were extracted from RSSI array: moving variance, number of peaks, location of peaks. The feature selection refers to [23] and [].

After the above feature extraction, the dimension of the features, except for the duration of motion and the number of peaks, was reduced by calculating seven statistical values over the subcarrier dimension. These seven statistical values are min, 25th percentile, medium, 75th percentile, max, mean and standard deviation. After the dimension reduction, the CSI feature of each antenna become a 43 * 1 vector, and the RSSI feature of each antenna become a 15 * 1 vector. Then three CSI feature vectors and three RSSI feature vectors into one 172 * 1 vector, which equals to 42*3 + 1 + 15*3. The 1 here is the period of time.

The feature vectors from all 520 files were combined together to form a feature matrix of size 520 * 172. Then the feature matrix was standardized by column, so that each feature had a center 0 and a standard deviation 1.

The feature matrix were randomly divided into two parts. 75% of the data was used as the training dataset, and 25% of the data was used as the testing dataset. By applying SVM model, as shown in figure 3.21, an accuracy of 85.38% was achieved. And the precision was 84.38%, the recall was 85.71%.

In order to explore whether the phase information will help with the SVM classifier, we trained another SVM model with phase information added. After calibrating the CSI phase, the same 7 features as the CSI Amplitude are extracted from the CSI phase, and then combined with the CSI amplitude feature and RSSI value feature to train the SVM model. 75% of the data was used as the training dataset, and 25% of the data was used as
Figure 3.21: SVM Result
the testing dataset. By applying SVM model, as shown in figure 3.22, after 10 runs, an average accuracy of 83.33% was achieved. And the average precision was 88.86%, the average recall was 79.46%.

![SVM Model With and Without Phase](image)

**Figure 3.22: SVM Model With and Without Phase**

### 3.1.3 Random Forest Algorithm

Next, we tried Random Forest algorithm. After the same data processing, feature extraction and data standardization procedure as we did in the SVM algorithm section, we did one more step to further reduce the dimension of feature matrix. That is, building a random forest and estimate predictor importance values by permuting out-of-bag observations among the trees. Figure 3.23 shows the average importance of each feature after 10 runs.
By calculating the Coefficient of Determination (R Squared) of the reduced model, we found that selecting all features with importance estimate higher than 0.3 could reduce the number of features from 172 to 25, and could lead to the highest R2 value of 0.78, which is slightly higher than the R2 of 0.74 for the full model. This result suggest that the reduced model is sufficient for prediction. Therefore, we select only the features with importance estimates higher than 0.3 in the training dataset as a new training dataset, and select only the features with importance estimates higher than 0.3 in the testing dataset as a new testing dataset. And then we used these two dataset to train a new random forest model and make predictions. We achieved an average 91.54% accuracy, a precision of 96.72%, and a recall of 86.76%.

We also explored how phase information would help with the random forest model. The procedure is, after adding phase values, we measured the importance of features as shown in figure 3.25. We found that selecting all features with importance higher than 0.3 can lead to the highest R-squared. And the number of features was reduced from 298 (172 + 42*3) to 23. The selected features are from: RSSI’s moving variance, number of peaks. CSI amplitude’s entropy, upper and lower bound. CSI phase’s median absolute deviation.

By using these selecting features, we trained a new random forest model and make pre-
CHAPTER 3. OUR WORKS

Figure 3.24: RF Result

Figure 3.25: Predictor Importance Estimates - Phase Added
dictions. As shown in figure 3.26, we achieved an average 94.87% accuracy, a precision of 94.80%, and a recall of 94.75%.

![Figure 3.26: RF Model With and Without Phase](image)

3.1.4 K-Nearest Neighbors Algorithm

Next, we tried K-Nearest Neighbors algorithm. After the same data processing, feature extraction and data standardization procedure as we did in the SVM algorithm section, we did one more step to explore the k values. That is to try the k values from 1 to 31. For each k value, we used the 4-fold cross-validation method to split the data and calculate the average accuracy. From the above figure, we know that choosing k value of 3 can get the highest accuracy. Therefore, we fed the training dataset into a KNN classifier whose k = 3, and performed prediction. We achieved a 89.23% accuracy, a precision of 88.24%,
and a recall of 90.91%.

We also add phase information to the feature matrix, and built a new KNN model. We found that in the data set with phase added, the use of selected features can make the KNN model have better performance, as shown in figure 3.29. Therefore, we use the selected features and K=9 to build the KNN model.

From this KNN model, as shown in figure 3.30, we achieved an average 83.85% accuracy, a precision of 84.29%, and a recall of 85.51%.

