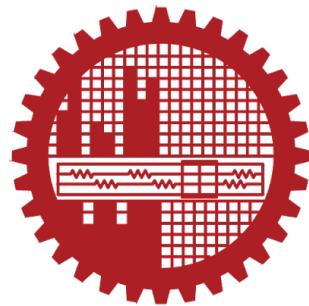


**DESIGNING EFFICIENT AND
IMPLEMENTATION-FRIENDLY CONVOLUTION
NEURAL NETWORK FOR SIGN LANGUAGE
RECOGNITION USING WIFI CSI DATA**

by

Joy Saha
1706189

Bachelor of Science
IN
ELECTRICAL AND ELECTRONIC ENGINEERING



Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh

May, 2023

Certification

On the 16th of May 2023, Joy Saha achieved a significant academic milestone in his pursuit of a Bachelor of Science degree in Electrical and Electronic Engineering. His thesis, entitled "Enhancing Sign Language Recognition through CSI Data from Wifi Using Convolutional Neural Networks," was accepted as satisfactory, marking the culmination of his diligent and arduous research efforts. Joy Saha's unwavering dedication and hard work in his chosen field of study led to the successful approval of his thesis. His research showcased his proficiency in innovative technologies such as convolutional neural networks, which have the potential to transform the field of sign language recognition. This accomplishment positions Joy Saha as a potential future leader in the field and reflects his passion for Electrical and Electronic Engineering.

Supervisor:

Dr. Hafiz Imtiaz
Associate Professor
BUET

Co-supervisor:

Dr. Tahsina Farah Sanam
Assistant Professor
BUET

Candidate's Declaration

This is to certify that the work presented in this thesis entitled, “Designing Efficient And Implementation-Friendly Convolution Neural Network For Sign Language Recognition Using WiFi CSI Data”, is the outcome of the research carried out by Joy Saha under the supervision of Dr. Hafiz Imtiaz, Associate Professor, Department of Electrical and Electronic Engineering (EEE), Bangladesh University of Engineering and Technology (BUET), Dhaka-1000, Bangladesh. and Dr. Tahsina Farah Sanam, Assistant Professor, Institute of Appropriate Technology at Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh.

It is also declared that neither this thesis nor any part of this has been submitted anywhere else for the award of any degree, diploma, or other qualifications.

Signature of the Candidate

Joy Saha
1706189

Dedication

It is with a heavy heart that I express my dedication to my thesis in memory of my beloved uncle, who has passed away. Throughout my academic journey, his unwavering love, unfaltering assistance, and encouragement have served as a source of inspiration for me. Although he is no longer with us, his belief in my abilities and his guidance continue to guide me and shape my accomplishments.

As I reflect on my achievements, I cannot help but recognize the profound impact my uncle has had on my life. He played a significant role in shaping the person I am today, and his absence is deeply felt. I honor his memory and wish that he could have witnessed this important milestone in my life.

It is my hope that this dedication serves as a tribute to his life and legacy, and that his unwavering support and love continue to inspire me as I embark on future endeavors. His memory will forever be cherished, and his influence on my life will never be forgotten.

Acknowledgement

I am immensely grateful to have had the opportunity to work under the guidance of two exceptional mentors and supervisors, Dr. Hafiz Imtiaz and Dr. Tahsina Farah Sanam. Their extensive expertise and knowledge in the field of CSI-based sign language detection have been instrumental in shaping me into a proficient researcher. Their consistent encouragement, guidance, and support have been pivotal in my academic journey.

Dr. Hafiz Imtiaz and Dr. Tahsina Farah Sanam are renowned experts in the field of signal processing and machine learning, and their guidance has been invaluable in my research endeavors. Their keen insights and vast experience in this field have helped me to navigate through the complexities of CSI-based sign language detection with ease.

Their dedication towards ensuring my academic success has been unwavering, and I am grateful for the numerous opportunities that they have provided me with to enhance my research skills. Their mentorship has been an exceptional learning experience, and I am confident that the knowledge and skills I have gained under their tutelage will serve me well in my future research endeavors.

Once again, I extend my sincere appreciation and thanks to Dr. Hafiz Imtiaz and Dr. Tahsina Farah Sanam for their guidance, support, and expertise, which have been critical in shaping me into a capable researcher.

Contents

Certification	ii
Candidate's Declaration	iii
Dedication	iv
Acknowledgement	v
List of Figures	viii
List of Tables	x
Abstract	xi
1 Introduction	2
1.1 Related Work	3
1.2 Technical Challenges in CSI based Sign Language Detection	4
1.3 Research Motivation and Contributions	6
1.4 Thesis Organization	7
2 Preliminaries	8
2.1 Multi-path Propagation Model	8
2.2 Overview of MIMO-OFDM Channel	9
2.3 Overview of Channel State Information (CSI)	11
3 CSI Data Analysis	12
3.1 Multiple Antenna Data Analysis	12
3.2 Subcarriers	13
3.3 Effect of Subject	15
3.4 Correlation between Sub-carriers	17
4 CSI Data Preprocessing	20
4.1 Amplitude Correction	21
4.2 Phase Correction	23

4.3 Scaling The CSI Data	29
5 Sign Gesture Recognition Models	31
5.1 Reference Models	31
5.1.1 Support Vector Machine (SVM)	31
5.1.2 Random Forest (RF)	32
5.1.3 K-Nearest Neighbors Algorithm (KNN)	33
5.1.4 Gaussian Naive Bayes	34
5.1.5 Artificial Neural Network	35
6 CNN Implementation and Performance Evaluation	37
6.1 CNN Implemenation	37
6.1.1 Gesture Recognition Algorithm and CNN Layers	37
6.1.2 Input Layer	40
6.1.3 Convolution Layer	41
6.1.4 Batch Normalization Layer	42
6.1.5 ReLu Layer	44
6.1.6 Pooling Layer	46
6.1.7 Dropout Layer	47
6.1.8 Fully Connected Layer	48
6.1.9 Classification Layer	50
6.2 Performance Evaluation	51
7 Conclusion	53
8 Future Directions	54
References	56

List of Figures

1.1	Comparison of different sign language recognition technologies	3
1.2	CSI Amplitude before and after Noise reduction	5
1.3	CSI Access Point(AP) and Data collection	5
1.4	24 Static Hand Gestures for American Sign Language Letters	6
2.1	Multi-path Propagation Model	9
2.2	Illustration of three MIMO concepts	10
3.1	CSI Amplitude and Phase Variation with antenna index	13
3.2	CSI Amplitude and Phase Variation with Sub-Carrier	14
3.3	CSI Amplitude and Phase Variation with Subject	15
3.4	CSI Amplitude and Phase Variation with Environment	16
3.5	Floor plan and measurement settings of the lab and home environments	17
3.6	Amplitude Correlation between the Sub-carriers	18
3.7	Phase Correlation between the Sub-carriers	18
4.1	Effect of window size in MAF	21
4.2	Correlation Coefficient After and before MAF	22
4.3	Correlation Coefficient Heatmap After and before MAF	23
4.4	Effect of Amplitude with MAF	24
4.5	Correlation Coefficient Heatmap After and before Phase Correction . .	25
4.6	Effect of Phase Correction	27
4.7	Phase correction effect on Antenna	28
4.8	Phase correction effect comparison	29
4.9	Scaling Technique Performance Comparison	30
5.1	SVM Performance	32
5.2	Random Forest Classifier Performance	33
5.3	KNN Performance	34
5.4	GNB Performance Evaluation	35
5.5	Artifical Neural network	36

6.1	Architecture of the CNN network	39
6.2	An example of two-dimensional convolution	41
6.3	Impact of BatchNormalization Layer	44
6.4	Different type of ReLu function	44
6.5	Perfromance of Different ReLu function	45
6.6	Pooling Layer Operation	46
6.7	Perfromance of Pooling Layer	47
6.8	Perfomrance w/ and w/o Dropout Layer	48
6.9	Performance Dip with Fully Connected Layer	49
6.10	Increase of Performance using top N prediction	50

List of Tables

6.1	The Recognition Accuracies of the Proposed Model vs Other Approaches	51
6.2	Comparison of different Sign Language Recognition Technologies . . .	52

Abstract

SignFi is an innovative technology that utilizes WiFi signals to accurately recognize a wide range of sign language gestures, including those involving the head, arm, hand, and fingers. With a recognition capability for 276 sign gestures, SignFi surpasses existing WiFi-based systems and offers enhanced gesture diversity. By collecting Channel State Information (CSI) measurements and applying pre-processing techniques, SignFi achieves accurate sign gesture classification using a Convolutional Neural Network (CNN). It has been evaluated in lab and home environments, achieving impressive average recognition accuracies of 97.64% and 98.85% respectively. In a separate evaluation involving five users, SignFi achieved a recognition rate of 84.63% for 7,500 instances of 150 sign gestures in a lab setting. This technology has significant potential for improving accessibility for individuals relying on sign language as their primary mode of communication. SignFi showcases the successful fusion of wireless signals and machine learning, opening doors for innovative applications in sign language recognition.

Chapter 1

Introduction

Sign language is an essential mode of communication for millions of people around the world who are deaf or hard of hearing. According to World Deaf Federation the number is roughly 70 million. Many hearing people also use it as their First or Second language. It is noteworthy that ASL is gaining recognition as an important language, with increasing numbers of people choosing to learn it. This is demonstrated by the consistent rise in ASL enrollments from 2002 to 2013, as reported by a survey conducted by the Modern Language Association [1]. It is particularly encouraging that many colleges and universities are accepting ASL as a foreign language credit, which can have positive implications for both deaf and hard-of-hearing individuals and hearing individuals seeking to communicate with and understand this population. However, traditional methods of sign language detection and recognition are often limited in accuracy and accessibility, creating a significant barrier for individuals who rely on sign language as their primary means of communication [2]. A sign language recognition system can help break the communication barrier between the Deaf community and those unfamiliar with sign language. Recent advancements in wireless technology have paved the way for new and innovative approaches to sign language detection and recognition. One such approach is based on Channel State Information (CSI), which refers to the wireless signal characteristics that change as a result of the physical movement of an object or person.

CSI-based sign language detection uses the unique features of CSI measurements to identify and recognize sign language gestures with a high degree of accuracy. By capturing wireless signals during the performance of sign language gestures, CSI-based detection provides a non-invasive and intuitive means of sign language recognition that can greatly improve accessibility for individuals who rely on sign language.

This technology has the potential to revolutionize the way sign language is detected

and recognized, paving the way for more accessible and accurate communication for individuals who are deaf or hard of hearing. In this paper, we present SignFi, a novel CSI-based sign language detection technology that utilizes Convolutional Neural Networks (CNNs) for gesture classification. We evaluate the performance of SignFi in both lab and home environments and demonstrate its ability to accurately recognize a wide range of sign language gestures, including those involving the head, arm, hand, and fingers.

1.1 Related Work

While there are existing sign language recognition systems that utilize cameras [3], Kinect [4–6], Leap Motion [7–10], and gloves with motion sensors [11], these methods have limitations such as being subject to lighting conditions or being intrusive. In addition, some systems can only recognize finger gestures or are highly sensitive to the distance and displacement of the sensors and human.

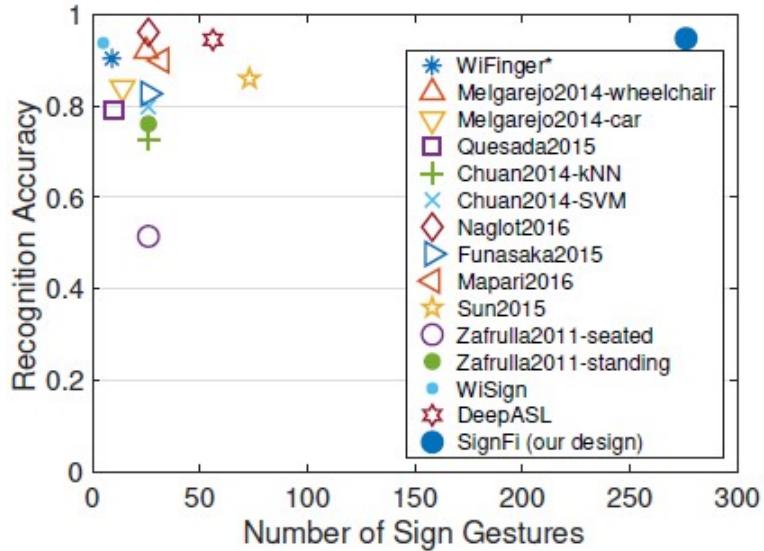


Figure 1.1: Comparison of different sign language recognition technologies

Based on the plot presented in the paper [12], it is evident that our proposed model, Signfi, outperforms other models in terms of accuracy for both small and large number of classes. This indicates that our model is not only capable of handling datasets with a limited number of classes but also scales well to datasets with a larger number of classes. It is important to note that other models may have performed satisfactorily for small number of classes, but they have not been tested on datasets with a larger number

of classes. Overall, the results demonstrate the effectiveness and versatility of Signfi as a classification model for datasets with varying numbers of classes.

To address these limitations, some researchers have explored using Channel State Information (CSI) of WiFi to recognize hand [12–15] and finger [16, 17] gestures in a non-intrusive way, with some success in recognizing simple American Sign Language (ASL) gestures. However, recognizing a larger set of nearly 300 basic sign gestures frequently used in daily life remains a challenge.

Existing classification algorithms, such as k-Nearest Neighbor (kNN) with Dynamic Time Wrapping (DTW) [16, 17], have low recognition accuracy and are computationally expensive when dealing with a large number of classes. For example, when tested in a lab environment using CSI traces of 276 sign gestures, the average recognition accuracy of kNN with DTW was only 68%. Moreover, the testing stage took an extremely long time when dealing with nearly 300 possible classes. Therefore, new classification algorithms are needed for sign gesture recognition using WiFi that can provide high recognition accuracy and low computational cost during testing.

1.2 Technical Challenges in CSI based Sign Language Detection

CSI (Channel State Information) based sign language detection is a promising approach that uses wireless signals to recognize sign language gestures. However, it also presents some technical challenges that need to be addressed for its successful implementation. Some of these challenges are:

1. **Noise and interference:** Wireless signals can be affected by environmental factors, such as interference from other devices, multipath propagation, and fading. This can introduce noise and distortions in the CSI measurements, making it harder to accurately detect sign language gestures as shown in fig[1.2].
2. **Limited CSI availability:** CSI measurements are usually limited to the access points (APs) or Wi-Fi devices(fig[1.3]) that are in range of the sign language user. This can result in sparse and incomplete CSI data, especially in crowded areas where multiple users are using the same wireless network.

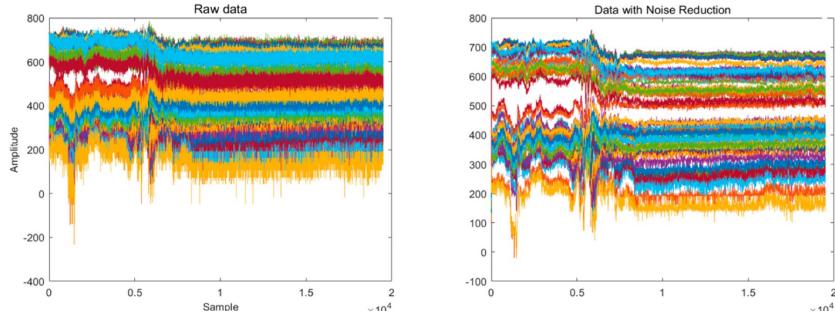


Figure 1.2: CSI Amplitude before and after Noise reduction each sub carrier is represented by a color. [18]

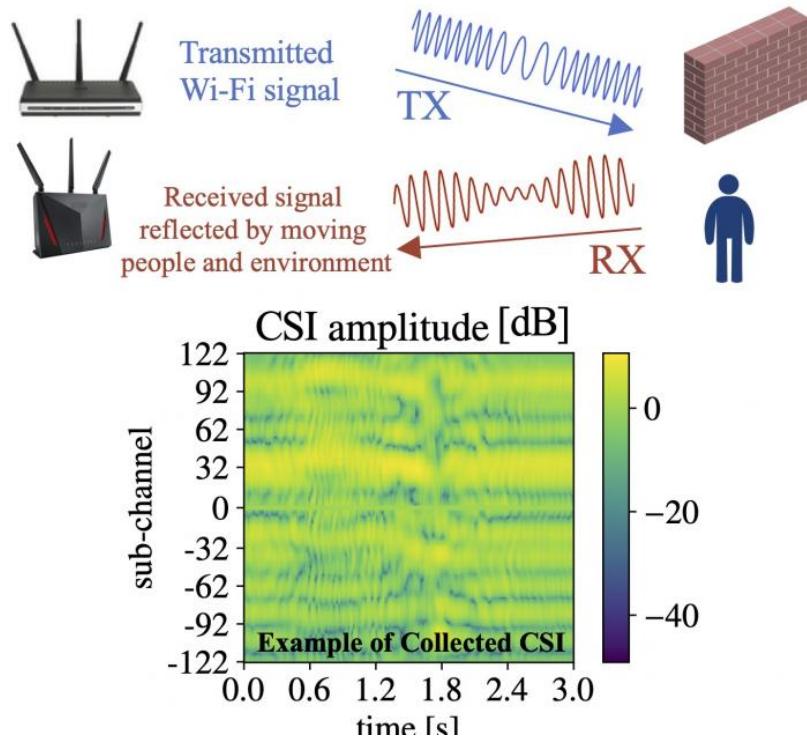


Figure 1.3: CSI Access Point(AP) and Data collection [19]

3. **Data preprocessing:** The raw CSI data needs to be pre-processed before it can be used for sign language detection. This involves filtering out noise, correcting phase offsets, and normalizing the data. This preprocessing step can be time-consuming and requires specialized knowledge in signal processing. The pre-processing part is shown in **Chapter[4]**.
4. **Gesture variability:** Sign language gestures can vary in shape, size, and speed, depending on the individual user and the context. This can make it challenging to develop a robust CSI-based sign language detection system that can accurately recognize a wide range of gestures. Fig[1.4] shows variation of hand,finger ges-

tures with different words.



