Robust CSI based Smart Human Sensing

WANG DAZHUO

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

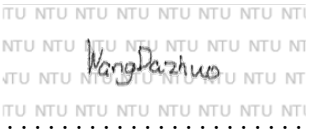
2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

17th Aug. 2022

Date



WANG DAZHUO

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

17th Aug. 2022

.....

Date

.....

Prof. Lihua Xie

Authorship Attribution Statement

This thesis contains material from 3 paper(s) published or under review in the following peer-reviewed journal(s) in which I am listed as an author.

Chapter 3 is published as D. Wang, J. Yang, W. Cui, L. Xie and S. Sun, "Multimodal CSI-Based Human Activity Recognition Using GANs," in IEEE Internet of Things Journal, vol. 8, no. 24, pp. 17345-17355, 15 Dec.15, 2021, doi: 10.1109/JIOT.2021.3080401.

The contributions of the co-authors are as follows:

- I proposed the idea, designed the system model, performed the experiments, and prepared the manuscript drafts.
- The manuscript was revised by Prof Xie, Dr. Sun, Dr. Yang and Dr. Cui.
- Dr. Yang and Dr. Cui provided guidance during the system design and the experiments.

Chapter 4 is published as D. Wang, J. Yang, W. Cui, L. Xie and S. Sun, "CAUTION: A Robust WiFi-based Human Authentication System via Few-shot Open-set Gait Recognition," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2022.3156099.

The contributions of the co-authors are as follows:

- I proposed the idea, designed the system model, performed the experiments, and prepared the manuscript drafts.
- The manuscript was revised by Prof Xie, Dr. Sun, Dr. Yang and Dr. Cui.
- Dr. Yang and Dr. Cui provided guidance during the system design and the experiments.

Chapter 5 is submitted as D. Wang, J. Yang, W. Cui, L. Xie and S. Sun, "AirFi: Empowering WiFi-based Passive Human Gesture Recognition to Unseen Environment via Domain Generalization," to IEEE Transactions on Mobile Computing.

The contributions of the co-authors are as follows:

- I proposed the idea, designed the system model, performed the experiments, and prepared the manuscript drafts.
- The manuscript was revised by Prof Xie, Dr. Sun, Dr. Yang and Dr. Cui.
- Dr. Yang and Dr. Cui provided guidance during the system design and the experiments.

17th Aug. 2022

. . . .

Date

. . . .

WANG DAZHUO

Acknowledgements

First and foremost, I wish to express my deepest gratitude to my supervisors, Prof Xie Lihua and Dr. Sun Sumei, for their supervision and care. Their prefessional suggestions and kind encouragement guided me throughout my entire Ph.D study. Whenever I had any doubts during my research, they always offered their help and time. They even sacrificed their personal time after work to help me improve my research abilities. Without their guidence, I could not have achieved these research progress.

My gratitude also goes to Dr Yang Jianfei and Dr Cui wei for inspiring me with their incisive insights in their research fields. Discussions with them always brought me new ideas and thoughts. They helped me design the experiments and revise my manuscript. Their experiences and knowledge helped me to improve my research ability a lot.

Meanwhile, I would like to express my gratitude to my colleagures Muqing Cao, and Thien Hoang Nguyen. The sharing of interesting things in our life has always been a great comfort during the hard time in my research study. Our collaboration also inspired me a lot in my research study.

Last but not least, I sincerely thank my family. My parents give me their love and support throughout my life. My wife is always there whenever I need encouragement and support. I Love you all.

"If I had one hour to save the world, I would spend 55 minutes defining the problem and only five minutes finding the solution."

—Einstein, Albert

To my dear family

Abstract

Sensing technology plays an important role in modern smart cities. Different sensing techniques have enabled systems with the ability to observe and interact with the surrounding environments. Leveraging the existing infrastructure of WiFi technology, Channel State Information (CSI) based human activity recognition has received great attention in recent years due to its advantages in privacy protection, insensitivity to illumination, and no requirement for wearable devices. Various CSI based smart sensing systems have been proposed. Though existing CSI-based smart sensing has achieved much progress, there are still some challenges that need to be addressed.

Firstly, existing CSI-based sensing systems normally suffer from great performances degradation under environmental dynamics. Besides, collecting enough new CSI data to retrain the system may not be possible in some situations. To address these issues, we propose a Multimodal Channel State Information Based Human Activity Recognition (HAR) system named MCBAR. It utilizes the current infrastructures of WiFi technology and measures the Channel State Information of human behaviors. It addresses the problem of performance degradation due to dynamic environment settings. Specially, we manage to address the problem that the CSI data of some rarely-performed activities cause non-uniform distribution of the unlabelled data collected. A generative adversarial network is applied to improve the diversity of the CSI dataset. The distribution of CSI data in the new environment setting can be approximated using a multimodal generator. It generates fake CSI data to improve the diversity of training CSI data, which leads to a better knowledge transfer. It also equips MCBAR with the ability to recognize activities in different CSI patterns due to dynamic environments. Compared to existing CSI-based HAR systems, MCBAR has higher stability. It is designed to adapt to new environment settings only with non-uniformly distributed unlabelled CSI data. The experimental results demonstrate that our system outperforms existing CSI-based HAR systems in the ever-changing environment settings.

Secondly, existing CSI-based smart sensing systems must be trained with CSI data from a deployed environment to adapt to the new environment. They use massive unlabeled high-quality data from the new environment, which is usually unavailable in practice. Besides, sometimes it is not possible to collect data from the deployed environments due to privacy concerns. Therefore, we propose a novel augmented environment-invariant robust WiFi gesture recognition system named AirFi that deals with the issue of environment dependency from a new perspective. The AirFi is a novel domain generalization framework that learns the critical part of CSI regardless of different environments and generalizes the model to unseen scenarios, which does not require collecting any data for adaptation to the new environment. AirFi extracts the common features from several training environment settings and minimizes the distribution differences among them. The feature is further augmented to be more robust to environments. Moreover, the system can be further improved by few-shot learning techniques. AirFi is able to work in different environments without acquiring any CSI data from the new environment. The experimental results demonstrate that our system remains robust in the new environment and outperforms the compared systems.

Thirdly, many CSI-based smart sensing systems need a great amount of training data, especially for those taking advantages of deep learning technologies. For example, some human authentication systems may have many users. If the systems are trained with many CSI samples for one individual user, the total number of CSI training data required is very large. It definitely affects the training efficiency and scalability of these systems. We propose a CSI-based human authentication system (CAUTION). It measures the distinctive gait features of each user via CSI to authenticate them. Few-shot learning technology is leveraged for the model construction. By this, CAUTION can be trained with only a few CSI samples collected. These CSI data are then downsampled on the feature space. CAUTION calculates central points of different classes based on them. Besides, CAUTION is able to detect strangers with an intruder threshold. The optimization of the threshold needs no CSI data from strangers, which is more suitable for real world situations. We test CAUTION in multiple environments and compare it with other advanced CSI user authentication systems. Results show that CAUTION can perform better than compared systems with a limited amount of training CSI data.

Finally, after the CSI-based smart sensing systems are deployed, they can give decent performances for a period of time. However, in real world, the surrounding environments are usually dynamic. The ever-changing surroundings can cause significant performance degradation. Retraining the system with CSI data collected in the new environment settings leads to catastrophic forgetting of previous training knowledge. Some existing CSI-based smart sensing systems address this issue by retraining the system with all stored CSI data, which increases the training cost and requires a longer training time. It is also limited by hardware storage. In this thesis, we propose a new life-long learning CSI-based human activity recognition system named LICAR. It uses the simulated meta CSI training data from CSI augmentation generator to reduce the distribution difference between each set of CSI training data. Most importantly, LICAR is added with a parameter updating buffer. It is optimized with the CSI data from the new settings meanwhile keep the knowledge for the previous CSI training tasks by updating different parameters selectively. LICAR is equipped with the life-long learning ability without the requirement of previous CSI data. The training process only involves the current CSI training dataset. From the experiment, the results show that LICAR is able to provide effective life-long learning. It outperforms the compared systems and remains to be robust under dynamic environment settings.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbols

$\mathcal{R}^n$	the n-dimensional Euclidean space
$\mathcal{H}$	the Hilbert space
$\|\cdot\|$	the 2-norm of a vector or matrix in Euclidean space
$G(\cdot;\theta)$	a function G parameterized by θ
$\mu(\cdot)$	mean map
$x \sim P$	a sample x follows the distribution P
$\otimes$	the Kronecker product
$\langle \cdot, \cdot \rangle$	the inner product of two vectors
$MMD(x, y)$	the Maximum Mean Discrepency between x and y
$E[X]$	mathematical expection of a random variable X
$Cov(x)$	the covariance of x
$x_{i,k}$	the i-th component of a vector x at time k
$\bar{x}$	the vector with the average of all components of x as each element
$\mathcal{C}$	the average space, i.e., $span\{\mathbf{1}\}$
p(x)	the probability distribution of x
$F(x)$	Fisher Information of variable x
$O(\cdot)$	order of magnitude or ergodic convergence rate (running average)

Acronyms

IoT	Internet of Things
SVM	Support Vector Machine
DNN	Deep Neural Networks

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
CSI	Channel State Information
LSTM	Long Short-Term Memory
DA	Domain Adaption
DT	Domain Transfer
DG	Domain Generalization
KNN	K-Nearest Neighbours
WiFi	Wireless Local Area Network
RSSI	Received Signal Strength Indicator
PCA	Principal Component Analysis
DTW	Dynamic Time Warping
SVD	Singular Value Decomposition
MMD	Maximum Mean Discrepancy
GAN	Generative Adversarial Networks
LOS	Line-Of-Sight
CIR	Channel Impulse Response
SGD	Stochastic Gradient Descent
CV	Computer Vision
i.i.d.	independent and identically distributed
$a.s.$	almost sure convergence of a random sequence

Chapter 1

Introduction

1.1 Motivations and Objective

Smart sensing plays a vital role in smart buildings and smart homes. It has important applications in energy usage estimation, human activity recognition and indoor security surveillance. With the rapid development of the Internet of Things (IoT), the occupancy information can be obtained by the cognitive computing technologies, which are enabled by an integration of different types of sensors, controllers and machine learning techniques. Various applications are enabled, of which the occupancy sensing is a significant one [1]. Occupancy sensing means retrieving users information and monitoring the situation of indoor environments by analyzing sensor data from different sensor networks in smart buildings. Occupancy sensing using IoT devices is able to actively obtain the information of users including their identities, activities and even gestures. It provides useful information for a smart system, so that it can make judgments or provide feedback accordingly.

Prevailing techniques in this field mainly include vision-based techniques and wearable device based techniques. Vision-based techniques which provide the highest granularity of activity recognition via computer vision algorithms have received much attention, however, their limitations include that the brightness of the environment affects the performance significantly [2]. The installation of camera also raises privacy concerns, which is usually not passable in smart homes. For wearable devices based techniques, RF tags [3] and accelerometers [4] are used. These incur extra expense and they are user-unfriendly since users have to carry the devices

with them. Radio frequency based approaches show their advantages. They do not consume much energy. However, it requires many RF links in a single room, which causes high installation cost. Therefore, its availability is affected.

Smart human sensing using WiFi technology receives great attention recently [5–7]. It is enabled by Channel State Information (CSI) [8–10]. CSI is a fine-grained measurement at the physical layer from a subcarrier channel. It can be used to measure the multipath effects of surroundings. It is able to capture the impacts of users' behaviors on the propagated WiFi signal on different subcarriers.

Compared to vision-based techniques and wearable sensor based techniques, CSI-based human activity recognition has many advantages. As the WiFi technique has been developed rapidly, the indoor coverage of wireless networks has become more and more widespread. By leveraging the existing infrastructure available in buildings and at homes, we do not need to install specific sensors, RF links or camera devices. It offers more convenience and privacy. Besides, it does not require line of sight or illumination requirements.

For smart sensing based on wireless signals or other technologies, machine learning is widely used, which helps to achieve high accuracy and good generalization capacity[11]. Nevertheless, there are still many challenges to be addressed in order to perform efficient and robust CSI-based smart human sensing.

One of the current challenges faced by CSI-based sensing is the degradation of performances under environmental dynamics. With training CSI samples, an accurate CSI-sensing model can be trained by taking advantages of machine learning. It can be used for user activity recognition, user authentication or indoor localization based on the variation of CSI samples caused by users. These variations provide CSI samples with different features. However, these features may vary due to environmental dynamics. Systems trained with data from one environment setting may not be functional in other environment settings. For example, the measured WiFi signals, CSI data, could be heavily distorted by walls, furniture placement, individual heterogeneity and etc. These structured noises hinder the generalization of deep models with regard to domain differences. In brief, the model trained in an environment cannot be directly employed in another one, which impedes the practical applications of sensor-based smart sensing techniques.

Besides, existing machine learning based CSI sensing systems normally require large amounts of data to train the system model. However, collecting such amount of data is usually time-consuming and labor-intensive. Supervised learning-based systems need labelled data for their training, which makes the situation even worse. Furthermore, it is not user-friendly to collect many data from users. It will also increase the cost of system utilization and the training period. The scalability of the systems is also affected. In certain situations, it is not even possible to acquire any CSI data for training purposes due to privacy concerns.

Furthermore, existing CSI based smart sensing systems lack of efficient life-long learning ability. After a CSI-based smart sensing system, which takes the advantages of deep learning techniques to build its system model, is trained in the target environment settings, it is able to perform its designed functions such as activity recognition, gesture recognition, user authentication and etc for some time. It can give very decent performances as long as the target environment settings remain static. However, in real life, the surrounding environments are usually dynamic. The ever-changing environments affect the performances of the deployed system greatly, as the CSI samples of the same activities or users are influenced by the surrounding environments as well. Simply retraining the system with new CSI samples leads to catastrophic forgetting, which means the system model may forget about its previous tasks. The CSI system should be able to learn new tasks with new CSI data meanwhile still maintain good performances on previous tasks to adapt to the ever-changing environment settings.

In summary, the objective of this Ph.D. research is to develop CSI based smart sensing systems to perform robust smart sensing under environment dynamics using CSI data and reduce the amount of CSI data required for model construction to improve the training efficiency.

1.2 Major Contributions

Our main contributions can be stated as follows:

- *Multimodal CSI-based Human Activity Recognition (HAR) using GANs*: CSI-based sensing systems normally suffer from great performances degradation

under environmental dynamics. Besides, collecting enough new CSI data to retrain the system may not be possible in some situations. To address this issue, we propose a Multimodal CSI HAR system. It measures different human activities using CSI data. It addresses the performances degradation of WiFi-based human recognition systems due to environmental dynamics.

- *Empowering WiFi-based Passive Human Gesture Recognition to Unseen Environment via Domain Generalization*: Existing CSI-based smart sensing systems must be trained with CSI data from the deployed environment to adapt to the new environment settings. They use massive unlabeled high-quality data from the new environment, which is usually unavailable in practice. Besides, sometimes it is not possible to collect data from the deployed environments due to privacy concerns. Therefore, we propose a novel augmented environment-invariant robust WiFi gesture recognition system named AirFi that deals with the issue of environment dependency from a new perspective. The AirFi is a novel domain generalization framework that learns the critical part of CSI regardless of different environments and generalizes the model to unseen scenarios, which does not require collecting any data for adaptation to the new environment.

- *Robust WiFi-based Human Authentication System via Few-shot Open-set Recognition*: CSI-based human authentication systems in the literature require many CSI samples to train deep learning networks models, and are not able to detect unknown strangers. To solve this problem, we propose a CSI-based human authentication system (CAUTION). It utilizes distinctive gait features of each individual user via CSI data to recognize different human users. Leveraging few-shot learning, CAUTION constructs its model with only a few CSI samples.

- *CSI-based Life-long Smart Sensing System*: The ever-changing surroundings can cause significant performance degradation. Retraining the system with CSI data collected in the new environment settings leads to catastrophic forgetting of previous training knowledge. We propose a new life-long learning CSI-based human activity recognition system named LICAR. LICAR uses the simulated meta CSI training data from the CSI augmentation generator to reduce the distribution difference between each set of CSI training data. Most importantly, LICAR is added with a parameter updating buffer. It

is optimized with the CSI data from the new settings meanwhile keep the knowledge for the previous CSI training tasks by updating different parameters selectively. LICAR is equipped with the life-long learning ability without the requirement of previous CSI data.

1.3 Outline of the Thesis

Chapter 1 introduces the motivation and objective of this thesis. Chapter 2 reviews the related literature. In Chapter 3, we develop a multimodal CSI based activity recognition system, which is able to transfer the CSI data to different environment settings using a generative adversarial network in a multimodal manner. In Chapter 4, we generalize a training CSI based gesture recognition system to unseen environments using domain generalization. In Chapter 5, we build a novel CSI based user authentication system. Via few-shot learning, the system model can be constructed using a few CSI samples. In Chapter 6, we address the issue of performance degradation of CSI-based sensing systems in dynamic environment settings using life-long learning techniques. Chapter 7 concludes the thesis and discusses future works.

Chapter 2

Literature Review

2.1 Overview of Sensing Techniques

A variety of sensing techniques have been applied to a wide range of applications. Systems are equipped with the ability to observe the surrounding environments and interact with them accordingly. These sensing techniques provide great intelligence and convenience to the construction of modern smart cities. In this section, various sensing technologies are summarized and analyzed.

2.1.1 Vision Based Sensing Techniques

Vision based sensing techniques normally require deployment of cameras. Based on the pictures and video captured, those systems sense and interact with the surrounding environments. For example, closed-circuit television (CCTV) is widely used nowadays for surveillance purpose. The rapid development of computer vision and machine learning enables the vision-based sensing system to perform various complicated tasks such as object recognition and detection, activity recognition, and semantic segmentation. It has high accuracy and granularity. It is applied in many fields and achieves good performances. However, it requires line-of-sight and necessary illumination conditions, otherwise its performance can be severely affected. Besides, in certain places, the installation of cameras may raise privacy concerns. The camera can also induce additional costs.

Vision based sensing techniques have given decent performances in different sensing jobs. The reference [12] proposes a method that combines a relevance model with general object recognition. It manages to recognize the objects and scene. Based on the relevance model between different objects and scene calculated by a Bayesian network, it can improve the detection accuracy. [13] develops a vision-based solution for identifying and locating household objects in users daily life. In [14], a vision-based moving objects detection system is proposed. In [15], an Object Detection model based on the Single Shot Detector (SSD) algorithm is proposed and can be utilized for different real-world scenes.

