

2018

An Approach to Finding Parking Space Using the CSI-based WiFi Technology

Yunfan Zhang
South Dakota State University

Follow this and additional works at: <https://openprairie.sdstate.edu/etd>



Part of the [Computer Engineering Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Zhang, Yunfan, "An Approach to Finding Parking Space Using the CSI-based WiFi Technology" (2018). *Electronic Theses and Dissertations*. 2440.
<https://openprairie.sdstate.edu/etd/2440>

This Thesis - Open Access is brought to you for free and open access by Open PRAIRIE: Open Public Research Access Institutional Repository and Information Exchange. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Open PRAIRIE: Open Public Research Access Institutional Repository and Information Exchange. For more information, please contact michael.biondo@sdstate.edu.

AN APPROACH TO FINDING PARKING SPACE USING THE CSI-BASED WIFI
TECHNOLOGY

BY
YUNFAN ZHANG

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Computer Science

South Dakota State University

2018

AN APPROACH TO FINDING PARKING SPACE USING THE CSI-BASED WIFI

TECHNOLOGY

YUNFAN ZHANG

This thesis is approved as a credible and independent investigation by a candidate for the Master of Science degree and is acceptable for meeting the thesis requirements for this degree. Acceptance of this thesis does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Myounggyu Won, Ph.D.
Thesis Advisor

Date

Steven Hietpaš, Ph.D.
Head, Department of EECS

Date

Kimchel C. Doerner, Ph.D.
Dean, Graduate School

Date

ACKNOWLEDGMENTS

First of all, special thanks to my thesis advisor Dr. Won. He gave me a lot of technical and economic supports for my thesis. He always guided me to the right direction so that I can do my research in the right way. Many Thanks!

I would like to thanks to Dr. Shin and Dr. Liu. They gave me a lot constructive comments for my thesis during my preliminary presentation so that I can improve my thesis's quality. They also gave me some suggestions for my performance during the presentation. These suggestions will help me a lot in the future. And also, I would like to thanks to the Dr. Shaukat. He took time out of his busy schedule and gave me a lot help. Thanks!

I am also grateful to my parents. The moral and economic supports they gave me are the motive power that help me overcome the difficulties and achieve the objective of my study.

And finally, I would like to thanks to all the faculty members of Electrical Engineering and Computer Science department and South Dakota State University. They helped me a lot during the school days. Their assistances helped me solve a lot of my problems

Thank you all for your encouragement!

TABLE OF CONTENTS

LIST OF FIGURES	vii
ABBREVIATIONS	ix
ABSTRACT	x
Chapters	1
1 Introduction	1
2 Related Work	5
2.1 Parking Solutions	5
2.2 Environment Sensing Based on CSI	7
2.2.1 Human Body Detection	9
2.2.2 Indoor Localization	11
2.2.3 Activity Recognition	11
3 Preliminaries	14
3.1 Real-time LOS and NLOS Identification	14
3.2 Channel State Information	14
3.3 Channel Impulse Response	17
3.4 Multi-Input and Multi-Output Technology	19
3.5 Environment Sensing Technology Based on Fingerprint Information	20
4 System Design	22
4.1 Methodology	22
4.2 System Overview	25
4.3 CSI Collection	27

4.3.1	Hardware Configuration	27
4.3.2	Comparison Between Two Modes	28
4.4	Pre-processing of WiFi CSI Data	30
4.4.1	De-noising Processing	30
4.4.2	Phase Pre-processing	31
4.4.3	Processing Procedures	34
4.5	Feature Extraction	36
4.5.1	Extraction Principles	36
4.5.2	Phase Variance For Different Empty Parking Slots	37
4.5.3	Distribution patterns of phase variance	42
4.6	Training and Testing	44
4.6.1	Methodology	44
4.6.2	MATLAB-SVM Classifier	47
4.6.3	Multiple Classification Problems in LIBSVM	50
4.6.4	Setting the Training Data and Test Data	55
5	Experiments	57
5.1	Experimental Setup	57
5.2	Experimental Scheme	57
5.3	System Interface	60
6	Result Analysis	62
6.1	Classification Accuracy	62
6.2	Effects of Receiver Antenna	64

7 Conclusion 66

References 68

LIST OF FIGURES

3.1	Illustration of LOS and NLOS	15
3.2	The Component of WiFi CSI	15
3.3	Channel State Information.	17
3.4	Total Distortion at the Receiver by Multipath Effects	17
4.1	Phase shifts of CSI.	25
4.2	Illustration for Approach Overview	26
4.3	Illustration for Out-mode.	28
4.4	Illustration for In-mode.	29
4.5	Special case of Out-mode	30
4.6	A raw phase diagram.	32
4.7	After low-pass filter phase diagram.	32
4.8	Comparison between the raw phase and pre-processed phase.	33
4.9	The data of a <code>csi_trace</code>	34
4.10	Parts of codes of Using <code>get_scaled_csi</code> function	35
4.11	Codes of setting time	35
4.12	Codes of setting filters	36
4.13	Variances of phase for parking lots with 9 empty slots and 10 empty slots	38
4.14	Variances of phase for parking lots with 1 empty slots and 0 empty slots	39
4.15	The scatter graph of phase variances	40
4.16	The 3D scatter graph of phase variances	41

4.17	Phase variances of all cases	43
4.18	All case phase variances	43
4.19	CDF of phase variance for varying numbers of empty slots	44
4.20	The graph drawn by using SVM	49
4.21	The process of training and testing data with SVM	50
4.22	The Algorithm of Multiple Classifications	51
4.23	Illustration for classification using Linear SVM	52
4.24	Map 2D data to 3D data with Kernel function	54
4.25	Non-linear SVM	54
4.26	The Figure after using SVM	56
5.1	Snapshot of experimental setting at department parking lot	58
5.2	Transmitter Interface	59
5.3	Receiver Interface	59
5.4	System Interface	60
6.1	Confusion matrix that represents classification accuracy	63
6.2	Classification accuracy with and without a tolerance of one empty slot	64
6.3	Classification accuracy for different receiver antenna	65

ABBREVIATIONS

AP	Access Point
ANN	Artificial Neural Network
CSI	Channel State Information
CIR	Channel Impulse Response
CFR	Channel Frequency Response
DNN	Deep Neural Networks
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
KNN	K-Nearest Neighbor
LOS	Line-of-Sight
LNPM	Log Normal Propagation Model
MAC	Medium Access Control
MIMO	Multiple Input Multiple Output
NLOS	Non-Line-of-Sight
NIC	Network Interface Card
OFDM	Orthogonal Frequency Division Multiplexing
RSSI	Received Signal Strength Indicator
SVM	Support Vector Machine
WLAN	Wireless Local Area Network

ABSTRACT

AN APPROACH TO FINDING PARKING SPACE USING THE CSI-BASED WIFI
TECHNOLOGY

YUNFAN ZHANG

2018

With ever-increasing number of vehicles and shortages of parking spaces, parking has always been a very important issue in transportation. It is necessary to use advanced intelligent technologies to help drivers finding parking spaces quickly.

In this thesis, an approach to finding empty spaces in parking lots using the CSI-based WiFi technology is presented. First, the channel state information (CSI) of received WiFi signals is analyzed. The features of CSI data that are strongly correlated with the number of empty slots in parking lots are identified and extracted. A machine learning technique to perform multi-class classification that categorizes the input data into classes representing the number of empty slots is employed. A prototype system of the proposed approach is developed. Experiments are performed and it is shown that the system is feasible.

Compared with traditional approaches based on magnetic sensors deployed on individual parking slots, the proposed approach is non-intrusive as it does not require to install specialized devices in a parking lot, and is cost-effective since it utilizes either existing WiFi infrastructure or only a pair of WiFi devices. As a result, the average classification accuracy of system is 80.8%, and the accuracy is improved to 93.8% with a tolerance of one empty slot.

Chapter 1

Introduction

In recent years, the intelligent technology has been widely used in almost all aspects of transportation. For example, the intelligent vehicle scheduling system can greatly reduce traffic congestion and the electronic toll collection system can charge for passing vehicles without parking by using wireless sensor technology. As a part of transportation, parking has always been a very important issue. On the one hand, the number of vehicles has increased significantly, but there is no corresponding increase in parking space. As a result, it is very difficult to find a parking space in city centers and frequently-visited places. On the other hand, in large parking lots, it is also difficult for drivers to know exactly where there is a vacant parking space, which can easily cause drivers mental stress and affect the traffic and even the safety of pedestrians. It has been reported that 30% of city traffic is due to the drivers cruising to find parking spaces. In addition, cruising to find parking spaces wastes drivers time and gas, which lead to economic loss [1]. Therefore, it is necessary to use various advanced intelligent technologies to help vehicle drivers to find parking spaces accurately and quickly.

Considering the importance of the parking issue, the problem on finding parking space has attracted much attention. A number of parking solutions have been proposed. Most approaches utilize magnetic sensors to monitor parking occupancy [3, 4, 5, 6]. However, these systems are required to install sensors on the individual parking slot in order to detect available parking spaces and guide drivers

to areas with vacancies. Some solutions adopted a camera to improve the accuracy of the sensor-based approaches [7]. A major issue of these sensor-based solutions is, however, the high cost to purchase, install, and maintain the sensors and associated backend systems. Smart solutions have been developed that do not rely on deployment of a large number of sensors [8, 9, 10]. Parknet utilizes an ultrasonic rangefinder that is attached to a car to detect the parking occupancy [8]. Other driver oriented solutions [9, 10] exploited the smartphone sensors to detect empty parking slots. However, the success of these solutions is substantially dependent on driver participation since it is an individual driver (*i.e.*, their smartphones) who notifies other drivers of whether he or she has left the parking lot.

With the increasingly mature and widespread deployment of WiFi, researchers have been seeking to use WiFi signals for environment sensing beyond a basic communication service. Especially with the availability of channel state information (CSI) in common commercial devices, CSI-based environment sensing is studied and some pioneer works have been proposed. Compared with received signal strength, CSI is a finer-grained descriptor of the wireless channel, which can discriminate multipath components, thus holding the potential for more robust and reliable environment sensing. Utilizing the characteristics of the CSI-based environment sensing, the CSI-based WiFi technology can provide a feasible solution to address the issue of finding empty space at parking lots.

In this thesis, we present an approach to finding empty spaces in parking lots using the CSI-based WiFi technology. This thesis analyzes the channel state

information (CSI) of received WiFi signals, derives the features of CSI data that are strongly correlated with the number of empty slots in a parking lot, and adopts a machine learning technique to perform multi-class classification that categorizes the input data into classes representing the number of empty slots. We find the correlation between the phase variance of subcarriers of a WiFi signal and the number of empty slots, which is caused by the cars obstructing the line of sight (LOS) path between the transmitter and the receiver, generating higher refractions, diffractions, and reflections which lead to greater randomness in the phase of received signals [11]. The experiment shows that it is feasible to use CSI-based WiFi technology to find the parking space.

In terms of cost, the solution using CSI-based WiFi technology is more economical. Most indoor parking areas have set up WiFi infrastructure that supports CSI data collection. Even though for parking lots with no WiFi infrastructure, with a pair of COTS WiFi devices such as a laptop or any other system, *e.g.*, Raspberry Pi equipped with a WiFi network interface card, it can be easily implemented. Although depending on the size of a parking lot, WiFi range extenders, or additional WiFi devices may be employed, the cost is still very low. For example, considering the cost of \$200 for a transmitter, and about \$100 for a WiFi access point, if we need to deploy 5 pairs of WiFi devices to cover the entire parking lot, the cost will be less than \$1,500. In the Parknet system [8], each parking space needs one ultrasonic sensor, and each sensor is worth \$400. Suppose 10 sensors are required, it will cost \$4,000, which is much more expensive than the system we designed.

The proposed approach is non-intrusive as it does not require to install specialized devices in a parking lot, and is cost-effective since it utilizes either existing WiFi infrastructure or only a pair of WiFi devices.

To sum up, the contributions of this thesis are summarized as follows.

- The design of a unique WiFi-based system finding parking space is proposed.
- A useful feature that is strongly correlated with the number of empty slots is identified for effective classification of the number of empty slots.
- A prototype system of the proposed approach is developed.
- Experiments are performed to validate the performance of the proposed approach, and the classification accuracy is analyzed.

Chapter 2

Related Work

2.1 Parking Solutions

In this section, the literature specially focusing on parking solutions is reviewed. Urban street-parking would greatly benefit society by reducing traffic congestion. In the early parking solutions, most approaches utilized magnetic sensors to monitor parking occupancy [7, 3, 4, 5]. The key idea is to detect magnetic field changes when a parking vehicle changes its position. At the same time, advanced signal processing algorithms or other tools were developed to improve the accuracy of magnetic sensors [6]. For example, in Streeline Company a camera is utilized to compensate for the errors in magnetic sensor data [7]. These magnetic-sensor-based solutions, however, involve high cost, and they are inherently intrusive solutions since installation of sensors equipped with wireless communication modules at each parking slot as well as the backend server is required for analysis of the sensor data.

There have been efforts for developing solutions that do not rely on deployment of a large number of sensors throughout a parking lot [21, 8]. Some solutions develop specialized equipment for monitoring parking vacancies. For example, Mathur *et al.* attached an ultrasonic range finder with a GPS module to a car to determine the spot occupancy [8]. Collected data were aggregated to build a map of parking availability [8]. Other solutions utilized smartphone sensors. ParkSense is a smartphone app that is specially designed for sensing the event that a

driver has vacated a parking spot [12]. However, without sufficiently high participation of drivers, the effectiveness of parking occupancy monitoring is significantly decreased. Soubam *et al.* proposed a similar approach that detected the “parking” and “unparking” event based on identification of the user “walking” and “driving” states using smart phone sensors [9]. Carnelli *et al.* developed a sensor fusion method to detect the parking activity event using the accelerometer and magnetometer of a smartphone [10]. Kriet *et al.* solved the same problem and used all available sensors of a smartphone to maximize the event detection accuracy at the cost of reduced battery life [13]. In addition, there are other approaches beyond utilization of smartphone sensors. Crowd sourcing techniques have been proposed to gauge the parking lot occupancy [14, 15]. Smartphone sensor data are collected from a crowd of drivers, and parking recommendations are generated to arriving drivers [24]. Huang *et al.* improved the precision of parking solutions by applying machine learning techniques [16]. Other works were focused on finding an effective trajectory that improves the chance of reaching at an empty parking slot [17, 18].

Unlike the magnetic-based approaches as mentioned above, the approach presented in this thesis is cost-effective and non-intrusive as it does not demand for purchase and installation of specialized equipment in a parking lot. In addition, compared with the smartphone-based and/or crowdsourcing-based methods, the success of this approach is not dependent on the driver participation rates, and it does not mandate drivers to install an app for parking lot monitoring. This approach is in line of recent research trends that leverage the strength of machine learning

techniques in that it fusions the wireless communication technology with a machine learning algorithm.