Figure 1.4: 24 Static Hand Gestures for American Sign Language Letters
[20]

5. Privacy concerns: CSI-based sign language detection requires access to wireless signals, which can raise privacy concerns. Users may be uncomfortable with the idea of their wireless signals being used to detect their sign language gestures, especially if they are not aware of the process or have not explicitly given their consent.

Addressing these technical challenges requires a combination of signal processing techniques, machine learning algorithms, and user-centered design principles. Researchers need to develop robust and accurate models that can handle noisy and incomplete CSI data while addressing user privacy concerns. Additionally, designing user-friendly interfaces that can provide real-time feedback to sign language users is also important to ensure the success of CSI-based sign language detection systems.

1.3 Research Motivation and Contributions

CSI (Channel State Information) based sign language detection is a relatively new and promising area of research that aims to use wireless signals to recognize sign language gestures. The motivation for this research is to provide an alternative to existing sign language recognition techniques, which often rely on expensive and complex sensors or require physical contact with the user.

CSI-based sign language detection has several potential advantages over traditional approaches. It can be performed using off-the-shelf Wi-Fi devices, making it more affordable and accessible to a wider range of users. It can also be performed in real-time and without requiring physical contact, making it more convenient and comfortable for users.

The contribution of research in CSI-based sign language detection is twofold:

1. **Technical Contribution:** Research in this area aims to develop accurate and reliable models that can detect sign language gestures using CSI measurements. This involves developing algorithms and techniques that can preprocess the raw CSI data, extract relevant features, and classify the gestures using machine learning models.
2. **Social Contribution:** The ultimate goal of research in CSI-based sign language detection is to provide a more inclusive and accessible communication technology for the deaf and hard-of-hearing community. By enabling real-time communication in sign language through Wi-Fi networks, it can help to break down barriers to communication and improve the quality of life for sign language users.

CSI-based sign language detection has the potential to improve the lives of deaf individuals and promote inclusivity. It enables better communication, fosters diversity, and creates new opportunities in the tech industry. Investing in research and development in this field is vital for a more accessible and equitable society.

1.4 Thesis Organization

The remaining sections of this thesis are organized as follows:

In Chapter 2, the initial concepts of the research are presented. The analysis of CSI data, including its visualization and associated effects, is discussed in Chapter 3. Chapter 4 outlines the algorithms used for data preprocessing. Several classification methods are introduced in Chapter 5. The implementation and performance evaluation of the CNN model are described in Chapter 6. Chapter 6 provides a comprehensive performance evaluation of our proposed model, comparing it with existing models and related work. Finally, Chapter 7 offers a conclusion of our work, while Chapter 8 discusses future work.

Chapter 2

Preliminaries

2.1 Multi-path Propagation Model

Multipath propagation is a phenomenon that occurs when wireless signals are reflected, refracted, or diffracted by objects in the propagation environment, resulting in multiple versions of the signal arriving at the receiver. This can cause signal interference, distortion, and fading, as shown in fig[2.1], which can affect the performance of wireless communication systems.

A multipath propagation model is a mathematical representation of the physical processes that give rise to multipath propagation. These models describe how the electromagnetic waves propagate through the environment, interact with objects, and arrive at the receiver. Multipath propagation models are essential for designing and optimizing wireless communication systems, as they allow engineers to predict the behavior of wireless signals in different propagation environments. One common multipath propagation model is the Rayleigh fading model, which assumes that the multipath components of a wireless signal follow a Rayleigh distribution. This model is widely used in wireless communication systems because it is simple and can be easily implemented. However, it does not account for the effects of line-of-sight (LOS) paths, which can be important in some propagation environments.

Another widely used multipath propagation model is the Rician fading model, which includes both LOS and non-LOS components in the signal. This model is more complex than the Rayleigh fading model but provides a more accurate representation of the signal behavior in certain propagation environments.

In addition to these models, there are many other multipath propagation models that have been developed to describe specific propagation environments or wireless commu-

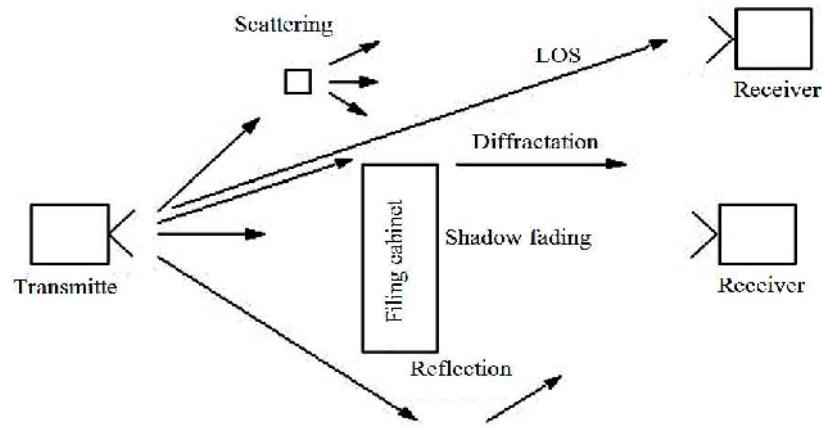


Figure 2.1: Multi-path Propagation Model
[21]

nication scenarios. These models can range from simple empirical models to complex physics-based models, depending on the level of accuracy required for the specific application.

Overall, multi-path propagation models play a critical role in designing and optimizing wireless communication systems, and their accuracy is essential for ensuring reliable and efficient wireless communication in a wide range of applications.

2.2 Overview of MIMO-OFDM Channel

MIMO-OFDM (Multiple Input Multiple Output-Orthogonal Frequency Division Multiplexing) is a wireless communication technology that combines MIMO and OFDM to improve the efficiency and reliability of wireless communication systems. In a MIMO-OFDM channel, multiple antennas are used at both the transmitter and receiver, and the data is transmitted and received using multiple subcarriers.

By leveraging spatial, frequency, and temporal diversities(fig[2.2]), MIMO-OFDM systems can improve the system's performance in terms of data rate, coverage, reliability, and resistance to fading and interference. These diversity techniques help mitigate the impact of wireless channel impairments, resulting in more reliable and efficient wireless communication. These advantages are achieved through several key features of the MIMO-OFDM channel:

1. **Spatial Diversity:** Multiple antennas are used at both the transmitter and receiver to create multiple spatial paths for the transmission of data. This provides spatial diversity, which improves the reliability of the wireless communication system

by reducing the effects of fading and interference.

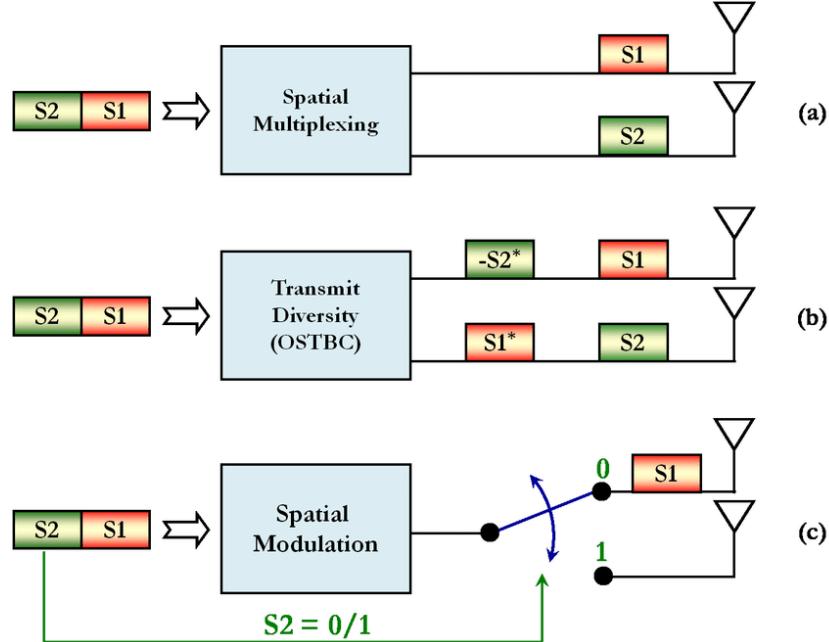


Figure 2.2: Illustration of three MIMO concepts
: (a) spatial-multiplexing;(b)transmit-diversity; and(c) spatial modulation [22]

2. **Frequency Diversity:** The data is transmitted and received using multiple sub-carriers in the OFDM system, which provides frequency diversity. This improves the reliability of the wireless communication system by reducing the effects of frequency-selective fading.
3. **Interference Reduction:** MIMO-OFDM technology can use advanced signal processing techniques to reduce interference and increase the signal-to-noise ratio. This improves the overall reliability and efficiency of the wireless communication system.
4. **Beamforming:** MIMO-OFDM technology can use beamforming techniques to improve the efficiency and reliability of wireless communication by focusing the transmission and reception of signals towards specific antennas or locations.

Overall, MIMO-OFDM technology is a powerful tool for improving the efficiency and reliability of wireless communication systems, and it is widely used in many modern wireless communication standards, such as Wi-Fi, 4G LTE, and 5G.

2.3 Overview of Channel State Information (CSI)

Channel State Information (CSI) is a critical parameter in wireless communication systems that characterizes the properties of the wireless channel between the transmitter and receiver. It is typically expressed in terms of a complex channel matrix that describes the amplitude and phase of the wireless channel for each antenna element.

The channel matrix H can be expressed as:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdot & \cdot & \cdot & \cdot & h_{1M} \\ h_{21} & h_{22} & \cdot & \cdot & \cdot & \cdot & h_{2M} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ h_{N1} & h_{N2} & \cdot & \cdot & \cdot & \cdot & h_{NM} \end{bmatrix}$$

where $N_t=M$ is the number of transmit antennas, $N_r=N$ is the number of receive antennas, and h_{ij} represents the complex gain of the wireless channel between the i th receive antenna and j th transmit antenna.

CSI is typically estimated using pilot signals that are transmitted from the transmitter and received by the receiver. The received pilot signals are used to estimate the channel matrix, which can then be used to equalize the data signals and improve the reliability and efficiency of the wireless communication system.

One common metric used to evaluate the quality of the estimated CSI is the Mean Square Error (MSE) between the estimated channel matrix and the true channel matrix. The MSE is defined as:

$$MSE = E \left[\left\| H_{est} - H_{true} \right\|^2 \right]$$

where H_{est} is the estimated channel matrix and H_{true} is the true channel matrix.

CSI can be used for a variety of purposes in wireless communication systems, including beamforming, spatial multiplexing, and interference management. It is a critical parameter in modern wireless communication standards such as 4G LTE and 5G, and its accurate estimation and use are essential for achieving reliable and efficient wireless communication.

Chapter 3

CSI Data Analysis

3.1 Multiple Antenna Data Analysis

CSI data analysis is a crucial aspect of wireless communication system design and optimization. It involves the extraction of information about the wireless channel's amplitude, phase, and frequency response, and the evaluation of its quality and reliability.

In wireless communication systems with multiple antennas, the CSI data can vary with the antenna index, as different antennas can experience different levels of multi-path fading, polarization, or other environmental factors. To account for this variation, antenna diversity techniques such as MIMO, space-time coding, or beamforming can be used to improve signal quality and reduce the effects of channel variations.

Multiple antenna CSI data analysis can be performed using various techniques, such as PCA or SVD, which can extract the most significant components of the data and reduce its dimensionality. It can also involve the evaluation of metrics such as the MSE or SNR, which can help assess the accuracy and reliability of the CSI estimation and identify sources of error or interference.

The insights gained from multiple antenna CSI data analysis can be used to optimize the performance and reliability of wireless communication systems. For example, it can be used to design antenna arrays or beam-forming algorithms that maximize signal strength or minimize interference, optimize power allocation or modulation schemes based on the channel's properties, or reduce the effects of channel variations using diversity techniques.

Two plots are given in fig[3.1] for Sign Word '**'Delicious'**' and '**'Sorry'**' which gives us an idea about the variation of Amplitude and Phase of the CSI data with antenna. These variations impact system performance and require techniques such as channel estima-

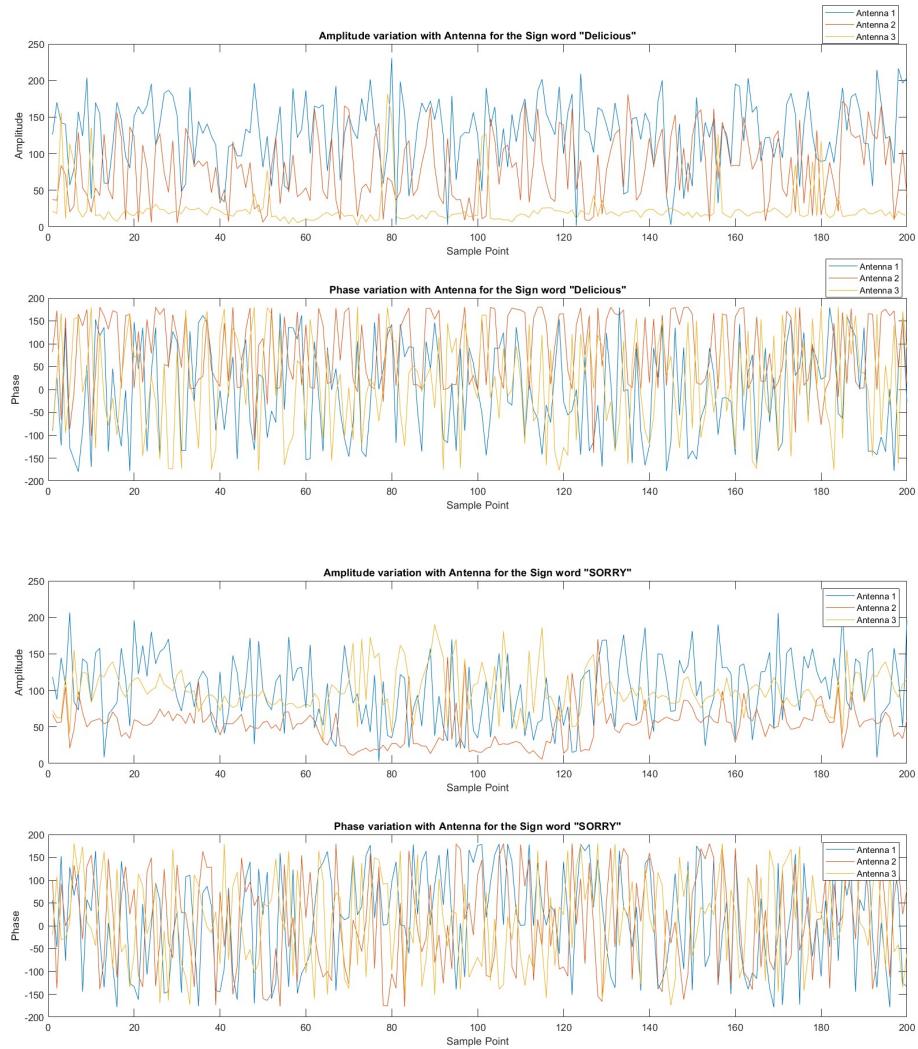


Figure 3.1: CSI Amplitude and Phase Variation with antenna index

tion and calibration to mitigate their effects.

In short, multiple antenna CSI data analysis is a powerful tool for understanding the properties of a wireless channel and designing optimized communication systems that are efficient, reliable, and robust.

3.2 Subcarriers

In wireless communication systems that use OFDM (Orthogonal Frequency Division Multiplexing) modulation, the transmitted data is divided into multiple sub-carriers that are transmitted simultaneously. The sub-carriers are spaced apart at regular intervals and are orthogonal to each other, which allows for efficient use of the available bandwidth and minimizes interference between the sub-carriers.