2.1.2 Wearable Sensors Based Sensing Techniques

Wearable device-based sensing techniques are also very popular nowadays. They normally have the ability to measure the velocity, acceleration, gravitational forces and moving directions. Many modern devices such as smart phone, watch and etc are equipped with these functionalities. They can collect the data stream from their users and use a statistical model to predict the users' behaviors. Wearable device-based sensing techniques can perform users activity recognition and localization. However, it still has certain disadvantages. Firstly, it can only identify simple activities by treating the user as one whole unit. For very precise gesture recognition, it does not perform well. Besides, it mainly uses a statistical model for activities recognition, which needs empirical data for model construction. Wearable devices induce extra costs. Users have to carry these devices with them, which is not user-friendly. Wearing these devices may also cause some privacy concerns in some situations.

In [16], the authors develop a novel wearable system based on a new set of 20 computationally efficient features and the Random Forest classifier to recognize human activities. In [17], a wearable intelligence device for activity monitoring applications is presented. It is able to recognize activities with data extracted from accelerometer and a camera. Acceleration data are inputted to a trained model with nine categories. The image sensors use multiple optical flow vectors computed as features for defining an activity. Besides activity recognition, it can also be utilized for localization purpose. [18] presents a novel method for mapping and localization in indoor environments using a wearable gesture interface, which

consists of an infrared proximity sensor and a dual axis accelerometer. In reference [19], the authors propose a memory efficient localization system. It takes the advantages of a KNN classifier with SVM model.

2.1.3 WiFi Based Sensing Techniques

WiFi-based smart sensing systems do not require deployment of additional hardware or any wearable devices. Nowadays, WiFi access points are available in most of the commercial and residual buildings. WiFi modules are embedded in all kinds of IoT devices including Televisions, smart speakers, and power switches. As a result, different CSI-based sensing applications by analyzing fine-grained Channel State Information (CSI) data become feasible. Recent literature has witnessed many successful employment of CSI measurements for various applications. In [20], the authors build a CSI-based gesture recognition system by mapping the sequences of positive or negative doppler shifts to human gesture. Anti-Fall [5] has utilized both phase and amplitude of CSI features as salient features of human behaviours, which are the inputs to the support vector machine. DeepFi [6] studied CSI-based fingerprinting for user monitoring, which makes use of a deep learning network with labelled CSI as the training data. RT-Fall [7] is another a pattern based fall detection system, which learns different signal patterns and uses Support Vector Machine (SVM) to classify different received signal into different categories determined during the training phase. Reference [21] uses the CNN model trained on a natural image classification task to Wi-Fi gesture recognition, which can distinctly reduce the amount of training data required. WiGest [22] identifies gestures according to the variation of RSS, but RSS is vulnerable to environmental dynamics. Moreover, CSI performs excellent granularity with regard to sensing ability. Smokey [23] utilizes CSI samples to monitor smoking in public area. WiKey [24] simulates keyboard inputs by using wireless signals. Though WiFi-based sensing techniques have achieved good performance and are widely used in many applications, there are still some challenges in this field. With training and testing data from the same data domain, these systems can perform very well. However, the sensing environment could have various changes in the real world, such as layout of furniture and equipment, positions of surrounding people, etc. As a result, the testing data may have different features from training data. This can affect the performance of the systems significantly. Besides, WiFi-based

smart sensing techniques which take the advantages of machine learning normally requires a large number of training data. It affects the scalability and training cost of the WiFi-based smart sensing technique. After the systems are deployed, it need to adapt to the dynamic changes of the surrounding environments. However, simply retraining the system using data collected in the ever-changing environment may cause performance degradation of previous tasks.

2.2 Channel State Information

There are two kinds of information at the physical layer of wireless communications: Received Signal Strength Indicator (RSSI) and Channel State Information (CSI). RSSI indicates the process of signal propagation using signal strength. State-of-the-art smart sensing system PAWS [25] uses the measured RSSI data for human activity recognition. However only using the signal strength is not enough to fully describe the signal propagation.

CSI has received more attentions in recent years. It enables WiFi-based sensing technology, since CSI varies with the changes of surrounding environment. CSI is a fine-grained measurement at the physical layer as a subcarrier-level channel measurement [8–10], offering exquisite channel description. During the propagation of wireless signal, channel state information is a representation of the channel properties of a communication link. Wireless signals are affected by physical environment and surrounding humans, which results in signal reflections, diffractions and scattering [26]. These effects can be captured using CSI so that it can reveal various influences of surrounding environment. It can also reflect the complex multipath effect caused by intruders motion due to its frequency diversity and capture small scale multipath propagation of WiFi signal from transmitters to receivers over multiple subcarriers. Modern WiFi devices adopt Orthogonal Frequency Division Multiplexing (OFDM) at the physical layer. They follow the 802.11n/ac protocol and allow multiple input, multiple output (MIMO) with multiple antennae. As a result, CSI fully describes fine-grained characteristics of wireless signals including the effects of time delay, amplitude attenuation and phase shift of multiple paths on each communication subcarrier, which has a higher resolution. The WiFi signals can be modeled as Channel Impulse Response (CIR) $h(\tau)$ in time domain:

$$h(\tau) = \sum_{l=1}^{L} a_l e^{j\phi_l} \delta(\tau - \tau_l), \tag{2.1}$$

where $\delta(\tau)$ is the Dirac delta function. Among the L multipath components, α_l is the amplitude of the l_{th} component and ϕ_l is the phase information, τ_l represents the time delay. Due to limited bandwidth of WiFi, the OFDM receiver is able to provide a sampled version of the signal spectrum at subcarrier level. By comprising amplitude attenuation and phase through complex number, the sampled signal spectrum can be written as:

$$H_i = \|H_i\| e^{j\angle H_i}, \tag{2.2}$$

where $\|H_i\|$ is the amplitude of i_{th} subcarrier, while $\angle H_i$ is the phase of i_{th} subcarrier. Theoretically, though CSI phase information is supposed to be more robust with fewer variations, it may perform otherwise due to hardware imperfections and environmental variations [8]. We observe a perturbation of 100 kHz for 5 GHz band on devices. In practical applications, it brings difficulty in calibration accomplishing and denoising. Therefore, we choose the amplitude information for our systems design.

2.3 Overview of Learning Techniques

Using sensing data, modern machine learning techniques help to extract high-level representations of various patterns and achieve remarkable performance. Besides, with the help of transfer learning, knowledge from one domain can be transferred to different domains, which helps to generalize the systems across different environment settings. To tackle the issue that most current systems require large amount of training data, few-shot learning also inspires us to build systems which can be trained with as few data as possible.

2.3.1 Deep Representation Learning

For feature extractions on high dimensional data, deep learning is very crucial. Normally, the dimensions of CSI data samples are very large. For instance, in our experiments the dimensions of the CSI raw data is 342 by 500. To perform a direct representation learning on data samples of this size can be very difficult and time consuming. To downsample the high dimensional CSI data into low dimensional CSI data, deep learning techniques can be utilized. Convolutional Neural Network (CNN) can be applied to extract the spatial features [27]. CNN is a very popular technique in the deep learning field. It has been widely used in the field of computer vision and achieved remarkable performances for object detection and segmentation [28]. It consists of convolution layers, pooling layers, activation and fully connected layers. The structure of CNN is firstly proposed in [27]. The convolutional layers downsample the input and extract features according to the kernel sizes. With the extracted features, the pooling layer can subsample the feature codes and decrease the code sizes. The fully connected layers map the feature codes to output layers and works as a classifier. Based on the optimization objective function, the networks will calculate the corresponding loss for each predicted output and ground truth. Then the whole network is optimized by the back propagation. To improve the networks for capturing both short-term and long-term representation patterns, the Recurrent Neural Network (RNN) was proposed [29]. Long Short-Term Memory is one particular design of RNN. Several WiFi-based smart sensing techniques have utilized this network to build their system model [30, 31]. However, they normally require a large number of training CSI samples and long training time, which is not acceptable in some situations.

2.3.2 Transfer Learning

Transfer learning [32] aims to use limited training data to solve problems in related or even new domains. The initial idea is to solve the problem of lack of labeled data. By making use of unlabeled data, the model can generalize better. As transfer learning develops fast, more and more problems emerge, of which a crucial one is domain adaptation [33]. It aims to make predictions in a target domain by learning a related source domain but with different distribution. Domain adaptation can be achieved by many approaches. Pan et al. proposes transfer

component analysis [33] to learn a latent representation by minimizing the maximum mean discrepancy. It can also be dealt with by some adversarial strategies, such as adversarial discriminative domain adaptation [34]. Recently, generative adversarial networks (GAN) [35] also received increasing attentions. In GAN, the generative model is built against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution. The generative model can be thought of as a completion between a team of counterfeiters who manage to produce fake currency, while the discriminative model can be seen as the police who manage to distinguish between genuine currency and fake currency from these counterfeiters. The competition drives both parties to improve their skills until the fake currency becomes indistinguishable from the genuine currency. GAN can be classified into two categories: supervised learning GAN and semi-supervised learning GAN. For supervised learning GAN [36–39], the mapping is learned using a training set of aligned pairs of data from two domains. However, to align data pairs, it is necessary to label data from different domains, which is very difficult in practice due to limited time and labor. For semi-supervised GAN [40–44], it does not require labelled pairs between data from different domain. Semi-supervised GANs are typically applied to scenarios where unlabeled data are abundant [45, 46]. However, for our cases, unlabeled data are generally limited. To address this issue, inspired by GAN applications on image-to-image translation, we use GAN to translate data from source domain (where data could be collected relatively easily, such as manufacturing places) to target domain (where data is difficult to collect, such as users places). With the help of this, enough data from different domains can be acquired. This is very helpful when we deal with problems, such as environmental dynamics and users heterogeneity. With data-to-data translation using GAN, it is possible to acquire enough sensor data from different environment settings and users, which can provide more information for classifier training.

2.3.3 Few Shot Learning

Deep learning models have achieved great success in recognition tasks [47–49]. Many smart sensing systems utilize deep learning to build their system models. However, large amounts of data and many iterations are needed to train their parameters, which reduce their scalability to new classes due to annotation cost.

Besides, training a deep learning model with large amount data may cause overfitting. To address this problem, few-shot learning is seen as an alternative to deep learning.

Few-shot learning aims to recognize novel visual categories from very few labelled examples. Unlike deep learning models which extract features from large amounts of data to construct a detailed model of all classes, the few-shot learning model is built by comparing the similarity between the training data and new data. There has been a recent resurgence of interest in few-shot learning. Reference [50] learns a Mahalanobis distance to maximize K Nearest Neighbor (KNN) accuracy in the transformed space. Reference [51] maximizes KNN accuracy using a neural network. Reference [52] also aims to optimize KNN accuracy with a hinge loss that encourages the local neighborhood of a point to contain other points with the same label. To improve the system availability, the number of training data needed is reduced to be as few as possible. In our case, most existing machine learning based smart sensing systems require large amounts of data for training. Collecting large amounts of data can be very expensive and even impossible in some situations, which poses a great challenge to current systems. Inspired by few-shot learning, we use very few samples for model construction, which reduces the cost of model construction and improves its scalability to new classes.

2.3.4 Domain Transfer and Domain Generalization

In the last few years, great success has been achieved by machine learning. The corresponding works have benefited many real-world applications including the CSI-based human sensing field [53]. However, it takes a lot of resources to collect and annotate each dataset for new tasks. Especially when the number of samples and domains are very large, it can be an extremely resource-consuming and time-consuming process. Besides, sufficient data samples will not always be available in certain circumstances. For example, it is not user-friendly to collect large numbers of data from users when the systems are deployed in users' places. This motivates the research works on reusing a trained model in a new domain. Domain adaption is one of the methods proposed to achieve this goal.

Recent works focus on transferring network representations from the source domain where labeled data datasets are easy to acquire to a target domain where labeled

data is sparse or even non-existent [34]. Reference [54] proposes a new CNN architecture that introduces an adaptation layer and an additional domain confusion loss, to learn a representation that is both semantically meaningful and domain invariant. In [55], a new Deep Adaptation Network architecture is proposed which generalizes deep convolutional neural network to the domain adaptation scenario. Reference [44] makes a shared-latent space assumption and proposes an unsupervised image-to-image translation framework based on Coupled GANs. In [41], a multimodal unsupervised image-to-image translation framework is proposed by assuming that the image representation can be decomposed into a content code that is domain-invariant, and a style code that captures domain-specific properties. Both [44] and [41] are able to transfer data from one domain into another domain without changing the categories of the data samples. In [56], the CoGAN learns a joint distribution of images in the two domains from images drawn separately from the marginal distributions of the individual domains by enforcing a simple weight-sharing constraint. The main strategy is to guide feature learning by minimizing the difference between the source and target feature distributions. Some other methods also manage to minimize the maximum mean discrepancy (MMD) loss for this purpose [57]. Domain adaption has achieved great success and benefited many systems and applications. However, the limitation of domain adaption is that it still needs many data samples from the target domain in order to perform the domain transfer of system models. In many other scenarios, there may not be any data of the target domain during the training phase, but the system is still needed to build a precise model for a totally new target domain. To address this problem, domain generalization is proposed.

DG leverages the labeled data from multiple source domains to learn a universal representation, which is expected to generalize well for an unseen target domain [58]. DG is firstly introduced in [59]. They identify an appropriate reproducing kernel Hilbert space and optimize a regularized empirical risk over the space. Then in [60], a discriminative framework is used to directly exploit dataset bias during training. In [35], the authors propose a new framework for estimating generative models via an adversarial process using a generative model and a discriminative model. Reference [61] utilizes maximum mean discrepancy (MMD), which leads to a simple objective that can be interpreted as matching all orders of statistics between datasets and samples from the model. Reference [62] leverages deep neural

networks for domain-invariant representation learning and achieves end-to-end conditional invariant deep domain generalization. The above works inspire us to use several environments where data is relatively easier to collect as source domains. Then we generalize and apply the trained model into a new environment which is regarded as the target domain. As it is difficult to collect large numbers of CSI data to train a generalized system model in our research problem, we utilize the data and feature augmentation techniques to improve the model generalization.

2.3.5 Life Long Learning

When a sensing system is deployed in the target environment, it can perform designed tasks accurately under the static environment setting. While in a real world situation, the surrounding environments are usually more dynamic than static. The frequent changes caused by furniture layout and human users lead to serious performance degradation. In order to adapt to different environment settings and maintain its performances, it requires the ability of life-long learning. To achieve the long-term robust, one popular method is to frequently update the system model by retraining with new CSI samples. In [63], an automated unsupervised retraining algorithm is designed. The system first judges if the environment is dynamic based on the incoming testing CSI series after preprocessing. Then an event detector is applied to the testing feature generated from the raw CSI series to determine which event is happening if the proposed dynamic detector detects dynamics in the environment. Finally, the system will collect new training sequences from the new testing series to update the system model.

For existing CSI based smart sensing systems which construct their system model with deep learning models, the usual assumption is that the training data for all tasks are available. For example, when we train a CNN based CSI smart sensing system, we normally take for granted that CSI data from all required settings are available. However, with the increasing numbers of tasks, the training difficulty with all stored data also grows. A new problem arises where we add new capabilities to a Convolutional Neural Network (CNN), but the training data for its existing capabilities are unavailable. Meanwhile, they are required to learn new tasks and still maintain the performances on previous tasks without suffering catastrophic forgetting.

A Learning without Forgetting method is proposed in [64]. It trains its network model using data for the new task only while preserving capabilities on previous tasks. Compared to other commonly used feature extraction and fine-tuning adaption techniques, the learning without forgetting method performs similarly to multitask learning. In reference [65], a new concept of a neural network capable of combining supervised convolutional learning is proposed. The network model is inspired by the human brain that it leverages the concept of spike-timing-dependent plasticity (STDP) to enable continual learning and prevent catastrophic forgetting. In [66], a neural structure optimization component and a parameter learning component are combined to build a deep neural network continual learning model. In [67], the authors add intelligent synapses into their model. The synapse accumulates task relevant information over time, and transfers the knowledge to rapidly store new memories during the network training to prevent catastrophic forgetting. In [68], a Dynamically Expandable Network (DEN) is proposed, which can decide the network capacity and learn a compact overlapping knowledge sharing structure among tasks. [69] proposes a net scheme called elastic weight consolidation to make a compromise between different tasks. In [70], the authors propose to learn object detectors incrementally. A new distillation loss is designed to minimize the discrepancy between responses for old classes from the original and the updated networks.

2.4 CSI Based Systems Under Environment Dynamics

There are some research works on CSI-based human activity recognition under environmental dynamics, and they can be classified into two categories: statistical signal information based techniques and transfer learning based techniques.

2.4.1 Statistical Information Based Techniques

Statistical signal information based techniques aim to extract statistical signal information which mitigates the interference from environmental dynamics. Based on a large number of experiments and CSI data collected, they build signal models

which are more representative and general under different environment settings. Though CSI-based HAR systems can handle some interference from the surroundings, strong domain knowledge is required. The model parameters also have to be tuned with large amounts of experiments. Furthermore, it is hard to map CSI statistic directly to different human activities. The training of such kinds of systems can be very difficult as it does not have many adjustable parameters. The generalized model trained without CSI data from the deployed environment may not fit to the new environment well.

References [71, 72] are able to detect human breathing and moving by Fresnel Zone model. Reference [73] measures the human movement speed via CSI. Then it links different speeds with different human activities. It takes the advantage of Discrete wavelet transform (DWT) for frequencies detection on multiple time scales. As the frequencies of signal detected via DWT are more stable under different environments. Reference [20] uses Doppler shift in WiFi signals to detect the directions of human motions with respect to the receiver. If users are getting closer to the receiver, a positive Doppler shift will be received. While users are moving further from the receiver, a negative shift will be generated. The system can perform activities recognition depending on the sequences received of Doppler shifts. WiAnti [74] utilizes different subcarriers selectively for HAR. Subcarriers with weak correlation are chosen as the input for the system to reduce the co-channel interference. In [75], authors use Angle Difference of Arrival to remove the environmental information and only keep the information of human activities.

2.4.2 Transfer Learning Based Techniques

Transfer learning is receiving more attention in the CSI-based HAR field. It involves CSI data from both the source domain (training environment setting) and the target domain (testing environment setting) to optimize the system, which equips itself with the ability to adapt to the new setting.