2.2 Environment Sensing Based on CSI

The wireless LAN based on Wi-Fi technology has been widely used in the room and the services. In addition, it is not limited to information communication. Many applications such as human detection, indoor localization, through wall tomography, activity recognition, are emerging in the field of wireless local area network applications. In the indoor environment, transmitted wireless signals do not usually reach the receiver along a direct path. In fact, the received signal is usually the superposition of signals from the furniture, human body, and other obstacles, and is diffracted and scattered. This phenomenon is known as the multipath effect. Physical space limits the spread of radio signals. In contrast, the wireless signals can also be used to sense the physical environment in which they pass. Whether it is an environmental object (such as a wall, furniture) or a human body, its location and movement can modulate the wireless signal. Thus a periodic signal or variable signal is formed. Based on the signal analysis, it can infer perception of the environment.

In fact, the wireless signal environment perception is not novel. The received signal strength (Received Signal, Strength, RSS) has been used as multiple received signal average value that can be obtained directly by most existing wireless equipment. It has been widely used in various sensing applications in the past 20 years. In general, RSS is used to sense channel quality and its coarse granularity and

high volatility make it unsuitable for the accurate perception of multipath rich indoor environments, such as the indoor location and human body detection. Experimental results show that in typical indoor environments, the RSS received by a stationary receiver may exhibit a fluctuation of 5 dB [19]. For making up the unreliability of RSS, Wilson and Patwa proposed a RTI (Radio Tomography Imaging, RTI) concept, which deploys the sensor networks in the target area and uses the dense deployment of sensor nodes to reduce the multipath effect [20].

Since RSS has only the feature of Medium Access Control (MAC) layer, the latest technology research has begun to shift to more fine-grained wireless channel features. Traditionally, to implement accurate measurement of the wireless channel usually require specialized equipment such as vector analyzer or software radio, which limits its wide application [21, 22, 23]. In recent years, with application of the orthogonal frequency division multiplexing technology (Orthogonal, Frequency, Division, Multiplexing, OFDM) in the wireless LAN standards, the sampling of the spectrum of signal can be easily obtained at present. Specifically, with existing Intel 5300 wireless network card and the modified driver program, the sampled channel frequency response (Channel Frequency Response, CFR) can be obtained in the form of the channel state information (Channel State, Information, CSI) [24].

Compared with the RSS, the CSI can distinguish multipath components to a certain extent. Conceptually, CSI and RSS just like the relationship between rainbow and white light. In the CSI, the signal components at different wavelengths are separated [25]. Compared with RSS, the main advantage of CSI is the channel

information on each subcarrier can be estimated. The characteristics of frequency selective fading of WiFi channel can be described. In addition, the CSI contains the amplitude and phase information of each subcarrier. Thus, it can provide more rich frequency domain information. As a corresponding signal of RSS in physical layer, CSI is expected to achieve more accurate and reliable environment perception, and has attracted more and more attention of researchers in the past 5 years. Since 2010, researchers from Massachusetts Institute of Technology, Stanford University, University of Washington, Hong Kong University Science Technology, Tsinghua University, *etc.* have carried out a lot of relevant research on CSI, and attempt to develop a context sensing application system based on CSI.

According to the different ways of using CSI, this thesis divides the existing work into four categories: Human body detection, Indoor localization, Sight path identification and Activity recognition.

2.2.1 Human Body Detection

Human body detection refers to the process of detecting the presence of an individual in the target area, which is critical to many applications. These include security warning, intrusion detection, border detection, and detection of survivors in fires or earthquakes. In order to obtain higher detection rate and anti-narrowband interference, Xiao *et al.* implemented a device independent indoor motion detection system (Fine-grained device free motion detection, FIMD) [25]. FIMD can extract better temporal stability from CSI and achieve higher detection rates. FIMD

constructs a K dimension vector of CFR amplitude histogram as the fingerprint and uses the Earth Mover's Distance (EMD) to classify the fingerprints [26]. In a large number of studies on passive detection, most existing methods used only the amplitude information of CSI without studying phase information. Therefore, Qian *et al.* proposed a PADS system for passive human detection using all the information (amplitude and phase) of CSI [27]. Because the original phase information obtained by using commercial equipment is randomly distributed in the feasible region, PADS firstly makes linear transformations for the original CSI to reduce the influences of random noise. Thus, it obtains the useful phase information. Then it uses the outlier filtering method to remove biased observations. Because of using the full information of CSI, especially the sensitive phase features, the experimental results show that PADS can detect moving human bodies with different walking speeds, and the performance is superior to the traditional system based on RSS/CSI. Based on the deep research on the signal fading caused by human motion, Wu *et al.* proposed a unified scheme, DeMan, for non invasive detection of mobile and stationary human bodies using commercial WiFi devices [23]. DeMan uses both CSI amplitude and phase information to detect moving individuals and considers human breathing as an innate indicator of stationary human bodies. Specifically, DeMan calculates the correlation coefficients matrices of CSI amplitude and phase respectively and extracts their respective maximum feature values to form two-dimensional features for inferring the existence of moving human beings

2.2.2 Indoor Localization

In recent years, the demand for indoor positioning grows with each passing day, which prompts many researchers dedicate to study on indoor positioning and many achievements have been obtained. Xiao *et al.* studied the feasibility of implementing passive indoor location using fine-grained CSI and designed Pilot system [29]. Since CSI reflects the frequency diversity characteristic of broadband channels, Pilot uniquely identifies the user's location by monitoring the shift of the CSI characteristic mode. Pilot uses the probabilistic algorithm to match the detected abnormal CSI with the fingerprint database to estimate the location of the user. Abdel-Nasser *et al.* proposed an accurate WLAN location system called MonoPHY, which is based on single stream and device-independent [30]. MonoPHY mainly contains two operation stages: offline training stage and online positioning stage. In the online training stage, MonoPHY records the user's CSI at different locations. In addition, Sabek *et al.* proposed a WLAN location system called MonoStream based on a device-independent and single stream [31]. MonoStream modeled the passive location problem as an object recognition problem. It regards the CSI files in different locations as different images and extracts features that can reflect the location of users

2.2.3 Activity Recognition

Recognizing people's indoor activities is not just the location of people, and it is more practical to be used other tasks, such as health monitoring and health

management. By utilizing the widely deployed WiFi infrastructure and fine grained CSI, Wang *et al.* designed a device-independent and location oriented activity identification system called E-eyes [34]. Since a number of important indoor activities occur at specific locations such as cooking in the kitchen, dining at table, etc., E-eyes only needs to collect CSI files for specific activities that take place at these locations. The basic idea of E-eyes is to match the acquired CSI features with the existing active files. The core modules include the activity identification module, the file construction module, and update module. Specifically, E-eyes first uses mobile variance threshold technique to distinguish walking and standing activities, and segment the CSI trajectory by using mobile variance, and finally recognize activities by computing the similarity of each CSI segment and the pre-constructed activities documents.

Han *et al.* designed a device-independent passive fall detection system WiFall, which is based on time stability and frequency diversity of CSI [35]. Since the static human body does not cause temporal changes of wireless signals, WiFall uses anomaly detection algorithms based on LOF (Local Outlier Factor, LOF) to detect human motion and isolate the corresponding abnormal patterns. Since the anomaly detection module labels various human movements such as walking, standing and falling, WiFall uses single class support vector machines(SVM) to classify the features in the abnormal modules. Xi *et al.* proposed the use of CSI for population statistics, and designed a device independent population count system FCC [36]. The concept of FCC comes from the high sensitivity of CSI to environmental change. The main

problem that the FCC needs to solve is to find metrics that can accurately describe the monotonic relationship between CSI changes and the number of moving people.

Chapter 3

Preliminaries

3.1 Real-time LOS and NLOS Identification

Line-of-sight (LOS) propagation is a characteristic of electromagnetic radiation or acoustic wave propagation, which means waves which travel in a direct path from the source to the receiver. Electromagnetic transmission includes light emissions traveling in a straight line. The rays or waves may be diffracted, refracted, reflected, or absorbed by the atmosphere and obstructions with material and generally cannot travel over the horizon or behind obstacles.

When the transmitter and the receiver are due to furniture, walls and other obstacles no apparent path, the wireless signal reaches the receiving end after scattering, diffraction and reflection. Different paths have different time delays, phases and amplitudes. This phenomenon is called NLOS (Non-line-of-sight, NLOS). NLOS are radio transmissions across a path that is partially obstructed, usually by a physical object in the innermost Fresnel zone. Fig. 3.1 shows the difference between LOS and NLOS.

3.2 Channel State Information

Currently, the orthogonal frequency division multiplexing (OFDM) modulation scheme is employed in the physical layer of WiFi standards. According to OFDM, a high data-rate stream is divided into close-spaced subcarriers, which

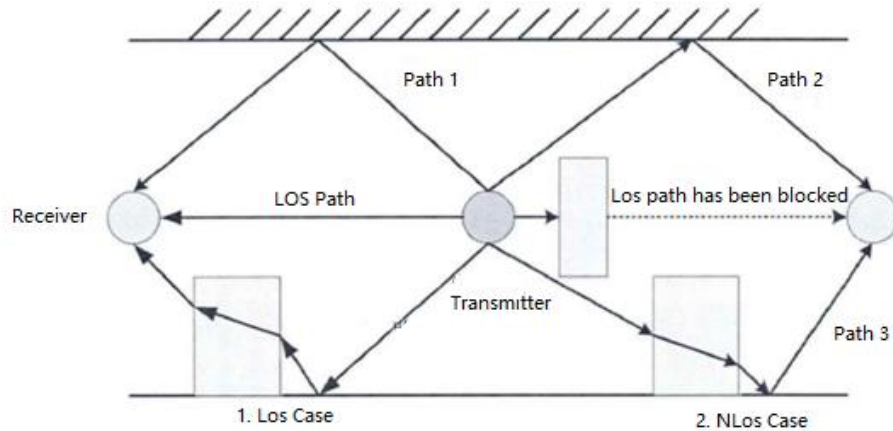


Figure 3.1: Illustration of LOS and NLOS

makes it robust against frequency selective fading. At present, most WiFi systems adopt Received Signal Strength Indicator (RSSI) and CSI as receiving signals. RSSI and CSI refer to the characterization of these OFDM subcarriers. CSI has been increasingly utilized for various applications, *e.g.*, human activity recognition [37, 38], localization [39], and traffic monitoring [40].

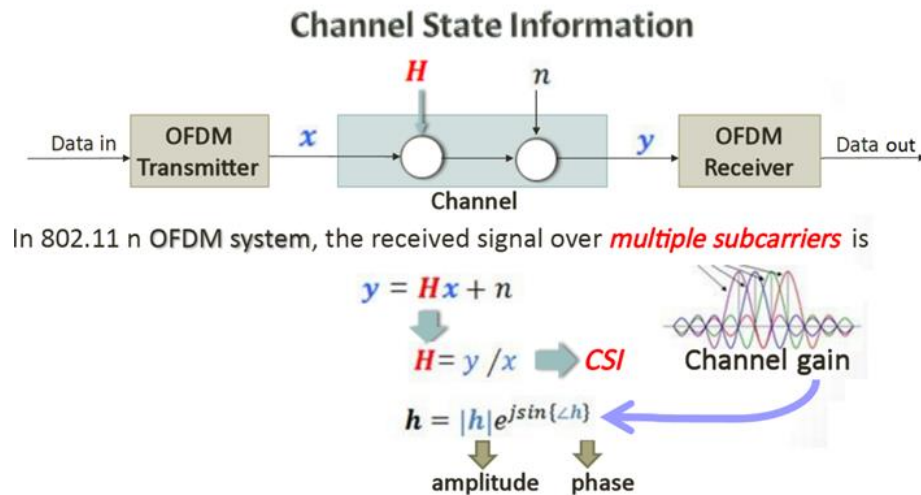


Figure 3.2: The Component of WiFi CSI

Recently, some common IEEE 802.11n standard commercial wireless cards

(such as Intel 5300) can provide detailed amplitude and phase information for different subcarriers in the form of CSI as shown in Fig. 3.2 [49] and Fig. 3.3. CSI is better than RSSI because it includes much more information. CSI represents the properties of the channel as follows [16]:

$$y = H \cdot x + n \quad (3.1)$$

Here x is the transmitted signal; y is the received signal; n is the channel noise; and H is a matrix that represents the frequency responses for subcarriers of all streams. For example, given M receiver antennas, N transmitter antennas, and W subcarriers, the dimension of matrix H would be $M \times N \times W$. Matrix H can be viewed as a vector of subcarrier groups:

$$H = [H_1, H_2, \dots, H_w] \quad (3.2)$$

The $i - th$ subcarrier group H_i contains a set of channel state information for the $i - th$ subcarrier signals received via different transmitter-receiver antenna pairs.

Let's denote the CSI of the $i - th$ subcarrier received via the receiver antenna m and the transmitter antenna n by CSI_{mn}^i . CSI_{mn}^i can be defined as follows.

$$CSI_{mn}^i = |h|e^{j\phi} \quad (3.3)$$

Here $|h|$, and ϕ are the amplitude and the phase information of the $i - th$ subcarrier

signal between the receiver antenna m and the transmitter antenna n .

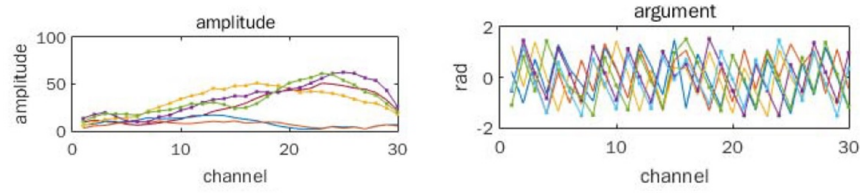


Figure 3.3: Channel State Information.

3.3 Channel Impulse Response

In the practical wireless channel, the wireless signal is disturbed by vehicles, walls, the other obstacles, and terrain material, which makes the signal occur reflection, diffraction and be scattered by the receiver so that there is a plurality of signal transmission paths between the transmitter and the receiver. Different paths of the signal have different time delays, attenuations and frequency dispersions as shown in Fig. 3.4. The total distortion generated at receiver by the superimposed signal is called as multipath effects.

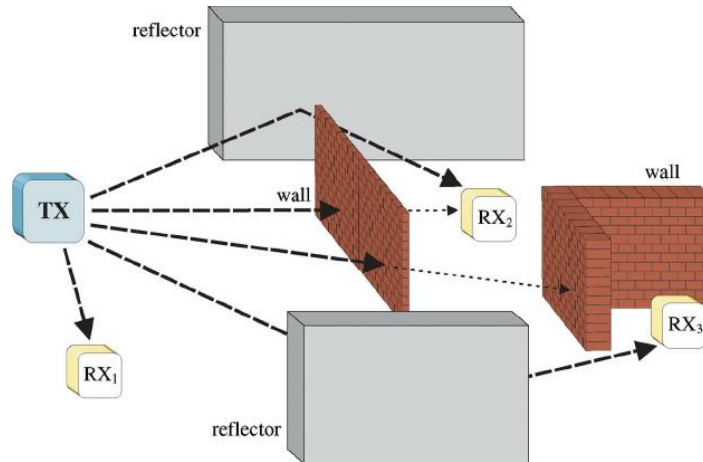


Figure 3.4: Total Distortion at the Receiver by Multipath Effects

Due to the time-varying of channel, the channel multipath effect will change with time delays. Hence, the channel can be described by time delays generated on all paths, frequency fading and Doppler frequency shift, which can be presented as the channel impulse response (Channel Impulse Response, CIR). In the indoor wireless communication, IEEE 802.11 wireless communication using burst transmission.