In CSI data, each sub-carrier can be considered as an independent channel, and the channel state information for each sub-carrier can be estimated separately. This enables finer-grained analysis of the wireless channel and can provide insights into the frequency-dependent properties of the channel. The CSI data for each sub-carrier can

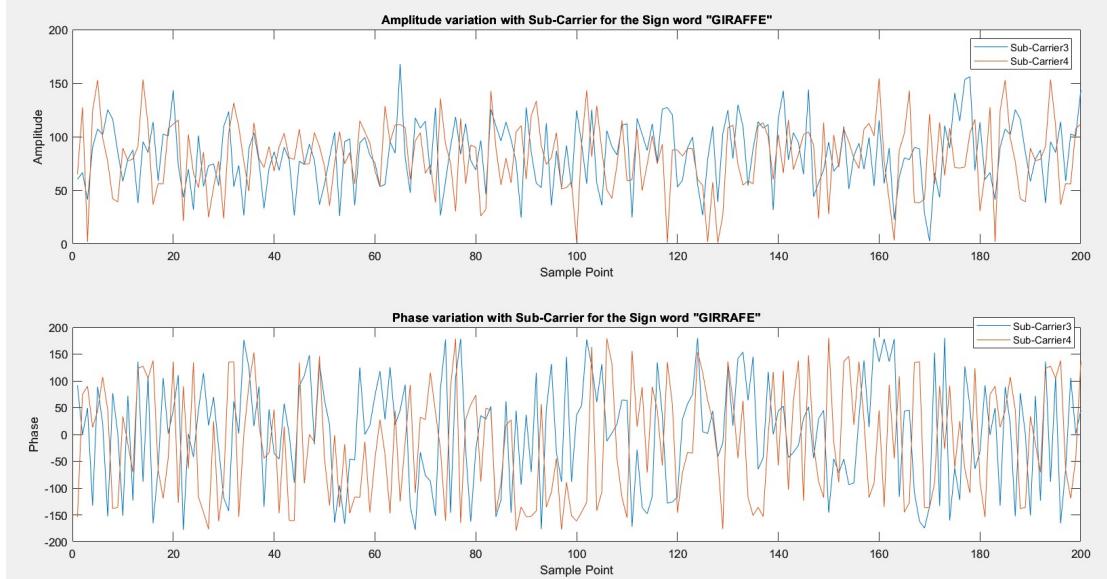


Figure 3.2: CSI Amplitude and Phase Variation with Sub-Carrier

be represented as a complex number that describes the amplitude and phase of the wireless channel for that sub-carrier. The CSI data for all sub-carriers can be organized into a matrix that represents the frequency response of the wireless channel.

By analyzing the CSI data for each sub-carrier, it is possible to identify frequency-dependent phenomena such as multi-path propagation, frequency-selective fading, and interference. This information can be used to design equalization algorithms that compensate for these effects and improve the reliability and efficiency of the wireless communication system.

In wireless communication systems that use Orthogonal Frequency Division Multiplexing (OFDM), the amplitude and phase can vary across different subcarriers.

In fig[3.2], we can see the variation of amplitude and phase for the sign word 'Giraffe' for sub-carrier 3 and 4.

Amplitude variation refers to changes in the magnitude or power level of the signal across subcarriers. This variation can be caused by various factors such as channel fading, interference, or frequency-selective effects. Due to these variations, the received signal power may differ for different subcarriers, resulting in amplitude differences.

Phase variation, on the other hand, relates to changes in the phase angle of the signal

across subcarriers. This variation can be caused by differences in propagation delays, frequency offsets, or multipath effects. As a result, the phase of the received signal can exhibit differences between subcarriers.

Both amplitude and phase variations across subcarriers can have an impact on system performance. In OFDM, equalization techniques such as adaptive filtering or frequency-domain equalization are employed to compensate for these variations and restore the original signal.

It's important to note that the specific amplitude and phase variations with subcarriers can vary depending on the wireless channel conditions, modulation scheme, and the presence of any impairments or interference.

3.3 Effect of Subject

In CSI data, the subject refers to the object or person that is present in the environment where the wireless communication system is operating. The subject can have a significant impact on the wireless channel and can affect the accuracy and reliability of the CSI data.

For example, the presence of a human body or other objects in the environment can cause attenuation or reflection of the wireless signal, leading to changes in the amplitude and phase of the CSI data. This can result in errors or interference in the wireless communication system and can affect the performance and reliability of the system.

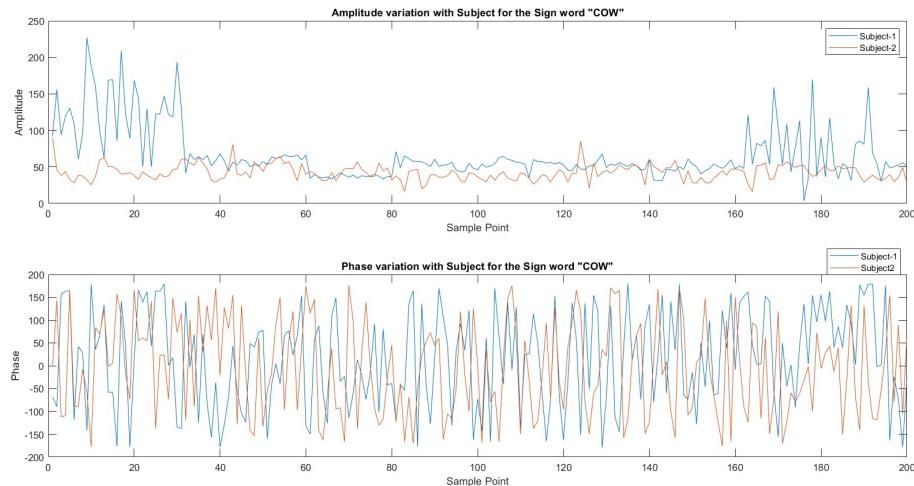


Figure 3.3: CSI Amplitude and Phase Variation with Subject

;

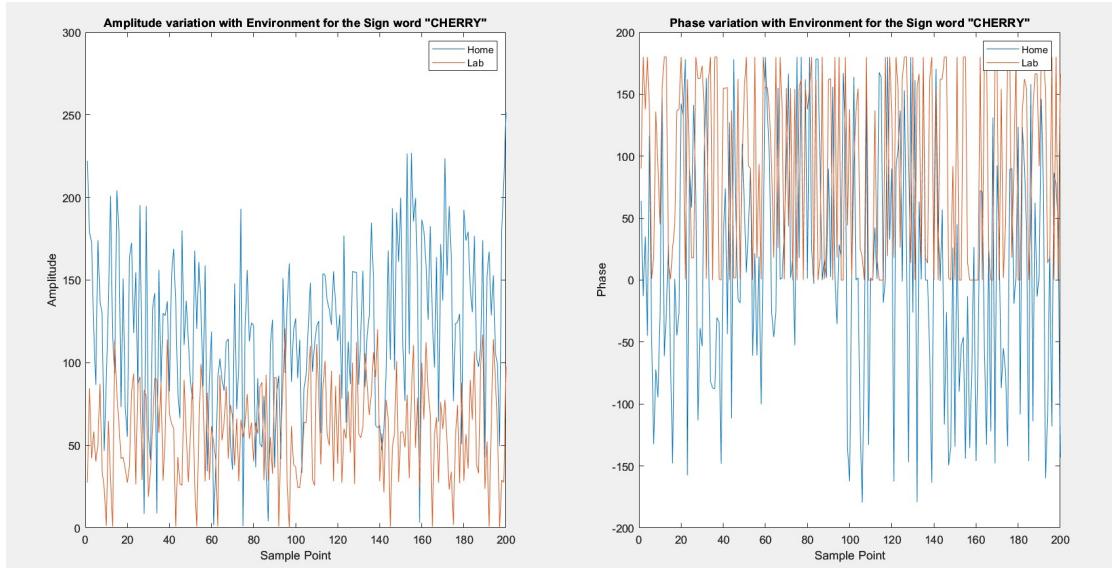


Figure 3.4: CSI Amplitude and Phase Variation with Environment

;

The effect of the subject on the CSI data can vary depending on factors such as the distance between the subject and the antennas, the size and shape of the subject, and the orientation of the subject relative to the antennas. For example, a large object such as a wall or furniture may cause more attenuation and reflection than a smaller object such as a person. The CSI data vary with subject(fig[3.3]) and also with the environment(fig[3.4]).The amplitude and phase of the sign wave word "CHERRY" may vary depending on the environment in which the data is collected, whether it is in a laboratory or a home setting, due to differences in the surrounding objects and setup present in each location,see fig[3.5].

To mitigate the effect of the subject on the CSI data, it is important to carefully design the wireless communication system and the environment where it is operating. This may involve optimizing the placement of the antennas, using directional antennas or beamforming algorithms to reduce interference from objects in the environment, or using frequency diversity techniques to improve the reliability of the wireless channel. The effect of the subject on the CSI data is an important consideration in the design and operation of wireless communication systems. By carefully analyzing and mitigating the impact of the subject, it is possible to improve the accuracy and reliability of the CSI data and optimize the performance of the wireless communication system.

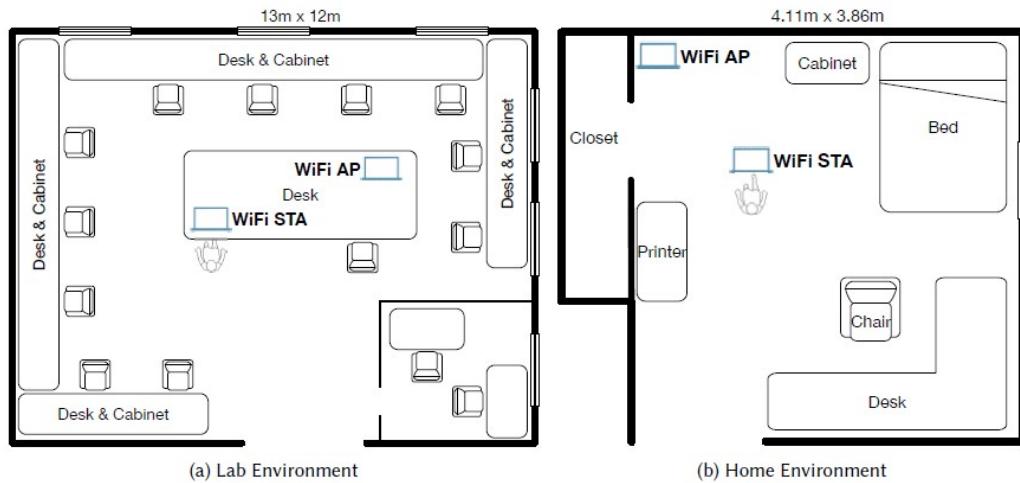


Figure 3.5: Floor plan and measurement settings of the lab and home environments

3.4 Correlation between Sub-carriers

The correlation coefficient is a statistical measure of the strength of a linear relationship between two variables. Its values can range from -1 to 1. A correlation coefficient of -1 describes a perfect negative, or inverse, correlation, with values in one series rising as those in the other decline, and vice versa. A coefficient of 1 shows a perfect positive correlation, or a direct relationship. A correlation coefficient of 0 means there is no linear relationship.

$$\rho = \frac{\text{Cov}(X, Y)}{\sigma_x \sigma_y}$$

wherw,

ρ =Pearson product-moment correlation coefficient

$\text{Cov}(x,y)$ =covariance of variables x and y

$\sigma(x)$ =standard deviation of x

$\sigma(y)$ =standard deviation of y

This can be elaborated as

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

where,

r=correlation co-efficient

n=number of observation

Correlation between sub-carriers in CSI (Channel State Information) data refers to the extent to which the amplitude and phase of the channel response of one sub-carrier is related to that of another sub-carrier in the OFDM signal.

In wireless communication systems that use OFDM modulation, sub-carrier correlation can have a significant impact on the accuracy and reliability of the CSI data. This is because the sub-carriers are closely spaced in frequency, which can result in inter-carrier interference (ICI) and inter-symbol interference (ISI). These interferences can cause the channel response of one sub-carrier to be correlated with that of its neighboring sub-carriers.

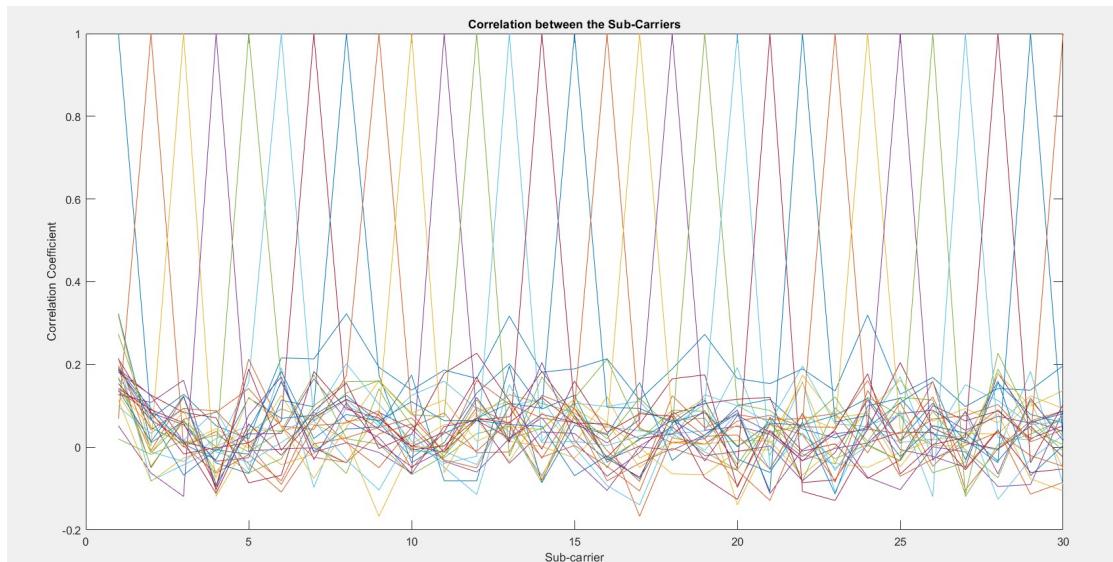


Figure 3.6: Amplitude Correlation between the Sub-carriers

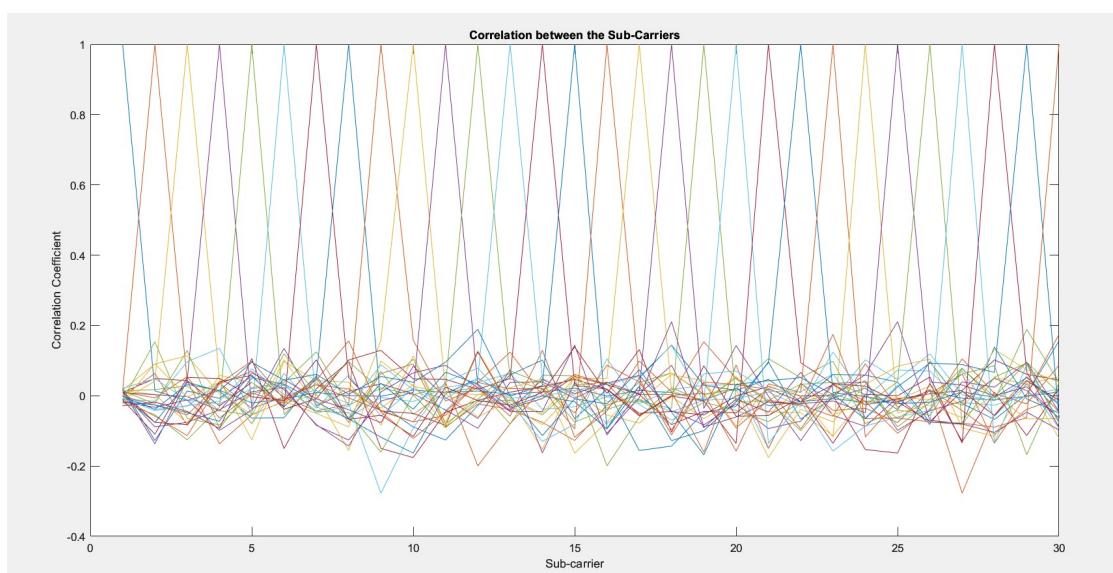


Figure 3.7: Phase Correlation between the Sub-carriers

As a result, the correlation between sub-carriers in CSI data can result in inaccurate channel estimation, which can ultimately lead to a degraded performance of the wireless communication system. To mitigate the effects of sub-carrier correlation, techniques such as frequency diversity, time diversity, or space diversity can be used. These techniques help to reduce the impact of the correlation between sub-carriers, resulting in more accurate CSI data.

The plots depict the characteristics of the sign word "SCARED". It is observed that there is a weak correlation between the amplitude(fig[3.6]) and phase(fig3.7) among the subscribers, as indicated by the correlation coefficient being close to zero.

It is crucial to consider the correlation between sub-carriers in CSI data for the optimization of the performance of wireless communication systems that use OFDM modulation. By mitigating the effects of sub-carrier correlation, more accurate and reliable CSI data can be obtained, leading to a better performance of the wireless communication system.

Chapter 4

CSI Data Preprocessing

CSI (Channel State Information) data preprocessing is an essential step in analyzing wireless communication systems. CSI data provides information about the channel between the transmitter and the receiver, such as the amplitude, phase, and delay of the signal, which can be used for various purposes, such as signal strength estimation, localization, and gesture recognition.

The preprocessing of CSI data involves several steps, which may vary depending on the specific application. The first step is to extract the CSI data from the raw signal, which may require specialized hardware and software tools. Once the CSI data is extracted, it is necessary to clean and filter the data to remove any noise or interference.

Next, the data may need to be transformed or normalized to a common scale, so that it can be compared and analyzed accurately. This may involve techniques such as signal normalization, Fourier transform, or wavelet transform.

Another important step in CSI data preprocessing is feature extraction, where meaningful features are extracted from the data to be used for analysis or classification. These features may include statistical properties of the signal, such as mean, variance, and skewness, or more advanced techniques such as Principal Component Analysis (PCA) or Independent Component Analysis (ICA).

Overall, CSI data preprocessing is a critical step in analyzing wireless communication systems, and the quality and accuracy of the results depend heavily on the preprocessing techniques used. Proper preprocessing can help improve the performance and reliability of applications such as gesture recognition, indoor localization, and wireless communication system optimization.