[21] builds a CSI based gesture recognition using a CNN model. It decreases the amount of CSI training data significantly. [76] utilizes the domain transfer to adapt its system model into different environments with labelled data collected from the new environment. [77] uses a roaming model to transfer the model into different environment with data translation. It collects labelled data from the new

environment and learns the translation mapping between CSI data between the different environments. Then it can retrain its system model with the translated CSI data. In [63], it measures the normalized distance between the CSI data from different environments. When the normalized distance is larger than a predefined threshold. The system retrains itself with the new CSI data. [78] also retrains the system when surroundings changes take place. It clusters the new CSI data and requires user inputs to retrain the system. [79] utilizes GAN to generate fake CSI data with two generators. The system model can adapt to the new environment using fake data. A teaching network is used to transfer the knowledge to a new network [80]. [81] constructs a standard Siamese structure to remove the background noise from the received CSI data.

System can adapt to different settings with the help of transfer learning. It can transfer the knowledge across different domains. The robustness of systems can be improved via transfer learning. However, existing CSI-based HAR systems still suffer from some drawbacks. Many of them have to be manually activated. Labelled data from the new environment are necessary to retrain the systems, which is not realistic. [63, 76] removes the manual activation by adding a threshold into the system. But to set the value of the threshold requires strong domain knowledge and the value can be very different when environment settings change. [79] greatly improve the previous works by using unlabelled CSI data to train the system model. However, it does not address the problem that the collected data from the new environment may be non-uniformly distributed. The fake data generated can be unbalanced due to this reason. The unimodal generator is not able to solve this problem. This greatly affects the system performances. To address this issue, the generators in the system model should be able to generate diverse fake CSI data under different dynamic settings, which helps the system to approximate the distribution of CSI data in other environments better.

2.5 CSI Based Human Authentication

In recent years, user authentication has become increasingly vital due to the growing concern of user security and privacy leakage. There have been some approaches on user authentication. Vision-based approaches use camera to identify suspicious

objects [82, 83]. References [84, 85] take the advantages of Radio Frequency Identification (RFID) systems to identify different users based on the RF tags on their bodies. However, for vision-based approaches, they are affected by illumination line and require line of sight. Besides, intruders may not wear RF tags on their bodies all the time which limits the use of RF Identification approaches. Though utilizing radar can achieve a decent performance, but it is expensive and restricted in scale, which hinder its civilian use.

Research has found that the gait of different human possesses unique feature patterns in the CSI measured by wireless systems. Human gaits possess distinct features for each individual person in the CSI data due to the physical characteristic differences. Therefore, the CSI data of human gaits can be used for user authentication. Many works have studied CSI based user authentication. In this section, we categorize them into two groups: Manually Extracted Feature (MEF) based methods and Network Self-learning (NSL) based methods.

MEF based methods select several signal parameters in the CSI sequences. They are clustered together to form the feature matrices. Systems are trained using these feature matrices. WFID [86] performs device-free user authentication via analyzing subcarrier-amplitude frequency (SAF) on CSI measurements. FreeSense [87] uses Principle Components Analysis to extract features of different users' gaits for the user identification purpose. WiFiU [88] uses fine-grained gait patterns as indicators to recognize different users such as walking speed, gait cycle time, footstep length and movement speed. Reference [89] computes the maximum and minimum amplitudes, skewness, mean kurtosis and standard deviation of the received signal in the time domain, and it also computes spectrogram magnitude, percentile frequency components, and spectrogram difference between time windows in the frequency domain. Wiwho [90] calculates maximum, minimum, mean, median, standard deviation, skewness and kurtosis of received CSI data. GateID [91] selects mean, max, min, skewness, kurtosis, variance and mean crossing rate of CSI waveform. In addition, frequency domain features including normalized entropy, normalized energy and FFT peaks are also utilized.

NSL based methods take the advantages of deep learning to build their system models. CSI sequences are used directly to train the system model [92]. WiAu [76] utilizes convolutional neural network to extract features from the CSI data. For CSIID [93], the long short-term memory is applied. HumanFi [94] uses the

LSTM to extract feature codes of gait for user identification. Wihi also uses the same technique [95]. A multiple layer CNN is applied in NeuralWave for feature extractions [96].

For CSI-based user authentication, MEF based methods are proved to have very accurate performances. However, they still have some challenges. Firstly, in order to design the feature matrices, some expert knowledge is needed. The feature matrix may work well among certain user groups while for some other user groups, it may not work well. Most importantly, many feature matrices designed can only perform well when users walk in a straight line. The extracted biometrics feature can be very different if users walk in a curve. But for NSL based methods, they do not have these limitations. Systems can learn directly from the CSI data and fit well into different walk paths. The issue that most NSL based methods share is that they require a large amount of CSI data to train their systems. To improve the accuracy, complicated networks are used for model constructions. They need a lot of CSI data to train them. We can get enough data when we train the system model in the lab. However, in practice, collecting large amounts of data can be very user-unfriendly and incur additional system costs. The scalability can also be affected greatly. It is necessary to build a new CSI-based authentication system that can be trained using limited amounts of CSI data.

2.6 Conclusion

In this chapter, we firstly introduce the channel state information and the reason that it can be applied to CSI-based smart sensing. Secondly, we review works on CSI based smart sensing. Then we review works on transfer learning and few-shot learning. We discuss on how they can be applied to improve the current CSI-based smart sensing systems. After that, we discuss on how existing CSI-based smart sensing systems handle the environmental dynamics. Finally, we review some existing works on CSI-based authentication, and discuss their drawbacks and how they can be improved. Based on the literature reviews, we propose new CSI-based smart sensing systems to improve the limitations of current existing systems in later chapters.

Chapter 3

Multimodal CSI-based Human Activity Recognition using GANs

3.1 Introduction

The environments are very dynamic rather than static in the real life. Changes from the furniture layout, surrounding people can usually affect the environment settings. For example, in smart homes or offices, environmental dynamics can be brought by layouts and furniture changes. In factories and warehouses, the layout of large machines and goods can also cause environmental dynamics.

Given a CSI based HAR system, it can be trained well with large amounts of CSI data collected within the environments. While after the dynamic changes within the environment take place, it may barely recognize human activities as the CSI patterns are different in the new environment setting.

Therefore, it is important that a CSI based HAR system is able to remain robust under the environment dynamics. There are some existing works that address this problem. In [77, 79], they use domain adaption to adapt their model to new environment settings. However, their methods require large numbers of labelled CSI data from the new environment setting, which is not always possible in the real life.

This chapter studies the problem of overcoming the environmental dynamics with limited data from the target environment for a CSI-based smart sensing system.

Assume that only a limited amount of unlabeled CSI data from different settings of the deployed environment are given, we aim to build a robust CSI-based activity recognition system that is able to give decent performances in the ever-changing target environment.

To achieve this, we design a novel CSI-based HAR system MCBAR. It is able to maintain robustness under different environment settings. With small amounts of unlabelled CSI data collected from the new environment setting, MCBAR can adapt to the new setting by domain adaption. It utilizes two generators in its model to generate fake CSI data. The fake CSI data have similar features to those collected in the new environment setting. These fake data can be used to approximate the CSI data distribution in the new environment and train the system model of MCBAR. it reduces the required amount of CSI data collected from the new environment setting. The domain adaption of MCBAR to the new environment setting is in a diverse way due to the multimodal structure of its generator. This diverse adaption process helps the system to improve its robustness in different environment settings. The classifier is also able to be trained with diverse information of CSI data in different settings. Experimental results show that MCBAR remains robust under different testing environment settings and outperforms those compared systems.

The contributions of the chapter are summarized as follows:

- We propose a novel human activity recognition system MCBAR, which utilizes multimodal GAN generators to provide the system with diverse information and improve its ability to handle different environment settings. To the best of our knowledge, MCBAR is the first CSI application system that applies a multimodal GAN to handle environment dynamics.
- We improve the boosting generator by incorporating a marginal loss to make the generated outputs uniformly distributed. A new objective function of the classifier is proposed along with a new diverse fake data generation framework.
- Extensive experiments demonstrate that MCBAR outperforms state-of-the-art systems and overcomes environmental dynamics.

3.2 Problem Formulation

For CSI-based Human Activity Recognition (HAR), it is relatively easy for a CSI-based HAR system to achieve good performances when the surrounding environment is static. While in the real ever-changing environment settings, the dynamic changes affect the system's performance greatly. To maintain the system's robustness, it is required to adapt to the new environment settings. Existing methods need a large number of labelled data to retrain the system, which is hard to achieve in many real-life situations [77, 79].

Consider the original training environment setting and the new environment setting as the source domain and the target domain respectively. Suppose we collect labelled CSI data $\{X_l, Y\}$ from the source domain, where X_l is the collection of labelled CSI data and Y is the collection of the corresponding labels. We train the CSI-based HAR system within the source domain by optimizing the cross entropy loss. Then we only collect the unlabelled CSI dataset X_u from the target environment. The ultimate goal is that we adapt our system model from the source domain to the target domain only using unlabelled CSI data X_u, and the system model is able to give decent performances in the target domain which is the new environment setting.

3.3 System Overview of MCBAR

To address the issue that environment dynamics affect the performance of CSI-based HAR system, we propose a novel system MCBAR. It has three parts: two generators and one classification model as illustrated in Fig. 3.1. The first part is the boosting generator. It is trained to generate fake CSI data which are similar to real CSI data from the target domain during the boosting generator training phase. As only limited amount of unlabelled data from the target domain can be collected before the system training, the boosting generator is applied to generate more unlabelled fake data to help with the system training. The second part is the translation generator. The translation generator is trained to transfer domain knowledge from the source domain to the target domain with limited unlabelled data from the target domain. It takes all the labelled CSI data from the source domain and translates them into the target domain. As the translation process does

not change the variety of activities, translated outputs can inherit their original activity labels from source domain. The translation generator is used to improve the situation that certain activities from the target domain are not captured during the data collection. It provides a better approximation of the target domain CSI data distribution. Finally, both the real CSI data collected from the two domains and fake CSI data generated from two generators are used for the classification model.

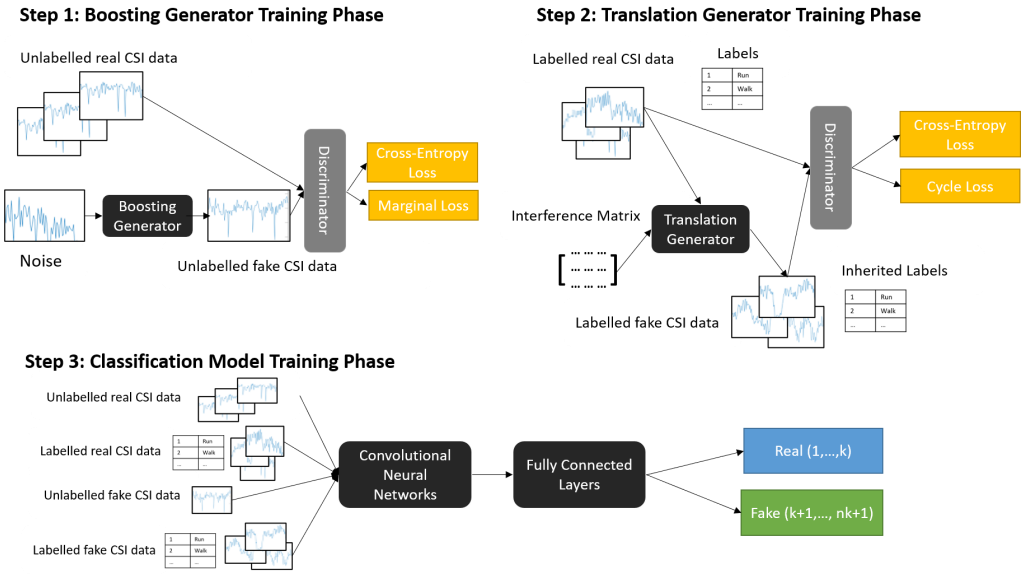


FIGURE 3.1: MCBAR System Chart

3.3.1 Boosting Generator

Suppose that we have two sets of real data: labelled data set $\{X_l, Y\}$ and unlabelled data set X_u. $\{X_l, Y\}$ is collected from the training environment which is the source domain and X_u is collected from the new environment which is the target domain. Let x_l be the CSI sample in X_l, and y is the activities' ground truth label in the label set Y. The CSI sample in the set X_u is denoted as x_u. The boosting generator G_{bo} is trained to generate unlabelled fake CSI data x_{fu} which is similar to x_u. In order to train the boosting generator, a discriminator D is used to distinguish real samples from generated samples and propagates back the loss of differences between real and generated CSI samples. The generator is optimized to minimize the differences between real samples and generated samples.

The unlabelled data collected from target domain may not be uniformly distributed as some activities are not performed by users regularly. We encourage the generated data to be more uniformly distributed as it can help the system fully explore the features of CSI data for all activities. To achieve this, we design the marginal loss $\mathcal{L}_m$ for training the boosting generator, which is the shannon entropy between the generated outputs of different activities. Shannon entropy is defined as the expected value of the information term carried by a sample from a given distribution. If the class distribution for a given element is uniform, it means we are uncertain on the class assigned to this element. In other words, we can enforce the model to increase the entropy of the batch-wise generated CSI samples to make them more uniformly distributed. This can be achieved by maximizing the shannon entropy of the generated CSI data, which is represented as

$$\mathcal{L}_m = H[\frac{1}{M}\sum_{i=1}^{M} p(y|G(x^i, D))], \tag{3.1}$$

where H is the shannon entropy, p is the class distribution, M is the number of generated data, i is the index of the generated data and y is the output class. By maximizing the marginal loss, the output data distribution can be more uniform. Combined with the entropy loss of minimizing the difference between real data and generated data, the boosting generator loss $\mathcal{L}_{G_{bo}}$ is given by

$$\mathcal{L}_{G_{bo}} = \min_{G_{bo}}[-\lambda\mathcal{L}_m + \mathcal{E}_{z\sim p_z}[log(1 - D(G_{bo}(x)))]], \tag{3.2}$$

where λ is the weight for the marginal loss and $\mathcal{E}$ is the entropy loss. We can use the trained boosting generator to generate fake data in the target domain.

3.3.2 Translation Generator

The translation generator consists of two parts: an encoder E and a decoder G. It is used to translate CSI data into different domains. To train the translation generator, it samples a random portion of CSI data from source domain and target domain and then translates the data from the source domain into the target domain. The discriminator is used to distinguish the translated outputs from target domain data. The distribution divergence can be minimized by adversarial training w.r.t the translation generator.

The translation generator in MCBAR is of a multimodal structure by leveraging style transfer techniques [41]. In [41], a new domain transferring technique using a style matrix is proposed. The style matrix is randomly simulated and used to provide diversity to the transfer process. Pictures transferred with different style matrices can have different styles. In our translation generator, a randomly generated interference matrix is used to simulate the environmental dynamics that may affect the CSI data. The translation generator can translate CSI data of one activity from the source domain into the target domain with different interference matrices. As a result, the CSI data of this activity interfered by different environmental dynamics in the target domain can be simulated. With the randomness brought by this interference matrix, CSI data can be translated into different domains in diverse ways. Taking the advantages of diverse translated CSI data, MCBAR has more information on CSI data under different environmental dynamics.

Let x_1 and x_2 represent two CSI samples from the source domain and target domain respectively. We denote the CSI content matrix as c and environmental dynamic interference matrix as s_j where j is the interference matrix index. Then the CSI data can be viewed as a combination form of c and s_j, such that $x_1 = G_1(c, s_1)$ and $x_2 = G_2(c, s_2)$ where G_1 and G_2 are decoders for the source domain and target domain respectively.

To translate x_1 into the target domain, we firstly use encoder E to encode x_1. The encoder E is used to extract the corresponding content matrix and interference matrix, $(c, s_1) = E(x_1)$. Then generator randomly simulates another interference matrix s_2. Decoder G_2 recombines it with c which is extracted from x_1 to get the translated data $x_{1\to2} = G_2(c, s_2)$. Then a discriminator D will be used to distinguish the generated sample from target domain data. In order to optimize the generator, we minimize the translation loss $\mathcal{L}_t$ as follows

$$\mathcal{L}_t = \min_{G_2} \mathcal{E}_{c,s_2}[log(1 - D(G_2(c, s_2))). \tag{3.3}$$

With different s_2, the translation generator can generate diverse translated outputs using the same c. Therefore using data from one domain, we can produce diverse fake data in another domain, which is corresponding to different environmental interference. This can help the system to handle various kinds of environmental dynamics and provide an approximation of the target domain data distribution.

Instead of generating deterministic translation results, the translation generator in our system is of a multimodal structure. The decoding of generated CSI data using decoder G involves the randomly simulated interference matrix. It provides diversity to the translation process. The translated CSI data has the same content with different features corresponding to different interference matrix. With the help of the multimodal structure, more generated data with different features are available for system training. This improves the system's robustness under different environmental dynamics.

The system also manages to reverse the translation which helps to push the convergence. We use an encoder to encode $x_{1\to2}$, $c^*, s_2^* = E_2(x_{1\to2})$ where c^* and s_2^* are newly extracted content and interference matrix. With c^*, we recombine it with s_1 extracted before. The reconstruction loss calculates the similarity between the recombined data and original x_1. This reconstruction loss helps the system to unify content code c between domains and push the convergence of the generator training. The reconstruction loss $\mathcal{L}_R$ is

$$\mathcal{L}_R = \mathcal{E}_{x_1}[\|G_1(c^*, s_1) - x_1\|_1]. \tag{3.4}$$

Combining the translation loss and reconstruction loss, we get the overall translation generator loss as

$$\mathcal{L}_{G_{tr}} = \mathcal{L}_t + \beta\mathcal{L}_R, \tag{3.5}$$

where β is the coefficient of reconstruction loss. After training the translation generator, we translate the rest data from the source domain to the target domain. The translated data x_{fl} can get its inherited label y' by inheriting their original labels y from the source domain. Besides, by using different s_2, we can generate multiple sets of x_{fl}. All these labelled generated data can be used to train our classifier.