CFR is the frequency domain form of CIR. It is the ratio of the frequency spectrum of the received signal and the spectrum of transmitted signals, including the amplitude and phase frequency response. In the known spectrum of received signal and transmit signal conditions, it can be calculated to reflect the multipath characteristics of CFR. Then through an inverse Fourier Transform Inverse (IFT) the response of CIR can be calculated, where CSI is the sampling version of CIR/CFR in the specific protocol.

By the modification of Linux kernel, CSItools can make the common WiFi device which is equipped with Intel 5300 network adapter (Network Interface Card, NIC) can obtain a version with 30 CFR samples in the form of CSI. Use a wireless card which is compatible to IEEE802.11 a/g/n to extract a set of CSI from each receiving data packet. Each group CSI represents amplitudes and phases of an Orthogonal Frequency Division Multiplex (OFDM).

For CFR under bandwidth of WiFi, sampling N points will get a set of CSI information, which is the sampling interval between the different subcarrier frequencies. Intel5300 wireless network cards work in high throughput model of

20MHz (HT mode). According to the IEEE802.11n protocol, the channel is divided into 56 subcarriers. The CSItools samples the 56 CFR OFDM subcarriers for 30 times. It works in group mode 2.

3.4 Multi-Input and Multi-Output Technology

Currently, the spectrum resource is becoming increasingly scarce. The use of multiple antennas and the multi-input and multi-output (MIMO) technology are the key features that the device based on 802.11n standard is different from the early 802.11 a/g equipment. MIMO technology uses multiple antennas to increase data throughput and transmission distance and not need to increase bandwidth and transmission power.

In the MIMO system, both the transmitter and receiver have multiple antennas and each receiver antenna and transmit antenna combination can be considered as a stream. All the streams of CSI can be represented as:

$$A = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1M} \\ H_{21} & H_{22} & \cdots & H_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ H_{N1} & H_{N2} & \cdots & H_{NM} \end{bmatrix} \quad (3.4)$$

M and N indicate the number of transmit antennas and receive antennas respectively. H_{ij} represents the CSI between the i -th receiving antenna and the j -th transmitting antenna. Because the paths of different flows are different in

indoor environment, the CSI received by different antennas is different too.

3.5 Environment Sensing Technology Based on Fingerprint Information

We know that the signal characteristics corresponding to different positions are different. For example, the CSI values are not the same at different locations. A lot of research is based on this point, using the RSS as a feature of the location fingerprint information. The positioning technology based on fingerprint information includes two procedures: training and positioning processes.

The training process completes the establishment of the fingerprint feature library. The positioning process uses the signal data that collected by the points to be measured to estimate the location of the points to be measured by matching algorithm or regression function. Typical fingerprint matching algorithms include Bias algorithm, KNN matching algorithm, and SVM matching algorithm.

At present, the positioning systems using SVM matching algorithm are mostly based on CSI. The basic principle is to use the CSI value of each anchor node as a training sample and to model them. Then, a separating hyperplane can be found, which can be denoted as formula 3.5:

$$w \cdot x + b = 0 \tag{3.5}$$

where w is the weight vector and b is a hyperplane offset constant. To implement

positioning, the following decision function must be passed.

$$f(X) = \text{sgn}\left\{\sum_{i=1}^n Y_i a_i X_i X + b_0\right\} \quad (3.6)$$

Among them, Y_i is the class label of support vector X_i ; a_i and b_0 are the parameters that are determined by the optimal SVM algorithm automatically. The signal strength of a given point to be measured X is used as the fingerprint input and substituted into formula 3-6. If the result is greater than zero, the point to be measured belongs to the area where it is located; otherwise it does not belong to the area where it is located. At last, the coordinate of the area where it is located is the coordinate of the points to be measured.

Chapter 4

System Design

4.1 Methodology

In 3.3, we introduce that the CSI signal contains a lot of information, including CFR amplitude, CFR phase and CIR, by means of which the desired fingerprint characteristics can be obtained. Hence, the CSI signal is one of the key issues we need to study.

It is well known that when the number of parking spaces changes, the signal strength received will change. The change is corresponded to the amplitude and phase of the received signal. In addition, there is a certain relation between the received signal energy and the location of known AP nodes. If $X(f, t)$ is denoted as the frequency domain of the transmitting signal, and $Y(f, t)$ as the frequency domain of the receiving signal, we can use the formula 4-1 to present the relationship.

$$Y(f, t) = H(f, t) \times X(f, t) \quad (4.1)$$

In formula 4-1, $H(f, t)$ is a CFR in time domain, which contains phase and amplitude signals. The physical basis for statistical features is that the spatial randomness of LOS and NLOS paths differs. NLOS paths typically involve richer reflections, diffractions and refractions. Therefore, signals transmitting through NLOS paths often behave more randomly, which manifests in both signal amplitudes and phases.

In this thesis, we investigate the phase information for two potential reasons.

First, as most amplitude based features implicitly assume certain distributions (*e.g.* Rician distribution [43]); large numbers of measurements are necessary for accurate distribution parameter estimation. Second, LOS/NLOS propagation is not the only factor that determines the extent of randomness in received amplitudes. Both the obstacle blockage of NLOS conditions and the long propagation distance can substantially attenuate signal amplitudes [44]. In contrast, phase shifts change periodically over propagation distances. As a result, the phase is more robust.

In above Section 2.2.2, it is introduced when people move indoors the phase variance of the CSI signal will become large. This thesis considers that this phenomenon is also one of the characteristics of parking spaces. How to use this feature to build the fingerprint characteristics of a parking query system is one of the contents we need to study. We know that the variance is an important tool for depicting data volatility. The greater the variance is, the greater the volatility of the data. Therefore, the phase variance of CSI is employed to characterize the fluctuations of CSI signal in different parking space conditions. The specific process is shown in equation 4.2.

$$[Var_1, Var_2, \dots, Var_n] = std([CSI_1, CSI_2, \dots, CSI_n]) \quad (4.2a)$$

$$\overline{Var} = \frac{1}{n} \sum_{i=1}^n Var_i \quad (4.2b)$$

where CSI_i is the CSI data of the i -th channels. Through getting the standard deviation and mean of CSI data of each channel, the CSI variance can be obtained.

It can be observed when the vacancy changes, because of the large fluctuation of the CSI signal, the phase variance of CSI will also fluctuate greatly.

With the change of empty spaces of vehicle sequence in the parking lot, the phase of CSI signal that received by the receiver is not same. By calculating the phase variance, we can preliminary estimate the number of the empty space. The system principle is shown in Fig. 4.1. Assume that there are transmitter T and receiver R and the T never stops the CSI signal transmission. If the vacancy state of vehicle parking lot changes from A to B, the signal propagation distance has an obvious difference between them. The CSI that received by the receiver also has different phase shifts. As a result, the phase that received is different, which is as shown in Fig. 4.1. Therefore, calculating the received phase variance of different subcarriers can be used to estimate the parking lot space.

- Transmitter: T_x Receiver: R_x

The propagation distance of multipath changes when the vacancy state of parking lot changes.

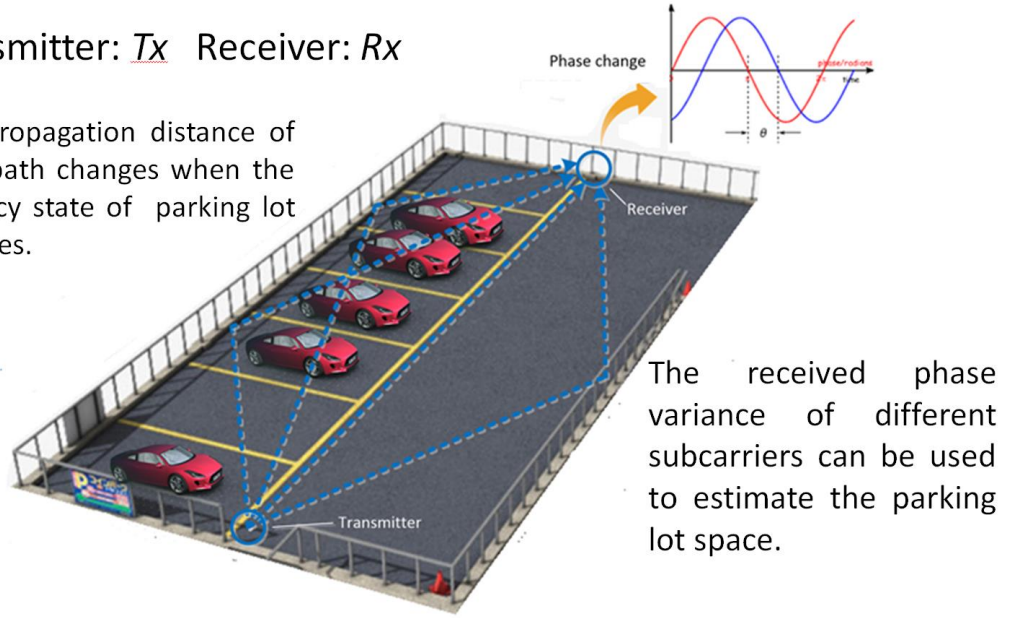


Figure 4.1: Phase shifts of CSI.

This thesis specifically focuses on the phase information based on the observation of unique phase variance of subcarriers depending on the number of vehicles obstructing the line of sight (LOS) path between the WiFi transmitter and the receiver. More specifically, the phase variance of subcarriers is employed as the major feature for training a model that classifies an input CSI data sequence into different classes representing the number of empty slots.

4.2 System Overview

In this section, an overview of the proposed approach is presented and then each system component is detailed.

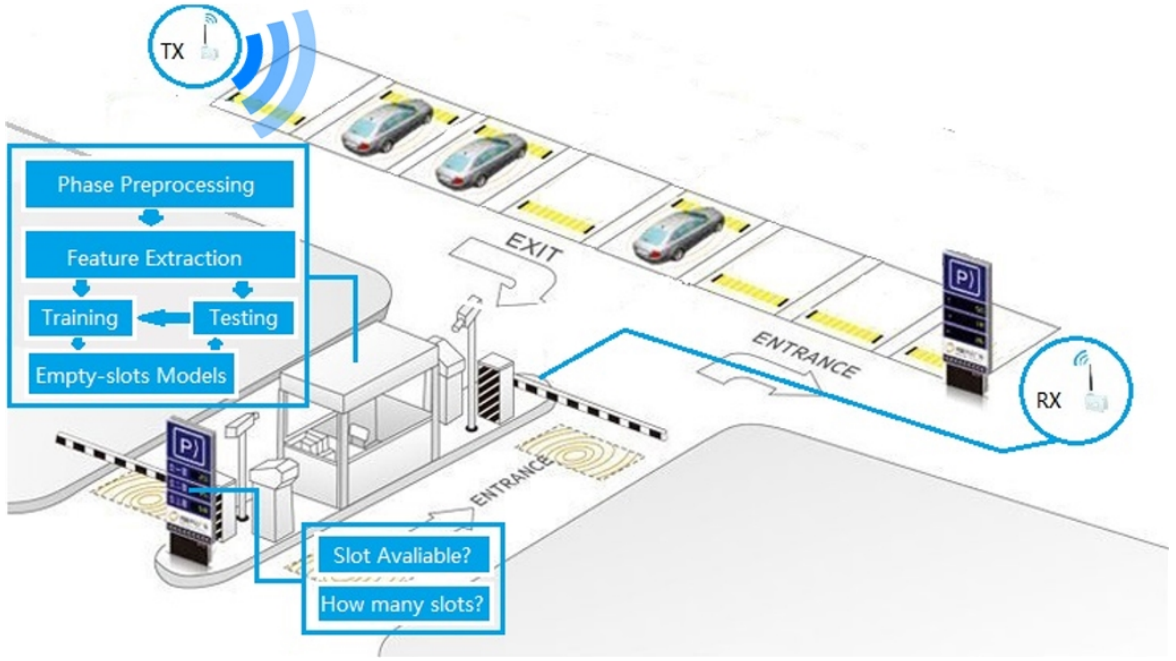


Figure 4.2: Illustration for Approach Overview

The proposed approach consists of five main components as shown in Fig. 4.2: CSI collection, phase preprocessing, feature extraction, training, and testing.

The CSI collection component collects the original CSI information from receivers. The phase pre-processing component obtains the phase information from the CSI data of a received WiFi signal, and performs a linear transformation to reduce the random noise. The primary task of the feature extraction module is to extract appropriate features from the preprocessed phase data and to organize the extracted features into an input file. Then the input file is sent to either the training module, or the testing module depending on the operation mode (*i.e.*, training mode or testing mode). The Training module is used to train a classification model that

Table 4.1: Hardware Configuration

Name	Number	Configuration	Use
Laptop	1	HP Elite 8730w	Transmitting WiFi packets
Laptop	1	HP Elite 8730w	Receiving WiFi packets
Network Interface Card	2	Intel 5300	Receive CSI data
Two Chairs	2	Folding Chair	Supporting laptops
Laptop	1	iMac	Analysis for data
Software	1	MATLAB	Constructing system

classifies the input data into classes representing the number of empty parking slots (*e.g.*, classes for 0 empty slot, ..., n empty slots).

More specifically, the phase variance of subcarriers is identified as a major feature based on the observation that vehicles interfering with the LOS path between the transmitter and receiver cause signal refractions, diffractions, and reflections, resulting in varying arrival times of subcarriers. Consequently, based on the generated classification model, the testing module can identify whether there is any empty parking slot or not. It also finds the number of empty parking slots.

4.3 CSI Collection

4.3.1 Hardware Configuration

The hardware configuration is as shown in Table 4.1.

A HP Elite 8730w model laptop is needed, which is equipped with the 2.53GHz Intel Core Extreme CPU Q9300 processor and 4GB RAM. This laptop is integrated with an Intel 5300 WiFi NIC and can transmit WiFi packets at a rate of 2500 packets/sec. It can run Ubuntu 14.04.04 with kernel version of Ubuntu 4.2.0-27.

Another laptop with the same configuration above is used to receive packets, to save the CSI data of the packets, and to perform classification of the number of empty parking slots. A parking lot is needed to perform experiments. A single row with 10 parking slots is occupied by the prototype system.

4.3.2 Comparison Between Two Modes

In this section, the effects of different positions of the transmitter and receiver on the system performance are investigated as follows.