In the paper [12], the provided datasets consist of CSI data, which are complex quan-

tities that involve both amplitude and phase. To process this data before feeding it into a neural network, we separated the amplitude and phase components. Each sample in the dataset contains 30 sub-carriers and 3 antennas, with a total of 200 data points.

4.1 Amplitude Correction

In signal processing, a moving average filter is a commonly used technique to suppress quick variations or noise in a signal. The moving average filter works by calculating the average of a subset of adjacent data points in the signal, where the subset size is typically a small odd integer such as 3, 5, or 7.

In the context of processing the amplitude data in the provided dataset, a moving average filter was used to smooth out the signal and remove any high-frequency variations or noise that may have been present. By taking the average of several adjacent data points, the filter can eliminate any rapid or irregular changes in the signal and produce a more stable and predictable output.

The moving average filter can be implemented using various techniques such as convolution, which involves sliding a window of fixed size across the data and calculating the average of the data points within the window. The filter's effectiveness depends on the window size and the choice of averaging function.

$$A = \frac{A_1 + A_2 + A_3 + \dots + A_N}{N}$$

Here, A_1, A_2 are amplitude points and N is size of the window.

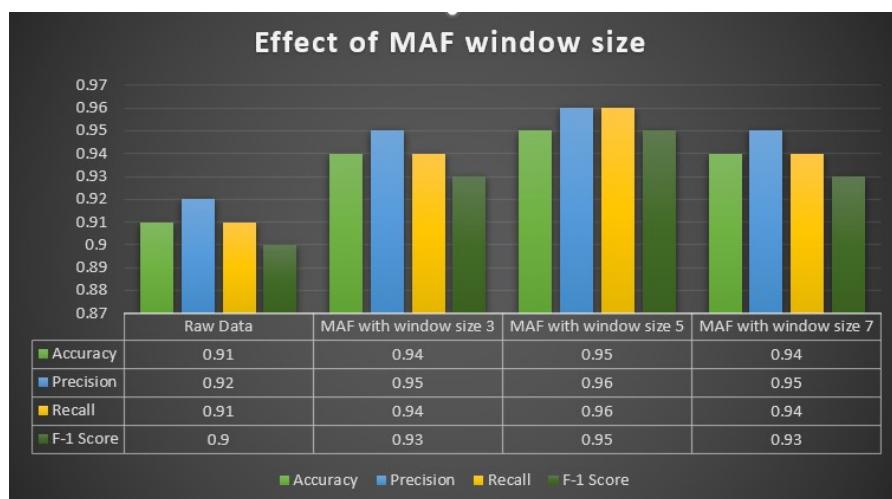


Figure 4.1: Effect of window size in MAF

In the data preprocessing step, we applied a Moving Average Filter (MAF) with a window size of 5. As it gives us the best performance as we can see from fig[4.1], which shows the classification report for various window size with no processing applied for the phase correction. MAF of window shows 1% better result in terms of accuracy where as 2-3% in terms of precision, recall and F-1 score from other window size MAF and no MAF applied to data acquired in lab environment. However, adjusting the window size can help achieve better noise rejection in the data. It is essential to note that using a window that is too large can result in the loss of useful information.

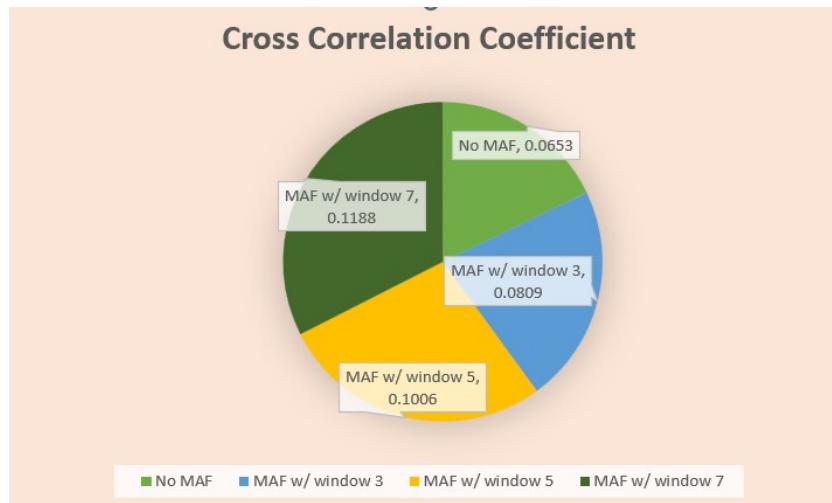


Figure 4.2: Correlation Coefficient After and before MAF

In the absence of moving average filtering (MAF), the subcarriers 13 and 15 of antenna 3 exhibit a notably weak correlation coefficient with the word "BROWN" in the home downlink dataset. According to the figure[4.2], the correlation coefficient shows an increase following the implementation of MAF. Specifically, using a window size of 7 for MAF results in a significant rise in the correlation coefficient about .0535 from the case when no MAF used.

The figure[4.3] shows the heatmap before and after MAF the right hand side heatmap greater coefficient with deep colour for the sign word "BEAUTIFUL" in the data collected from home environment.

This gives us an overview of effectiveness of MAF to filter out quick variation of the CSI data due to noise and other phenomena that may degrade the signal quality.

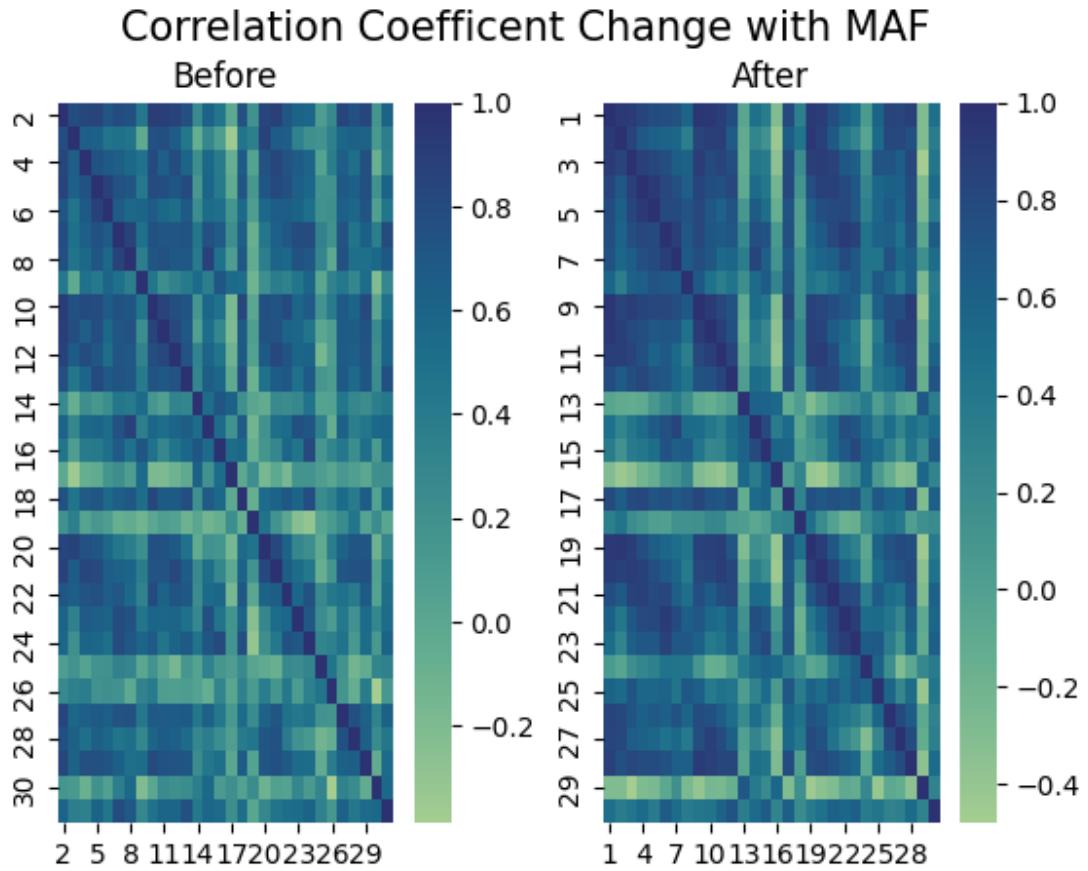


Figure 4.3: Correlation Coefficient Heatmap After and before MAF

To illustrate the effect of varying the window size in the MAF, we considered the word "Tea" and plotted the amplitude data for different window sizes, see fig[4.4]. The plot showed how the amplitude data was affected by the window size, with larger window sizes resulting in more smoothing of the signal and smaller window sizes preserving more details of the signal.

Therefore, choosing an appropriate window size for the MAF is a crucial step in the data preprocessing process, as it can significantly impact the accuracy and reliability of subsequent analysis or classification tasks.

4.2 Phase Correction

The CSI data extracted from the hardware, which is highly random, poses a challenge in accurately recognizing sign language gestures. This randomness is primarily caused by the lack of synchronization between the time and frequency of the transmitter and receiver components. To mitigate the errors arising from these hardware imperfections,

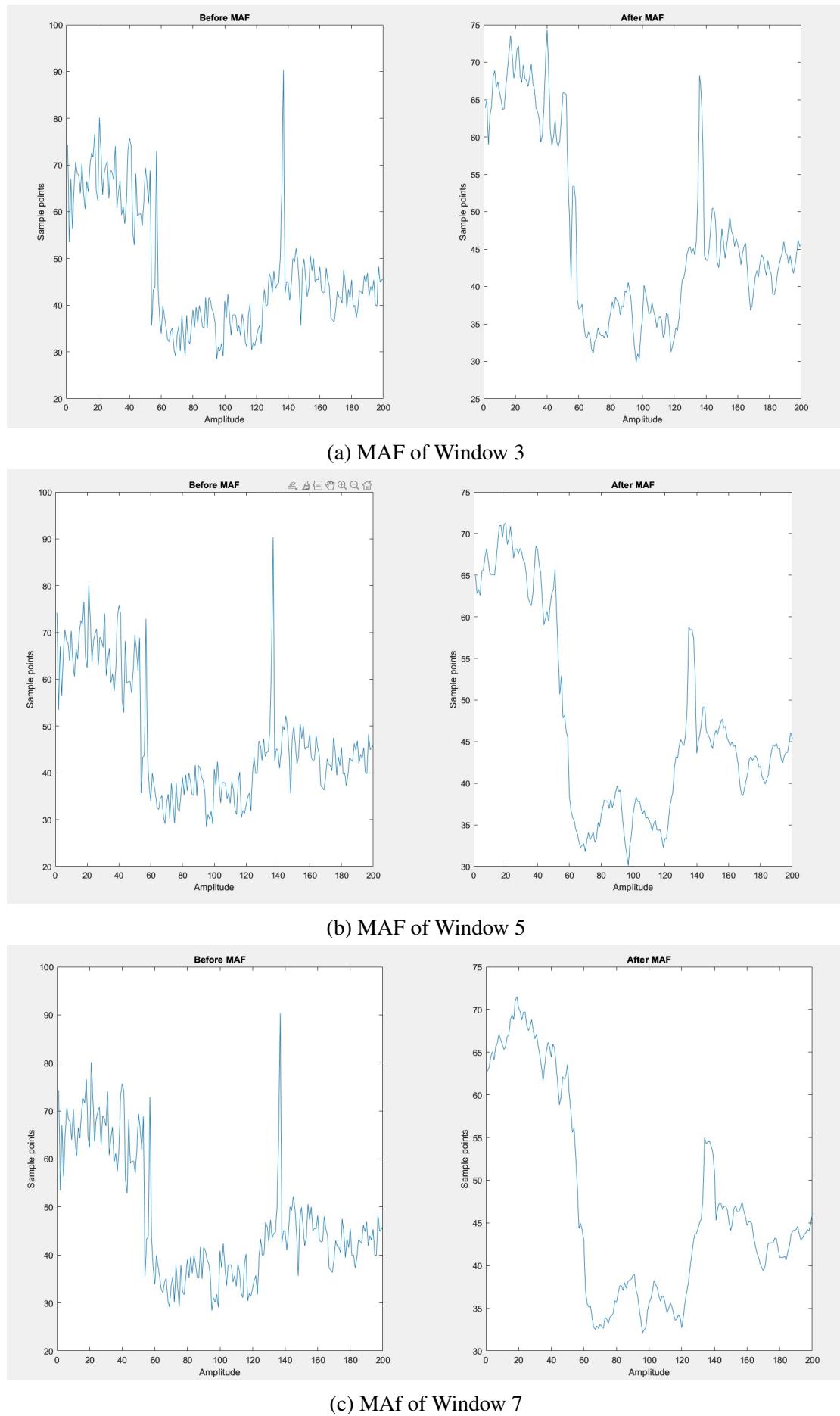


Figure 4.4: Effect of Amplitude with MAF

we utilize the discrepancy in phase values between two receiver antennas. By analyzing the phase difference between these antennas for consecutively received data packets, we can achieve a significantly more stable measure compared to the raw CSI data [23]. This approach aims to improve the accuracy of sign language gesture recognition by leveraging the reliable phase differences between receiver antennas. The measured CSI phase value θ_i of a subcarrier i can be mathematically represented as follows:

$$\theta_i = \phi_i + i(\lambda_{PB} + \lambda_{SF}) + \lambda_{CF}$$

In the expression of the measured CSI phase value θ_i for a subcarrier i , the original phase ϕ_i is influenced by channel propagation. The subcarrier index i identifies the specific subcarrier. Additionally, phase errors λ_{PB} , λ_{SF} , and λ_{CF} occur due to packet boundary detection (PBD), sampling frequency offset (SFO), and central frequency offset (CFO), respectively. Our objective is to determine the true phase value ϕ_i by mitigating the impact of these error parameters λ_{PB} , λ_{SF} , and λ_{CF} .

Correlation Coefficient Change with Phase Filtering

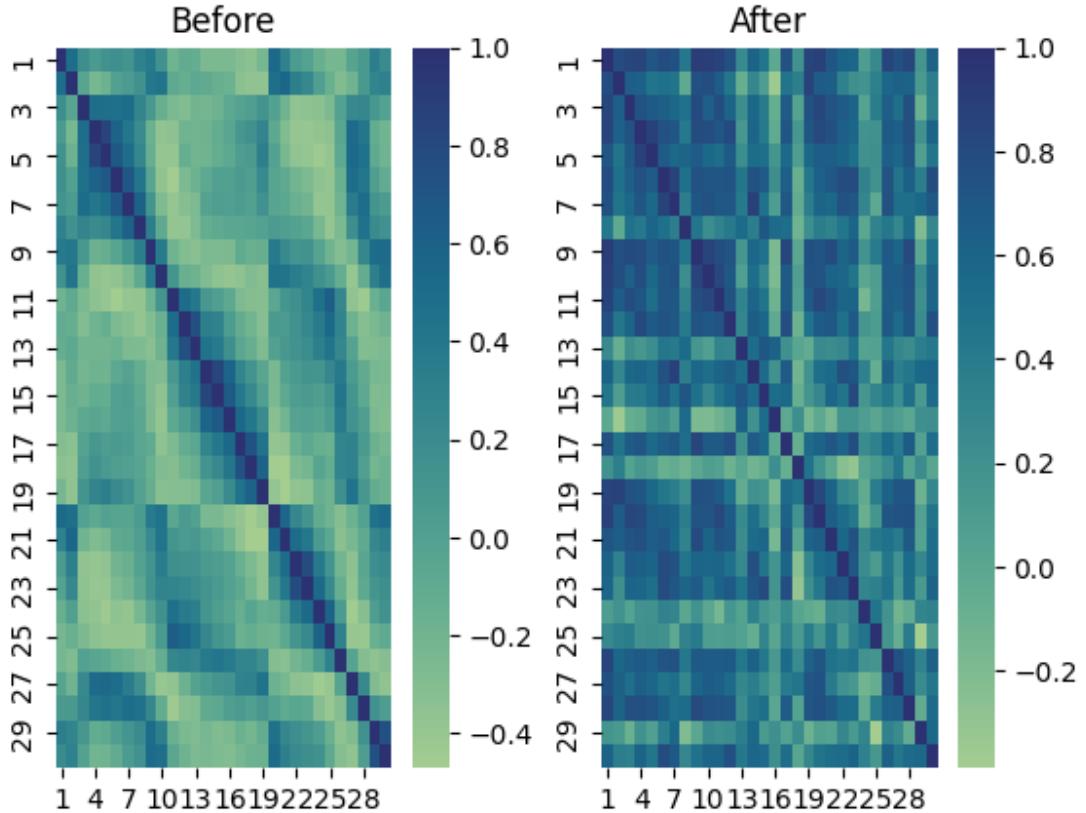


Figure 4.5: Correlation Coefficient Heatmap After and before Phase Correction

Phase error λ_{PB} is caused by the uncertainty in detecting the packet boundary, resulting in a time shift λ_{PB} . Phase error λ_{SF} is generated due to the offset between the sampling frequencies of the sender and the receiver. The hardware imperfection leads to incomplete compensation for the central frequency offset, causing CSI phase error λ_{CF} . These error parameters impact the measured CSI phase value θ_i .