3.3.3 Classification Model

The classification model consists of two parts: a CNN based feature extractor and a fully connected layer based classifier. CNN is widely used in solving classification problems. It reduces the needs of expert knowledge as its feature extractors can

effectively learn relevant features for high dimensional data. Our classification model consists of three convolutional layers $C(n_k \times n_k; n_{fm})$, where n_k is the kernel size and n_{fm} is the number of feature maps, three max pooling layers P and three fully connected layers F. These last three fully connected layers construct the classifier. The model architecture is represented by the shorthand notation: $C(5 \times 5; 32) \rightarrow P \rightarrow C(5 \times 5; 128) \rightarrow P \rightarrow C(5 \times 5; 128) \rightarrow P \rightarrow F \rightarrow F \rightarrow F$. We use leaky rectified linear units (leaky ReLus) in each layer. With the real data x_l and x_u, fake data x_{fu} from the boosting generator and n sets of diverse fake data x_{fl}, the classifier needs to classify data into $(n+1)k+1$ categories where k is the number of different activities that we want to classify. The first k classes are real data classes $(1, ..., k)$, and fake data class from translation generator are $(k+1, ..., nk)$ and 1 class for fake unlabelled data from boosting generator. The overall objective function $\mathcal{L}_o$ is as follows

$$\begin{aligned}\mathcal{L}_o = &- \mathcal{E}_{x_l,y} log[p(y|x_l, y < k+1)] \\ &- w_1 \mathcal{E}_{x_u} log[p(y < k+1|x_u)] \\ &- \sum_{a=1}^{n} w_{2,n} \mathcal{E}_{x_{fl,y'}} log[p(y'|x_{fl}, ak < y < (a+1)k+1)] \\ &- w_3 \mathcal{E}_{x_f u} log[p(y > nk|x_f u)],\end{aligned} \tag{3.6}$$

where a is the index of output data sets from translation generator, y and y' are labels from the source domain and target domain respectively. w_1, w_2 and w_3 are weights for each term in L_c and set by validation. The first term is to check whether x_l has been put in the correct class within the first k classes. The second term is to check whether x_u has been placed within the range of real data categories $(1, ..., k)$. The third term is the critical part. As the translation generator has generated n sets of diverse outputs, within each set the translated data should be put in the correct class in the range of labelled fake data classes. It provides the system with more information of possible CSI data for one class in different environment settings due to multiple possible dynamic changes. Instead of training with only one set of deterministic translated outputs, the classifier is trained with more diverse and balanced information generated from the boosting generator and the translation generator. This prevents insufficient training due to limited information and further improves the capability of the system in handling the ever-changing dynamic environment.

3.3.4 System Training

We summarize the training procedure of MCBAR. As shown in Algorithm 3.1, with the unlabelled data collected from the target environment setting, we firstly train the boosting generator. Then we use the trained boosting generator to generate more uniformly distributed unlabelled fake data. Using parts of data from the source domain and target domain, the translation generator learns to translate data from the source domain to the target domain. Having the translation generator trained, all data from the source domain can be translated to the target domain. These translated outputs can inherit their labels from the source domain. Finally, the classifier is trained using $x_u, x_{fu}, (x_l, y)$ and (x_{fl}, y').

Algorithm 3.1: Training Phase of MCBAR

Input: Labelled data (x_l, y), unlabelled data(x_u), weight parameter$w_1, w_{2,n}$, number of output data sets n, number of boosting generator training iterations $N_{itrTrans}$, number of translation generator training iterations N_{itrBo}, number of classifier training iterations N_{itrCl}

Initialize translation generator with parameter θ_{tr}, boosting generator with parameter θ_{bo} and classifier with parameter θ_c

while $num\_iteration \leqslant NitrBo$ **do**

1. Input the unlabelled data x_u;
2. Update θ_{bo} for training the boosting generator;
3. Generate x_fu using the trained boosting generator;

while $num\_iteration \leqslant NitrTrans$ **do**

1. Sample a batch of labelled data (x_l, y), real unlabelled data x_u and fake unlabelled datax_{fu};
2. Update θ_{tr} for training the translation generator;
3. Generate n sets of fake labelled data $(x_{fu}, y'$ using the trained translation generator, where $y' = y + nk$;

while $num\_iteration \leqslant NitrCl$ **do**

1. Input the unlabelled data x_u x_{fu}, and labelled data (x_l, y), (x_{fl}, y');
2. Update θ_c for training the classifier by minimizing the classification loss L_c;

3.4 Experiments

We test the performance of MCBAR in several experiments. The design of experiments, results and analysis are shown in this section.

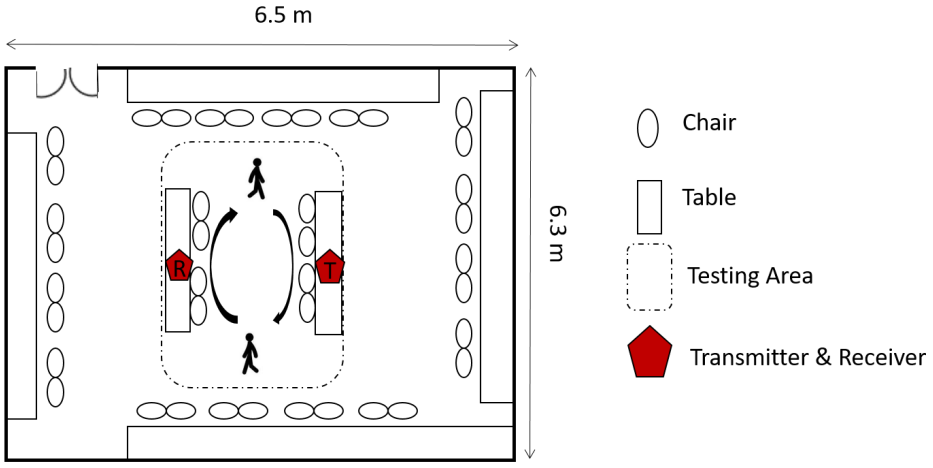
3.4.1 Environment Setup and Data Collection

We perform the experiments in two locations, a lab and a cubic office. Their layouts are illustrated in Fig. 3.2. As shown in Fig. 3.2, we use two routers in our experiments. The router with one antenna is used as the transmitter and the other router with three antennas is used as the receiver. We upgrade the firmware of them to build the CSI platform [97]. We set the router to work at 5 GHz with a bandwidth of 40MHz. The transmitter works as the master and the receiver works as the client. Six kinds of activities are tested in the experiments including running (r), walking (w), falling down (fa), boxing (b), circling arms (cc), and cleaning floor (cl).

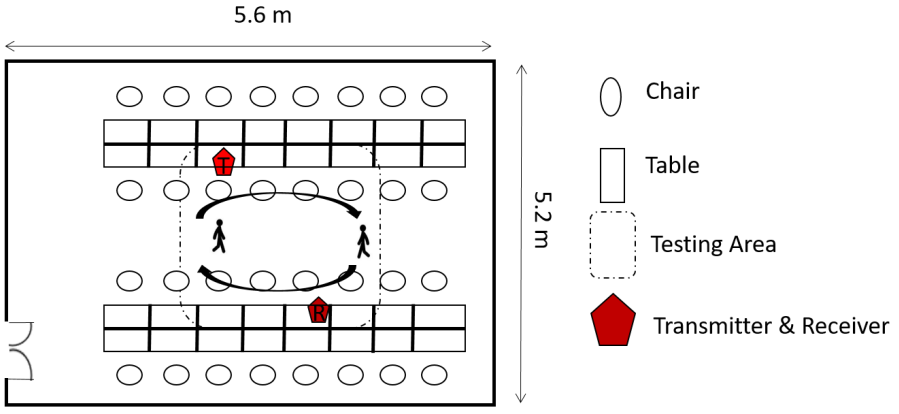
There are ten people that participate in the CSI data collection as the volunteers in the experiments. They performed the activities within the testing range that is shown in Fig. 3.2. During the process, the transmitter sends WiFi signal to the receiver. Our platform records the CSI data received. MCBAR only utilizes the amplitude of the CSI data and removes the phase information.

In order to test the robustness of MCBAR, we design four different environment settings. Every volunteer is required to perform all six activities in each setting at both lab and cubic office locations.

- Environment A (E-A): the primary setting of the environment, only one person performs six activities within the test area. The collected CSI data are used as source domain data and the corresponding label information is available for the training phase of MCBAR.
- Environment B (E-B): only one person performs six activities within the test area. The layouts are changed during the CSI data collection. We add six 27-inch iron plates as obstacles in the testing area. The positions of lab benches and chairs are moved to simulate environmental dynamics.
- Environment C (E-C): while one person performs six activities within the test area. The rest of the people perform actions such as standing, sitting, gesturing, and chatting around the testing area to simulate the dynamics changes of surrounding people.

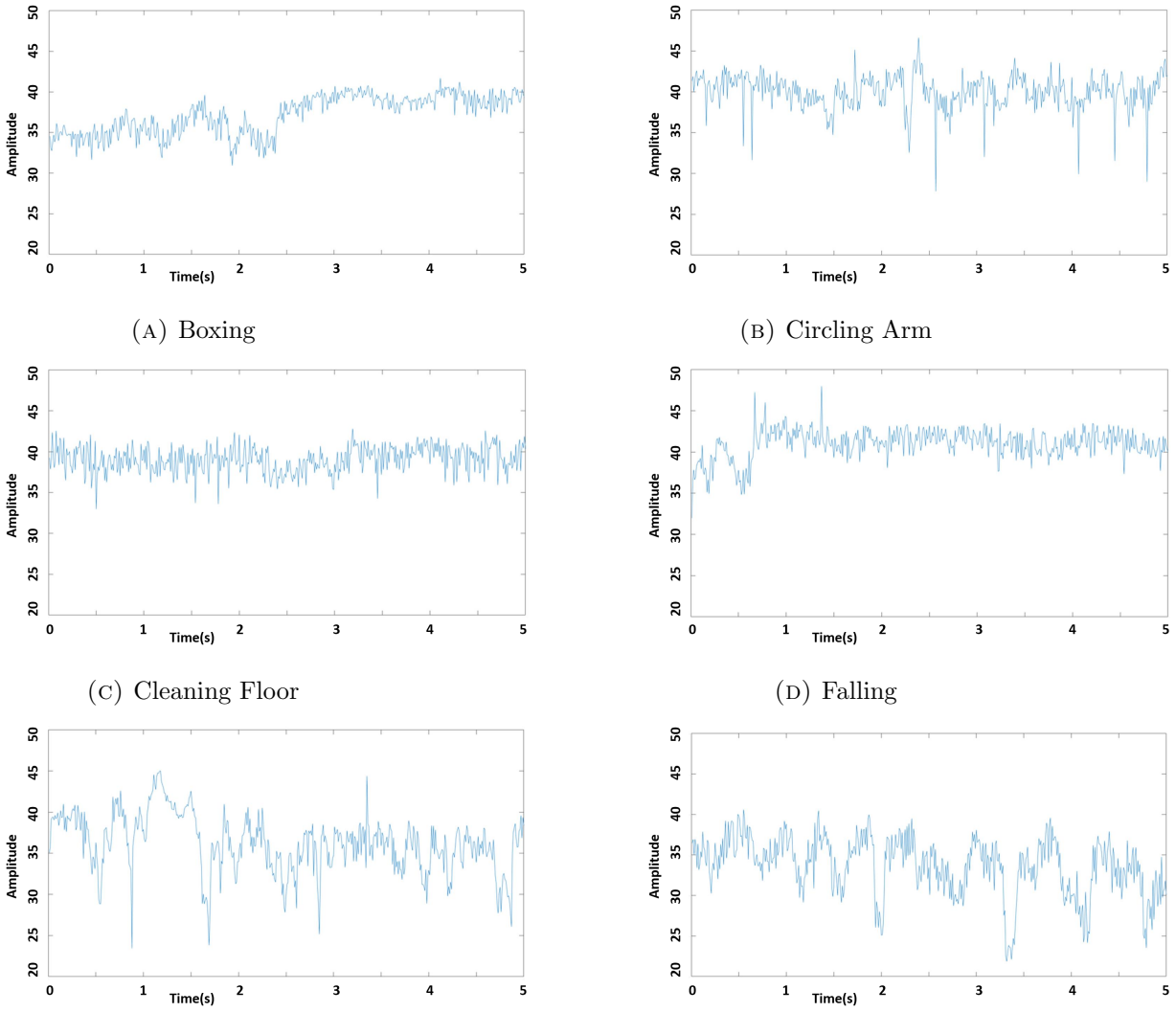


(A) Lab layout



(B) Cubic Layout

FIGURE 3.2: Layouts



(A) Boxing

(B) Circling Arm

(C) Cleaning Floor

(D) Falling

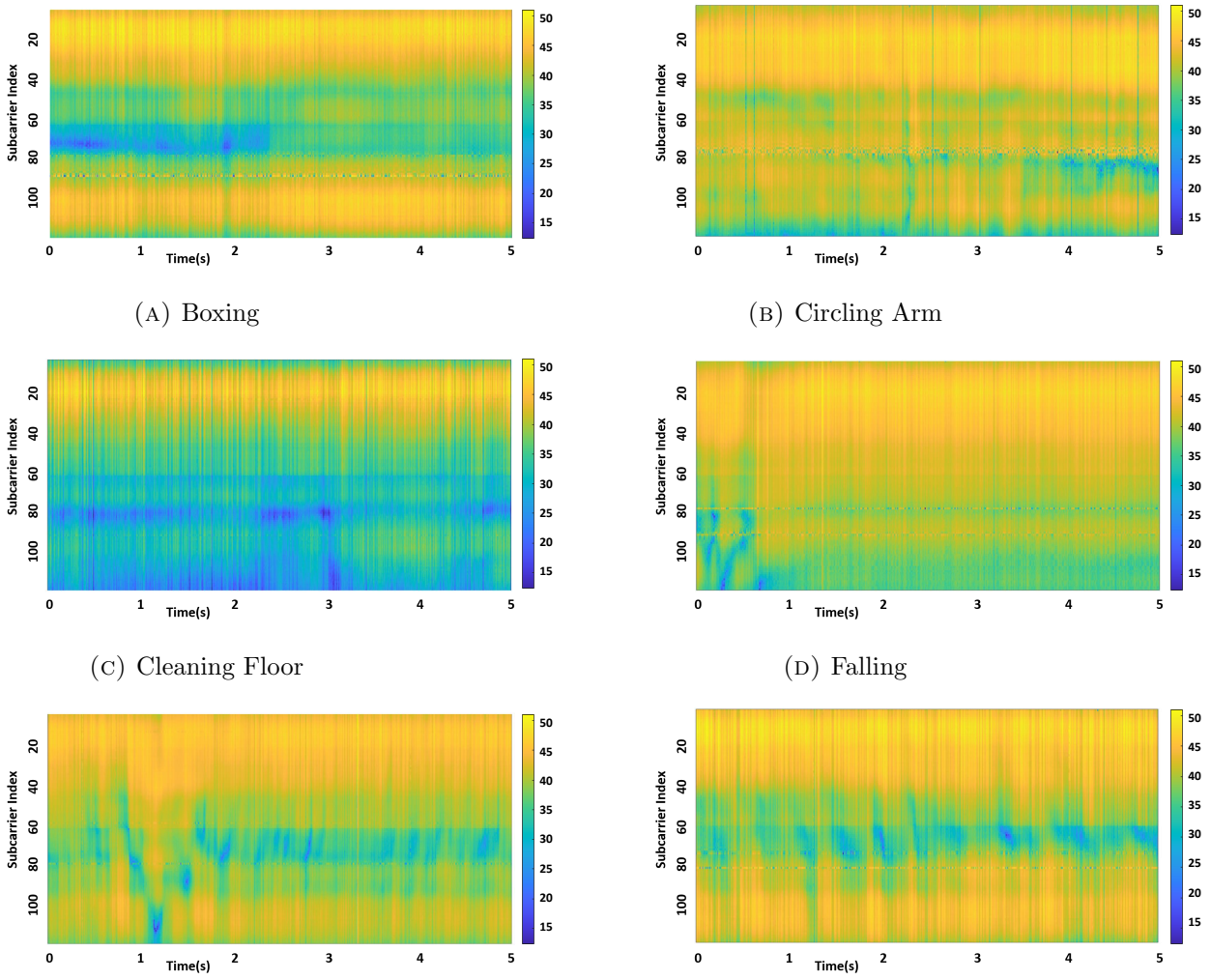
(E) Running

(F) Walking

FIGURE 3.3: CSI plots of 50th subcarrier under different activities

- Environment D (E-D): the interference from both layouts and surroundings are simulated in this setting.

Within one location, we collect 20000 CSI data per activity for each environment setting. The amplitude information is extracted for system training. We validate the hyperparameters w_1, $w_{2,n}$, w_3, β and λ using the validation data, and set their value to be $0.60, 0.10, 0.10, 0.15$ and 0.10 respectively. The validation data is only used for the validation of hyperparameters. They are not used for the system training. Each CSI sample includes 114 subcarriers, so the input size of each training sample is $114 \times 3 \times 500$.



(A) Boxing

(B) Circling Arm

(C) Cleaning Floor

(D) Falling

(E) Running

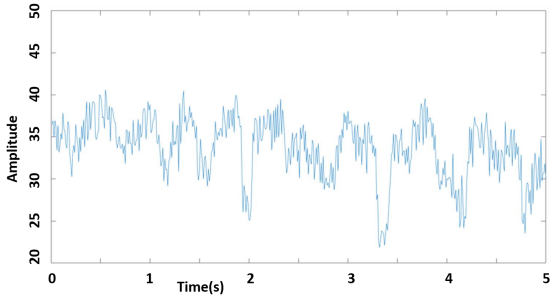
(F) Walking

FIGURE 3.4: CSI data w.r.t 114 subcarriers under different activities

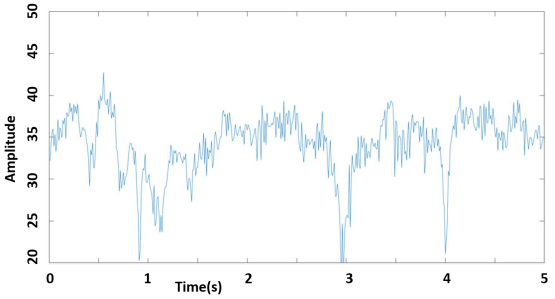
3.4.2 CSI Visualization

CSI data of different activities are visualized in this subsection including both real and fake CSI data from different domains. We plot them in the amplitude versus time form. The horizontal axis is the time scale and the vertical axis shows the amplitude value of each CSI sample.