- Out-mode

In this mode, the path of transmitter and receiver is the LOS. The transmitter and receiver are placed on the outer side of vehicle sequence as the Fig. 4.3.

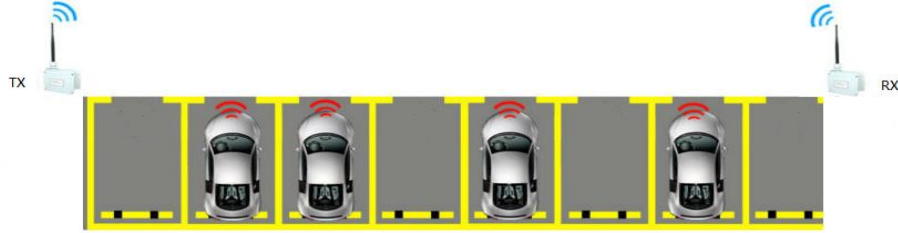


Figure 4.3: Illustration for Out-mode.

- In-mode

In this mode, both transmitter and receiver are placed at both sides of the vehicle arrangement. The path is NLOS. The signals from a transmitter pass through vehicles and arrive at the receiver. The distances between the vehicle and

the transmitter, and the receiver, are different, which are 3m, 6m, and 10m respectively as shown in the Fig. 4.4.



Figure 4.4: Illustration for In-mode.

After analyzing the collected data, we found that in the condition of using phase variance as the feature, Out-mode has more obvious differences than In-mode. In the In-mode, the data sampling rate is significantly lower than in the Out-mode. This is because in the Out-mode, the signal passes through the vehicle aslant, which will lead to stronger refraction and diffraction. Therefore, the change of phase variance is more obvious for different parking spaces. In the In-mode, if the distance is too near, such as 3m distance between the transmitter and the vehicle, the packet loss rate is greatly increased and the sampling rate decline as the vehicle made from metal has a strong shielding effect on the WiFi signal. If the distance is too far, for example, the receiver is 10m far from the vehicle; it is difficult to distinguish between different feature spaces. Although the sampling rate is not changed, the effect of refraction, diffraction and reflection is not obvious. Therefore, in the presented system, we place above instruments using Out-mode.

Sometimes the driver may park the vehicle not normally. For example, the vehicle is parked over the line of the parking space. In that condition, the signal will

be blocked by the vehicle and the out-mode will become in-mode. It will reduce the accuracy of the system. To avoid that situation, we placed the equipment 1 meter and 0.3 meter away from the border line of the parking lots. The figure 4.5 above shows the location of placement.

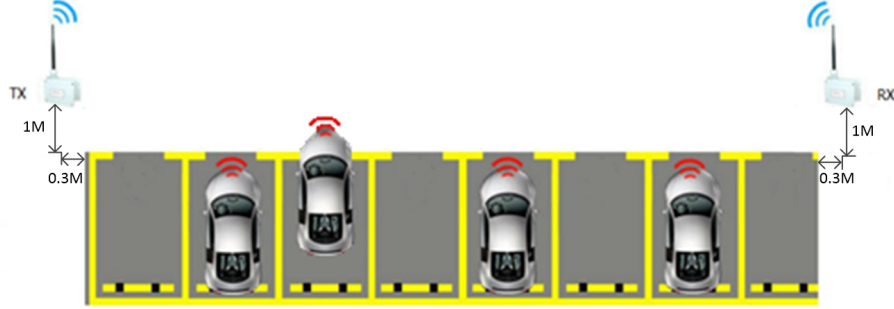


Figure 4.5: Special case of Out-mode

4.4 Pre-processing of WiFi CSI Data

The noise processing of CSI signal is a technical difficulty in this thesis. The CSI signal provided by existing commercial WiFi devices has very high noise. Fig. 4.6 illustrates the CSI signal collected with Intel 5300 NIC at the sampling rate of 2.5kHz. It can be seen from Fig. 4.6 that the signal graph has many burrs and is not smooth, which will have a great impact on the accuracy of positioning. It is necessary to remove noise from the collected CSI data.

4.4.1 De-noising Processing

Through the analysis of the parking lot CSI signal, we found that the background noise is mainly concentrated in the high phase part and the low pass

filter can remove high-frequency noise. Therefore, we consider using (Butterworth) low-pass filter to remove noise from the CSI signal. From the de-noising results as shown in Figure. 4.7, we can see that after filtering the CSI waveform is smoother than the original waveform. In addition, the local burrs are also reduced. However, there is still a noise pulse, which indicates that only using low pass filter, all noises in the CSI signal can't be eliminated because the CSI signal contains the impulse noise with high energy and high bandwidth. Hence, the low-pass filter's band-pass needs to be less than $1/20$ of sampling rate so that it can make the noise energy of the filtered CSI signal negligible. When the sampling rate is not high enough, the residual noise of the CSI signal will still make signal distortion after low pass filtering [45].

In order to solve the above problems, we further deal with the CSI signal after low-pass filtering.

4.4.2 Phase Pre-processing

One of the main sources of noise in the CSI signal is caused by the conversion of the internal state of WiFi NIC. It produces impulse noise and burst noise in the CSI signal.

Reducing random noises of the CSI data is crucial for achieving high classification performance. Since the phase information of the CSI data is used as a primary feature for identifying empty parking slots, the accuracy of the phase information needs to be paid attention. Hence, this section explains the details on how to mitigate the impact of random noises contained in the phase information of

CSI data.

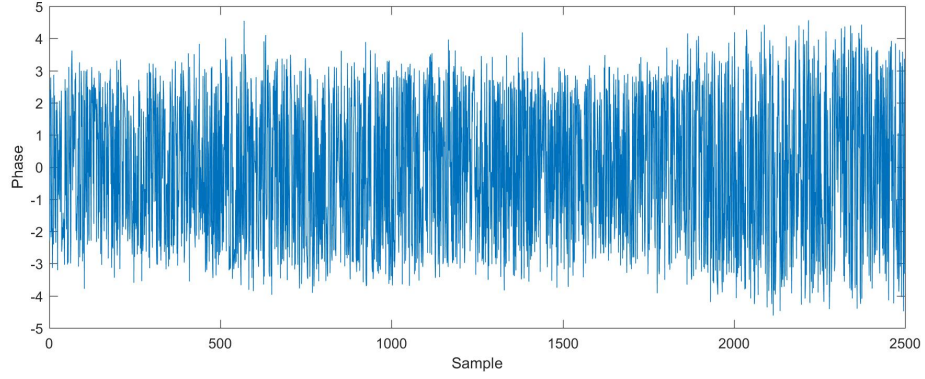


Figure 4.6: A raw phase diagram.

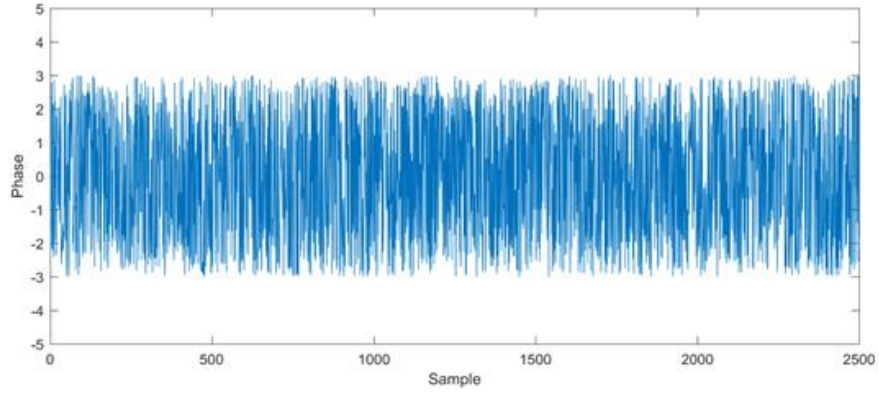


Figure 4.7: After low-pass filter phase diagram.

The measured phase of subcarrier f can be written as the following formula

4.3:

$$\hat{\phi}_f = \phi_f - 2\pi \frac{k_f}{N} \delta + \beta + Z \quad (4.3)$$

Here ϕ_f is the genuine phase, k_f is the subcarrier index, N is the Fast Fourier Transform size. There are two major sources of noise: δ is the time lag at the receiver, and β represents an unknown phase offset. Z is some measurement noises.

Based on [47], a linear transformation is performed to reduce the random noise. More specifically, the phase information of a subcarrier can be obtained according to the following calculation:

$$a = \frac{\hat{\phi}_f - \hat{\phi}_1}{2\pi F} \quad (4.4)$$

$$b = \frac{1}{F} \sum_{1 \leq f \leq F} \hat{\phi}_f \quad (4.5)$$

where F is the last subcarrier index (In this experiment, F is 30 as Intel 5300 NIC exports 30 subcarriers). And then, a simple linear transformation, *i.e.*, removes the timing offset at the receiver, and the unknown phase offset.

And then, a simple linear transformation, *i.e.*, $\phi_f = \hat{\phi}_f - af - b$ removes the timing offset at the receiver δ and the unknown phase offset β .

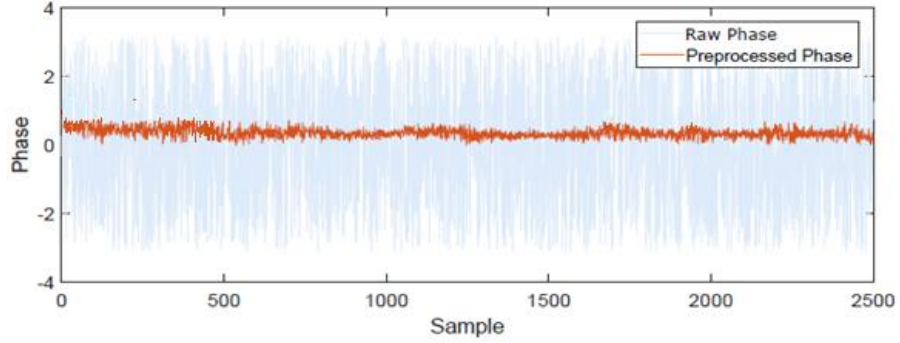


Figure 4.8: Comparison between the raw phase and pre-processed phase.

Fig. 4.8 shows the raw phase and the pre-processed phase of a subcarrier that was measured for 1 second at a rate of 2,500 samples/sec. As shown, the

random noises of the phase information have been effectively mitigated.

Once random noises are eliminated, phase values are normalized into the range of $[0, 1]$. The feature extraction from CSI data collected from varying scenarios with different number of empty slots can be implemented.

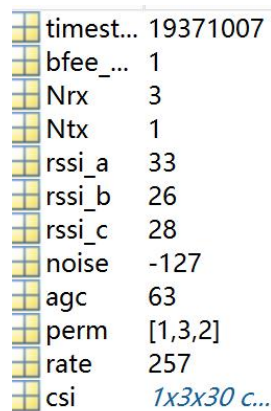
4.4.3 Processing Procedures

The pre-processing of CSI data proceeds according to following four procedures:

(1) Import to Matlab

Firstly, by using a toolkit called linux-80211n-csitool-supplementary provided by WiFi CSI tool the CSI data are imported into MATLAB cells. MATLAB is employed to process CSI data.

Then, use the function `read_bf_file` of linux-80211n-csitool-supplementary to extract the CSI data from `csi_trace`. The data of the `csi_trace` are shown in Fig. 4.9



timest...	19371007
bfee_...	1
Nrx	3
Ntx	1
rss_i_a	33
rss_i_b	26
rss_i_c	28
noise	-127
agc	63
perm	[1,3,2]
rate	257
csi	1x3x30 c...

Figure 4.9: The data of a `csi_trace`

The `csi_trace` is a cell that contains a lot of information. We only need CSI information. Hence, we need to use `get_scaled_csi` function to extract the CSI information into a new matrix so that we can process CSI data. The `get_scaled_csi` function is also included in the `linux-80211n-csitool-supplementary`. Parts of codes using `get_scaled_csi` function are shown as Fig. 4.10.

```

1- cd 'C:\DefeatEvil\Thesis\CSI\linux-80211n-csitool-supplementary\matlab';
2- csi_trace=read_bf_file('sample_data/921-empty3-out-6.dat');
3- for i=1:5000
4-     csi(i, :, :)=get_scaled_csi(csi_trace{i});
5- end
6- for i=1:5000
7-     for j=1:30
8-         empty3out6(i, j)=imag(csi(i, 1, j));
9-     end
10- end

```

Figure 4.10: Parts of codes of Using `get_scaled_csi` function

(2) Set time We need to set a time for analysis the data. As mentioned above Section, the receiver receives 2500 packets every second. If we want to analysis the data of 2 seconds, we need to set time to $2500 \times 2 = 5000$ because the time is the number of packets. The code of setting time procedures is as shown in Fig. 4.11.

```

1- for i=1:5000
2-     time1(i)=i;
3- end

```

Figure 4.11: Codes of setting time

(3) Set Filters

The raw CSI data has a lot of noise and we need to reduce them. We will use

the method we discussed above to reduce the noise.

The codes of setting filters is as shown in Fig. 4.12:

```

1 — □ for i=1:5000
2 —     a=0;
3 —     sum=0;
4 —     a=(empty3out6(i,30)-empty3out6(i,1))/(2*3.1415*30);
5 —     □ for j=1:30
6 —         sum=sum+empty3out6(i,j);
7 —     end
8 —     b=sum/30;
9 —     □ for f=1:30
10 —         Empty3out6(i,f)=empty3out6(i,1)-a*f-b;
11 —     end
12 — end

```

Figure 4.12: Codes of setting filters

4.5 Feature Extraction

4.5.1 Extraction Principles

Fundamentally constrained by the time resolution of COTS WiFi devices, it is infeasible to identify features by relying on only one channel snapshot and CIR based LOS. In this section, we extract distinctive statistical features from the basically unexplored phase information.

The physical foundation for statistical features is the difference of the spatial randomness of LOS and NLOS paths. NLOS paths typically involve abundant reflections, diffractions and refractions. Therefore, the signals transmitting through NLOS paths often behave more randomly, which reflects in both signal amplitudes and phases.

CSI measurements can provide phase information for each carrier. To

mitigate the impact of random noises, we first perform a linear transformation on the raw phases, as recommended in section 4.4.2.

Similar to the principle of amplitude based features, the extent of randomness can be revealed in statistics describing certain order of deviation. For real-time operations, we employ variances calibrating phases as candidate features due to its simplicity.

Although we cannot obtain the true phase except a calibrated measurement $\hat{\phi}_f$, we can demonstrate that the variances of filtered phases differ from true phases which are only related a multiple of frequency dependent constants. In addition, although it cannot be a distinctive feature of LOS recognition, the variance of phases does exhibit different trends under the LOS/NLOS propagation. Inspired by the preliminary measurements, we explore more conspicuous phase variance.

Thirty subcarriers of the WiFi signal are emitted at different positions in all directions. After a lot of reflection, diffraction and refraction, the phases of different subcarriers accepted by the receiver are different. Due to the different arrangement of vehicles in the parking lot, the degree of reflection, diffraction and refraction is also different. We can find the relationship between the phase variance and the parking space vacancy by the distribution of the phase variance of each packet.