### 3.1.5 Long Short-Term Memory

The last algorithm I tried is LSTM, which is long short-term memory. Because LSTM can process an entire sequence of data, we only did outlier detection and noise removal before inputting the data into LSTM, and did not do segmentation and feature extraction.
Figure 3.28: KNN Result

Figure 3.29: KNN-k values (phase added)
Figure 3.30: KNN Model With and Without Phase
• Sequence Layer: It takes as input prepossessed CSI data, i.e. the feature vector is a 93 dimensional vector which contains the raw CSI amplitude of each of the 90 subcarriers and raw RSSI amplitude of each of the 3 antennas.

• LSTM layer: 2 layer TensorFlow BasicLSTMCell with 50 nodes.

• Softmax layer: It normalizes and prepares data for classification known also as multi-class generalization.

• Classification layer: It applies cross entropy to classify and give the final output. For minimizing the cross entropy loss, Adam Optimizer is used with a batch size of 20 and a learning rate set to ten to the power of minus 3. [41]

From this model, We achieved an average accuracy of 76.92%, a precision of 77.42%, and a recall of 76.52%. When adding phase information, the accuracy was 60.58%, the precision was 63.15%, and the recall was 60.58%.

3.1.6 Summary

As our goal is to keep the accuracy high and maximize the precision, we can draw a conclusion that the random forest model outperformed other models. And we know that In CSI amplitude data, the most important features were the signal entropy, upper envelope, and lower envelope. In CSI phase data, the most important feature is the median absolute deviation. In RSSI data, the most important features were the moving variance and number of peaks. In addition, as shown in figure 3.32, we found that the phase information helped improved the accuracy and recall in RF model, but not very helpful in other models.
CHAPTER 3. OUR WORKS

**Figure 3.31: LSTM Result**

- Accuracy: 76.92% w/o CSI phase, 77.42% w/ CSI phase
- Precision: 60.58% w/o CSI phase, 63.15% w/ CSI phase
- Recall: 76.52% w/o CSI phase, 60.58% w/ CSI phase

**Figure 3.32: Gesture Recognition Result**

Average Statistics After 10 Runs
3.2 Reflector Angle Recognition

After the research on the dynamic human gesture dataset, we were wondering whether the CSI data could be used to monitor static object status [42]. If so, by connecting these two methods together, we can design a powerful system to perform door and window monitoring, human fall detection, and even bridge structure and traffic flow monitoring.

3.2.1 Data Collection

15 Degree Angle

The data was collected at a garage. As shown in figure 3, the angle between the reflector and the transmitter/receiver is 15 degrees. The access point is located at a height of 2 feet 1.5 inches, and the mobile station is at a height of 2 feet 2.5 inches. The reflector is a wooden board covered by metal skin. There are 3 data files collected in this experiment. The first data set includes data when the garage is empty, and when the reflector is present. The reason is that CSI data can be affected by the environment, so when the garage is empty, the data will provide us information about the environment, and then we can use this information as a baseline. The second and the third dataset only includes when the reflector is present.

The first data file was shown in figure 5 and figure 6. In this data file, The size of the CSI array is 44,991*90, and the size of the RSSI array is 44,991*3. When the index is approximately $2.4 \times 10^4$, the reflector was placed in the garage. It can be seen that the CSI and RSSI values have changed obviously after that.

In order to be close to the real usage scenario, we added some noise to the dataset, which is to let people and cars get in and out of the garage when collecting the data.
Figure 3.33: Experiment Setting - A15
CHAPTER 3. OUR WORKS

Figure 3.34: Picture of the Reflector

Figure 3.35: A15 - CSI Amplitude in the First Data file
Figure 3.36: A15 - CSI Phase in one of the packets
Figure 3.37: A15 - RSSI Array in the First Data File
30 Degree Angle

As shown in [3.38], the angle between the reflector and the transmitter/receiver is 30 degrees. There are also 3 data files collected in this experiment. The first data file was shown in [3.39] and [3.40].

![Figure 3.38: A30 - Experiment Setting]

45 Degree Angle

As shown in [3.41], the angle between the reflector and the transmitter/receiver is 45 degrees. There are also 3 data files collected in this experiment. The first data file was shown in [3.42] and [3.43].
Figure 3.39: A30 - CSI Amplitude in the First Data file
Figure 3.40: A30 - RSSI Array in the First Data File
Figure 3.41: A45 - Experiment Setting
Figure 3.42: A45 - CSI Amplitude in the First Data file
Figure 3.43: A45 - RSSI Array in the First Data File
60 Degree Angle

As shown in 3.44, the angle between the reflector and the transmitter/receiver is 60 degrees. There are also 3 data files collected in this experiment. The first data file was shown in 3.45 and 3.47.