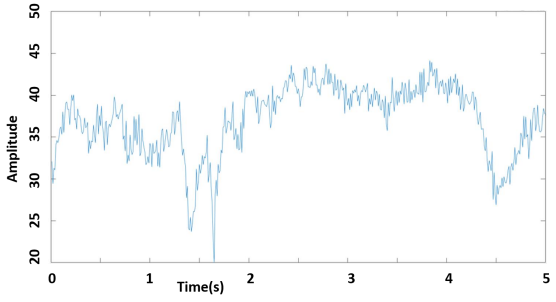
In Fig. 3.3, we show the influences of each activity on the CSI samples' amplitude by one subcarrier in detail using the amplitude variation versus time diagram. In Fig. 3.4, all 114 subcarriers are illustrated with heat diagrams. The amplitude variations of the human activities are shown over all subcarriers for a CSI sample. It is shown that different activities lead to different amplitude variations of the



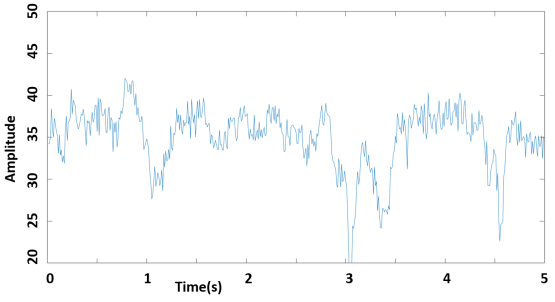
(A) Real CSI data of walking in source environment



(B) Real CSI data of walking in target environment

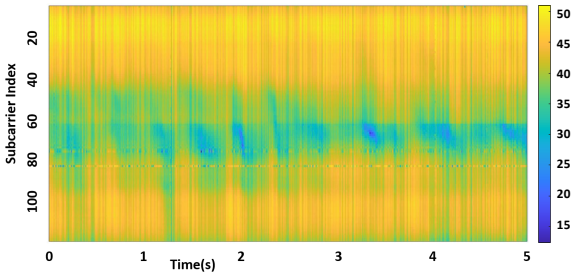


(C) Fake CSI data of walking in target environment

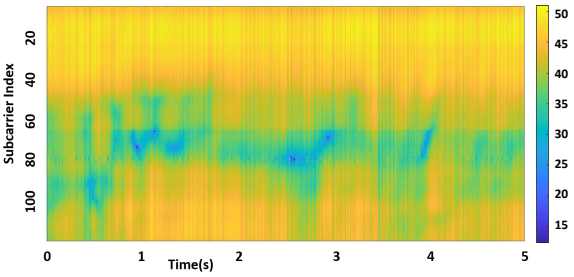


(D) Fake CSI data of walking in target environment

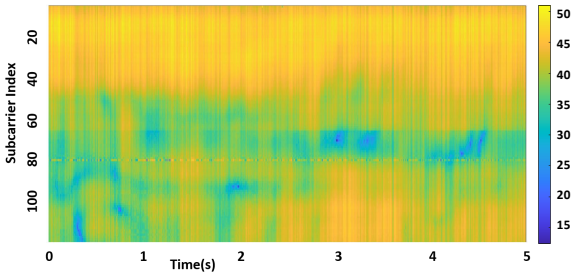
FIGURE 3.5: Real and fake CSI plots of 50th subcarrier of walking



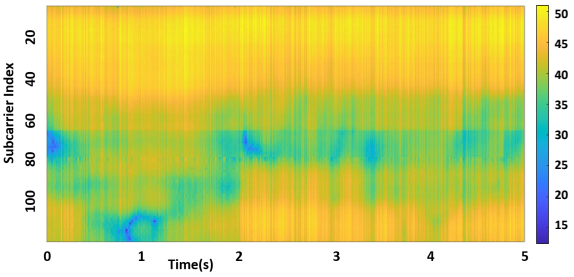
(A) Real CSI data of walking in source environment



(B) Real CSI data of walking in target environment



(C) Fake CSI data of walking in target environment



(D) Fake CSI data of walking in target environment

FIGURE 3.6: Real and fake CSI data w.r.t 114 subcarriers of walking

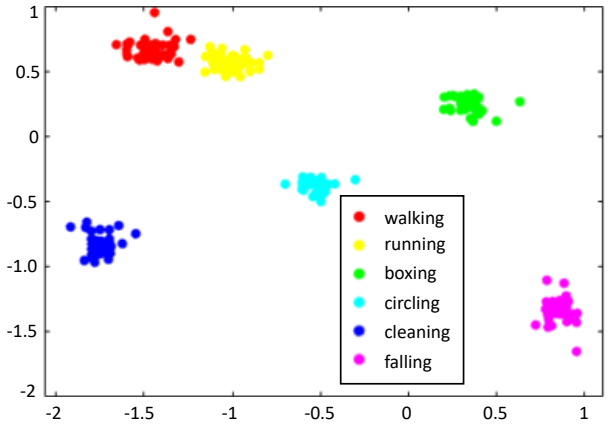


FIGURE 3.7: T-sne plotting of CSI features of six activities

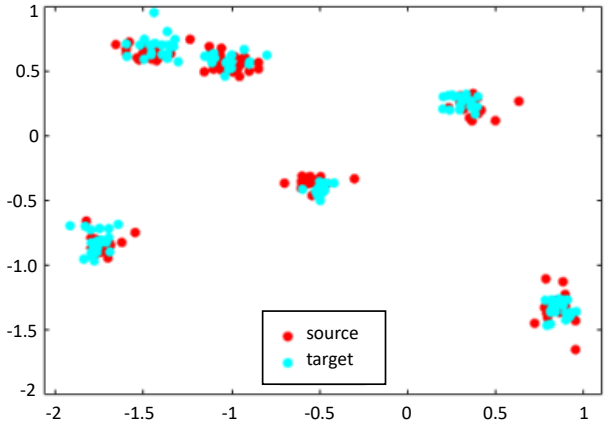


FIGURE 3.8: T-sne plotting of CSI features of source and target domain

received CSI data. With the help of this, CSI-based HAR systems can perform human activity recognition according to different amplitude variation features.

We use the CSI data of walking in Fig. 3.5 and Fig. 3.6 to show how fake CSI data contribute to MCBAR. Real CSI samples in both the source and the target domains are shown. Besides, there are two fake CSI data of walking simulated with two interference matrices using the same real CSI data from the source domain. These visualizations are able to show amplitude features of these CSI samples in detail. We are able to observe that for the same human activity of walking, different variations of amplitude can be observed on real CSI samples from different domains. It causes performance degradation of CSI-based HAR systems. In order to address this problem, we use the translation generator to simulate fake CSI samples. As shown in Fig. 3.5 and Fig. 3.6, the simulated fake CSI data have similar trends of

TABLE 3.1: Overall performances in lab

Environment Index	MCBAR	CNN	CSIGAN	CROSSSENSE	Percentage of unlabelled data
E-B	92.32%	23.21%	81.15%	74.42%	
E-C	92.61%	24.62%	82.21%	78.40%	40% unlabelled data
E-D	91.73%	26.30%	81.42%	76.30%	
E-B	92.71%	23.40%	83.01%	76.10%	
E-C	92.75%	24.10%	84.90%	79.12%	60% unlabelled data
E-D	91.96%	26.47%	82.75%	79.48%	
E-B	92.80%	24.77%	84.42%	78.46%	
E-C	93.91%	28.30%	85.14%	81.47%	80% unlabelled data
E-D	92.44%	22.44%	83.52%	80.01%	
E-B	93.73%	25.32%	87.21%	81.19%	
E-C	94.30%	28.47%	88.72%	82.30%	100% unlabelled data
E-D	93.11%	23.11%	83.91%	82.95%	

TABLE 3.2: Overall performances in cubic office

Environment Index	MCBAR	CNN	CSIGAN	CROSSSENSE	Percentage of unlabelled data
E-B	91.11%	21.49%	82.51%	72.19%	
E-C	93.60%	29.66%	81.46%	75.23%	40% unlabelled data
E-D	90.90%	25.12%	80.49%	74.51%	
E-B	92.10%	23.92%	83.40%	77.46%	
E-C	93.71%	29.90%	82.62%	76.82%	60% unlabelled data
E-D	91.55%	25.41%	83.42%	75.10%	
E-B	92.69%	25.20%	84.16%	78.24%	
E-C	94.21%	31.53%	83.59%	78.90%	80% unlabelled data
E-D	91.92%	27.80%	86.39%	78.40%	
E-B	94.22%	26.31%	86.22%	82.43%	
E-C	94.28%	32.42%	84.23%	79.29%	100% unlabelled data
E-D	92.70%	27.80%	88.22%	80.57%	

TABLE 3.3: Ablation study of MCBAR under different environment settings in lab

Environment Index	MCBAR	MCBAR w/o marginal loss	MCBAR w/o cycle loss	Percentage of unlabelled data
E-B	92.32%	85.41%	90.10%	
E-C	92.61%	86.48%	88.76%	40% unlabelled data
E-D	91.73%	86.11%	90.08%	
E-B	92.71%	86.85%	90.71%	
E-C	92.75%	86.91%	89.92%	60% unlabelled data
E-D	91.96%	86.93%	90.72%	
E-B	92.80%	88.52%	91.15%	
E-C	93.91%	87.76%	90.30%	80% unlabelled data
E-D	92.44%	87.40%	90.91%	
E-B	93.73%	90.30%	91.59%	
E-C	94.30%	89.20%	90.87%	100% unlabelled data
E-D	93.11%	88.15%	91.03%	

amplitude variations. They can be used to approximate the CSI data distribution in the target domain. With different interference matrices used, one CSI sample can be translated from the source domain to the target domain in diverse ways. Diverse translated fake CSI samples still possess similar features to those of real CSI samples from the target domain. The introduced interference does not destroy

TABLE 3.4: Ablation study of MCBAR under different environment settings in cubic office

Environment Index	MCBAR	MCBAR w/o marginal loss	MCBAR w/o cycle loss	Percentage of unlabelled data
E-B	91.11%	84.12%	87.51%	40% unlabelled data
E-C	93.60%	86.92%	88.97%	
E-D	90.90%	85.91%	89.89%	
E-B	92.10%	84.33%	89.64%	60% unlabelled data
E-C	93.71%	87.49%	89.32%	
E-D	91.55%	86.85%	90.57%	
E-B	92.69%	84.92%	90.12%	80% unlabelled data
E-C	94.21%	87.52%	90.51%	
E-D	91.92%	87.17%	90.68%	
E-B	94.22%	85.55%	92.33%	100% unlabelled data
E-C	94.28%	87.98%	90.98%	
E-D	92.70%	89.01%	90.92%	

the similar trend of amplitude variation. It improves the diversity of fake CSI data simulated, which contributes to the systems' robustness.

We visualize the feature clustering of CSI data in different domains in Fig. 3.7 and Fig. 3.8 using T-Distributed Stochastic Neighbor Embedding (T-SNE). T-SNE is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data. It is able to give an intuition of how the data is arranged in a high-dimensional space. We use T-SNE to visualize the CSI features in our experiments. Fig. 3.7 shows the feature clustering of different activities. CSI samples of the same activity are clustered in the same group. Fig. 3.8 illustrates that CSI data from different domains are mapped together. It shows that MCBAR can adapt its system model to different environment settings to avoid performance degradation.

3.4.3 Overall Evaluation

We perform an overall evaluation of MCBAR in this subsection. CrossSense [77], CSIGAN [79] and a purely CNN based CSI classification system [48] are selected for comparison purposes. CrossSense [77], CSIGAN [79] are two state-of-the-art CSI-based HAR systems to address the issue of environmental dynamics.

They also utilize the domain adaption to adapt their models to different environment settings. The CNN network is pre-trained to have over 90% accuracy within the environment setting A. It is used to show the influences of different environment settings on the system without any domain adaption ability. E-A is used

as the source domain. Labelled CSI data from E-A are available. E-B, E-C, and E-D are used as the target domains. Only unlabelled CSI data from these domains are available. In each target domain, different amounts of CSI data are used to perform domain adaption as we also investigate the impacts of different amounts of training data on systems' performances.

The results in lab are shown in Table 3.1 and the results in cubic office are shown in Table 3.2. MCBAR has the highest accuracy of above 90% among the compared systems in all environment settings. Through the performance degradation of the CNN under different environment settings, it can tell that the environmental dynamics affect the systems' performances greatly. MCBAR is able to perform well with small amounts of unlabelled CSI data collected as the boosting generator augments the real CSI data collected from the target domain. It requires fewer samples to perform domain adaption than the other compared systems. The training efficiency of MCBAR is higher than CSIGAN and CrossSense. Both tables show similar results. MCBAR is able to overcome environmental dynamics with no dependency on experimental locations.

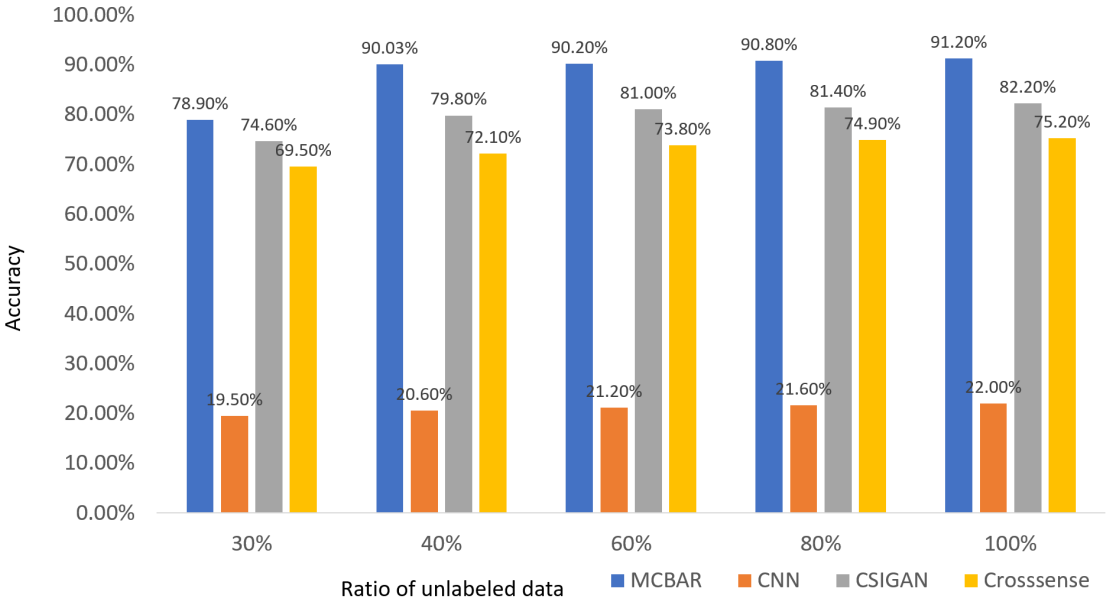


FIGURE 3.9: Accuracy of proposed network under unseen lab environment setting

We also test the performances of these experimental systems in an unseen environmental setting. Therefore, we do not provide any CSI samples from the target domain directly. Instead, we train these systems in other environment settings.

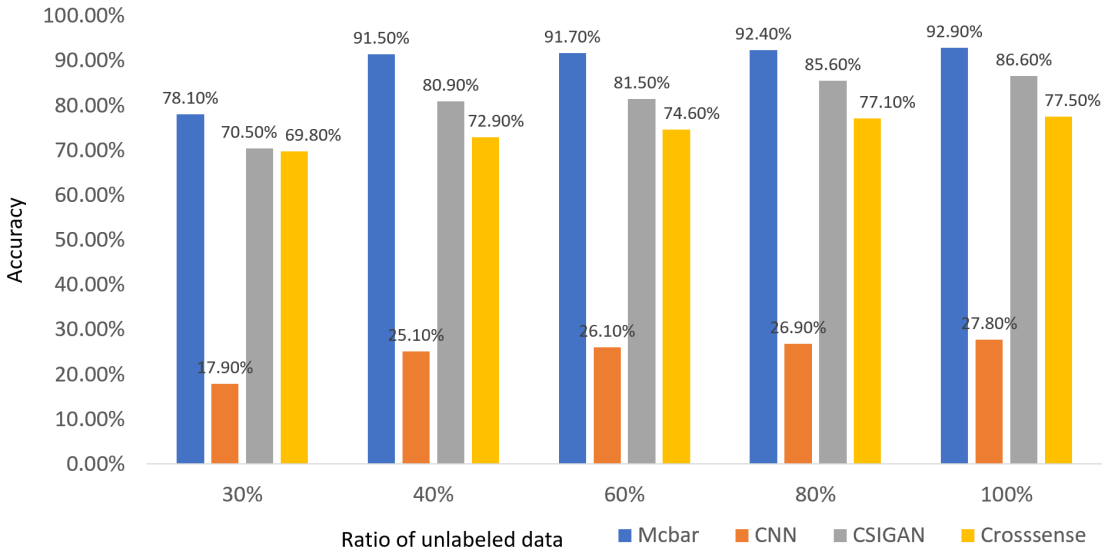


FIGURE 3.10: Accuracy of proposed network under unseen cubic environment setting

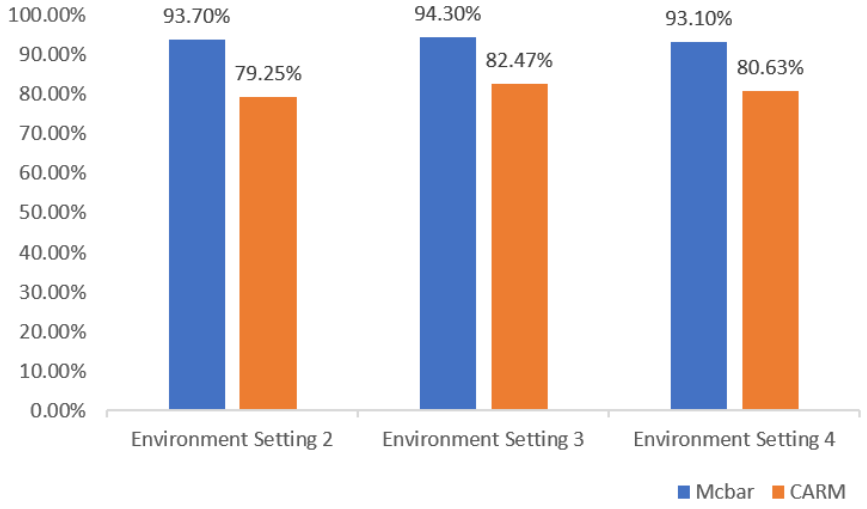


FIGURE 3.11: Accuracy of systems in lab

TABLE 3.5: Systems training time

Number of CSI samples	MCBAR	CARM	Time Saved
250	21.58s	30.42s	29.06%
500	40.29s	51.36s	21.55%
750	59.49s	86.55s	31.27%
1000	85.17s	113.12s	24.71%

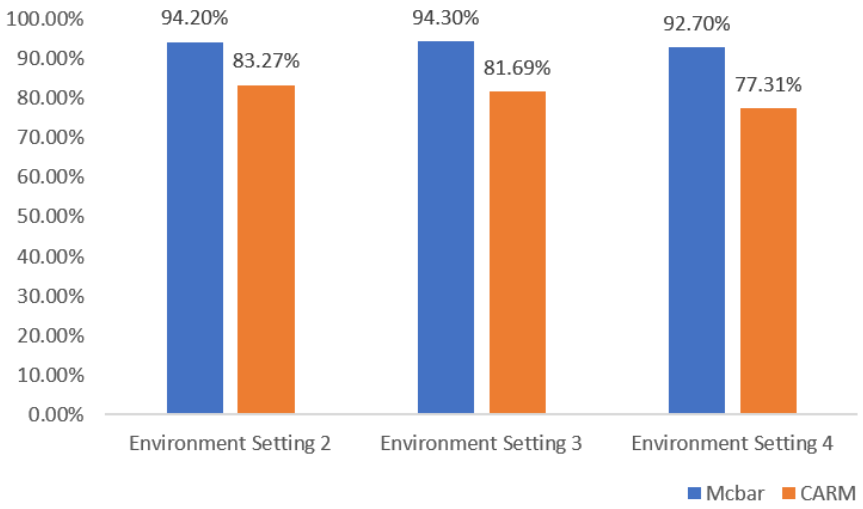


FIGURE 3.12: Accuracy of systems in cubic office

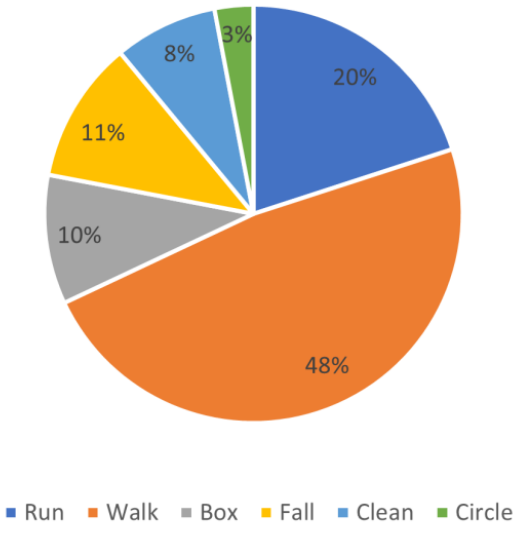
We use the labelled data from E-A and unlabelled data from E-D to train these systems. Then we test them in E-B and E-C, which are unseen new settings. Their performances are shown in Fig. 3.9 and Fig. 3.10.