4.5.2 Phase Variance For Different Empty Parking Slots

The challenge for developing the feature extraction module is to identify features that can effectively characterize and distinguish the number of empty slots

in a parking lot.

The key idea comes from the observation that more obstacles obstructing the line of sight (LOS) path will generate higher refractions, diffractions, and reections. As a result, the phase received causes greater randomness [44]. The feature extraction module exploits the strong correlation between the number of vehicles (*i.e.*, the number of empty slots) obstructing the LOS and the variance of phase, and selects it as a primary feature for classification.

Fig. 4.13 shows the phase variance of 400 WiFi packets for a parking lot with 9 empty slots, and for the same parking lot with 10 empty slots.

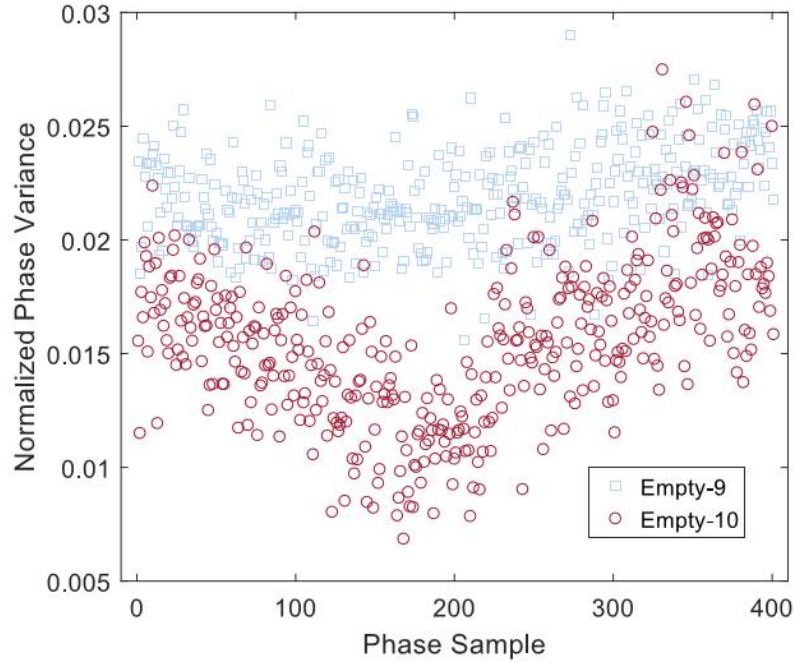


Figure 4.13: Variances of phase for parking lots with 9 empty slots and 10 empty slots

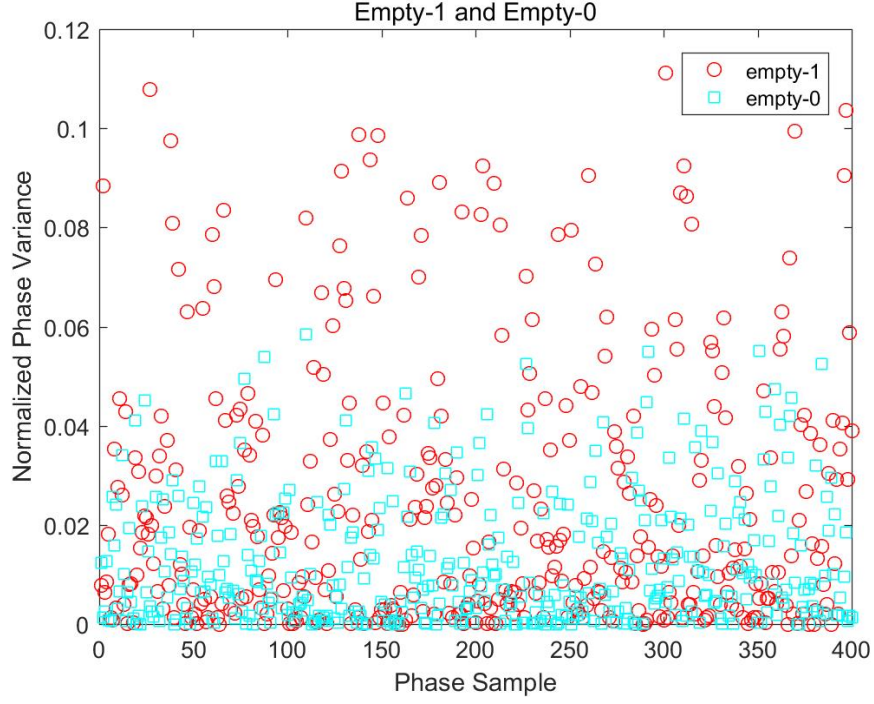


Figure 4.14: Variances of phase for parking lots with 1 empty slots and 0 empty slots

Fig. 4.14 shows the phase variance for 400 WiFi packets for a parking lot with 0 empty slot and for the same parking lot with 1 empty slot.

The phase variance is the feature to classify the empty space of parking lots. After getting the filtered phase information, we need to calculate the variance of them. The phase variance is the variance of 30 subcarriers of one packet. We can use the MATLAB function to implement this calculation.

1) Filtering of phase of CSI is as follow:

```
1 - varianceempty3out6=var( Fempty3out6,0,2);
```

2) Drawing graph

After the phase variance has been calculated, we can use scatter function to

draw a scatter graph so that we can find the difference intuitively. In order to identify the difference of features more obviously, we also create a 3D scatter graph so that we can find the difference of phase variance very clearly. In general, a large amount of data is selected to find the feature more obviously. The code of drawing the scatter graph is as follows:

```

1— box on;
2— %scatter(time1,empty1out1variance,'r');
3— %hold on;
4— scatter(time1,varianceempty0out1,'s','p');
5— hold on;
6— scatter(time1,varianceempty3out5,'r');
7— legend('empty-0','empty-3');
8— title('Empty-0 and Empty-3');
9— xlabel('Phase Sample');
10— ylabel('Normalized Phase Variance');

```

Fig. 4.15 shows that the phase variance of no empty slot and 4 empty slots is different very obviously. To find the difference more effectively, we draw a 3D graph as shown in Fig. 4.16.

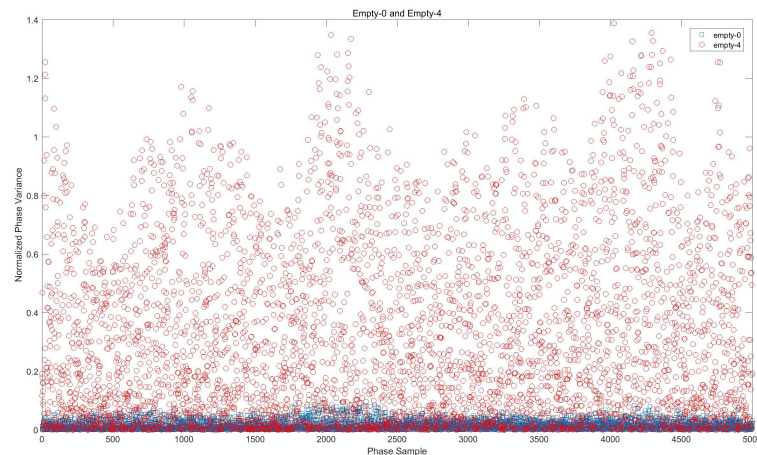


Figure 4.15: The scatter graph of phase variances

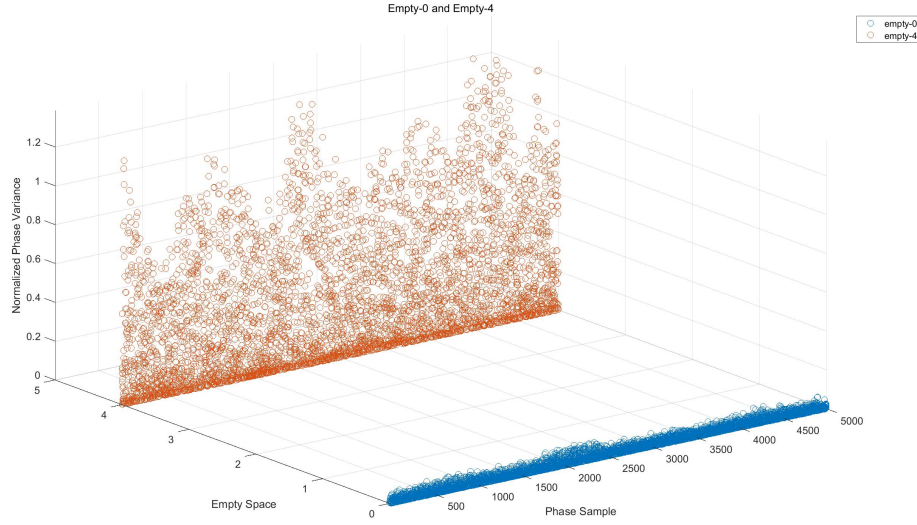


Figure 4.16: The 3D scatter graph of phase variances

Taking a parking lot with 9 empty slots and a parking lot with 10 empty slots as an example, we can observe their variances of phase in Fig. 4.11. The distributions of phase variance for the two scenarios are not clearly separated. On the other hand, mainly considering the phase variance values, it can be viewed that the phase variance of the scenario with 10 empty slots is smaller than that with 9 empty slots.

This particular observation demonstrates that using an individual value of phase variance as a feature is not appropriate; rather, the distribution patterns of a group of phase variance values collected over a period of time are used as features, which can obtain a better classification accuracy.

4.5.3 Distribution patterns of phase variance

We have analyzed all the cases and drawn two 3D scatter graphs as shown in Fig. 4.17 and 4.18. We find that the phase variance increases when the empty space of vehicle become more. The more the empty space is, the wider the distribution of phase variance is. The total distribution of phase variance presents a ladder type

The CDF graph of phase variance for varying numbers of empty slots is shown in Fig. 4.19, which is extracted from larger data sets (collecting CSI data of 270 seconds for each case. The total amount of samples is 6,750,000 samples). An interesting observation is that although the phase variances for the two classes of similar numbers of empty slots are very close to each other, distinctive patterns are found that allow us to use it for training a model with a group of phase variances.

Given the observations that indicate the effectiveness of the phase variance as a feature for classification, we determined to use it as the primary feature for running a machine learning algorithm, which will be explained in the following section.

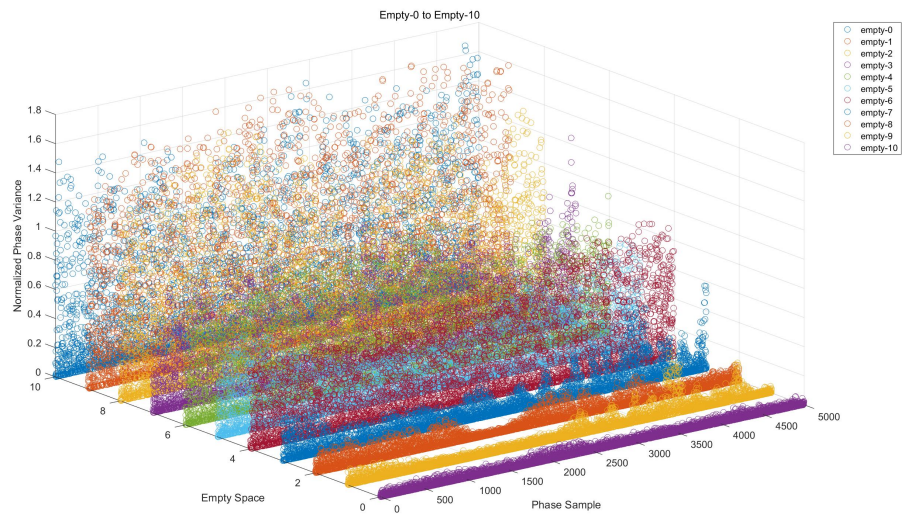


Figure 4.17: Phase variances of all cases

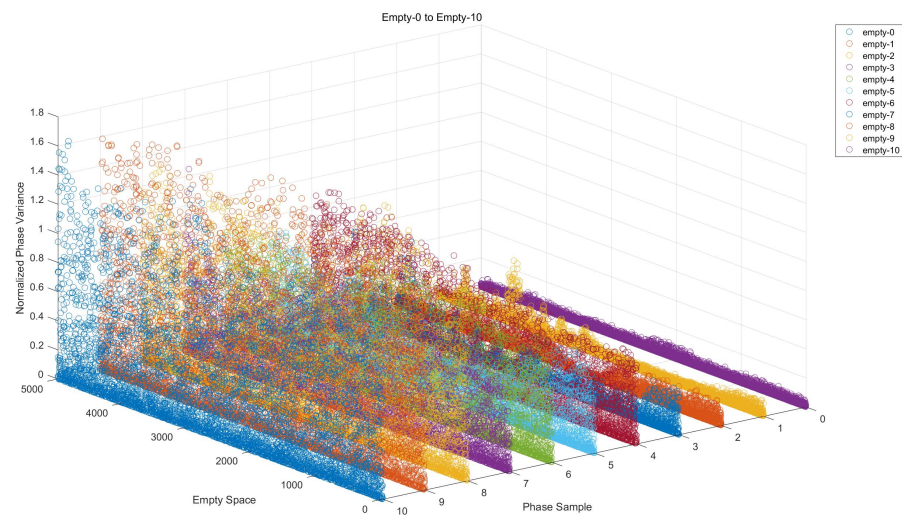


Figure 4.18: All case phase variances

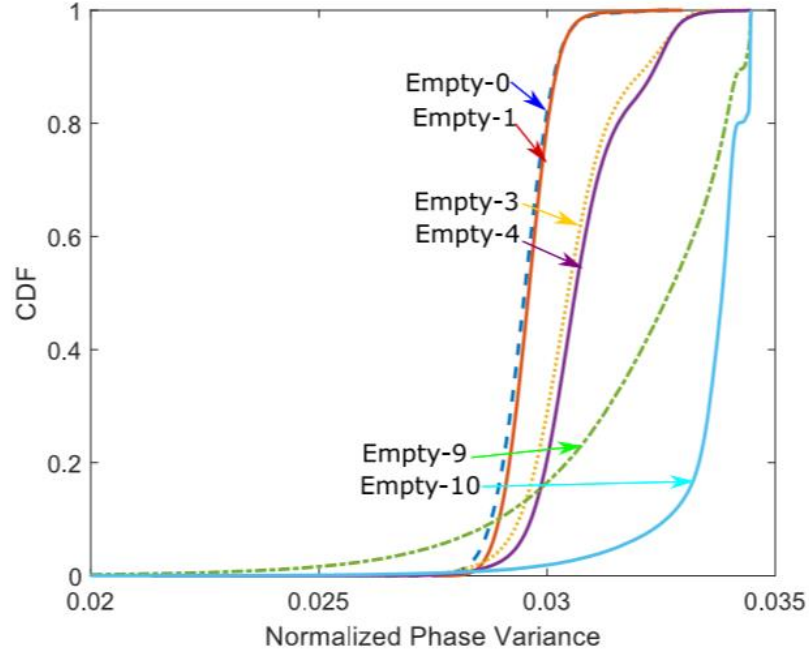


Figure 4.19: CDF of phase variance for varying numbers of empty slots

4.6 Training and Testing

4.6.1 Methodology

Obviously, using threshold-based approach will not work effectively for classification since the phase variances for different numbers of empty slots are not clearly separated. It is also because that the pattern of phase variances is used as a major indicator to distinguish from varying numbers of empty slots rather than using individual values of phase variance. Consequently, a machine learning approach is employed for more effective classification given the characteristics of the selected feature. More specifically, we adopt the Support Vector Machine (SVM) in training a

classification model that classifies an input data sequence consisting of a group of phase variance values into an appropriate class that represents different numbers of empty slots. The following will introduce some concepts on SVM.