According to the referenced work [20], the phase error parameters can be expressed as follows:

$$\begin{aligned}\lambda_{PB} &= 2\pi\Delta\tau N; \\ \lambda_{SF} &= 2\pi \frac{(T_r - T_t)}{T_t} \frac{T_s}{T_u}; \\ \lambda_{CF} &= 2\pi\Delta f T_s n;\end{aligned}$$

In these equations, N represents the FFT size, $\Delta\tau$ is the packet boundary detection delay, T_r and T_t are the sampling periods of the receiver and the transmitter, respectively, T_u is the data symbol length, T_s is the total length of the guard interval and the data symbol, n is the current packet sampling time offset. Δf is the difference in the center frequency between the transmitter and receiver.

Since the values of $\Delta\tau$, T_r , T_t , n, and Δf , in above equation are unknown and can vary between different packets, it becomes challenging to accurately detect the original phase based solely on the measured CSI phase. This limitation arises due to the fact that off-the-shelf devices provide only physical layer CSI data, which lacks the necessary information to properly determine the original phase. Furthermore, variations in $\Delta\tau$ and n over time introduce further uncertainty, leading to fluctuations in the phase errors λ_{PB} , λ_{SF} , and λ_{CF} .

The measured CSI phase values between two receiver antennas in an MIMO OFDM system exhibit stability on a specific subcarrier. This stability is attributed to the use of the same clock and down-converter frequency for the receiver antennas. Consequently, the central frequency difference, packet detection delay, and sampling period for the measured CSI phase remain consistent for that particular subcarrier [24]. As a result, the difference in CSI phase measured between the two antennas on subcarrier i can be approximated as follows:

$$\Delta\theta_i \approx \Delta\phi_i$$

In above equation, denoting $\Delta\phi_i$ as the phase difference of the original phase on subcarrier i, we observe that the impact of random phase errors is significantly reduced. This reduction occurs because the random terms $\Delta\tau$, T_r , T_t , n, and Δf , which are associated with the phase errors λ_{PB} , λ_{SF} , and λ_{CF} , are eliminated. As a result, across different

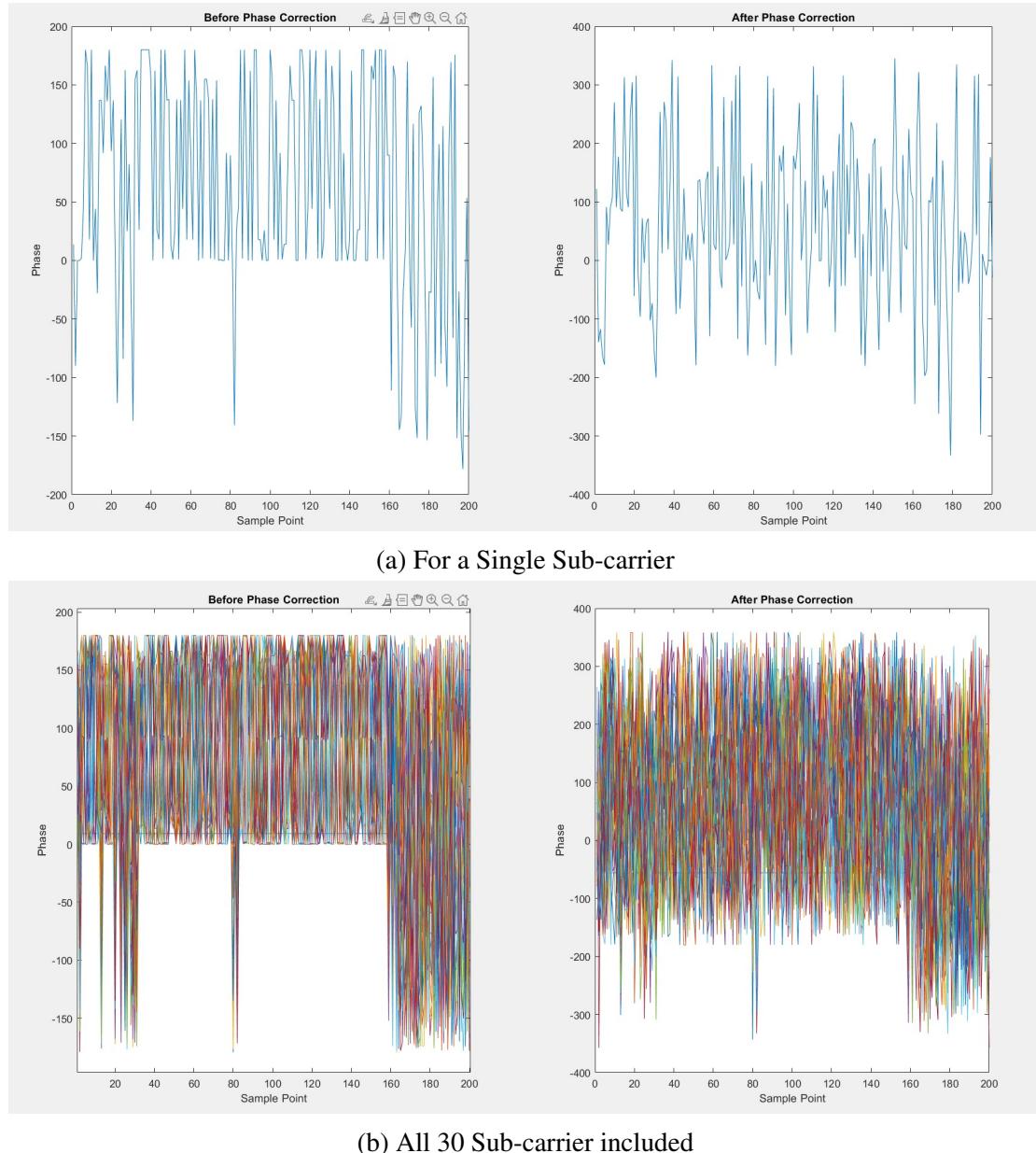


Figure 4.6: Effect of Phase Correction

packets, the value of $\Delta\theta_i$ becomes more stable compared to the measured CSI phase value.

We have implemented the phase correction technique described in the research paper [23]. To evaluate the effectiveness of this phase correction method, we generated plots comparing the phase values before and after the correction process. The specific sign word we focused on is "Thursday."

In the plots, we displayed the phase values for a single sub-carrier as well as for all the sub-carriers included, considering the same antenna index, see fig[4.6]. By visualizing the phase data before and after correction, we can observe the improvements achieved through the phase correction algorithm. The plot[4.5] shows that the sub-carriers have greater correlation co-efficient after applying phase correction in compared to when no phase correction is applied. The large percentage of the correlation coefficient lies in between .4-.6 implying strong correlation among the subcarrier. These plots provide a clear comparison, demonstrating how the phase correction mitigates the effects of phase errors caused by hardware imperfections and synchronization issues. The reduction in phase variations after correction signifies the successful elimination of unwanted noise and inconsistencies, resulting in more accurate and reliable phase values for the sign word "Thursday" across both single sub-carrier and all sub-carriers. The implementation of the phase correction technique outlined in the paper [23] has proven beneficial in enhancing the precision and robustness of sign language recognition, specifically for the word "Thursday."

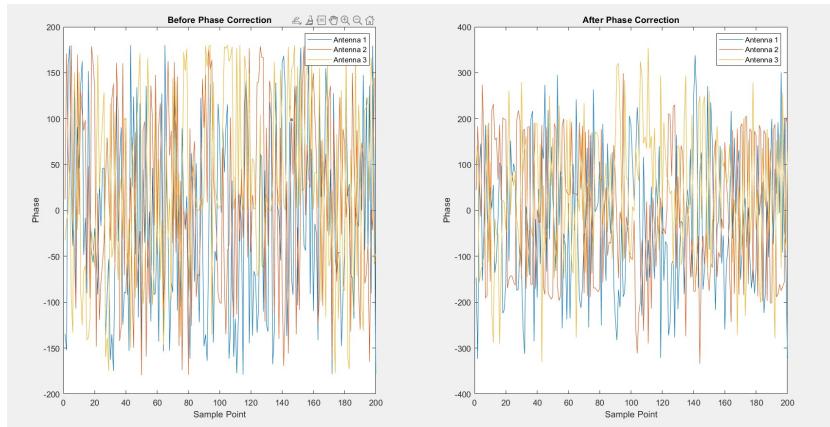


Figure 4.7: Phase correction effect on Antenna

Certainly! We have conducted an analysis of the phase before and after correction for the sign word "Where" specifically on a single subcarrier. The phase values for all the antennas involved are presented in the plots to demonstrate the impact of the correction process.

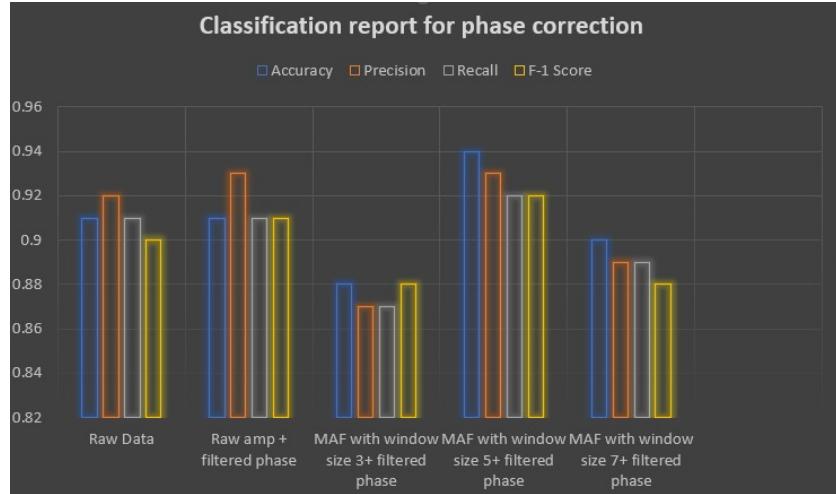


Figure 4.8: Phase correction effect comparison

By visually comparing the phase values before and after correction, we can observe the effectiveness of the phase correction technique in reducing the variations and improving the accuracy of the measured CSI phase.

The plot[4.7] depict the phase values for each antenna before and after correction, providing a clear indication of the improvements achieved. The reduction in phase discrepancies after the correction process highlights the successful elimination of hardware-induced errors and synchronization issues.

After performing phase correction on a home dataset and evaluating the classification results, it is observed that phase correction using a moving average filter (MAF) with a window size of 5 yields the best performance. The classification report provides insights into the accuracy, precision, recall, and F1-score for each class in the dataset. These metrics assess the correctness, completeness, and balance of the classification results. This indicates that the chosen phase correction method effectively enhances the classification accuracy and helps in achieving reliable results for the home dataset, see in fig[4.8].

4.3 Scaling The CSI Data

Scaling techniques are crucial in data preprocessing for machine learning tasks as they significantly impact model performance and accuracy. Various techniques are available to ensure proper scaling. Here are some key points about each scaling technique mentioned:

1. StandardScaler: This technique transforms features to have zero mean and unit variance, making them suitable for algorithms assuming a normal distribution. It

prevents dominance of features with larger magnitudes and enables fair comparisons among features.

2. MinMaxScaler: This technique rescales features to a specific range, preserving the shape of the distribution. It is useful when feature values need to be within a known range and maintains the relative relationships between data points.
3. MaxAbsScaler: This technique scales features to a range between -1 and 1 while preserving the sign of the data. It is useful for sparse data or features centered around zero, ensuring that the sign and relative relationships are maintained.

Scaling techniques ensure proper preparation of features for model training, promoting fair comparisons, reducing dominance, enhancing convergence, and improving overall model robustness. The selection of an appropriate scaling technique depends on the data's characteristics and the machine learning algorithm's requirements. The dataset,

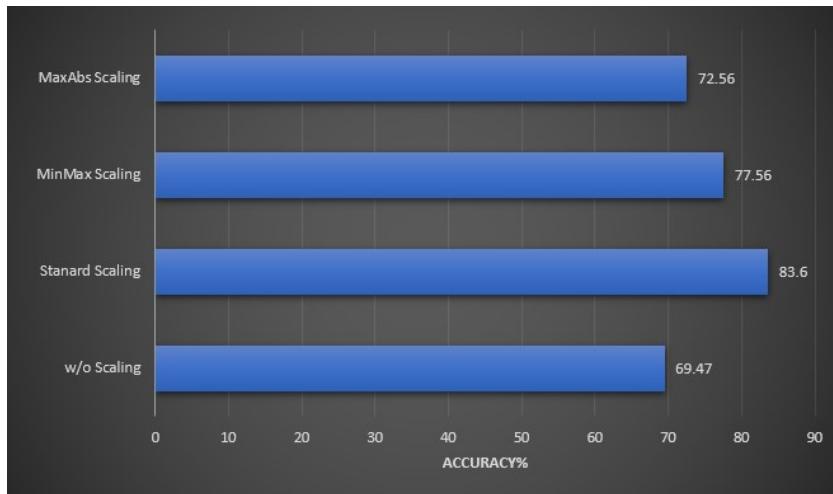


Figure 4.9: Scaling Technique Performance Comparison

which comprises 150 distinct classes and a total of 7,500 samples, the performance of various scaling techniques is illustrated in Figure 10. These samples were collected from five different users, with each user contributing 1,500 samples, and the data collection was conducted in a controlled laboratory environment. In Figure[4.9], we can observe the performance of various scaling techniques applied to the dataset. The performance analysis reveals that the StandardScaler exhibits superior performance compared to the other scaling techniques. It outperforms the MinMaxScaler and MaxAbsScaler techniques by approximately 6% and 11% respectively. These results highlight the effectiveness of standardizing the features to achieve better performance in the given dataset. The utilization of MinMax scaling resulted in a significant improvement of approximately 14% in accuracy compared to the scenario where no scaling was employed.

Chapter 5

Sign Gesture Recognition Models

5.1 Reference Models

Following the preprocessing in MATLAB, the data was imported into Python as a CSV file for further classification analysis. The data was structured into a matrix of dimensions n by 36000, where n represents the number of samples in the dataset. Each sample consisted of 36000 data points, which corresponded to the measurements from 30 subcarriers and 3 antennas. This arrangement resulted in a total of 18000 data points for amplitude and phase, with 600 data points per subcarrier and antenna combination. To evaluate the performance, various classifiers such as SVM, GNB, random forest, KNN, and ANN were employed.

5.1.1 Support Vector Machine (SVM)

SVM is a versatile machine learning algorithm used for classification and regression. It finds an optimal hyperplane that separates data points with the maximum margin. It can handle non-linear data using the kernel trick. SVM uses support vectors, which are closest points to the hyperplane. It has a regularization parameter (C) to control the trade-off between margin width and misclassifications. SVM is widely used in various applications for its ability to handle complex boundaries and resistance to overfitting.

The performance of Support Vector Machines (SVM) in the given scenario is significantly poor, with an accuracy of only 11% for the home environment and a mere 5% for the lab environment. These low accuracy rates suggest that the SVM model struggles to effectively classify the data in both settings. The poor performance of Support Vector Machines (SVM) in this case could be attributed to the large number of classes in the

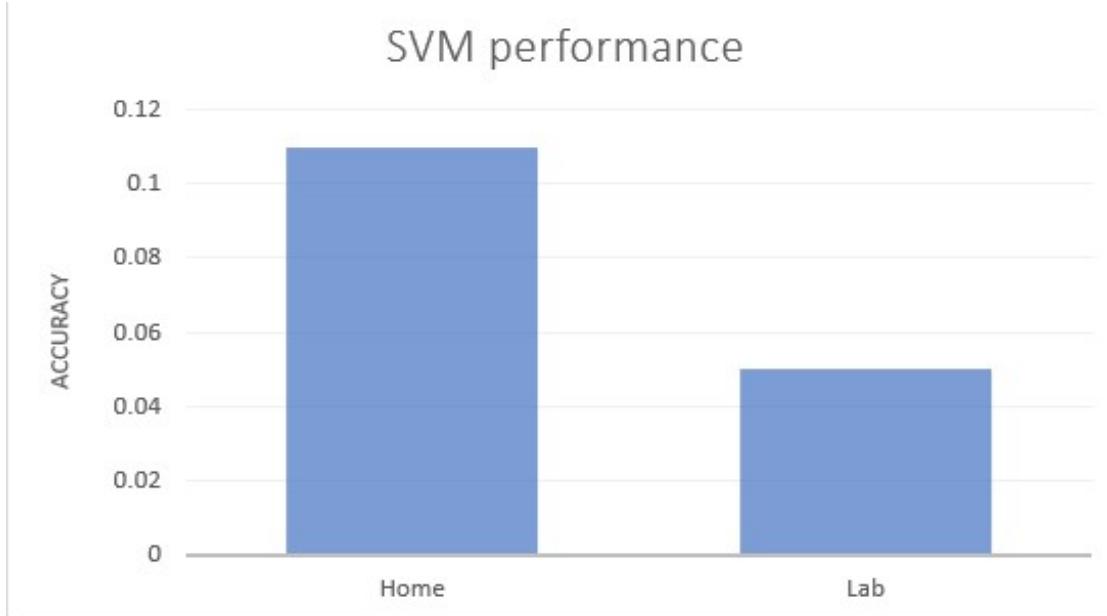


Figure 5.1: SVM Performance

dataset. SVMs work best in binary classification scenarios, where there are only two classes to distinguish. When dealing with a large number of classes, the complexity of the classification task increases, making it more challenging for SVMs to accurately separate the data. This highlights the need for alternative approaches or improvements in the SVM model to achieve better performance and accurate predictions in these environments.