MCBAR can still maintain its robustness with an accuracy of over 90%. However, the performances of CSIGAN and CrossSense decreases obviously. Without any CSI samples from the target domain, MCBAR can perform activity recognition as the translation generator has provided diverse CSI data information based on some related data domains. This also promotes a faster learning process in some related new settings.

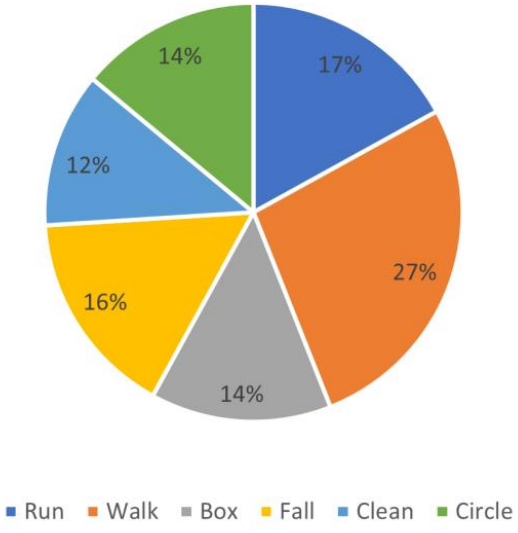
3.4.4 Comparison with Model Based HAR Systems

In this section, we compare our MCBAR with CARM [98] which achieves good performances in HAR using a model-based approach. We test their performances in both locations (lab and cubic office) under different environment settings and their training speeds.

As shown in Fig. 3.11 and Fig. 3.12, provided with same amount of training data, MCBAR performs better than the model-based approach under different environment settings. The generators in our system enhance the training data



(A) Distribution of fake data w/o marginal loss



(B) Distribution of fake data with marginal loss

FIGURE 3.13: Distribution of fake data from the boosting generator

in diversity and provide the classifier with more information for training which improve the system robustness.

We also measure the training time of these two systems with different numbers of training samples. The results are shown in Table 3.5. We also calculate the percentage of training time saved by using MCBAR. Compared to CARM, MCBAR shortens the training time by 21.55% to 31.27% using different amounts of training data, while provides higher accuracy.

3.4.5 Ablation Study

In MCBAR, we address the issue of non-uniformly distribution of target domain CSI data collected with the marginal loss. We also improve the convergence of the translation generator with the cycle loss. We compare their contribution to MCBAR's performances through the ablation study. The testing environment settings are still used as same as the overall evaluation.

The experimental results are shown in Table 3.3 and Table 3.4. Without the marginal loss, the performances of MCBAR decrease more obviously. As the collected CSI data from the target domain can be non-uniformly distributed, the fake CSI data generated will be greatly affected by this. Both training processes of the translation generator and classifier are influenced. The domain adaption has to be performed based on these non-uniformly distributed target domain samples in this case. Cycle loss also contributes to the system by unifying the extracted features between different domains and improves the system's robustness.

Besides, we visualize the distribution of fake CSI data generated from the boosting generator in Fig. 3.13. With the marginal loss, the generated fake CSI data are more balanced among different activities. While the fake CSI data generated without the marginal loss are centralized at those activities which users perform more frequently than others.

3.4.6 Public Benchmark

To demonstrate the robustness of our system, we also use public data SignFi [99] and FallDeFi [100] to test our system. SignFi collects CSI data about frequently

used sign language gestures. CSI measurements are collected in a lab environment. FallDeFi collects CSI data about human activities. Their experiments are conducted in a corridor and kitchen. Rather than our system which can provide 114 subcarrriers for each antenna, they use 802.11n CSI tool [10] which can only provide 30 subcarriers for each antenna. We keep the same size by duplication. We compare our system with CrossSense and CSIGAN. The results are shown in Fig. 3.14 and Fig. 3.15. The accuracies of our system tested under both datasets are the best when compared to other systems, which further demonstrates our system can effectively improve the performance of the CSI-based human activity recognition.

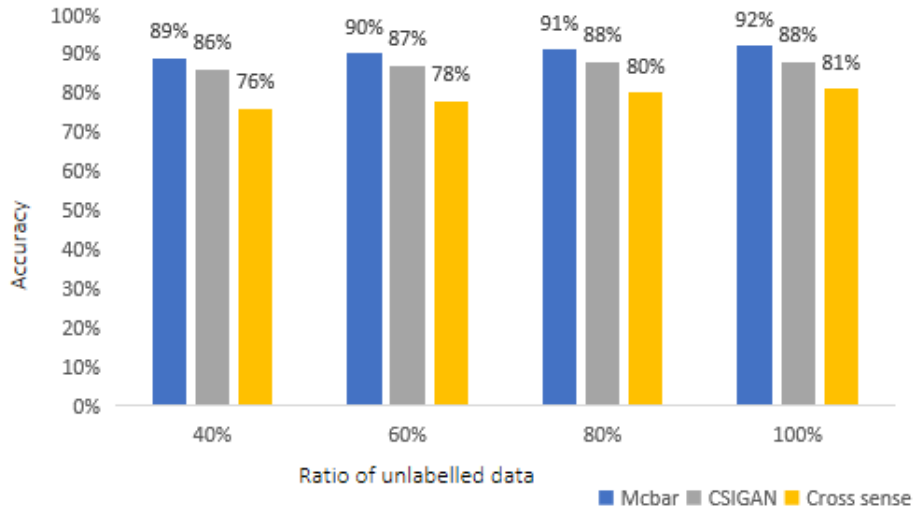


FIGURE 3.14: Overall Test using FallDeFi

3.5 Conclusion

This chapter discusses the issue of adapting a CSI-based HAR system to different environment settings with a limited amount of unlabelled CSI data to overcome environmental dynamics. We propose a novel CSI-based HAR system MCBAR. It is able to perform domain adaption to different new environment settings using fake CSI data simulated by its generators. The experimental results show that MCBAR is able to maintain its robustness under different environment settings and outperforms the state-of-the-art in this field.

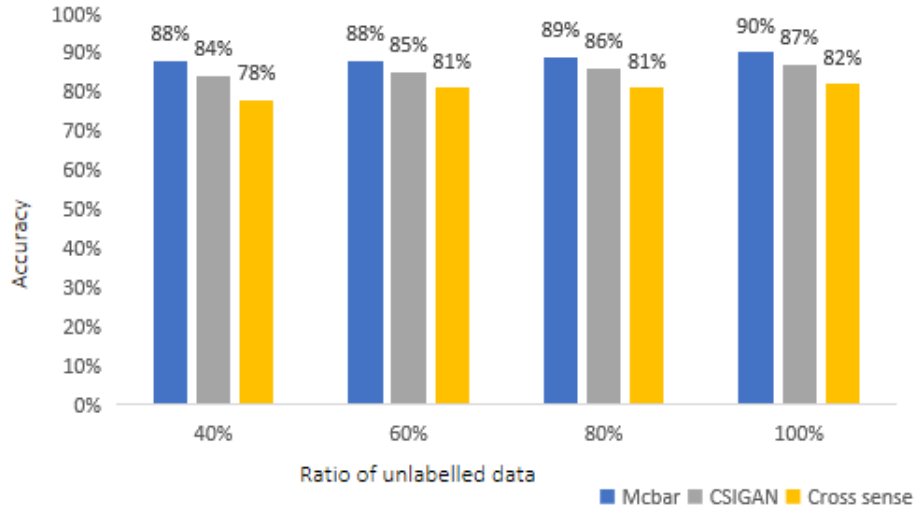


FIGURE 3.15: Overall Test using SignFi

Chapter 4

Empowering WiFi-based Passive Human Gesture Recognition in Unseen Environment via Domain Generalization

4.1 Introduction

In chapter 3, we have studied the problem that CSI-based sensing systems suffer from serious performance degradation under environment dynamics. Using multimodal domain transfer, our proposed system MCBAR is able to transfer the trained system model in the target environment settings and achieve robust performances in the target environment settings [101]. Some other works also take the advantages of DA to transfer their trained system model into different environments [77, 79, 102]. But the DA based CSI human behavior sensing systems require a large number of CSI samples from the new environment to perform domain adaption. Though we managed to solve this issue using simulated fake data, an adequate amount of CSI data from the new environment is still needed to generate fake data [101]. Acquiring CSI data from the target deployed environment settings may not be practical in many situations.

In this chapter, we address the issue that when CSI data from the target environment settings are not available, the CSI-based system is required to have decent performances and be robust once deployed in the target environment settings.

Inspired by the idea of Domain Generalization, we propose a novel Augmented environment-Invariant Robust WiFi gesture recognition system AirFi that aims to solve the performance degradation of a CSI-based smart sensing system in unseen environments. In our scenario, CSI data from the testing environment is not available. AirFi addresses this issue by generalizing its system model to a new environment setting with CSI data from multiple training environmental settings.

Firstly, AirFi uses an encoder to extract feature codes from CSI data collected in several training environment settings. Then with the extracted features mapped on the feature plane, AirFi minimizes the distribution differences between feature codes from different environment settings. Finally, the feature codes are used to train the classifier. In this way, AirFi is able to generalize its model to unseen environment settings. Besides, in order to enhance the system model training, an additional random prior distribution is introduced to the feature extraction process in an adversarial manner. It reduces the dependency between the model and training CSI data. Data augmentation and feature augmentation techniques are also applied to improve the system training. Experiments show that AirFi achieves decent performance and outperforms the benchmarking reference systems.

The contributions of the chapter are summarized as follows:

- We propose a CSI-based gesture recognition system AirFi that can generalize to a new environment without any new data by domain generalization. To the best of our knowledge, the AirFi is the first work that deals with the environment dependency issue without collecting new data or adapting the model in the new environment.
- For better generalization, we augment the CSI data and feature codes to improve their representativity. Unlike previous works which augment CSI data and features randomly, in AirFi the augmentation is designed to be more aggressive on the domain direction while less aggressive on the class-wise direction using a label dependent regularizer.

- In the new environment, by applying the few-shot learning technique the performance of our proposed system AirFi can be further improved with a few CSI data from the testing environment setting.
- Experiments show that our proposed system gives decent performances across different environments. With few-shot learning techniques, the performances are further improved.

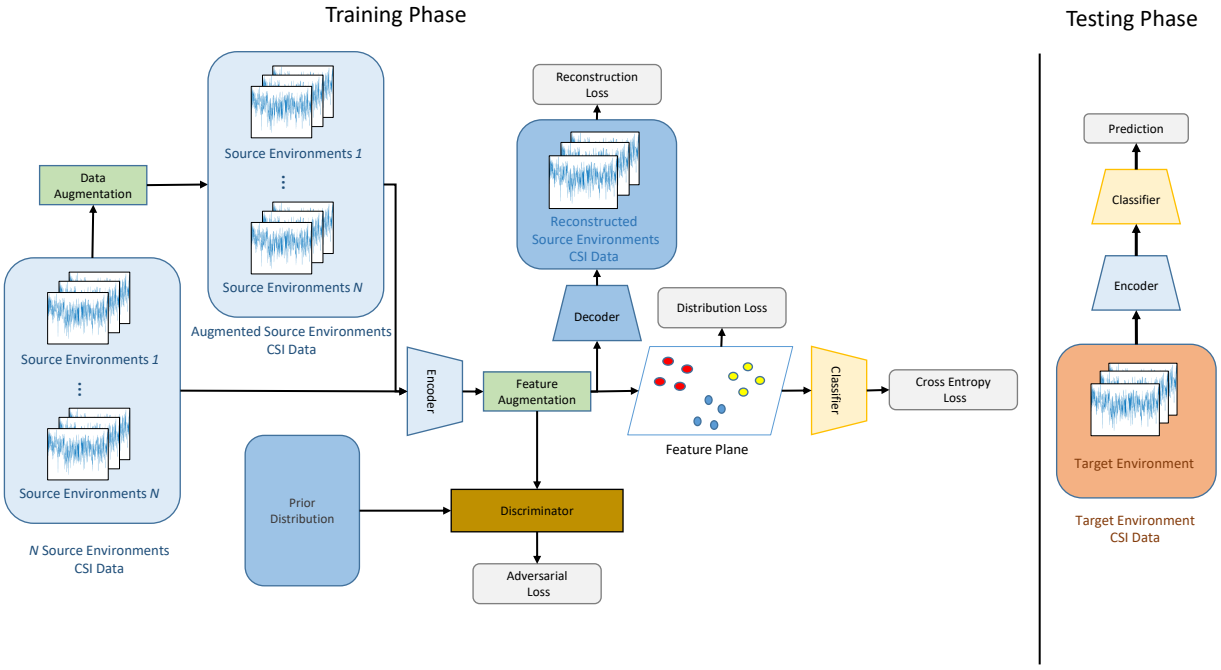


FIGURE 4.1: The structure of proposed AirFi system

4.2 Problem Formulation

In this chapter, we explore the issue of CSI-based gesture recognition. In order to adapt to a new environment, CSI data from the new environment are required for most existing systems [77, 79, 101, 102]. In the real world, it may not be possible to acquire any CSI data from the new environment. We aim to adapt our CSI-based gesture recognition system to an unseen environment and maintain its robustness without any new CSI data from the unseen environment.

Consider that we are able to collect CSI data from N training environments which are referred to as source domains. We collect N sets of CSI data (X^N, Y) from

N training environments. We refer to the new unseen environment as the target domain. We are not able to collect target domain CSI data X_t to adapt the system model, while it is required to perform robust gesture recognition immediately after being deployed in the target domain.

We use the same encoder to extract the feature codes Z^N of CSI data (X^N, Y) collected from N training environments. Then we map the feature codes Z^N onto the feature space. Then we minimize the maximum mean discrepancy (MMD) between N sets of feature codes Z from each environment. By minimizing the MMD, the distribution differences H between feature codes Z from different environments can be reduced. The ultimate goal in this chapter is to build a generalized CSI-based gesture recognition system using CSI data from different environments, and it can be applied to a new unseen environment directly.

4.3 System Overview of AIRFI

We illustrate our system AirFi in Fig 4.1. AirFi is composed of four stages: data augmentation, feature extraction & augmentation, domain generalization and classifier training. In order to generalize the trained model, a basic assumption is that there exists a feature space underlying different domains. In [101], it is proved that there is a common feature space between CSI samples of different human behavior from different environment settings. As shown in Fig 4.1, we collect data from N different environments which are referred to as source environments. Then we augment our collected CSI samples by adding an arbitrary Gaussian noise, which helps to generate more simulated CSI samples. AirFi uses an encoder to encode the collected data and simulated data and extract the down-sampled features. The extracted feature codes are also augmented to improve their diversity and projected onto the hidden codes space for further training. To avoid the issue of overfitting, we take the advantage of Adversarial Autoencoders [103]. We introduce a prior distribution to regularize the distribution of the feature codes using an adversarial training approach. The decoder is used to decode the feature codes back to source environment CSI data, which helps unify the extracted feature codes. The feature codes are mapped onto the feature space underlying all source environments. AirFi minimizes the distribution variance among different training environments based

on the MMD [57]. Finally, a classifier is trained to recognize different human gestures using the feature codes and their corresponding label information. We will present each part of AirFi in detail in the following sections.

4.3.1 Data Augmentation

CSI samples of human gestures are collected from N source environments where CSI data are relatively easier to acquire. These environments are referred to as training environments. To have more representative CSI data collections for a better generalization result, data from more domains should be collected if possible. However, due to limited time and human resources, it is very difficult and expensive to collect CSI data from all different environment settings. Furthermore, each environment is generally dynamic as well. In some previous works [79, 101, 104], Gaussian noise is widely used for data augmentation purpose. In [101], they build a multimodal CSI model for simulated CSI data generation. They found that by adding an arbitrary Gaussian noise to the collected CSI data, they are able to generate fake CSI data. The introduced Gaussian noise will not change the label of the CSI data, moreover, these fake CSI data can be used to approximate the distribution of the related CSI data in other environment settings.