(1) SVM

In machine learning, support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training samples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new samples to one category or the other, making it a non-probabilistic binary linear classifier.

An SVM model is a representation of the samples as points in space, mapped so that the samples of the separate categories are divided by a clear gap that is as wide as possible. New samples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering and is often used in industrial applications either

when data are not labeled or when only some data are labeled as a preprocessing for a classification pass.

(2) LIBSVM

In creating a classification model and predicting the class, LIBSVM and LIBLINEAR are usually used as an integrated software tool [46], which are two popular open source machine learning libraries for support vector classification, regression and distribution estimation. They support multi-class classification and were developed at National Taiwan University with C++ language.

LIBSVM implements the SMO(Sequential minimal optimization)algorithm for kernelized support vector machines, supporting classification and regression.

LIBLINEAR implements linear SVMs and logistic regression models trained using a coordinate descent algorithm.

The SVM learning code from both libraries is often reused in other open source machine learning toolkits, including GATE, KNIME, Orange and scikit-learn. Many bindings to it exist for programming languages such as Java, MATLAB and Ruby.

(3) Standard Cross-validation Technique

To obtain better classification results, the two parameters, *i.e.*, C and γ , are needed, which are usually decided by the standard cross-validation technique [47].

To train SVM problems, users must specify some parameters. LIBSVM provides a simple tool to check a grid of parameters. For each parameter setting, LIBSVM obtains Cross-Validation (CV) accuracy. Finally, the parameters with the

highest CV accuracy are returned.

The parameter selection tool assumes that the RBF (Gaussian) kernel is used although extensions to other kernels and SVM can be easily made. The RBF kernel takes the form. Hence, (C, γ) are parameters to be decided. Like formula 4.6.

$$k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \quad (4.6)$$

Users can provide a possible interval of C (or γ) with the grid space. Then, all grid points of (C, γ) are tried to find the one giving the highest CV accuracy. Users then use the best parameters to train the whole training set and generate the final model.

We do not consider more advanced parameter selection methods because for only two parameters (C and γ), the number of grid points is not too large. Further, because SVM problems under different (C, γ) , parameters are independent. LIBSVM provides a simple tool so that jobs can be run in a parallel (multicore, shared memory, or distributed) environment.

For multiclass classification, under a given (C, γ) , LIBSVM uses an one-against-one method to obtain the CV accuracy. Hence, the parameter selection tool suggests the same (C, γ) for all $k(k - 1)/2$ decision functions.

4.6.2 MATLAB-SVM Classifier

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment. A proprietary programming language developed by MathWorks,

MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

MATLAB provides many classifiers. To current understanding, classifiers of MATLAB include: K nearest neighbor classifier, Random forest classifier, Naive Bayesian, Ensemble learning method, Discriminant analysis classifier, and Support vector machine. In this thesis, we select SVM as the classifier to implement identification of empty parking slots.

MATLAB has a SVM classifier. In this thesis, we mainly use the following two functions to complete the classification tasks.

svmtrain function svmtrain is a function of training classification model. Its function format is as follow:

`SVMStruct = svmtrain (Training, Group, Name, Value).`

The input parameters are training data, training data corresponding group attribute, optional parameter name, optional parameter value. The output is a structure defined by MATLAB.

There are many optional parameters. Here we introduce two optional input parameters that following examples will use.

(1) kernel-function (kernel function): Optional kernel function includes: linear, quadratic, polynomial, RBF, MLP, @kfun, etc. If the type of kernel functions is not set, the linear kernel function is selected as the default kernel function.

(2) showplot (drawing): showplot is a Boolean value to indicate whether to

draw categorical data (here is the training data) and the line of classification, but this drawing function only supports the drawing of data with two eigenvalues, that is, two-dimensional point data (default value is false).

In svmtrain function, if showplot is set to true, the program will automatically draw two classes of points in the training data and the classification line obtained by the training data and marking which point is the support vector in different colors. The Fig. 4.20 is the graphs drawn by the application of showplot after a set of training data have been trained by the svmtrain function.

Among them, the red mark “+” represents the data in training set 1, and the green asterisk represents the data in training set 2. Support vectors are circled by mark “O”.

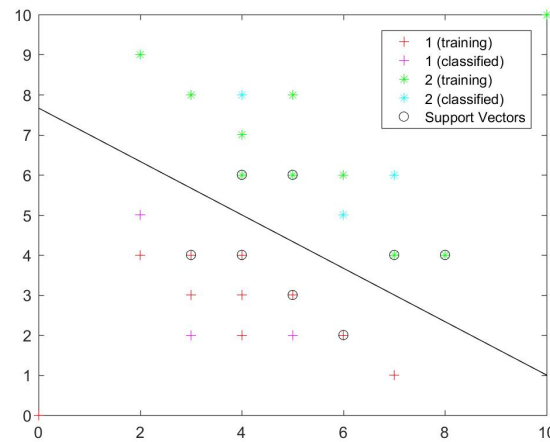


Figure 4.20: The graph drawn by using SVM

The Fig. 4.21 shows the process of training and test the CSI Data from two different classes. After the training and testing process has been done, an accurate pattern model has been created for the classification in the future.

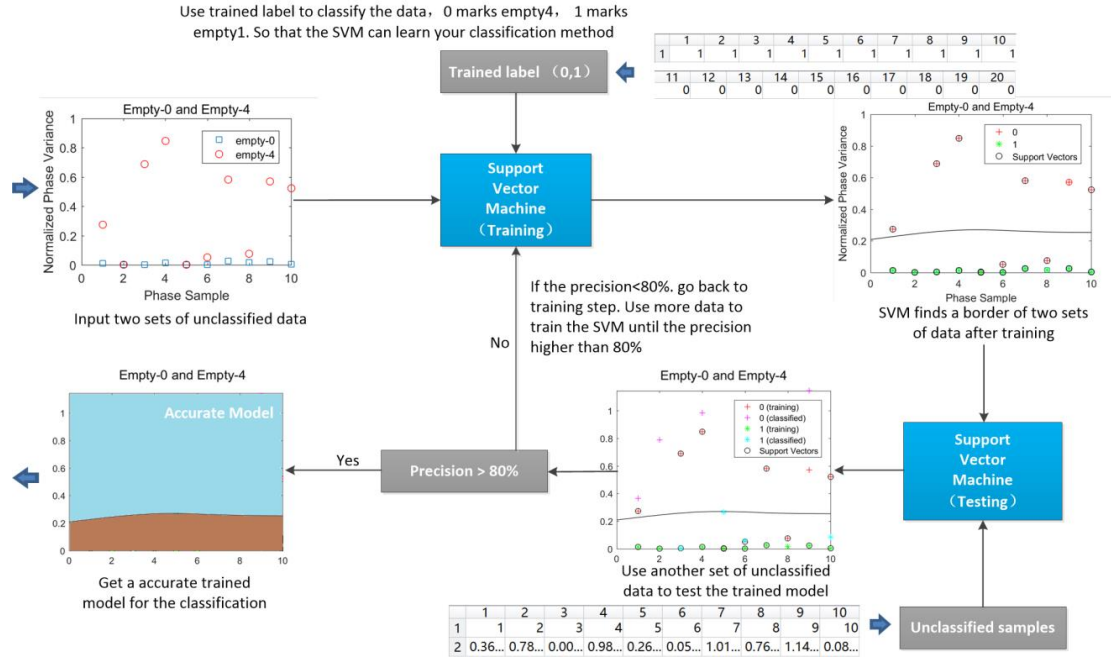


Figure 4.21: The process of training and testing data with SVM

4.6.3 Multiple Classification Problems in LIBSVM

Although SVM is a binary classifier, SVM can also be used to implement multi classification. The main method is to train multiple binary classifiers.

Given m classes, m binary classifiers are needed to be trained, in which the classifier i is to set the i class data as class 1 (positive), and all other $m-1$ classes except i class are commonly set as class 2 (negative). For each class, a binary classifier is needed to be trained. At last, we can have m classifiers.

The Fig. 4.22 shows the algorithm of the multiple classifications with SVM. If CSI match the trained module, the classifier will output 1, otherwise it will output 0.

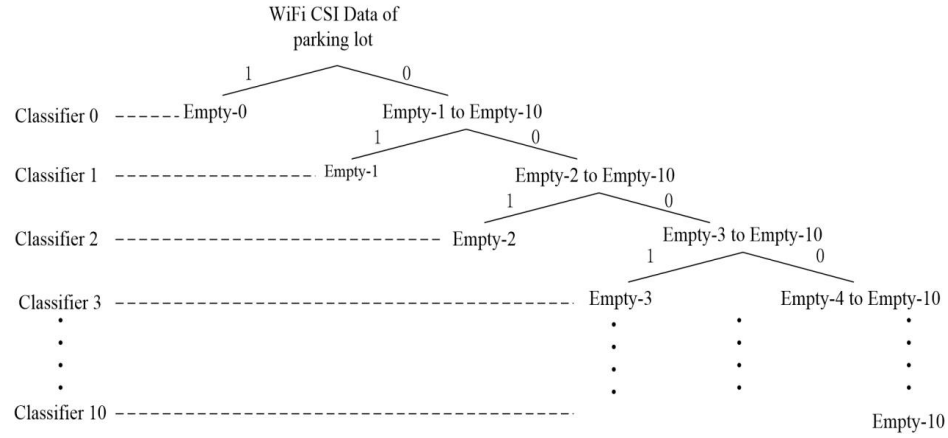


Figure 4.22: The Algorithm of Multiple Classifications

For data x that needs a classification, its category is determined by voting methods. For example, when the classifier i predicts the data x , if the positive result is obtained, it shows that the result of using classifier i to classify x is: x belongs to the class i . As a result, i won one. If the negative result is obtained, it shows that x belongs to other class except class i . Then, every class except i won one. The class which won the most votes will be the class attribute of X .

The following introduce the linear and Non-linear SVM

(1) Linear SVM

Linear SVM is used for two classification problems by finding a classification line (*e.g.* 2D straight line, three-dimensional plane, multidimensional hyperplane), which can divide the data into two categories. Using the linear function $f(x) = w * x + b$ the classifier can be constructed. The classification is a two-dimensional line as shown in Fig. 4.23.

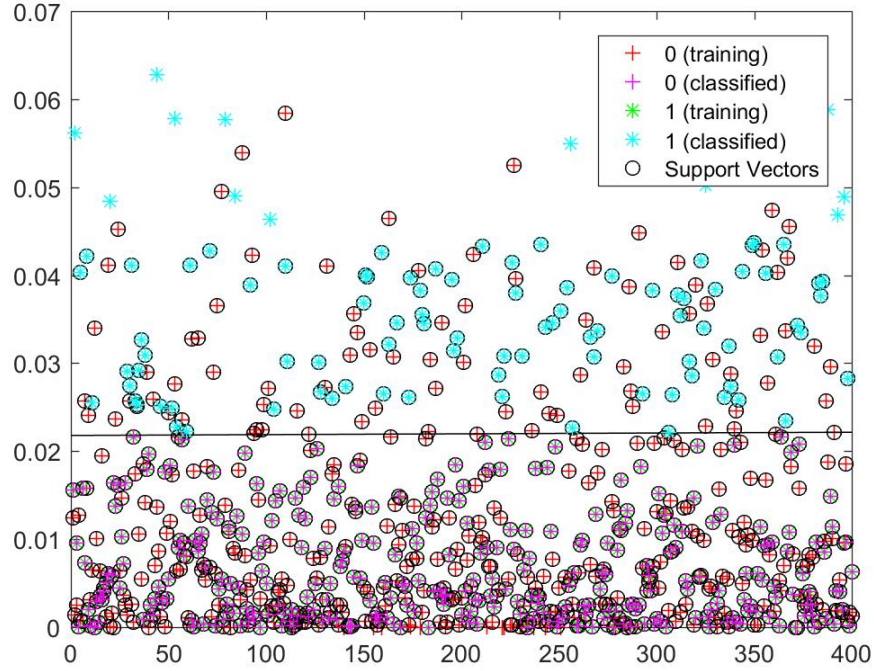


Figure 4.23: Illustration for classification using Linear SVM

(2) Non-linear SVM

For linear separable case as mentioned above, a linear function

$f(x) = w * x + b$ can be used to construct a classifier. The classification of data can be completed by finding a classification line (or surface) with maximum distance from the edge line to the nearest data point. However, we will encounter another problem, that is, if the data is not linearly separable, a two-dimensional linear, three-dimensional plane or multidimensional hyper plane cant classify them into two classes.

For the linear non-separable problem, the method of SVM is to convert the original linear inseparable data to a high dimensional space, so that it becomes

linearly separable. As shown in Fig. 4.25, some data are linearly inseparable. If they are converted into a high dimensional space, for example as indicated in the Fig. 4.24, converted from two-dimensional data into 3D, these data can classify into different categories. Therefore, the linear inseparable data needs to be mapped to the high dimension so that they can be linearly separable. Then, the linear SVM method is used to complete the classification.

The codes of mapping the 2D data to 3D data using kernel function are as follows:

```
1- sigma=2;
2- project = @(data, sigma) sum(exp(-(squareform( pdist(data, 'euclidean') .^ 2) ./ ( 2*sigma^2))));
3- blue_z = project(blue1', sigma);
4- red_z = project(red1', sigma);
```

Nevertheless, in the high dimensional transformation process, SVM doesn't actually implement high dimensional mapping. In fact, the maximum edge classification plane which is called the kernel function is found by a technique and applied to the original input data. Firstly, this technique allows us not need to know what is the mapping function. Only need to apply the selected kernel function on the original input data. Secondly, all the calculations are done on the low dimensional space of the original input data to avoid the high dimension operation. For this kernel function, there are several options, including Gauss polynomial kernel function, RBF kernel function, S kernel function and so on.

Through the analysis of the accuracy, we found that using nonlinear SVM has higher precise than using linear SVM. Hence, we will use nonlinear SVM in this thesis.