![Figure 3.44: A60 - Experiment Setting](image)

90 Degree Angle

As shown in 3.48, the angle between the reflector and the transmitter/receiver is 90 degrees. There are also 3 data files collected in this experiment. The first data file was shown in 3.49 and 3.51.
Figure 3.45: A60 - CSI Amplitude in the First Data file
Figure 3.46: A60 - CSI Phase in one of the packets
Figure 3.47: A60 - RSSI Array in the First Data File
Figure 3.48: A90 - Experiment Setting
Figure 3.49: A90 - CSI Amplitude in the First Data file
Figure 3.50: A90 - CSI Phase in one of the packets
Figure 3.51: A90 - RSSI Array in the First Data File
3.2.2 Findings

In this section, we will perform binary classification between A15 and the other angles one by one using CSI amplitude and RSSI values.

After collecting the data files, we first divide the data into two parts, one when the garage is empty, and the other when there is a reflector. When the garage is empty, we use the middle value of the CSI amplitude and the middle value of the RSSI values as the baselines. Then when the reflector exists, we subtract the baseline values from the CSI amplitudes and RSSI values at this time to reduce the influence of the environment on the data. Also, we performed outliers detection, noise removal and data standardization on the data.

A15 vs A90

As shown in 3.52, we achieved a 100% accuracy when performing binary classification between A15 and A90 by using decision tree algorithm with the Gini’s diversity index as the Split criterion. Feature 31 is the 31st subcarrier among the 90 subcarriers in the CSI amplitude array. This may be due to multipath effects and other reasons. When the angle of the reflector is 15 degrees, the normalized CSI signal amplitudes of the 31st subcarrier are all lower than -0.2dB (this dB is not referenced to 1 milliwatt, but refers to Intel’s internal reference level). When the angle is 90 degrees, the normalized CSI signal amplitudes of the 31st subcarrier are all greater than -0.2dB.
Figure 3.52: Binary Classification Between A15 and A90
A15 vs A60

As shown in 3.53, we also achieved a 100% accuracy when performing binary classification between A15 and A60. Feature 37 is the 37th subcarrier among the 90 subcarriers in the CSI amplitude array. When the angle of the reflector is 15 degrees, the normalized CSI signal amplitudes of the 37th amplitude subcarrier are all lower than -0.53dB. When the angle is 60 degrees, the normalized CSI signal amplitudes of the 37th subcarrier are all greater than -0.53dB.

Figure 3.53: Binary Classification Between A15 and A60
A15 vs A45

As shown in 3.54, we also achieved a 100% accuracy when performing binary classification between A15 and A45. Feature 34 is the 34th subcarrier among the 90 subcarriers in the CSI amplitude array. When the angle of the reflector is 15 degrees, the normalized CSI signal amplitudes of the 34th subcarrier are all lower than -0.21dB. When the angle is 45 degrees, the normalized CSI signal amplitudes of the 34th subcarrier are all greater than -0.21dB.
A15 vs A30

As shown in Figure 3.55, we also achieved a 100% accuracy when performing binary classification between A15 and A30. However, since the difference between 15 degrees and 30 degrees are smaller than the previous pairs, the layer of the decision tree increases. Feature 59, 87, 6 are the 59th, 87th, 6th subcarrier among the 90 subcarriers in the CSI amplitude array.

![Figure 3.55: Binary Classification Between A15 and A30](image)

Multi Classification Between All Angles

After comparing the angles between pairs, in order to get close to reality, we also performed a multi-class classification between all the angles using CSI amplitudes and RSSI values. In this way, when we observed a new data point, by using a multi-class classifier, we would know which angle the new observation point is closer to. We would be using a decision tree to build a classifier. And we would be using the Gini score as the
classification criteria. The Gini score indicates the probability of misclassification

\[ Gini = 1 - \sum_j p_j^2 \]

, where \( p_j \) is the probability of class \( j \). A gini score greater than zero implies that samples contained within that node belong to different classes. A gini score of zero means that the node is pure, that within that node only a single class of samples exist. Therefore, our goal is to lower the Gini score as much as possible. We fed the data into a decision tree. The value of sample represents how many samples will be classified at this node. The value list tells us how many samples at the given node fall into each category. The first element of the list shows the number of samples that belong to the 15 degree class, the second element of the list shows the number of samples that belong to the 30 degree class, and the third element in the list shows the number of samples that belong to the 45 degree class, so on so forth. The class value shows the prediction a given node will make and it can be determined from the value list. Whichever class occurs the most within the node will be selected as the class value. After training, we obtained a 6 layer tree with Gini scores of all leaves of 0.0. Also, after 10 runs, we achieved an average of 99.998% prediction accuracy, which is, for about 123,481 out of 123,483 data points, the decision tree made the correct prediction. In 10 runs, the decision tree of one run is drawn as shown in figure 3.56, all the attributes of the tree comes from CSI amplitude data instead of RSSI data, which aligns with our observation in last subsection that the CSI amplitude data is more sensitive to changes in object angles than RSSI data.