We denote the collected CSI data pairs from N different environments as (X^N, Y) where X is the collection of CSI samples and Y is the collection of gesture labels. To augment the datasets and improve its diversity, an arbitrary Gaussian noise is added to each CSI data sequence. By combining the original collected CSI data sequence and the Gaussian noise, a new simulated CSI data sequence is generated. As explained above, the introduction of Gaussian noise does not change the label of the CSI data. As in wireless signal transmission perspective, the received signal can be modeled as a combination of the transmitted signal multiplying the transmission channel matrix and the Gaussian noise. By combining Gaussian noise with the CSI data, it will not change the class label of the data in terms of gesture recognition purposes [41, 79, 101]. The new generated CSI datasets can help us to approximate the CSI data distribution in other environment settings, which improves the diversity of our CSI datasets and benefits the system training.

4.3.2 Feature Extraction via Adversarial Learning

Given the augmented CSI data, an encoder is used for feature extraction. For CSI data (X^N, Y) from N source environments, AirFi uses the same encoder Q to extract feature codes Z from them. The encoder is an adversarial autoencoder. When the input CSI data pass through each convolutional layer in the encoder, they are downsampled and feature codes are extracted.

As AirFi utilizes CSI data from all source environments to train its model, it may cause the overfitting issue during the training phase. The model trained may follow too closely to the given training data and has a strong dependency on the source environment CSI datasets. This will harm model generalization. Unless this issue is addressed, the training process of AirFi would be like a supervised learning with all source domains. It is important that the model is trained using all source domain data, meanwhile it does not have a strong dependency on the training data. Only in this way can the model learn the common feature space of CSI data from different environments and be generalized to other unseen environments. To address this issue, a prior distribution H is imposed as an additional domain besides the data source environment domains. Whenever the feature codes are extracted, a regularization code is also generated from the prior distribution. Both of them are sent to the discriminator D that is used to distinguish between the feature code and regularization code. This process is similar to the generative adversarial network. The adversarial loss L_{ad} is given by

$$L_{ad} = E_{h\sim p(h)}[\log D(h)] + E_{x\sim p(x)}[\log(1 - D(Q(x)))]. \tag{4.1}$$

By minimizing the adversarial loss, we impose the dependency of the feature codes extraction on the prior distribution. This can reduce the dependency of the system model on the training CSI data. With the introduction of the prior distribution, we expect that the issue of overfitting to the source domains data can be addressed. In theory, the prior distribution can be any arbitrary distribution [57]. It is introduced to enable the adversarial encoder to extract feature code with less dependency on original CSI data distribution. Therefore the trained model can generalize better to the testing environment. We use the Laplace distribution as the prior distritbution in AirFi. We compare between several popular distribution used in related works [57] and [35], such as Laplace distribution, Gaussian distribution and Uniform

distribution, and the Laplace distribution performs the best. Besides, a decoder P is also applied to decode the feature code back to the source domain CSI data form. The reconstruction loss L_{re} is given by

$$L_{re} = \sum_{n=1}^{N} \|Q(z)_n - X_n\|_2^2 \tag{4.2}$$

The reconstruction process can unify the content of feature codes encoded from different training environments and improve the level of generalization to other unseen environments.

4.3.3 Label Dependent Feature Augmentation

To further enhance the generalization ability of AirFi, we augment the feature codes. As shown in [105], besides data augmentation, feature augmentation is also able to improve the model generalization by improving the diversity of feature codes. Given the collection of feature code Z, which is extracted from the input CSI data X using the encoder Q. We have

$$z = Q(x). \tag{4.3}$$

Then we input the feature codes into the augmentation layers A. In [105], the feature codes are augmented by scaling and adding bias in their networks. The scaling changes the absolute difference between elements in the feature codes, while the bias changes the absolute mean value of the feature codes. In AirFi, after the feature extractor, the sampled CSI feature codes z are multiplied and added with random variables sampled from normal distributions. The collection of augmented feature codes is denoted as Z' given by

$$z' = A(z) = \alpha * z + \beta, \tag{4.4}$$

where α and β are the scale and bias hyperparameters, and sampled from two Gaussian distributions,

$$\alpha \sim N(1, \sigma_1), \tag{4.5}$$

$$\beta \sim N(0, \sigma_2), \tag{4.6}$$

where σ_1 and σ_2 are two scalar hyperparameters. We set $\sigma_1 = \sigma_2$ to reduce the number of hyperparameters. The perturbation introduced improves the diversity of feature codes and benefits the model generalization. However, one of the limitations brought by the feature augmentation is that the perturbation caused by the random noise may not follow the class-preserving direction. The augmented feature codes may lose some properties of their own behavior classes and are embedded with some new properties of other behavior classes. This will affect the model training and performances. In order to augment the feature codes to improve their diversity meanwhile preserving those feature properties of their own behavior classes. We add a label dependent regularizer ϵ to the augmentation layer in AirFi. The augmentation process becomes

$$z' = A(z) = \alpha * z + \beta + \epsilon. \tag{4.7}$$

The regularizer ϵ is sampled from a class-wise normal distribution $N(0, \Sigma_c)$, $c \in [1, C]$, where c is the gesture class index, C is the total number of gesture classes and Σ_c is the class-wise covariance. Σ_c is estimated and updated from every mini-batch of training data in a moving average manner

$$\Sigma_c = \lambda * \Sigma_c + (1 - \lambda) * Cov(z'|y = c), \tag{4.8}$$

where λ is the discount factor. The corresponding Σ_c is only updated when the label y of CSI data belongs to its own class c. During the training phase, AirFi calculates the class-wise covariances of data from each class and update the Σ_c of each class. Then the elements of ϵ are sampled as,

$$\epsilon_c \sim N(0, \Sigma_c). \tag{4.9}$$

With the additional regularizer, the feature codes are augmented more aggressively along the cross domain direction instead of the class-wise direction. Though the feature code is augmented with random variables, it is added with the covariance ϵ_c of its own gesture class c to preserve key properties of that particular class. As

a result, the augmented feature codes are similar to those original feature codes of their own classes and have some perturbation introduced by the random variables α and β. This improves the diversity of feature codes from different gesture classes and leads to better performances of AirFi. We perform an ablation study to test it in our experiments.

4.3.4 Domain Generalization

The feature codes are mapped onto the feature space for domain generalization. Denote the feature codes z from n_{th} source environment as Z_n with the distribution P_n of CSI data. To perform the mapping, a mean map operation $\mu(\cdot)$ is required to map the feature codes to a reproducing kernel Hilbert space [106], which is given as

$$\mu_P = E_{z\sim P}[k(z)], \tag{4.10}$$

where k is the kernel function. For AirFi, we use the Radius Bias Function (RBF) kernel, which is a well-known and commonly used characteristic kernel [57].

To achieve domain generalization of the system model, the mapped feature codes from different domains are supposed to be clustered together on the feature space. AirFi fulfills this purpose by minimizing the MMD between different distributions. The MMD between feature codes from two source environments can be measured by

$$MMD(Z_i, Z_j) = \|\mu_{P_i} - \mu_{P_j}\|. \tag{4.11}$$

By extending it from two source environments to multiple environments, the distribution variances between different feature domains is calculated as

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}\|\mu_{P_i}-\mu_P\| &= \frac{1}{N}\sum_{i=1}^{N}\|\mu_{P_i}-\frac{1}{N}\sum_{j=1}^{N}\mu_{P_j}\| \\ &= \frac{1}{N}\sum_{i=1}^{N}\|\sum_{j=1}^{N}\frac{1}{N}(\mu_{P_i}-\mu_{P_j}) \\ &\leqslant \frac{1}{N^2}\sum_{1\leqslant i,j\leqslant N} MMD(Z_i,Z_j),\end{aligned} \tag{4.12}$$

where μ_P is the mean distribution for all training environments. The indices of any two source environments are denoted by i and j. As shown in the equation, the distribution variances are upper bounded. Therefore we use the distribution regularization loss L_{MMD}

$$L_{MMD}(Z_1,...,Z_n) = \frac{1}{N^2}\sum_{1\leqslant i,j\leqslant N} MMD(Z_i,Z_j). \tag{4.13}$$

By minimizing L_{MMD}, the distribution variances between each source domain are also reduced. As a result, the model trained can be generalized between different domains.

4.3.5 Classifier Optimization

With the features identified, a classifier is added at the end of AirFi. The classifier consists of three fully connected layers F. With the generalized feature codes on the feature space and corresponding gesture labels, AirFi trains its classifier in a supervised learning manner. AirFi uses the cross-entropy loss to measure classification errors L_{ce}:

$$L_{ce} = -E_{z,y}\log[p(y|z)]. \tag{4.14}$$

4.3.6 Few Shot Learning

In our study, we find that though AirFi can achieve a decent performance without any training CSI samples from the target environments, a few CSI samples from the target environment can indeed help to further improve its performances. It is

possible that the feature codes encoded from testing environment CSI data may not be mapped closely to the distribution of source environment feature codes. To address this issue, a few labeled CSI samples from the testing environment can be very helpful. After AirFi is trained using the source domain data from the training environment, we input the testing environment CSI data and minimize the distribution difference between the source environment and testing environment CSI data with the new distribution regularizer loss L'_{MMD}.

$$L'_{MMD} = MMD(Z_{training}, Z_{testing}) \tag{4.15}$$

We retrain the system with the source environment, target environment distribution loss and classification cross-entropy loss L'_{ce} which also includes the testing environment labeled data X_t.

$$L'_{ce} = -E_{x_t,y_t} \log[p(y_t|x_t)]. \tag{4.16}$$

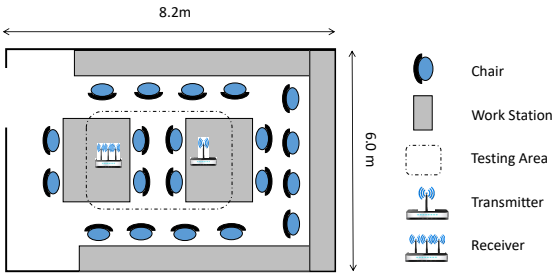
Then the total few shot learning loss L_f is given by

$$L_f = L'_{MMD} + L'_{ce} \tag{4.17}$$

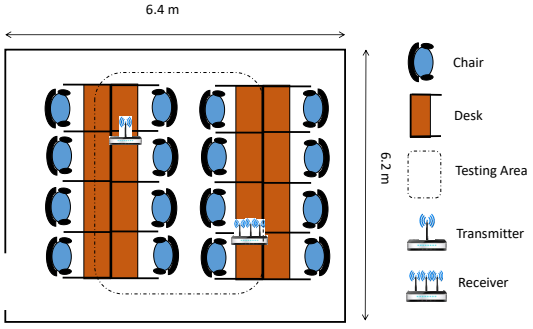
After the few-shot learning is added, the trained model has a better generalization ability on the target environment. We show the improvement in the following section.

4.4 Experiments

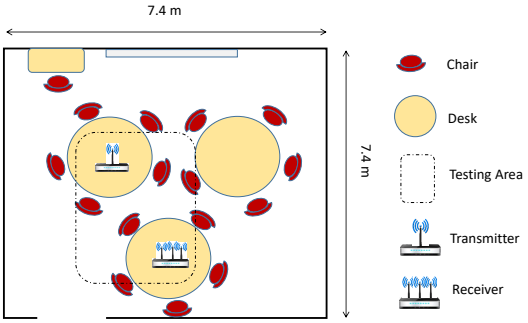
To be applied in a different environment, AirFi does not require any CSI data from the target environments as in the case for most existing systems. Using CSI data from several training environments, AirFi aims to build a generalized system model. In this section, we conduct multiple experiments to evaluate AirFi under different environments. Firstly, we introduce the experimental setup. Then we do an overall evaluation to compare AirFi with other CSI-based smart human sensing systems. Thirdly, we do an ablation study to investigate the impacts of different



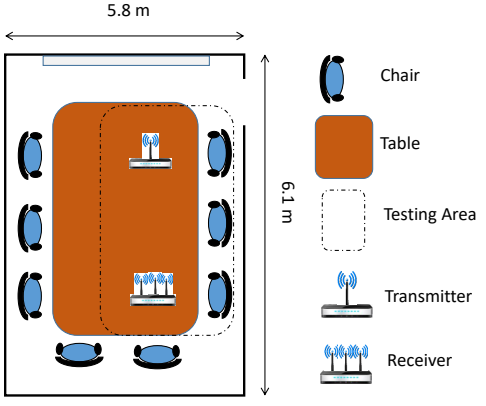
(A) Lab Layout



(B) Cubic Office Layout



(C) Tutorial Room Layout



(D) Meeting Room Layout

FIGURE 4.2: Layouts of experimental environments

TABLE 4.1: Overall Performances Evaluation

Sytems	Up&Down	Left&Right	Front&Back	Clap	Fist	Waving	Throw	Zoom	Environment Index
CNN Only [101]	31.23%	29.57%	36.55%	32.48%	29.63%	34.11%	30.86%	28.99%	
WiGr [102]	80.67%	81.44%	79.38%	82.04%	80.78%	78.63%	79.81%	80.52%	
WGRDTL [107]	70.85%	73.93%	73.47%	74.01%	72.14%	70.65%	71.46%	69.19%	ABC-D
Wi-Multi [108]	72.15%	71.89%	69.94%	71.28%	70.58%	72.23%	70.18%	70.63%	
AirFi	**89.32%**	**91.02%**	**89.47%**	**91.66%**	**90.85%**	**88.79%**	**92.37%**	**89.61%**	
CNN Only [101]	35.95%	39.11%	37.49%	46.24%	40.18%	41.79%	39.85%	40.43%	
WiGr [102]	76.39%	79.20%	80.71%	78.58%	81.84%	79.98%	79.34%	80.86%	
WGRDTL [107]	73.97%	73.55%	70.11%	73.16%	71.74%	70.06%	70.21%	72.63%	ABD-C
Wi-Multi [108]	74.14%	71.08%	70.83%	70.92%	69.71%	70.57%	69.24%	73.26%	
AirFi	**87.48%**	**91.25%**	**91.28%**	**90.75%**	**92.02%**	**90.79%**	**89.31%**	**90.63%**	
CNN Only [101]	31.01%	40.42%	34.15%	32.68%	36.76%	37.91%	35.31%	37.82%	
WiGr [102]	79.38%	82.57%	83.61%	79.99%	82.52%	81.64%	77.24%	80.36%	
WGRDTL [107]	71.13%	74.10%	76.97%	74.45%	75.23%	72.28%	70.34%	70.46%	ACD-B
Wi-Multi [108]	73.69%	71.25%	70.98%	72.89%	74.35%	72.47%	74.84%	71.06%	
AirFi	**88.16%**	**89.72%**	**92.78%**	**91.06%**	**90.34%**	**89.47%**	**88.32%**	**90.30%**	
CNN Only [101]	38.44%	37.98%	32.17%	30.19%	31.74%	32.68%	32.48%	36.37%	
WiGr [102]	76.52%	79.63%	80.46%	81.09%	76.87%	79.68%	80.50%	80.72%	
WGRDTL [107]	72.54%	73.71%	70.38%	72.02%	73.44%	72.18%	71.51%	72.43%	BCD-A
Wi-Multi [108]	71.84%	72.93%	74.42%	77.59%	73.28%	73.80%	72.17%	74.67%	
AirFi	**91.73%**	**90.86%**	**87.26%**	**88.67%**	**89.42 %**	**89.76%**	**90.92%**	**87.58%**	

TABLE 4.2: Ablation Study

Sytems	Up&Down	Left&Right	Front&Back	Clap	Fist	Waving	Throw	Zoom	Environment Index
CNN Only	31.23%	29.57%	36.55%	32.48%	29.63%	34.11%	30.86%	28.99%	
AirFi w/o Data Aug	85.15%	87.06%	83.47%	86.72%	85.96%	87.17%	85.63%	86.31%	ABC-D
AirFi w/o Fea Aug	84.34%	85.65%	84.98%	86.10%	83.57%	86.21%	85.67%	82.44%	
AirFi	**89.32%**	**91.02%**	**89.47%**	**91.66%**	**90.85%**	**88.79%**	**92.37%**	**89.61%**	
CNN Only	31.23%	29.57%	36.55%	32.48%	29.63%	34.11%	30.86%	28.99%	
AirFi w/o Data Aug	84.83%	86.94%	87.75%	85.28%	84.67%	85.13%	83.71%	86.14%	ABD-C
AirFi w/o Fea Aug	82.68%	85.14%	84.36%	86.11%	84.09%	83.76%	82.97%	84.59%	
AirFi	**87.48%**	**91.25%**	**91.28%**	**90.75%**	**92.02%**	**90.79%**	**89.31%**	**90.63%**	
CNN Only	31.23%	29.57%	36.55%	32.48%	29.63%	34.11%	30.86%	28.99%	
AirFi w/o Data Aug	82.87%	83.94%	84.72%	83.16%	84.71%	85.93%	83.66%	84.68%	ACD-B
AirFi w/o Fea Aug	83.91%	82.70%	83.05%	84.61%	83.14%	83.45%	83.17%	82.08%	
AirFi	**88.16%**	**89.72%**	**92.78%**	**91.06%**	**90.34%**	**89.47%**	**88.32%**	**90.30%**	
CNN Only	31.23%	29.57%	36.55%	32.48%	29.63%	34.11%	30.86%	28.99%	
AirFi w/o Data Aug	83.95%	82.27%	84.78%	85.49%	82.29%	84.77%	84.52%	83.96%	BCD-A
AirFi w/o Fea Aug	83.01%	82.94%	84.95%	82.39%	81.79%	83.28%	83.26%	84.73%	
AirFi	**91.73%**	**90.86%**	**87.26%**	**88.67%**	**89.42%**	**89.76%**	**90.92%**	**87.58%**	

components in AirFi. Besides, we test AirFi with few-shot learning added on and observe how it improves the performances. Finally, we use the T-SNE plots to show the distribution of hidden features with different system designs.