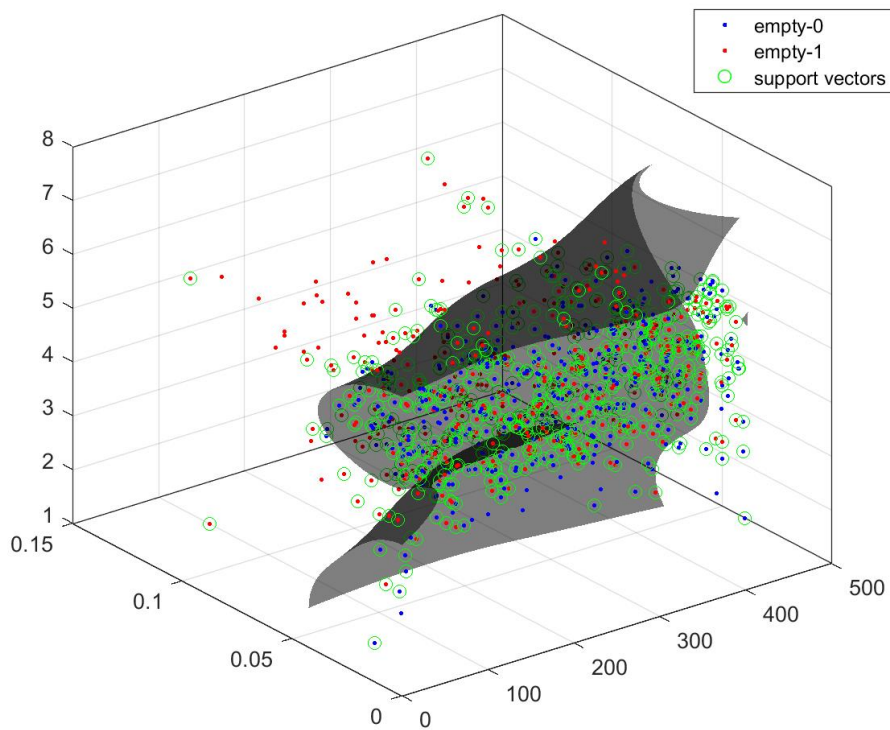


Figure 4.24: Map 2D data to 3D data with Kernel function

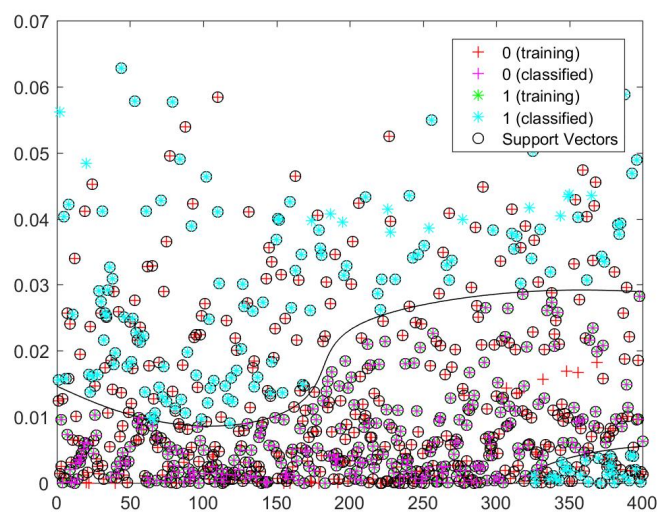


Figure 4.25: Non-linear SVM

4.6.4 Setting the Training Data and Test Data

Using SVM to implement the classification needs to train the selected SVM at first. It is necessary to set a group of train data and a group of test data to train the selected SVM. The train data include the train group and train label. The train label is the expected answer. The codes of setting train data and test data are as follows

(1) Training Data

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2	0.01...	0.00...	0.00...	0.01...	7.71...	6.19...	0.02...	0.01...	0.02...	0.00...	0.00...	0.03...	0.01...	5.95...	0.00...	0.01...	0.00...	0.00...

(2) Training Label

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```

1-  for i=1:5000
2-      atrain0_land3_5(i,1)=i;
3-      atrain0_land3_5(i,2)=varianceempty0out1(i);
4-  end
5-  for k=1:5000
6-      btrain0_land3_5(k,1)=k;
7-      btrain0_land3_5(k,2)=varianceempty3out5(k);
8-  end
9-  for k=1:5000
10-      test3_5(k,1)=k;
11-      test3_5(k,2)=varianceempty3out6(k);
12-  end
13-  ftrain0_land3_5=[atrain0_land3_5;btrain0_land3_5];
14-  for l=1:5000
15-      trainlabel(l)=0;
16-  end
17-  for m=5001:10000
18-      trainlabel(m)=3;
19-  end

```

We use svmtrain function to train the train data and train label, and use kernel-function and non-linear classification. Matlabs LIBSVM tool is employed to analyze and classify the data. The result is shown on Fig. 4.26

```

1 - svmModel=svmtrain(ftrain0_land3_5,trainlabel,'kernel_function','rbf','showplot',true);
2 - classification=svmclassify(svmModel,test3_5,'Showplot',true);

```

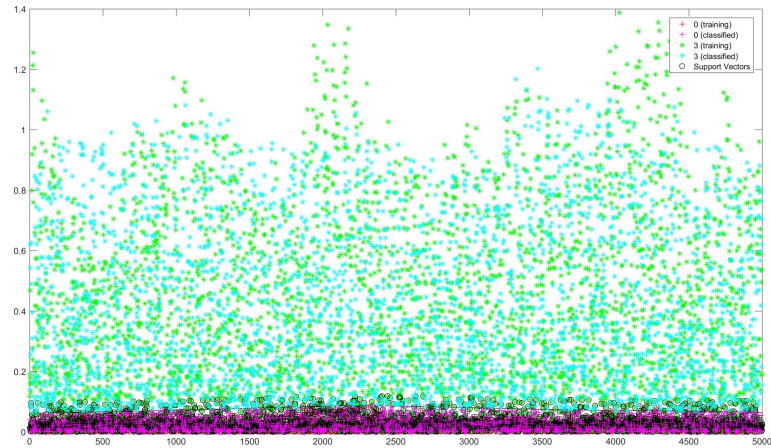


Figure 4.26: The Figure after using SVM

(3) Classification Precision

After the data is classified by SVM. It needs to find the precision of finding empty. All outputs are all stored in the matrix classification. Hence, we need to count the occurrence of the correct output for precision.

The code of Find Precision:

```

1 - count=0;
2 - for i=1:5000
3 -     if classification(i)==3
4 -         count=count+1;
5 -     end
6 - end
7 - fprintf(' the percision: %f\n', count/5000);

```

The result shows that the precision is 0.8422.

```

>> SVMclass
>> percision
The percision is: 0.842200

```

Chapter 5

Experiments

5.1 Experimental Setup

The experimental system is configured as follows:

- (1) A transmission device for CSI signals.

A HP Elite 8730w type laptop is needed, which is equipped with the 2.53GHz Intel Core Extreme CPU Q9300 processor and 4GB of RAM. A Intel 5300 WiFi NIC is integrated to transmit WiFi packets at a rate of 2500 packets/sec. The laptop run Ubuntu 14.04.04 with kernel version of Ubuntu 4.2.0-27.

- (2) A receiving device for CSI signals.

A laptop with the same configuration as above is used to receive packets, save the CSI data of the packets, and perform classification of the number of empty parking slots.

- (3) A parking lot is used as the experimental site

Considering parking and convenience, we selected a parking lot in front of our department building as the experimental site. At the parking lot, a single row with 10 parking spots was selected as the testing object.

5.2 Experimental Scheme

The laptop is placed at one end of each parking row and is approximately 50cm far from the parked car. In order to facilitate reliable packet reception the

laptop is situated about 40cm above the ground. When the parking slots change, the phase difference caused by parking cars between the transmitter and the receiver can be obtained. Fig. 5.1 shows the experimental scheme.

We collected 90 seconds of CSI data for each empty slot, *i.e.*, containing 225,000 samples.



Figure 5.1: Snapshot of experimental setting at department parking lot

We repeated the data collection for 5 times for each empty slot. Thus, a total of 450 seconds of CSI data containing 1,125,000 phase variance values were incorporated to define a class that represents a certain number of empty slot.

Fig. 5.2 and Fig. 5.3 show the operation interface of transmitter and receiver. On the transmitter total 450000 packets every 400s are transmitted. The size of a packet is 213 bytes. On the receiver, the laptop receives the signal and store in a log file.

```

zyf@zyf-HP-EliteBook-8730w: ~
zyf@zyf-HP-EliteBook-8730w:~$ sh runinjection.sh 450000 100 1 400
[sudo] password for zyf:
0x4101
Generating packet payloads
Initializing LORCON
Sending 450000 packets of size 100 (. every thousand)
.....50k
.....100k
.....150k
.....200k
.....250k
.....300k
.....350k
.....400k

```

Figure 5.2: Transmitter Interface

```

zyf@zyf-HP-EliteBook-8730w: ~
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
wrote 213 bytes [msgcnt=75500]
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213
received 213 bytes: id: 26 val: 1 seq: 0 clen: 213

```

Figure 5.3: Receiver Interface

In particular, another 90 seconds of CSI data for each empty slot were collected and used as the test data set. Collecting the full CSI data set for all empty slots was challenging. We had to wait until the desired number of empty slots was available since we did not have the control over the vehicles coming in and out of the parking lot.

In creating a classification model and predicting the class, the lib SVM was used [18]. In order to collect the CSI data efficiently, we also write a script so that we can reduce the time for inputting complex instruction. The script is as follows:

(1) Monitor script:

```
#!/bin/bash
cd /home/zyf/NewDisk/zyf/linux-80211n-csitol-supplementary/injection
sleep 1
./setup_monitor_csi.sh 64 HT20
sleep 1
echo "Successfully create $1.dat.Wait for inection..."
sudo ../netlink/log_to_file /home/zyf/NewDisk/zyf/CSI_DATA/$1.dat
```

(2) Injection script:

```
#!/bin/bash
cd linux-80211n-csitol-supplementary/injection
sleep 1
./setup_inject.sh 64 HT20
sleep 1
sudo echo 0x4101 |sudo tee /sys/kernel/debug/ieee80211/phy0/iwlwifi/
iwldvm/debug/monitor_tx_rate
sleep 1