5.1.2 Random Forest (RF)

A random forest classifier is an ensemble learning algorithm used for classification tasks. It combines multiple decision trees, each trained on a random subset of the data, to make predictions. By aggregating the predictions of individual trees, the random forest produces a final prediction. It is robust against overfitting, handles large feature sets, and can handle missing values. Random forests are widely used in various applications due to their versatility and good performance.

While the performance of the random forest classifier appears commendable, as indicated by Figure 5.2 for the lab dataset comprising 5520 samples with each class having 20 samples, its accuracy substantially declines to approximately 70% on other datasets. Consequently, this model does not engender a reliable outcome for achieving optimal performance.



Figure 5.2: Random Forest Classifier Performance

5.1.3 K-Nearest Neighbors Algorithm (KNN)

K-nearest neighbors (KNN) is a simple yet effective machine learning algorithm used for classification and regression tasks. In KNN, the prediction for a new data point is determined by considering its K nearest neighbors in the training dataset. The class or value of the majority of the neighbors is assigned to the new data point. KNN is non-parametric and lazy, meaning it does not make explicit assumptions about the underlying data distribution during training. It is easy to understand and implement but can be computationally expensive for large datasets.

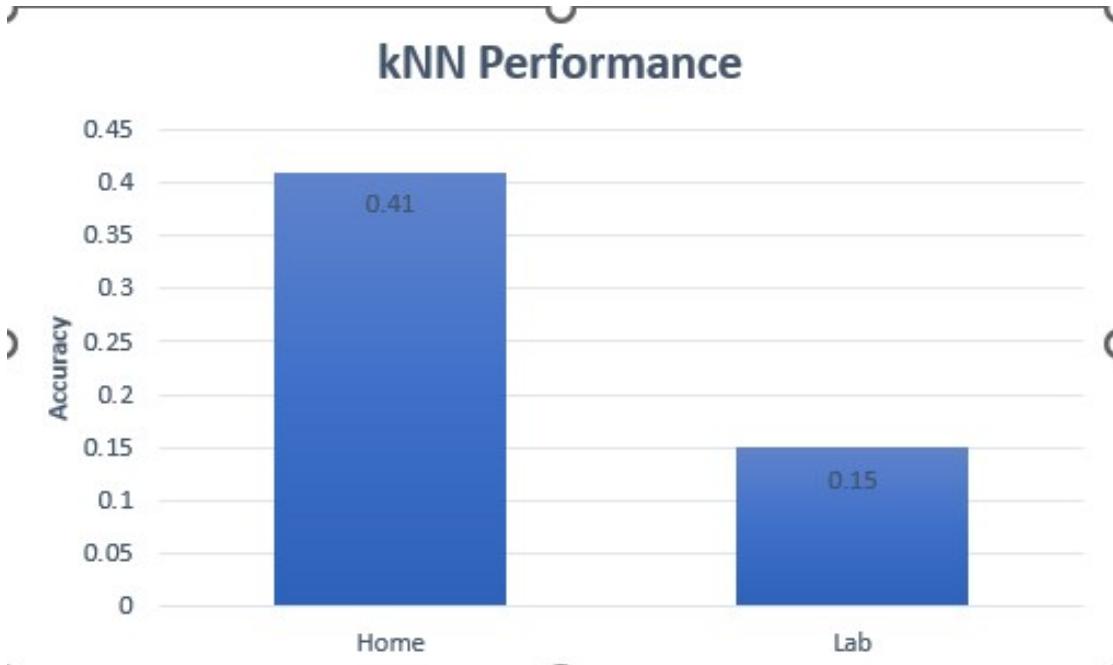


Figure 5.3: KNN Performance

Figure[5.3] reveals a notable decline in the performance accuracy of the K-nearest neighbors (KNN) algorithm, contrary to expectations of achieving a satisfactory rate of 98% or higher. Specifically, the accuracy for home environment stands at approximately 41%, which experiences a substantial drop of 26% when transitioning to the lab environment, ultimately reaching a disappointing 15% accuracy. These results highlight a subpar performance that can be deemed unsatisfactory in this context.

5.1.4 Gaussian Naive Bayes

Gaussian Naive Bayes is a machine learning algorithm used for classification tasks. It is based on the principles of Bayes' theorem and assumes that features follow a Gaussian (normal) distribution. The algorithm calculates the conditional probability of a class label given the feature values by estimating the mean and variance of each feature for each class. During prediction, it assigns the class label with the highest probability. Gaussian Naive Bayes is computationally efficient and performs well when the assumption of feature independence and Gaussian distribution holds. However, it may not perform well when the feature distribution deviates significantly from a Gaussian distribution or when there are strong correlations among features.



Figure 5.4: GNB Performance Evaluation

The observed performance of the Gaussian Naive Bayes (GNB) model, as indicated by Figure[5.4], presents a notable discrepancy from the K-nearest neighbors (KNN) algorithm. While the GNB model demonstrates a relatively higher accuracy of 55% in the lab environment, it falls short with an accuracy of approximately 31% in the home environment. These results contradict the trend observed with the KNN algorithm. Furthermore, it is important to note that the GNB model exhibits high accuracy during training, nearing 100%. However, the disparity between the training and testing accuracies suggests potential overfitting issues. Hence, the GNB model does not meet the desired expectations in terms of its performance, indicating the need for further evaluation or alternative models.

5.1.5 Artificial Neural Network

Artificial Neural Networks (ANN) are interconnected layers of neurons inspired by the human brain. They use weights and activation functions[5.5] to process data and learn patterns. During training, they adjust weights through backpropagation to minimize error. ANNs have input, hidden, and output layers, and can be deep with multiple hidden layers. They excel in tasks like image recognition and natural language processing, learning complex relationships from data through forward and backward propagation.

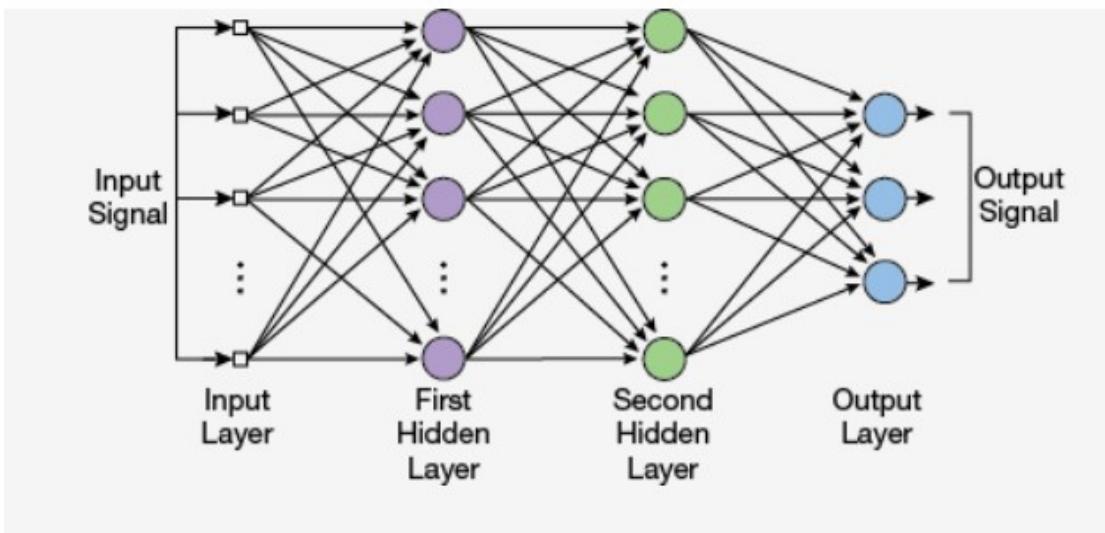


Figure 5.5: Artifical Neural network

[25]

Despite our attempts, the artificial neural network (ANN) we utilized did not yield satisfactory outcomes. Regrettably, the training accuracy consistently remained below 10%, providing us with limited useful information. We must explore alternative approaches or investigate potential factors, such as data quality, network architecture, hyperparameter settings, or training techniques, to overcome this obstacle and improve the performance of our ANN.

In order to improve the performance, it is necessary to employ more advanced classification techniques, such as Convolutional Neural Networks (CNN) with dropout and batch normalization. These techniques will be discussed in detail in the subsequent chapter, providing a comprehensive understanding of how they can enhance the model's accuracy and overall performance.

Chapter 6

CNN Implementation and Performance Evaluation

6.1 CNN Implementation

6.1.1 Gesture Recognition Algorithm and CNN Layers

A layer in neural network can be thought a function that takes an input, applies a set of learnable parameters (weights and biases), and produces an output. In the case of a convolutional neural network (CNN), a layer typically consists of multiple convolutional filters followed by non-linear activation functions and pooling operations. The i th layer of a n -layer neural network is given by,

$$y^{(i)} = g^{(i)}(W^{(i)}x^{(i)} + b^{(i)})$$

A neural network consists of multiple layers, where each layer takes an input, applies a weight matrix $W^{(i)}$ and a bias vector $b^{(i)}$, and passes the result through an activation function $g^{(i)}$. The output of the previous layer serves as the input $x^{(i)} = y^{(i-1)}$ for the current layer. The initial input of the network is denoted as $x^{(i)} = x$, and the final output is represented as $y^{(n)} = y$. In classification tasks, the output y contains labels corresponding to the input x .

In order for neural networks to learn the weights (W) and biases (b), an optimization algorithm is employed to minimize the cost function. In the case of SignFi, the algorithm used is Stochastic Gradient Descent with Momentum (SGDM). This optimization algorithm updates the weights and biases by taking small steps in the direction oppo-

site to the negative gradient of the loss function. The negative gradient represents the direction of steepest descent, and by moving in this direction, the algorithm aims to find the optimal values for the weights and biases that minimize the loss function. The inclusion of momentum in SGDM helps to accelerate convergence by adding a fraction of the previous update to the current update, enabling the algorithm to better navigate complex and uneven surfaces of the loss landscape.

The loss function can be written as:

$$\theta_{l+1} = \theta_l - \alpha \nabla E(\theta_l) + \gamma(\theta_{l+1} - \theta_l)$$

where θ is the parameter vector, l is the iteration index, α is the learning rate, $E(\theta)$ is the loss function, and γ momentum term.

The momentum term (γ) in SignFi controls the influence of the previous gradient step on the current iteration. In SignFi, a momentum term of 0.9 is used, which means that 90% of the previous update is incorporated into the current iteration.

To prevent overfitting, SignFi employs L2 regularization, which adds a regularization term for the weights to the loss function ($E(\theta)$). The regularized loss function, denoted as $E_R(\theta)$, is computed as the sum of the original loss function and the regularization term.

The regularization loss function:

$$E_R(\theta) = E(\theta) + \lambda \Omega(W)$$

The regularization term $\Omega(W)$ is calculated as $\Omega(W) = W^T W / 2$, where W represents the weights. This term encourages smaller values for the weights and helps control model complexity.

In SignFi, the regularization factor λ is set to 0.01, determining the strength of regularization. A higher value of λ increases the impact of the regularization term on the loss function, thus promoting more weight decay and a simpler model.

To incorporate convolution operations into a neural network, it becomes a Convolutional Neural Network (CNN). A CNN involves at least one layer that performs convolutions, enabling the network to capture spatial features in the data.

Each filter in a CNN layer performs convolution, which involves sliding over the input data (e.g., an image) and computing dot products between the filter weights and local patches of the input. This process captures spatial features present in the data.

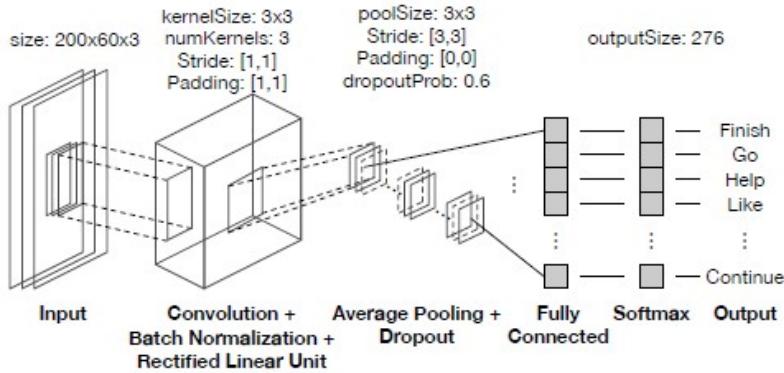


Figure 6.1: Architecture of the CNN network from the paper [12]

The output of each filter is then passed through a non-linear activation function, such as ReLU (Rectified Linear Unit), which introduces non-linearity into the network.

Pooling operations, such as max pooling or average pooling, are often used to down-sample the output of the convolutional layers, reducing the spatial dimensions while retaining important features. This helps in reducing the computational complexity and providing a form of translation invariance.

The overall architecture of a CNN is typically composed of multiple layers stacked together. Deeper architectures, like the 9-layer CNN mentioned in the paper [12] have a larger number of layers, enabling them to learn more complex and abstract features from the data. However, deeper networks also require more computational resources and training data to achieve good performance.

In our case, despite using a simpler one-layer CNN architecture, we were able to achieve comparable performance to the 9-layer CNN architecture presented in the paper [12]. This suggests that your simpler network was able to learn the relevant parameters and features efficiently, even with a large number of classes.

There are several benefits to using a simpler network architecture:

1. **Computational Efficiency:** Simpler networks with fewer layers and parameters require less computational resources for training and inference. This can be advantageous, especially in scenarios where there are constraints on computing power or time.
2. **Reduced Overfitting:** Simpler networks have a lower capacity to memorize noise or irrelevant patterns in the training data. This can help prevent overfitting, which occurs when a model becomes too specialized to the training data.

and performs poorly on unseen data. Simpler networks are often less prone to overfitting, as they have fewer parameters and are less likely to capture spurious correlations.

3. **Faster Training:** Training a simpler network is generally faster compared to training deeper architectures. With fewer layers and parameters, the optimization process can converge more quickly, allowing you to train and iterate on your models more efficiently.
4. **Interpretability:** Simpler networks are often easier to interpret and understand. With fewer layers, it is easier to trace the flow of information and interpret the learned features. This can be beneficial in fields where model interpretability is important, such as healthcare or finance.
5. **Generalization:** Simpler networks tend to generalize well to unseen data. They are less likely to overfit or memorize specific training examples, making them more robust when applied to new instances. This can lead to better performance on real-world data, as the model focuses on capturing more fundamental and generalizable features.

However, it's important to note that the choice of network architecture should be based on the specific problem, dataset, and resources available. Deeper networks with more layers may be necessary for tasks that require capturing intricate details or when dealing with highly complex datasets. The trade-off between simplicity and performance should be carefully considered in each case.

6.1.2 Input Layer

The CSI data for each sign gesture, consisting of CSI amplitude and phase, is organized as (3, 30, 200). This data is then reshaped into a tensor of size (200, 60, 3) by the input layer to be processed by the SignFi CNN, where:

1. 200 corresponds to number of CSI samples for each sign gesture.
2. 60 corresponds to the sub-carrier index ,30 for amplitude 30 for phase.
3. 3 represents the number of antenna.

The input layer is where the input data is fed into the CNN, and subsequent layers in the network process and transform the input data through operations like convolutions,

pooling, and non-linear activations to learn meaningful features and make predictions. It does not learn any parameter on its own. The input layer is where the input data is fed into the CNN, and subsequent layers in the network process and transform the input data through operations like convolutions, pooling, and non-linear activations to learn meaningful features and make predictions.

6.1.3 Convolution Layer

In SignFi, the convolutional layer applies two-dimensional convolutions instead of traditional matrix multiplications. The convolution operation can be termed as:

$$S(i, j) = (I * K)(i, J) = \sum_m \sum_n I(m, n)K(i - m, j - n)$$

Here I is the input and K is the kernel.

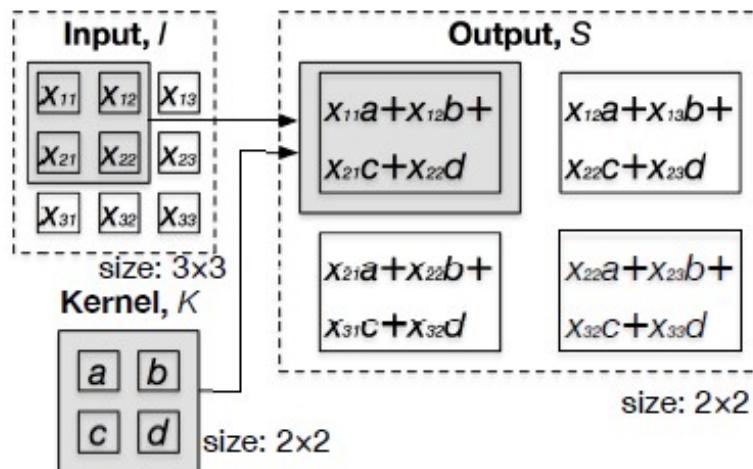


Figure 6.2: An example of two-dimensional convolution with a 2×2 kernel and stride of 1, from the paper [12]

Convolutional layers in Convolutional Neural Networks (CNNs) have various parameters that influence their behavior and performance. Here are some important parameters:

- 1. Kernel/Filter Size:** The kernel or filter is a small matrix that performs the convolution operation on the input data. The size of the kernel determines the receptive field or the area of input data that the filter covers at a time. Common kernel sizes are 3×3 , 5×5 , or 7×7 . Larger kernels capture more complex features but increase the number of parameters. SignFi uses kernel size of 3×3 .