We trained another decision tree model with CSI phase information added. So the feature matrix now includes CSI amplitude (label as ‘CSI Feature 1-90’), CSI phase (label as ‘CSI Phase 1-90’), and RSSI (label as ‘RSSI 1-3’). As shown in figure 3.57, the node parameters on the tree all come from the CSI amplitude, so it seems that the CSI phase information and RSSI information have little effect on classification.

In order to further confirm that CSI phase and RSSI have little effect on classification, we
CHAPTER 3. OUR WORKS

Figure 3.56: Multi Classification Between All Angles

Figure 3.57: Multi Classification Between All Angles – Phase Added
need to exclude the influence of feature order on the selection of features by the classifier. Because there may be two features with the same Gini value, but the classifier selects the top one, which reduces the chance of the feature with the bottom being selected. In this model, we arranged the order of the features as [CSI Phase, RSSI, CSI Amplitude]. The CSI amplitude was moved to the end of the feature list. However, as shown in figure 3.58, the node parameters still all come from the CSI amplitude. Therefore, we can conclude that CSI phase and RSSI are not as useful as CSI amplitude in this classification.

![Figure 3.58: Multi Classification Between All Angles– Phase Added – Feature Order Adjusted](image)

In order to determine which subcarriers played the most important role in angle detection, we counted the feature appearances from the three decision trees above, as shown in figure 3.59. We found that most of the features come from antenna 2, and the role of the subcarriers in the middle is more important than the role of the subcarriers at both ends.

We select features that appeared no less than 2 times and use these 7 features to train a new decision tree, as shown in figure 3.60. Feature 37 represents subcarrier 7 from antenna 2, feature 50 represents subcarrier 20 from antenna 2, feature 80 represents subcarrier 20.
Figure 3.59: Feature Occurrence

from antenna 3, and so on. After 10 runs, we also achieved an average prediction accuracy of 99.99%, which is roughly equal to the accuracy when using all functions. Therefore, we can conclude that these 7 subcarriers are most sensitive to the angle change.

Figure 3.60: Multi Classification - Selected Features
Summary

We achieved all 100% accuracy in the previous 4 pairs of binary classification and 99.99% accuracy in the multi-class classification between all angles. We found that the smaller the difference between angles, the more features are required to perform classification. We also found it interested that all the features that play a key role in classification come from CSI amplitude, not CSI phase or RSSI. This shows that CSI amplitude is more sensitive to changes in object angles than CSI phase and RSSI.
Chapter 4

Conclusions

To sum up, after our research, we found that CSI data can be used to perform object detection and gesture recognition, and by applying decision tree algorithm and random forest algorithm, we can achieve good accuracy in classification. With these two techniques, we can perform door monitoring and contact-less remote interaction without wearing any device, which will bring us a great, bright and convenient future home!
References


[7] Mohammed AA Al-qaness, Mohamed Abd Elaziz, Sunghwan Kim, Ahmed A Ewees, Aaqif Afzaal Abbasi, Yousif A Alhaj, and Ammar Hawbani. Channel state informa-


[14] Zhi-Ping Jiang, Wei Xi, Xiangyang Li, Shaojie Tang, Ji-Zhong Zhao, Jin-Song Han, Kun Zhao, Zhi Wang, and Bo Xiao. Communicating is crowdsourcing: Wi-fi indoor


detection of moving targets with dynamic speed using phy layer information. In 2014
20th IEEE international conference on parallel and distributed systems (ICPADS), pages

[34] Xing Zhihao Jiang Weipeng, Liu Yongjun. A kind of phase alignment and device,
CN201610552140.9A.

modity wifi devices. In IEEE INFOCOM 2017 - IEEE Conference on Computer Commu-
ications, pages 1–9, 2017.

[36] Xuyu Wang, Lingjun Gao, and Shiwen Mao. Csi phase fingerprinting for indoor loca-
larization with a deep learning approach. IEEE Internet of Things Journal, 3(6):1113–1123,
2016.

[37] Xiaochao Dang, Xiong Si, Zhanjun Hao, and Yaning Huang. A novel passive indoor
localization method by fusion csi amplitude and phase information. Sensors, 19(4):875,
2019.

localization with 5 ghz wi-fi. In 2017 IEEE International Conference on Communications
(ICC), pages 1–6, 2017.

[39] Xun Wang, Ke Sun, Ting Zhao, Wei Wang, and Qing Gu. Dynamic speed warping:
Similarity-based one-shot learning for device-free gesture signals. In IEEE INFOCOM

[40] Jay Prakash, Zhijian Yang, Yu-Lin Wei, Haitham Hassanieh, and Romit Roy Choud-
hury. Earsense: earphones as a teeth activity sensor. In Proceedings of the 26th Annual

[41] Neena Damodaran, Elis Haruni, Muyassar Kokhkharova, and Jörg Schäfer. Device free
human activity and fall recognition using wifi channel state information (csi). CCF