sudo ./random_packets $1 $2 1 $3|
```

5.3 System Interface

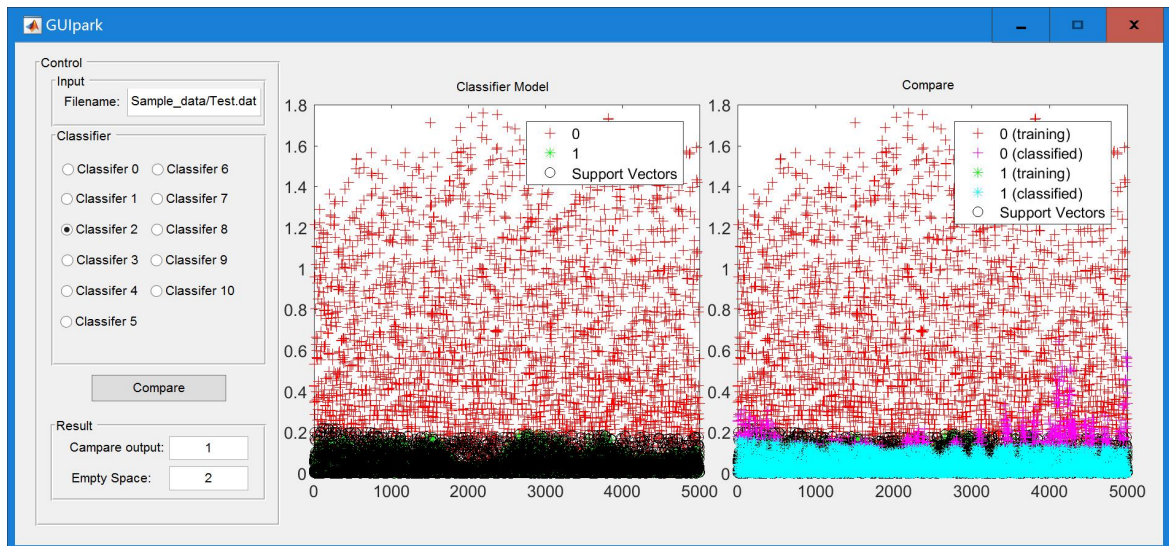


Figure 5.4: System Interface

The Fig. 5.4 above shows the interface of the prototypical WiFi parking space finding system. Firstly, you need to input the test CSI data and choose

classifier. Then click compare button. The figure on the left shows the train model which is the classifier that you selected. The right figure shows the comparison between the test data and trained model. If the testing data's pattern matches the trained model, the classifier of system will output 1 and the number of empty space will output. Otherwise, it will output 0. Each time, the system will begin from classifier 0. If the output of classifier is 0, the system will change to next classifier automatically until the output is 1.

If the scale of the parking lot is very large, for example, it has 100 empty parking lots. The amount of empty cases will be very large, if the code of each classifier is fixed, the work load will be very high and it will increase the cost. So in the future, we will develop a dynamic coding system to create classifier dynamically to reduce the cost and work load of the system

And also, we will study the rules of the phase variance between different empty space. For example, we will find the relationship between phase variance of 1 empty and 2 empty so that we can imply the other amount of empty space by this relationship. The cost will be reduced significantly by this approach.

Chapter 6

Result Analysis

In this section, the performance of the proposed approach is evaluated and the experimental results are analyzed. Our main task is to find empty parking slots. Hence, the classification accuracy for varying numbers of empty slots is important. The following will discuss on the classification accuracy in detail. In addition, the effect of using different receiver antennas is also discussed.

6.1 Classification Accuracy

The classification accuracy of this approach is presented in the form of a confusion matrix as shown in Fig. 6.1

The x-axis of the confusion matrix represents the actual class, and the y-axis indicates the predicted class. For example, a matrix entry at position (x, y) means the number of cases in percentage that input CSI data is classified into class y when the actual class is x.

First of all, it can be observed from the confusion matrix as shown in Fig. 6 that we can calculate the average classification accuracy by adding matrix entries at positions $(x, x), 0 \leq x \leq 10$ and dividing the result by 11, which is 80.8. The obtained average accuracy is not much impressive, compared with magnetic sensor-based approaches which have reported the average accuracy of 95% ~ 98% [5]. However, if we consider the high cost of the sensor-based methods, this approach is still a competitive solution especially for small businesses who may not be able to

afford the huge investment for parking lot management.

Furthermore, the results show that this approach is actually a very powerful solution if a tolerance of only a single empty slot is permitted. More specifically, we can observe from the results that this approach does not generate errors that are too much deviated from the ground truth. For example, while the accuracy for the class of 0 empty slot was just 70%, the majority of errors (*i.e.*, 22%) were associated with only differences of one or two empty slots compared with the ground truth. Fig. 6.2 exhibits the classification accuracy when errors of a single empty slot are tolerated.

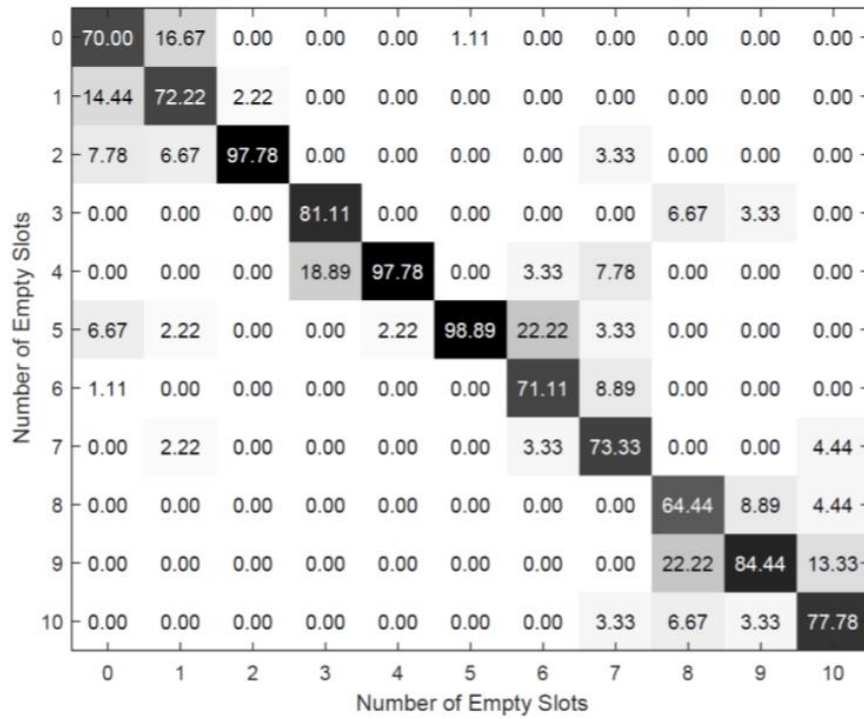


Figure 6.1: Confusion matrix that represents classification accuracy

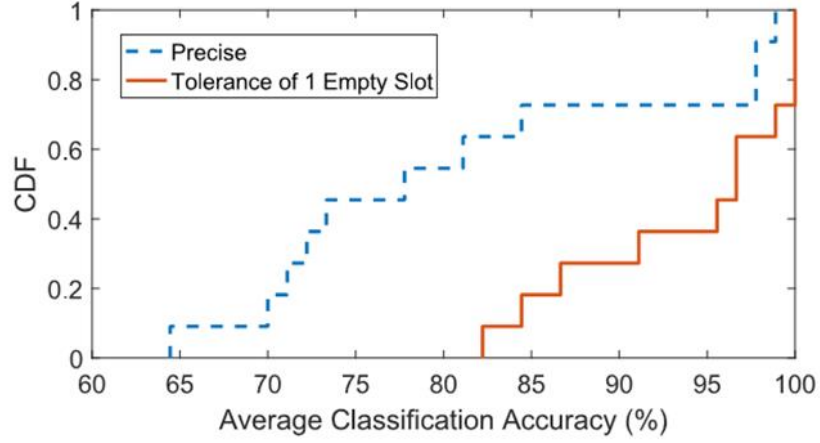


Figure 6.2: Classification accuracy with and without a tolerance of one empty slot

As shown, the average accuracy spiked up to 93.8%, which is quite comparable even with the sensor-based approach. In addition, the results suggest that if the operator is interested to provide only the information about whether the parking lot is full or not, the average accuracy is increased much higher up to 95.7%.

We attribute the low classification accuracy for certain number of empty slots to surrounding obstacles and moving objects that were present during CSI data collection. We suggest to address this problem by employing a directional antenna with a more strategic deployment of the transmitter receiver pair.

6.2 Effects of Receiver Antenna

The Intel 5300 WiFi NIC is equipped with three receiver antennas and one transmitter antenna, allowing for $2,500 \times 3$ samples to be collected with a single WiFi packet transmission assuming the sample rate of 2,500 samples/sec.

The question is whether or not to integrate all the samples received via the three antennas in training the classification model. Assuming the negligible distance between receiver antennas, theoretically, we should obtain very similar results regardless of receiver antennas.

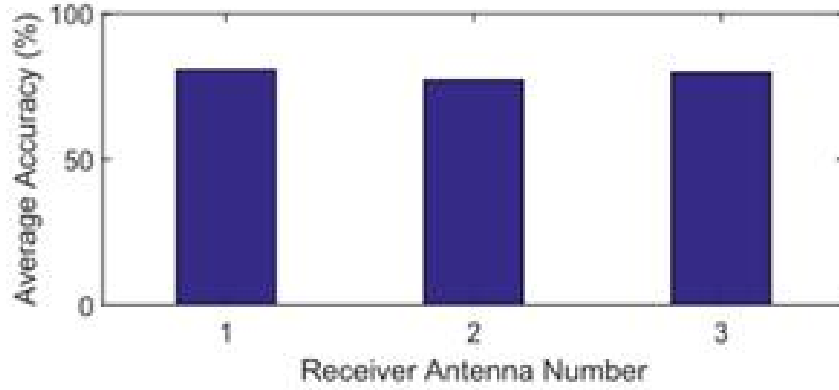


Figure 6.3: Classification accuracy for different receiver antenna

In our experiments, however, we found that aggregating the CSI data of all receivers resulted in lower classification accuracy, and that the average classification accuracy was different by up to 4.7% depending on which receiver antenna we used (as shown in Fig. 6.3).

We attribute the reason for the varying classification accuracy to the different hardware conditions of the receivers, potentially an issue specific to our equipment. Consequently, we used the CSI data received via the receiver antenna1.

Chapter 7

Conclusion

In this thesis, a new approach to finding empty parking slots is presented, which is a WiFi-based parking solution. The unique feature of WiFi CSI data for discerning the number of empty parking slots has been identified, which then was used to build a classification model for a machine learning algorithm that effectively detects parking occupancy. This approach demonstrates the potential of exploiting ubiquitous WiFi signals to implement a transportation tool leading to dramatically reduced cost without requiring development of specialized equipment nor intrusive deployment of it.

In addition, this research warrants a number of interesting future works. For example, the accuracy can be improved by using the directional antenna in order to minimize the impact of surrounding obstacles and moving objects. The classifier can be generated dynamically. An analysis on the effect of deployment position and posture of the WiFi transmitter and the receiver will be an interesting topic. Other useful features in addition to the phase variance can be identified to further improve the classification accuracy.

The future work will discover a compound features such as phase amplitude based features and signal intensity based features like RSS. Analyze the performance of these new features and apply these features on the future system to improve the accuracy of finding the number empty spaces.

Another important future work is to test the performance of this approach in

the real world .By testing more different and abnormal cases such as vehicle type, weather conditions to evaluate the robustness of this approach.

At last, the presented system will be integrated in the mobile device like smartphone. The driver can download the app of this system to receive the condition of empty spaces in parking lots by their smartphone. The information of empty spaces in parking lots is collected by the WiFi equipment in the parking lots and transmits to the main server by cellular mobile network. The main server will analysis and transmit the parking lots condition to smartphones.

References

- [1] D. C. Shoup, “Cruising for parking,” *Transport Policy*, vol. 13, no. 6, pp. 479-486, 2006.
- [2] “Parking problem,”
<http://www.jpost.com/Israel-News/New-Tech/Parking-the-problem-and-the-solution-396545>, accessed: 2017-09-01.
- [3] “Sf park project,” <http://sfpark.org/>, accessed: 2017-05-01
- [4] “Urbiotica,” <http://www.urbiotica.com/en/>, accessed: 2017-05-01.
- [5] “Fastprk,” <http://www.fastprk.com/>, accessed: 2017-05-01.
- [6] “Fastprk2,” <http://www.fastprk2.eu/the-project/>, accessed: 2017-05-01.
- [7] “Street line,” <http://www.streetline.com>, accessed: 2017-05-01.
- [8] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, “Parknet: drive-by sensing of road-side parking statistics,” in *Proc. of MobiSys*, 2010.
- [9] S. Soubam, D. Banerjee, V. Naik, and D. Chakraborty, “Bluepark: tracking parking and un-parking events in indoor garages,” in *Proc. Of ICDCN*, 2016.
- [10] P. Carnelli, J. Yeh, M. Sooriyabandara, and A. Khan, “Parkus: A novel vehicle parking detection system,” in *Twenty-Ninth IAAI Conference*, 2017.

- [11] M. Idris, Y. Leng, E. Tamil, N. Noor, and Z. Razak, “ park system: a review of smart parking system and its technology,” *Inf. Technol. J*, vol. 8, no. 2, pp. 101-113, 2009.
- [12] S. Nawaz, C. Efstratiou, and C. Mascolo, “Parksense: A smartphone based sensing system for on-street parking,” in *Proc. of MobiCom*, 2013.
- [13] J.-G. Krieg, G. Jakllari, H. Toma, and A.-L. Beylot, “Unlocking the smartphones senses for smart city parking,” in *Proc. of ICC*, 2016.
- [14] J. Cherian, J. Luo, H. Guo, S.-S. Ho, and R. Wisbrun, “Parkgauge: Gauging the occupancy of parking garages with crowdsensed parking characteristics,” in *Proc. of MDM*, 2016.
- [15] A. Nandugudi, T. Ki, C. Nuesle, and G. Challen, “Pocketparker: Pocketsourcing parking lot availability,” in *Proc. of PerCom*, 2014.
- [16] Y. Huang, Y.-C. Chen, C.-W. You, D.-X. Wu, Y.-L. Chen, K.-L. Hua, and J. Y.-J. Hsu, “Toward an easy deployable outdoor parking systemlessons from long-term deployment,” in *Proc. of PerCom*, 2017.
- [17] A. Hakeem, N. Gehani, X. Ding, R. Curtmola, and C. Borcea, “On the-fly curbside parking assignment,” in *Proceedings of the 8th EAI International Conference on Mobile Computing, Applications and Services. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)*, 2016.

- [18] N. Djuric, M. Grbovic, and S. Vucetic, "Parkassistant: An algorithm for guiding a car to a parking spot," in Transportation Research Board 95th Annual Meeting, pp. 16-24, 2016.
- [19] WU K, XIAO J, YI Y, "FILA: Fine-grained indoor localization," IEEE INFOCOM, pp. 2210-2218, 2012.
- [20] WILSON J, PATWARI N, "See through walls: Motion tracking using variance-based radio tomography networks," IEEE Transactions on Mobile Computing vol. 10, no. 5, pp.612-621,2011.
- [21] NEGUIZIAN C, NEGUIZIAN V, "Indoor fingerprinting geolocation using wavelet-based features extracted from the channel impulse response in conjunction with an artificial neural network," IEEE International Symposium on Industrial Electronics, pp. 2028-2032 2007
- [22] PATWARI N, KASERASK, "Robust location distinction using temporal link signatures," ACM MobiCom, pp. 111-122,2007.
- [23] ZHANG J,FIROOZ M H,PATWARI N, "Advancing wireless link signatures for location distinction," ACM MobiCom, pp. 26-37, 2008.
- [24] HALPERIN D, HU W,SHETH A, "Tool release: Gathering 802.11n traces with channel state information," ACM SIGCOMM Computer Communication review, vol. 41, no. 1, p. 53 , 2011.

- [25] XIAO J, WU K, YI Y, “FIMD: Fine-grained device free motion detection,”
IEEE ICPADS, pp. 229-235,2012.
- [26] ZHOU Z, YANG Z, WU C “Towards omnidirectional passive human detection,”
IEEE INFOCOM,pp.3057 3065,2013.
- [27] RUBNER Y, TOMASI C, GUIBAS L J, “The earth mover’s distance as a
metric for image retrieval, Springer International Journal on Computer, Vision,”
vol.40, no.3, : pp. 99 -1212000
- [28] QIAN K WU C YANG Z “PADS: Passive detection of moving targets with
dynamic speed using PHY layer information,” IEEE ICPADS,pp. 1-8,2014.
- [29] XIAO J, WU K, YI Y, “Pilot: Passive device-free indoor localization using
channel state information,” IEEE ICDCS, pp.236-245,2013.
- [30] ABDEL-NASSER H, SAMIR R, SABEK I, “MonoPHY: Monostream-based
devicefree WLAN localization via physical layer information,” IEEE WCNC,
pp.4546-4551,2013.
- [31] SABEK I, YOUSSEF M, “MonoStream:A minimal-hardware high accuracy
device-free WLAN localization system,” [http: arxiv.org/abs/1308.0768](http://arxiv.org/abs/1308.0768), 2013.
- [32] ZHOU Z, YANG Z, WU C “LiFi:Line-Of-Sight identification with WiFi,” IEEE
INFOCOM,pp.2688 -2696,2014.
- [33] WU C, YANG Z, ZHOU Z, “PhaseU: ealtime LOS identification with WiFi,”
IEEE INFOCOM, pp.2038 -2046,2015.

- [34] WANG Y, LIU J, CHEN Y, “E-eyes: Devicefree location-oriented activity identification using fine-grained WiFi signatures,” ACM MobiCom, pp. 617-628, 2014.
- [35] HAN C, WU K, WANG Y, “WiFall: Devicefree fall detection by wireless networks,” IEEE INFOCOM, pp. 271-279, 2014.
- [36] XI W, ZHAO J, LI X, “Electronic frog eye: Counting crowd using WiFi,” IEEE INFOCOM, pp. 361-369, 2014.
- [37] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, “Keystroke recognition using wifi signals,” in Proc. of MobiCom.
- [38] J. Zhang, B. Wei, W. Hu, and S. S. Kanhere, “Wifi-id: Human identification using wifi signal,” in Proc. of DCOSS, 2016.
- [39] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, “E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures,” in Proc. of Mobicom, 2014.
- [40] M. Won, S. Zhang, and S. H. Son, “Witrafic: Low-cost and non-intrusive traffic monitoring system using wifi,” in Proc. of ICCCN, 2017.
- [41] Y. Zeng, P. H. Pathak, C. Xu, and P. Mohapatra, “Your ap knows how you move: fine-grained device motion recognition through wifi,” in Proc. of HotWireless, 2014.

- [42] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 Packet Delivery from Wireless Channel Measurements," in ACM SIGCOMM, 2010.
- [43] F. Benedetto, G. Giunta, A. Toscano, and L. Vegni, "Dynamic LOS/NLOS Statistical Discrimination of Wireless Mobile Channels," in IEEE VTC, 2007.
- [44] C. Wu, Z. Yang, Z. Zhou, K. Qian, Y. Liu, and M. Liu, "Phaseu: Realtime los identification with wi," in Proc. of INFOCOM, 2015.
- [45] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the mona lisa: spot localization using phy layer information," in Proc.of MobiSys, 2012.
- [46] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," ACM transactions on intelligent systems and technology (TIST), vol. 2, no. 3, p. 27, 2011
- [47] –, "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 27:127:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [48] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," IEEE transactions on Neural Networks, vol. 13, no. 2, pp. 415-425, 2002.
- [49] "<http://slideplayer.com/slide/1448980/>"