2. **Stride:** The stride determines the step size at which the kernel slides over the input data. A stride of 1 means the kernel moves one unit at a time, while a stride of 2 moves two units at a time. Larger strides reduce the spatial dimensions of the output feature map but decrease the amount of computation. Smaller strides preserve more spatial information. In SignFi, the convolutional layers utilize stride with a value of 1 in both the vertical and horizontal directions.
3. **Padding:** Padding is the process of adding extra border pixels to the input data in order to preserve spatial dimensions and prevent information loss during convolutional operations. It is commonly used to ensure that the output feature map has the same spatial dimensions as the input or to control the reduction in feature map size.

In SignFi, a padding value of 1 is employed in both the vertical and horizontal directions of the convolutional layer. This means that an additional column and row of zeros are added along the edges of the original input. The purpose of this padding strategy is to maintain the output size of the convolutional layer and ensure that all inputs are treated equally during the convolution operation.

4. **Number of Filters:** The number of filters in a convolutional layer determines the number of output channels or feature maps. Each filter learns to detect specific patterns or features in the input. Increasing the number of filters allows the network to learn more diverse and complex features but also increases computational requirements. Here, in this method it is 3.

These parameters can be adjusted based on the specific task and requirements. The selection of kernel size, stride, padding, and number of filters depends on the complexity of the dataset, the desired level of feature extraction, and the trade-off between computational efficiency and information preservation.

By tuning these parameters and combining multiple convolutional layers, CNNs can effectively learn hierarchical representations and extract meaningful features from input data for various computer vision tasks.

6.1.4 Batch Normalization Layer

Batch normalization is a technique commonly used in deep learning models to improve the training process and the overall performance of neural networks. It addresses the problem of internal covariate shift, which refers to the change in the distribution of network activations during training.

The basic idea behind batch normalization is to normalize the inputs of each layer across a mini-batch of training examples. This is achieved by subtracting the batch mean and dividing by the batch standard deviation. The normalization process is applied independently to each feature dimension, ensuring that the network sees a more consistent and stable distribution of inputs.

The Normalized Activation is:

$$\hat{x} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

where μ_B and σ_B are the mean and variance of the mini-batch. In case of near-zero variances, a very small number ϵ , which is 10^{-6} in SignFi, is used to improve numerical stability. The output of the batch normalization layer is

$$y_i = \kappa \hat{x}_i + \rho$$

where κ is the scale factor, ρ is the offset, and \hat{x}_i is the normalized activation in equation. Both κ and ρ are learnable parameters that are updated during training. To take full advantage of batch normalization, SignFi shuffles the training data after each training epoch. By normalizing the inputs, batch normalization has several benefits. First, it reduces the impact of the vanishing and exploding gradient problems, making it easier for the network to learn. It helps alleviate the issue of saturating nonlinearities by keeping the inputs within a reasonable range.

Second, batch normalization acts as a regularizer, reducing the reliance on other regularization techniques such as dropout or weight decay. It introduces a small amount of noise by normalizing each mini-batch independently, which helps to prevent overfitting and improve the generalization ability of the network. Additionally, batch normalization reduces the sensitivity to the initial values of network parameters. It allows for faster convergence during training and helps to stabilize the learning process by reducing the internal covariate shift. As a result, networks with batch normalization layers often require fewer iterations to converge.

Moreover, batch normalization has been found to have a beneficial effect on the gradients flowing backward through the network. It smooths the optimization landscape, making it less likely for the network to get stuck in poor local minima and helping to speed up the training process.

While traditionally batch normalization has been applied to fully connected and convolutional layers, it can also be used with recurrent neural networks (RNNs) by normalizing the hidden states over time steps. This is known as recurrent batch normalization

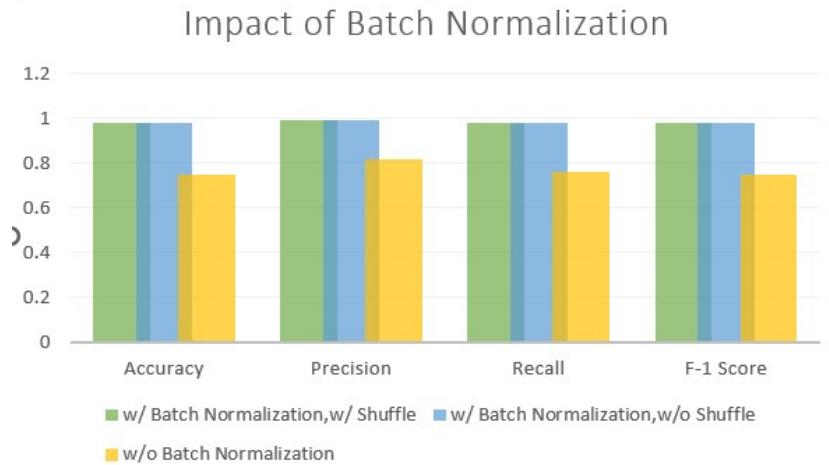


Figure 6.3: Impact of BatchNormalization Layer

and has been shown to improve the training of RNNs and their ability to capture long-term dependencies.

Batch normalization is essential for maintaining high accuracy, precision, recall, and F-1 score in deep learning models. From figure[6.3] it is evident that without batch normalization, there is a significant drop of approximately 23% in accuracy, recall, and F-1 score, and a 17% drop in precision. These decreases indicate that the model struggles to make accurate predictions and identify positive cases when the activations are not properly normalized. Although shuffling the batches during training helps reduce bias, it has a lesser impact on performance compared to the normalization provided by batch normalization.

6.1.5 ReLu Layer

The ReLU (Rectified Linear Unit) layer is a commonly used activation function in deep learning models. It introduces non-linearity to the network by mapping negative values to zero and leaving positive values unchanged. In other words, it replaces negative activations with zero and keeps positive activations as they are. The main advantage

$$\begin{array}{l}
 \text{ReLU:} \\
 g(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (12)
 \end{array}
 \quad
 \left| \begin{array}{l}
 \text{Leaky ReLU:} \\
 g(x) = \begin{cases} x & \text{if } x \geq 0 \\ scale * x & \text{if } x < 0 \end{cases} \quad (13)
 \end{array} \right.
 \quad
 \left| \begin{array}{l}
 \text{Clipped ReLU:} \\
 g(x) = \begin{cases} ceiling & \text{if } x > ceiling \\ x & \text{if } 0 \leq x \leq ceiling \\ 0 & \text{if } x < 0 \end{cases}
 \end{array} \right.$$

Figure 6.4: Different type of ReLu function

[12]

of the ReLU layer is its simplicity and computational efficiency compared to other activation functions. It is easy to compute and does not involve complex mathematical operations.

ReLU helps neural networks learn complex patterns and representations by allowing them to model non-linear relationships between inputs and outputs. It has been shown to work well in various deep learning architectures and has contributed to the success of convolutional neural networks (CNNs) in computer vision tasks. One drawback of

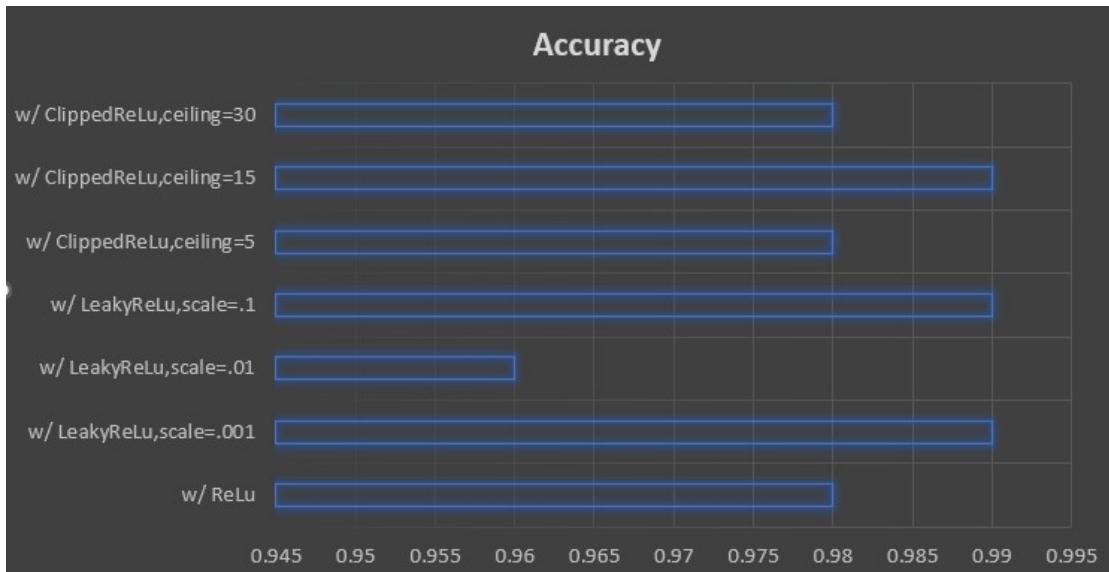


Figure 6.5: Performance of Different ReLU function

ReLU is the "dying ReLU" problem, where some neurons may become permanently inactive during training and never recover. This can occur when a large gradient flows through a ReLU neuron, causing it to remain in the zero-activation state. To mitigate this issue, variants of ReLU have been proposed, such as Leaky ReLU, Clipped ReLU, and Exponential Linear Units (ELUs), which introduce small positive slopes for negative inputs.

During our model experimentation, we tested various versions of the ReLU activation function with different parameter settings(see fig[6.5]). Despite the fundamental simplicity of ReLU, the observed impact on overall performance was relatively insignificant, resulting in only a marginal variation of approximately 2-3%. Among the different versions tested, the Leaky ReLU with scale values of 0.001 and 0.1, as well as the Clipped ReLU with a ceiling set at 15, consistently exhibited superior performance in the home environment, boasting an impressive accuracy rate of nearly 99%. Such remarkable results are highly satisfying and demonstrate the effectiveness of these particular ReLU configurations.

6.1.6 Pooling Layer

The average pooling layer in CNNs downscales input feature maps by dividing them into non-overlapping rectangular regions and computing the average value within each region. It reduces the spatial dimensions, making the model more efficient and faster. This downsampling operation helps control overfitting and provides regularization. However, it may discard detailed spatial information, so alternative pooling methods such as max pooling can be used when precise localization or fine-grained analysis is required. The basic operation of pooling layer is shown in fig[6.6], collected from the website [26]. In max pool it takes the largest value of selected region.

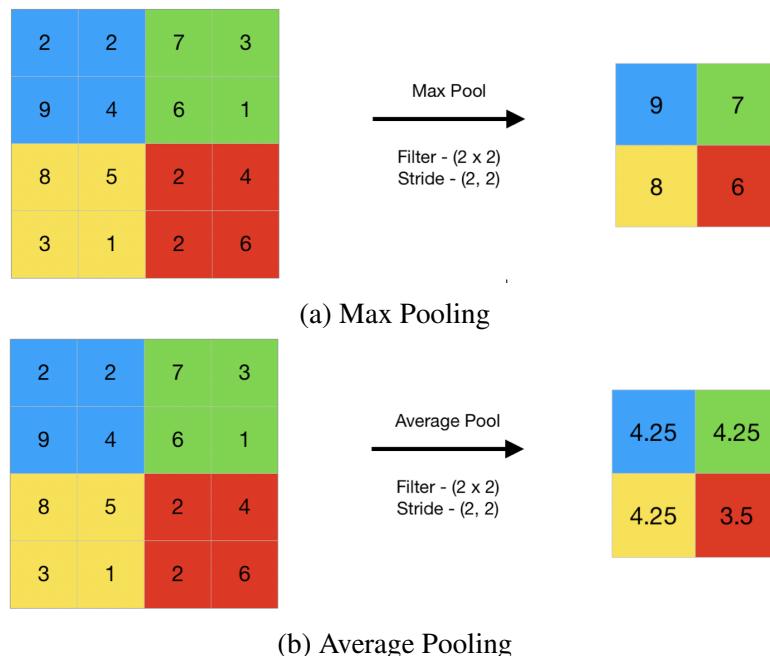


Figure 6.6: Pooling Layer Operation

In deep learning models, the convolutional layer, ReLU layer, and average-pooling layer are commonly combined into a single unit. These units are then connected to each other to handle large and complex datasets. Multiple of these units can be stacked together to create a deep neural network architecture. This configuration allows the network to learn hierarchical features, extract meaningful representations, and downscale the input data efficiently.

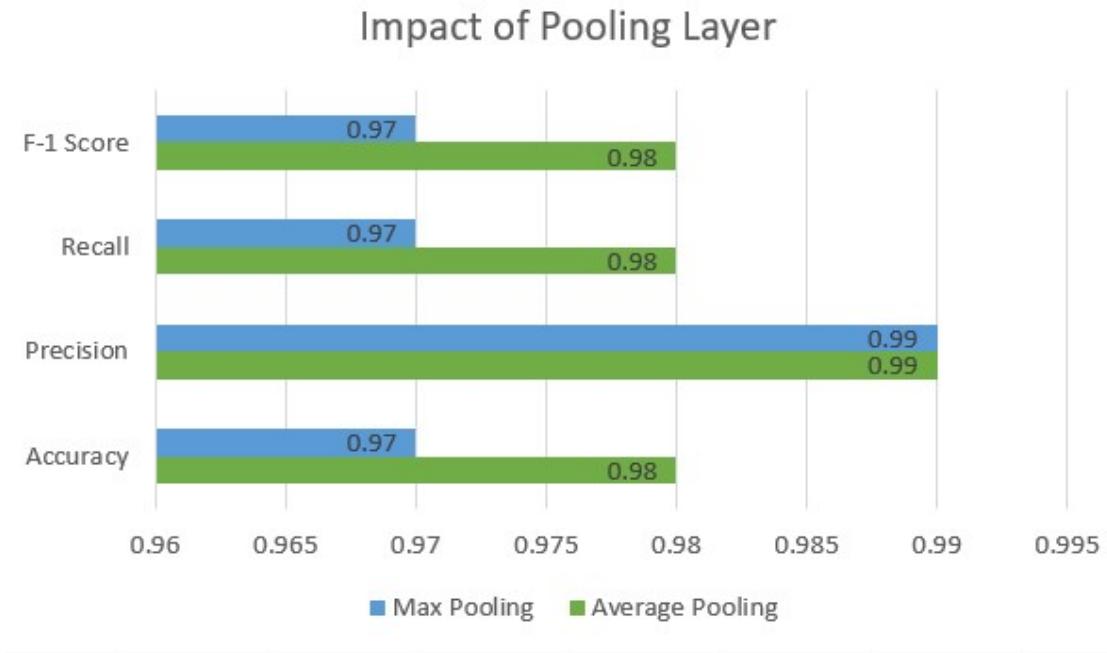


Figure 6.7: Performance of Pooling Layer

In our model, we compared the performances of max pooling and average pooling techniques, as shown in Figure[6.7]. We found that average pooling slightly outperformed max pooling for this particular dataset and model. Although both pooling layers had similar precision, average pooling showed a slight advantage, approximately 1%, in terms of recall, F-1 score, and accuracy. This indicates that average pooling was able to capture slightly more relevant information and achieve slightly better overall performance on the dataset.

6.1.7 Dropout Layer

The dropout layer plays a vital role as a regularization technique in deep learning models, specifically addressing the issue of overfitting. Its purpose is to randomly deactivate a portion of the input units by setting them to zero during the training phase. By doing so, dropout introduces noise and discourages co-adaptation among neurons, promoting model robustness and enhancing generalization capabilities.

The stochastic nature of dropout forces individual units to become more informative, as they cannot rely on the presence of other units. This encourages the model to learn more diverse and representative features, preventing over-reliance on specific subsets of neurons. This increased diversity in learning helps the model generalize better to unseen data.

During the inference or testing phase, the dropout layer is typically turned off, and the remaining weights are scaled. This scaling ensures that the expected output of the network remains consistent, even without dropout. By enabling this consistent behavior during inference, dropout does not introduce unnecessary randomness into the predictions.

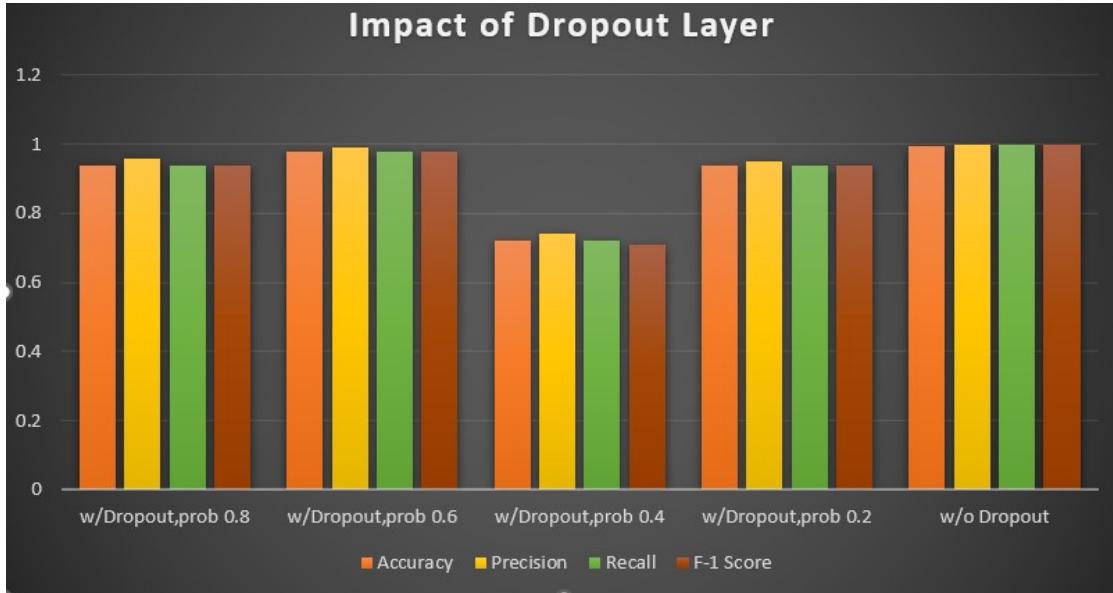


Figure 6.8: Performance w/ and w/o Dropout Layer

Based on the data presented in Figure[6.8], it can be observed that the dropout probability has a substantial impact on the model's performance. The results indicate that a dropout probability of 0.6 yields the most favorable outcomes, showcasing the highest efficacy across various performance metrics. In contrast, a lower dropout probability of 0.4 leads to a significant decline of approximately 23-24% in accuracy and other associated parameters. Conversely, when the dropout probability is set to 0.8 or omitted entirely, the model's performance remains relatively proximate to the optimal level achieved with a probability of 0.6, with a marginal decrease of approximately 2-3%. These findings underscore the significance of meticulously selecting the dropout probability to attain superior model performance.

6.1.8 Fully Connected Layer

The fully connected layer, also known as the dense layer, is a fundamental component of deep learning models. It connects every neuron from the previous layer to every neuron in the current layer. In other words, each neuron in a fully connected layer receives input from all the neurons in the preceding layer.

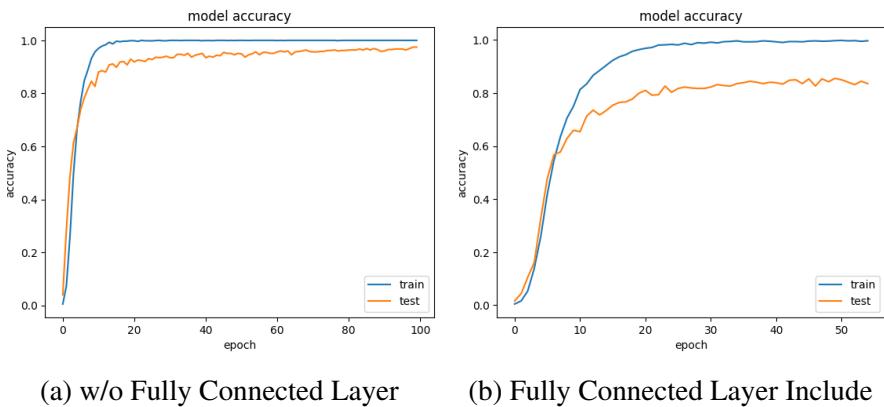


Figure 6.9: Performance Dip with Fully Connected Layer

The purpose of the fully connected layer is to learn non-linear combinations of the input features, enabling the model to capture complex patterns and make predictions. It acts as a mathematical transformation that maps the input data to the desired output space.

During training, the parameters of the fully connected layer, including weights and biases, are learned through optimization algorithms such as gradient descent. The layer's output is typically passed through an activation function to introduce non-linearity and enhance the model's representational power.

The fully connected layer is commonly used in the final layers of a neural network for tasks such as classification or regression. It can also be employed in intermediate layers, acting as a feature extractor and learning higher-level representations.

As illustrated in Figure[6.9], the inclusion of a fully connected dense layer with a dimensionality of 276, matching the number of classes in our significant model, leads to a substantial decline in performance. This observation holds true when considering the dataset collected from home environments, which consists of a total of 2760 samples, with each sign gesture having 10 samples.

Remarkably, despite the balanced distribution of 10 samples per sign gesture, the addition of the fully connected layer results in a noteworthy decrease of approximately 17 to 18% in model performance compared to the scenario where the fully connected layer is not included.

Furthermore, it is worth noting that both scenarios exhibit high training accuracy, surpassing 99%. However, when the fully connected layer is present, the test accuracy significantly drops to 81%, indicating a pronounced case of overfitting.

These findings highlight the importance of careful architectural choices. The signif-

ificant decrease in performance and increased overfitting associated with the inclusion of the fully connected layer demonstrate the necessity of selecting appropriate model components to ensure optimal performance and generalization and we excluded this in our final model.

6.1.9 Classification Layer

The classification layer is an essential component of deep learning models used for classification tasks. It serves as the final layer and is responsible for mapping the output from previous layers to the desired number of classes. The classification layer applies an activation function, often softmax, to generate a probability distribution across the classes. This distribution enables the model to assign probabilities to each class and make accurate predictions. By training the model with the classification layer, it learns to correctly classify input data into different categories.

In particular, the softmax layer is commonly used in deep learning models for multi-class classification. It takes a vector of real numbers as input and transforms them into probabilities that represent the likelihood of each class. This probabilistic representation allows the model to make confident predictions by assigning probabilities to different classes. Typically positioned at the end of the network, the softmax layer is often coupled with the cross-entropy loss function during training to optimize the model's classification performance.



Figure 6.10: Increase of Performance using top N prediction

Figure 6.10 reveals a noteworthy enhancement in model performance by approximately 2% when employing the top-3 categorical accuracy as the evaluation metric,

Table 6.1: The Recognition Accuracies of the Proposed Model vs Other Approaches

Method	Home	Lab	5-users
SignFi	98.91%	98.01%	86.67%
General DNN	99.89%	99.98%	
LSTM	74.06%	76.33%	56.9%
ABLSTM	95.44%	96.2%	73.83%
Proposed Model	97.64%	98.85%	84.63%

as opposed to the conventional single-label accuracy. This metric allows for multiple correct classifications by considering the top three predicted labels. By accommodating ambiguity and complexity inherent in certain classification tasks, the top-n categorical accuracy furnishes a more nuanced appraisal of the model's predictive prowess. The observed increase in accuracy attests to the metric's efficacy in capturing a broader spectrum of potential correct predictions, thus refining the model's overall predictive capabilities.

6.2 Performance Evaluation

To assess the effectiveness of our proposed model, we conducted an accuracy comparison with other deep learning CSI-based recognition approaches that are capable of automatically extracting features, see table[6.1].

Paper [12] introduced a 9-layer CNN model that achieved impressive accuracy rates of approximately 98% for both lab and home environments, as well as 87% for the lab environment with five users. This model utilized preprocessing techniques prior to feeding the data into the CNN architecture.

In contrast, Paper [27] proposed a DNN model that did not require any preprocessing before feeding the data into the classification algorithm. Notably, LSTM [28] and ABLSTM [29] models showcased exceptional performance in WiFi-based CSI human activity recognition, as they are specifically designed for time series data. Since CSI data exhibit temporal dependency, these models effectively captured and utilized the temporal patterns within the data.

The accuracy of the lab environment significantly decreased due to the inability to maintain a consistent environment, as explained by the autho[[12]]. This was evident when collecting data for the fifth user, which occurred four months later and was subject to different conditions.

Table 6.2: Comparison of different Sign Language Recognition Technologies

Comparisoon Technologies	Signal/Device used	Intrusive?	Granularity	Gesture Type	Recognition Algorithm	Number of Sign Gestures	Recognition Accuracy
Zafrulla2011	Kinect,gloves and sensors	Yes	Hand/Finger	Static	HMM	26	51.5%(seated) 56.12%(standing)
Sun2015 Pigou2015 Huang2015	Kinect	No	Hand/Finger	Dynamic	SVM CNN CNN	73 20 25	86.0% 91.7% 94.2%
Chuan2014					kNN,SVM	26	72.78%(kNN) 79.83%(SVM)
Quesada2015 Funasaka2015 Mapari2016 Naglot2016	Leap Motion	No	Finger	Static	SVM SVM MLP MLP+BP	10 26 26 26	79.17% 82.71% 90% 96.15%
DeepASL	Leap Motion	No	Hand/Finger	Dynamic	RNN	56	94.5%
Savur2015	sEMG Sensor	Yes	Finger	Static	SVM	26	91%(offline) 82.3%(real-time)
WifiSign WifiFinger Melgarejo2014	Wifi(2.4/5 GHz)	No	Hand Finger Hand/Finger	Dynamic Static	SVM kNN+DTW kNN+DTW	5 9 25(wheelchari); 14(car)	93.8% 90.4% 92%(wheelchair) 84%(car)
SignFi	WiFi(5GHz)	No	Head/Arm/ Hand/Finger	Dynamic	CNN	276	98%(lab),98%(home) 94%(lab+home)
Our Model	WiFi(5GHz)	No	Head/Arm/ Hand/Finger	Dynamic	CNN	276	98%(lab),99%(lab)

Table 6.2 presents a comparison between our signfi model and other existing methods of sign language gesture technologies. The results clearly demonstrate the superior performance of our model compared to other technologies currently available. Our model is capable of recognizing a larger number of classes and utilizes an already established infrastructure. Moreover, this technology is non-intrusive and it combines gestures from the hand, head, arm, and finger. In the case of Wi-Fi operating at 2.4 GHz [13, 16], it utilizes 5 sub-carriers, whereas signfi employs 30 sub-carriers [12] giving more information of the channel being used.

Due to computational limitations, we were unable to conduct the lab+home experiment, which consisted of 8280 samples. Additionally, we were unable to perform the 5th fold cross-validation for the larger dataset, which contained 7500 samples.

Chapter 7

Conclusion

The WiFi-based CSI sign language gesture recognition system utilized a 1-layer CNN model and pre-processing techniques, achieving impressive accuracy rates of approximately 97.64% and 98.85% in the home and lab environments, respectively. However, when dealing with a larger dataset of 7,500 samples from five users, the system faced challenges, resulting in a decreased accuracy of 84.63% in the lab environment. The decreased accuracy can be attributed to changes in data collection conditions over a four-month interval, emphasizing the importance of maintaining consistent conditions during data collection. To address these challenges, future research can focus on refining the model, collecting diverse datasets, and implementing techniques like automatic segmentation using RNN and LSTM models to improve accuracy and robustness in real-life scenarios.

Chapter 8

Future Directions

The utilization of WiFi-based CSI (Channel State Information) data for sign language recognition opens up several potential future directions. Here are some possible areas of exploration:

1. **Improved accuracy and robustness:** Researchers and engineers may continue to focus on enhancing the accuracy and robustness of sign language recognition systems. This can involve refining computer vision algorithms, developing more advanced machine learning models, and collecting larger and more diverse sign language datasets for training.
2. **Real-time recognition:** Efforts may be directed towards achieving real-time sign language recognition, enabling instantaneous translation and communication between sign language users and non-sign language users. This would require optimizing algorithms for faster processing, leveraging parallel computing, and utilizing hardware acceleration techniques.
3. **Gesture recognition:** While sign language primarily relies on hand movements and gestures, future research may explore incorporating facial expressions, body movements, and other non-manual components of sign language into recognition systems. This could lead to more comprehensive and accurate interpretations of sign language communication.
4. **Mobile and wearable applications:** Sign language recognition systems could be further adapted for mobile devices and wearables, such as smartphones, tablets, smartwatches, or augmented reality glasses. This would enable portable and accessible sign language translation and communication tools that can be readily available to users in various contexts.

5. **Cross-lingual sign language recognition:** Sign language recognition systems may expand to support multiple sign languages, allowing users to communicate across different sign language communities. This would involve developing models and datasets for different sign languages, as well as addressing the challenges of variations and regional differences within each language.
6. **User-friendly interfaces:** User interfaces for sign language recognition systems may become more intuitive and user-friendly, catering to the specific needs and preferences of sign language users. This can involve designing visually appealing and easy-to-understand interfaces, incorporating haptic feedback, or integrating sign language recognition with other assistive technologies.
7. **Integration with natural language processing:** The integration of sign language recognition with natural language processing (NLP) techniques can enhance the capabilities of sign language communication systems. This would enable translation between sign language and spoken/written languages, facilitating seamless interaction between sign language users and non-sign language users.

By focusing on these future directions, researchers can advance the field of sign language recognition using WiFi-based CSI data, making it more accurate, robust, and applicable in real-world scenarios, ultimately improving accessibility and communication for the deaf and hard-of-hearing communities.

References

- [1] D. Goldberg, D. Looney, and N. Lusin, “Enrollments in languages other than english in united states institutions of higher education, fall 2013.” in *Modern Language Association*. ERIC, 2015.
- [2] <https://wfdeaf.org/our-work/human-rights-of-the-deaf/>, May 2023.
- [3] A. Kuznetsova, L. Leal-Taixé, and B. Rosenhahn, “Real-time sign language recognition using a consumer depth camera,” in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 83–90.
- [4] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, “Sign language recognition using convolutional neural networks,” in *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13*. Springer, 2015, pp. 572–578.
- [5] J. Huang, W. Zhou, H. Li, and W. Li, “Sign language recognition using 3d convolutional neural networks,” in *2015 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2015, pp. 1–6.
- [6] C. Sun, T. Zhang, and C. Xu, “Latent support vector machine modeling for sign language recognition with kinect,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 2, pp. 1–20, 2015.
- [7] R. B. Mapari and G. Kharat, “American static signs recognition using leap motion sensor,” in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 2016, pp. 1–5.
- [8] B. Fang, J. Co, and M. Zhang, “Deepasl: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation,” in *Proceedings of the 15th ACM conference on embedded network sensor systems*, 2017, pp. 1–13.
- [9] M. Funasaka, Y. Ishikawa, M. Takata, and K. Joe, “Sign language recognition using leap motion controller,” in *Proceedings of the International Conference on*

- Parallel and Distributed Processing Techniques and Applications (PDPTA).* The Steering Committee of The World Congress in Computer Science, Computer . . . , 2015, p. 263.
- [10] D. Naglot and M. Kulkarni, “Real time sign language recognition using the leap motion controller,” in *2016 international conference on inventive computation technologies (ICICT)*, vol. 3. IEEE, 2016, pp. 1–5.
 - [11] U. of Washington., “Signaloud demo,” <https://youtu.be/4uY-MyoRq4c>, May 2023.
 - [12] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, “Signfi: Sign language recognition using wifi,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–21, 2018.
 - [13] P. Melgarejo, X. Zhang, P. Ramanathan, and D. Chu, “Leveraging directional antenna capabilities for fine-grained gesture recognition,” in *Proceedings of the 2014 ACM International Joint Conference on pervasive and ubiquitous computing*, 2014, pp. 541–551.
 - [14] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, “Whole-home gesture recognition using wireless signals,” in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 27–38.
 - [15] J. Shang and J. Wu, “A robust sign language recognition system with multiple wi-fi devices,” in *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*, 2017, pp. 19–24.
 - [16] S. Tan and J. Yang, “Wifinger: Leveraging commodity wifi for fine-grained finger gesture recognition,” in *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*, 2016, pp. 201–210.
 - [17] H. Li, W. Yang, J. Wang, Y. Xu, and L. Huang, “Wifinger: Talk to your smart devices with finger-grained gesture,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 250–261.
 - [18] R. Guo, H. Li, D. Han, and R. Liu, “Feasibility analysis of using channel state information (csi) acquired from wi-fi routers for construction worker fall detection,” *International Journal of Environmental Research and Public Health*, vol. 20, no. 6, 2023. [Online]. Available: <https://www.mdpi.com/1660-4601/20/6/4998>

- [19] F. Meneghello, N. Dal Fabbro, D. Garlisi, I. Tinnirello, and M. Rossi, “Csi dataset for wireless human sensing on 80 mhz wi-fi channels,” 2022. [Online]. Available: <https://dx.doi.org/10.21227/xbhv-f125>
- [20] D. Chakraborty, D. Garg, A. Ghosh, and J. H. Chan, “Trigger detection system for american sign language using deep convolutional neural networks,” in *Proceedings of the 10th International Conference on Advances in Information Technology*, 2018, pp. 1–6.
- [21] N. Sabri, S. Aljunid, R. Ahmad, M. AbdulMalek, A. Yahya, R. Kamaruddin, and M. Salim, “Performance evaluation of wireless sensor network channel in agricultural application,” *American Journal of Applied Sciences*, vol. 9, pp. 141–151, 01 2012.
- [22] M. Di Renzo, H. Haas, A. Ghayeb, L. Hanzo, and S. Sugiura, *Spatial Modulation for Multiple-Antenna Communication*, 11 2016.
- [23] T. F. Sanam and H. Godrich, “Fuseloc: A cca based information fusion for indoor localization using csi phase and amplitude of wifi signals,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7565–7569.
- [24] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, “Phaser: Enabling phased array signal processing on commodity wifi access points,” in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014, pp. 153–164.
- [25] https://www.aiche.org/sites/default/files/cep/inline/2018-06-01-Feature/2018-06-01-Introduction-to-Deep-Learning-Part-1/images/fig_06.png, accessed: 2013-05-13.
- [26] <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>, accessed: 2013-05-12.
- [27] B. Wei, K. Li, C. Luo, W. Xu, J. Zhang, and K. Zhang, “No need of data pre-processing: A general framework for radio-based device-free context awareness,” *ACM Transactions on Internet of Things*, vol. 2, no. 4, pp. 1–26, 2021.
- [28] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, “A survey on behavior recognition using wifi channel state information,” *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98–104, 2017.

- [29] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, “Wifi csi based passive human activity recognition using attention based blstm,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2714–2724, 2018.