TABLE 4.3: Few-Shot learning test in lab

Sytems	Up&Down	Left&Right	Front&Back	Clap	Fist	Waving	Throw	Zoom	Environment Index
CNN Only [101]	39.57%	41.08%	47.11%	34.95%	43.89%	41.97%	43.44%	37.96%	
WiGr [102]	86.94%	87.28%	87.43%	87.62%	86.15%	84.60%	89.37%	90.07%	
WGRDTL [107]	80.19%	78.88%	76.58%	79.78%	80.67%	81.85%	79.54%	78.27%	ABC-D
Wi-Multi [108]	78.95%	79.52%	77.91%	81.49%	78.18%	80.56%	80.02%	78.49%	
AirFi	**94.63%**	**95.21%**	**93.55%**	**93.17%**	**91.68%**	**96.14%**	**95.24%**	**93.54%**	
CNN Only [101]	45.93%	41.08%	35.12%	40.58%	38.94%	44.421%	37.01%	43.18%	
WiGr [102]	85.99%	87.19%	86.38%	86.92%	87.37%	88.96%	88.00%	87.25%	
WGRDTL [107]	79.63%	81.71%	78.45%	78.93%	79.48%	80.16%	81.04%	71.67%	ABD-C
Wi-Multi [108]	78.59%	78.61%	79.38%	79.11%	81.07%	77.86%	80.19%	82.93%	
AirFi	**93.89%**	**93.61%**	**94.29%**	**93.28%**	**93.47%**	**93.68%**	**94.78%**	**94.13%**	
CNN Only [101]	48.15%	34.18%	42.69%	39.48%	41.56%	42.00%	38.14%	40.82%	
WiGr [102]	84.60%	88.47%	89.15%	90.81%	89.39%	90.82%	89.49%	87.30%	
WGRDTL [107]	81.98%	82.19%	79.58%	77.14%	82.84%	80.14%	78.92%	83.86%	ACD-B
Wi-Multi [108]	78.49%	79.18%	77.39%	74.81%	80.18%	81.44%	78.68%	80.49%	
AirFi	**92.69%**	**92.11%**	**94.27%**	**95.34%**	**93.74%**	**95.08%**	**94.18%**	**94.57%**	
CNN Only [101]	41.33%	47.86%	42.17%	39.68%	39.42%	43.71%	43.62%	44.76%	
WiGr [102]	86.07%	86.32%	88.95%	87.60%	87.23%	90.27%	89.08%	88.81%	
WGRDTL [107]	79.31%	82.18%	83.86%	79.41%	80.09%	79.47%	78.86%	81.10%	BCD-A
Wi-Multi [108]	75.89%	77.21%	82.12%	82.63%	80.76%	81.97%	79.55%	81.83%	
AirFi	**94.58%**	**93.89%**	**94.79%**	**92.88%**	**93.75%**	**95.86%**	**95.33%**	**94.17%**	

4.4.1 Environment Setup and Data Collection

AirFi is designed to be generalizable to different environment settings. In order to test its ability, our experiments are performed in four different environment settings: lab, cubic office, meeting room and tutorial room. We select them as their layouts and furniture are very different from each other, which can be used to test the performances of compared systems in different environments. Their layouts are shown in Fig 4.2. In each location, two routers are used. One router is the transmitter (one antenna), and the other router is the receiver (three antennas). We have upgraded the firmware of both routers to our CSI enabled platform for data collection. The transmitter is operated in 802.11n AP mode at 5 GHz with a 40 MHz bandwidth and the receiver is connected to the transmitter in client mode. The detail of the environments is as follows.

- Environment A (lab environment). The furniture in the lab is mainly lab benches and chairs. The routers are placed on two opposite lab benches. The volunteer performs different human gestures, while sitting in the middle between the two lab benches.

- Environment B (cubic office environment). The furniture in the cubic office is mainly cubical desks and chairs. The two routers are placed on two different desks as shown in the layout figure. The volunteer performs different gestures in the middle area.
- Environment C (tutorial room environment). The furniture is mainly round table and chairs. There is also one big screen on the wall. The two routers are placed on two tables. The volunteer performs different gestures in the testing area.
- Environment D (meeting room environment). There is a big table in the center of the room. Chairs are put around the table. We place two routers at one side of the table, and the volunteer performs different gestures beside the table.

We have selected 8 volunteers aging from 19 to 27 to participate in our experiments. 5 of them are males and 3 of them are females. Our experiments involve 8 categories of human gestures including up & down, left & right, back & forward, clap, fist, circling, throw, zoom. Each volunteer performs different gestures while the transmitter sends signal packets to the receiver. The CSI enabled platform [97] measures and stores the CSI data.

For each gesture, 200 CSI samples are recorded at each experimental location. In total, each gesture has 800 CSI samples collected. AirFi only uses their amplitude information for gestures recognition. There are 114 subcarriers of each CSI sample received by our router platform during the collection. The input size of our CSI data is $114 \times 3 \times 500$.

4.4.2 Overall Evaluation

We compare the system AirFi with state-of-the-art CSI-based gesture recognition systems. For the compared systems, we select WiGr, WGRDTL and Wi-Multi [102, 107, 108]. The compared systems are selected as they also have the ability to adapt to the new environment by taking the advantage of domain transfer. Besides, we remove the generator components of MCBAR [101] and only keep the feature extractor and classifier components which are basically convolutional neural

networks (CNN). The remaining CNN is able to perform accurate human behavior recognition within one environment. Actually the CNN structure is widely used for classification purpose in many CSI-based human sensing systems. However, it does not have fitting ability to adapt into an unseen environment. We train the CNN to have over 90% accuracy in the training environment, then we deploy it in the testing environment together with other compared systems. To test the ability of each method in adapting to a new environment, we use the CSI data from three environments as the source environment data and the CSI data from the left environment as the testing environment data. For example, when environment A, B, C are used as training environment, D will be used as the testing environment, and denote the setting as ABC-D. All four different combinations are tested in the experiment. The testing environment data are not available for any systems including AirFi during the training phase.

The experiments results are shown in Table 4.1. As shown in the table, AirFi outperforms all other compared systems in all testing environments with an accuracy of around 90%. The obvious performance degradation of CNN demonstrates that there are large variations of collected CSI data from different environment settings. Though the compared systems are also designed to adapt to the a environment, they need a large number of testing environment data to perform the domain adaption. As in our scenario, all systems are not able to acquire the testing domain data, their models are not able to be functional as they should be. WiGr has the second best performance with an overall accuracy of over 80%. In order for it to adapt to a new environment better, it requires CSI data from the target environment. However, this is not provided in the experiment. AirFi takes the advantage of domain generalization. It manages to extract the common feature codes from these source environment data and generalize them on the feature space by minimizing the feature codes distribution differences. The training of AirFi does not need any CSI data from the testing environment.

4.4.3 Ablation Study

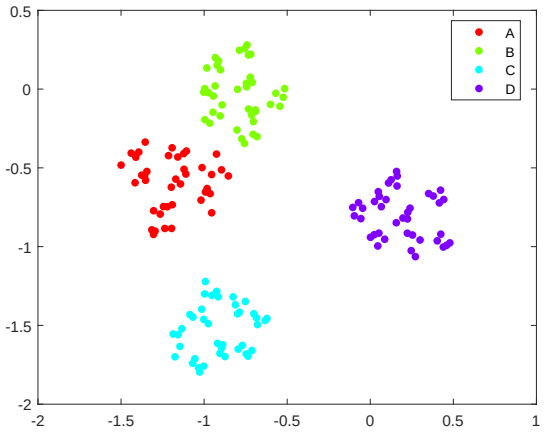
In order to equip the system AirFi with the ability to generalize to new environment settings, we take the advantage of domain generalization. We also augment our CSI datasets and feature codes to improve the performance. We use an ablation

test to study how each component contributes to the system AirFi. We compare the performances between the pure CNN, AirFi without data augmentation, AirFi without feature augmentation and complete AirFi. The experiments setups are the same as the previous overall evaluation. In each test, three of them are used as the training environments and the remaining one as the testing environment.

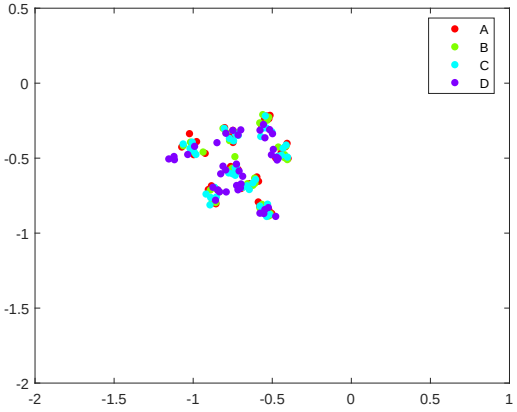
The testing results are shown in Table 4.2. CNN performs worst among the compared systems, while AirFi is still able to generalize its system model which is the most important ability of AirFi. To generalize the system model of AirFi, it needs the CSI data from different training environments. Both CSI data and feature augmentation are used to further enhance the generalization ability. Without these two techniques, the remaining parts of AirFi can still generalize its model and outperform the CNN which does not have any ability to adapt to a new environment. For AirFi without either data augmentation or feature augmentation, their accuracies all get worse compared to the complete AirFi. In other words, both augmentation techniques contribute to the performances of AirFi. It is observed that after removing either data augmentation or feature augmentation, the performance degradation of the two compared systems are very close to each other, which is about 4% to 6%. The missing of feature augmentation affects AirFi slightly more than data augmentation. In fact, both augmentation techniques make the feature codes of training to be more representative. With diverse feature codes, AirFi is able to generalize better on the feature space. The data augmentation improves the diversity of CSI training datasets. It generates more simulated CSI data so that more feature codes can be extracted from these generated CSI data. On the other hand, the feature augmentation works directly on the feature codes. It makes the feature codes more representative. With the help of these two augmentation techniques, the feature codes generalized on feature space are more diverse and AirFi has a higher possibility to generalize to a new environment setting.

4.4.4 Few Shot Learning Adds On

In the previous evaluations, CSI data from the testing domain environments are totally not available during the training phase. We also explore the situation that only a few CSI data from the testing domain environments are used for system training. As for the compared systems, they use domain adaption techniques which



(A) T-SNE plotting of CSI features without generalization



(B) T-SNE plotting of CSI features with generalization

FIGURE 4.3: T-SNE plotting of CSI features distribution

require a large number of CSI data to transfer their model to the new environment. We improve AirFi with a few-shot learning technique added on so that the generalized model can be further enhanced using a few labeled CSI samples. For the evaluation, this time, 10 CSI samples from the testing domain environments are available during the training phase for all systems. The environment settings and compared systems are the same as they are in the overall evaluation.

The results are shown in Table 4.3. Obvious improvement of performances can be observed for all systems. AirFi still outperforms the other compared systems. As the amount of the given CSI data is very small, the compared systems do not have enough CSI data to fully retrain their models. As WiGr is also equipped with the few-shot learning property in their prototypical model, it performs the second best among the compared systems. While for AirFi, its system model is already generalized to different environments. By applying the few shot learning

techniques, AirFi is able to improve its performances with a limited amount of data and generalize even better to the testing environment. AirFi manages to minimize the distribution difference between the given CSI data from the testing environments and the training environments, which can be achieved with small amounts of CSI data.

4.4.5 Distribution Visualization

To better understand how AirFi can generalize its model to different environments, we use the T-SNE plotting to visualize the distribution of feature codes [109]. We plot the hidden features of CSI data from four environments, which are environment A to D.

As shown in Fig 4.3a, for a trained system without domain generalization which can be a convolutional neural network, the hidden features of CSI data from one environment are gathered together while they are away from those of other different environments. When it is applied to a new environment, its model cannot recognize CSI data as the distribution between them is very large. For AirFi which is equipped with the ability of domain generalization, feature codes from different environment settings are gathered together as their distribution differences are minimized during the training phase. This is shown in Fig 4.3b. As a result, the model trained has a high probability to generalize to the new environment.

4.5 Conclusion

This chapter studies the problem that a CSI-based human sensing system suffers from serious performance degradation under new environment settings. Specifically, we assume that no CSI training data from the new environment settings are available. Our proposed system AirFi is required to achieve robustness under unseen environment settings. To deal with this problem, AirFi takes the advantage of domain generalization to train a generalized model that can be applied to different environments. The training process of AirFi does not require CSI data from the target environment settings which is more suitable for the real-world situation.

The experimental results show that AirFi outperforms the state-of-the-art in this field.

Chapter 5

Robust WiFi-based Human Authentication System via Few-shot Open-set Recognition

5.1 Introduction

We have addressed the issue of performance degradation under different environment settings in previous chapters. Focusing on CSI-based activity and gesture recognition applications, we showed that our approaches can improve the systems' robustness under dynamic environment settings.

There are some new remaining challenges for other CSI-based smart sensing systems. For example, CSI-based user authentication is receiving more attention in recent years. WiFi signal reflected by a walking human generates distinctive variation due to the differences in physical characteristics and gait of different people. As a result, the biometrics differences of gaits among different users can be captured by the CSI. Most existing CSI based user authentication systems construct their system model using deep learning technology [76, 93, 95]. They can achieve good performances with a large amount of CSI data to train their systems. However, collecting a large number of data is time-consuming and labor intensive. It increases the setup costs and training time, which affects the scalability of these systems. It may also lead to the overfitting problem. Besides, existing works on

CSI-based human authentication lack effective training to enable systems' ability to detect unknown illegal users.

In this chapter, we address two problems. Firstly, we aim to build a CSI-based user authentication system model with very limited amounts of CSI data while it can perform accurate user authentication works by improving its training efficiency. Secondly, we propose an effective anomaly detection strategy for CSI-based user authentication systems. Without any information given for the unknown intruders, the system must be able to distinguish between them and existing users within the system.

To address these issues, we propose a new CSI-based user authentication system CAUTION. It takes the advantages of few-shot learning technology. Compared to existing works which utilize deep learning technology, it is able to construct the system model with only a few CSI samples. Besides, with the open-set system design, CAUTION is able to detect intruders with no prior knowledge of them.

CAUTION extracts feature codes from the received CSI samples and map them on the feature space. Then it calculates central points of each user class using these feature codes. CAUTION identifies different users according to the Euclidean distance between the downsampled CSI data and central points. Besides, it is also designed to detect intruders. After the CSI samples are received and mapped onto the feature space. CAUTION computes a distance ratio based on its Euclidean distances to different central points. The ratio is used to measure the similarity to different user profiles. CAUTION detects unseen strangers by comparing the ratio with an intruder threshold. The intruder threshold can be optimized without any prior CSI data from intruders, which is very realistic. Extensive experiments are conducted to test CAUTION. Results show that CAUTION is able to perform accurate user authentication with a limited amount of CSI data provided.

The contributions of the chapter are summarized as follows:

- We propose a novel human authentication system CAUTION that overcomes the issue of requiring massive labelled training data in existing works. CAUTION is able to construct the system model using a limited number of training data by few-shot learning techniques.

- We propose an anomaly detection strategy for open-set human authentication problem by comparing Euclidean distance ratios in the feature space. Such strategy can be applied without any prior knowledge of intruders' data.
- Extensive experiments demonstrate that CAUTION outperforms state-of-the-art systems on user authentication using a limited amount of CSI training samples, and CAUTION can recognize the intruder case efficiently and accurately.

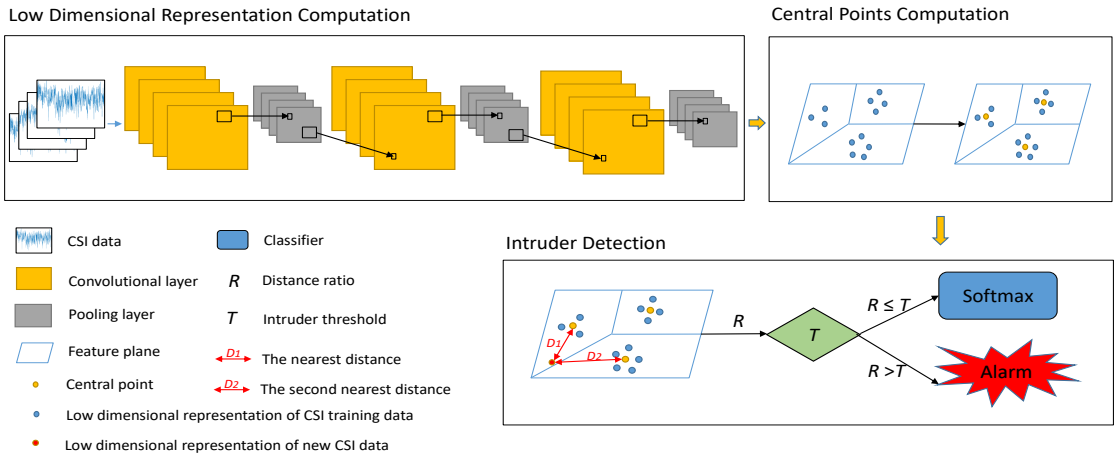


FIGURE 5.1: Architecture of the proposed Caution system

5.2 Problem Formulation

We investigate the problem of CSI-based user authentication in this chapter. Most current CSI-based user authentication systems require a large amount of CSI data to train their models, which is time-consuming and labor intensive [76, 93, 95]. Moreover, it affects the scalability of these systems. To address this problem, it is necessary to reduce the amount of CSI training data. Besides, CSI-based user authentication systems lack effective anomaly detection strategy for strangers. They are only able to detect illegal users whose CSI data are involved in the training process. While for unseen strangers, they are not able to detect them effectively.

Given the situation that, for K different users, only a small amount of CSI data $\{X, Y\}$ is collected for each user. $\{x_i\}$ is the CSI sample set of the i_{th} users' gait

and $y_i \in \{1, ..., K\}$ is the labels of users. We aim to use these CSI samples to train the system model to authenticate users.

Besides, we also design our system to be able to detect unseen strangers based on an intruder threshold T. For a new CSI sample x_n, we calculate its similarity score R based on a distance function d. Our system decides whether it belongs to existing users' classes k or strangers as follows:

$$\theta(x_n) = \begin{cases} k, & R \leq T, \\ Stranger, & R > T \end{cases} \tag{5.1}$$

where θ is the set of model parameters. We aim to find the best value of the intruder threshold T to optimize the system.

5.3 System Overview of CAUTION

In this section, we introduce our system CAUTION. As shown in Fig. 5.1, CAUTION is composed of two parts: a feature extractor and a feature plane. The feature extractor downsamples large dimensional CSI data from different users and converts them into low dimensional representations. Having these low dimensional representations on the feature plane, CAUTION calculates the central point for each user as their CSI profiles. Therefore, when new CSI data is received, the distributions of the users recognition are given based on the Euclidean distances between the low dimensional representations of new CSI data and each central point.

Different from previous works which mainly use deep neural networks with many training samples to construct detailed models for different users and then generate the classification distributions, CAUTION uses much less CSI samples to build users' profile on the feature plane. The amount of data required for system training is greatly reduced as CAUTION does not compute a detailed model for different users based on their features. Instead, by taking advantages of few-shot learning it performs user classification by comparing the similarity between the received data and CSI profiles based on the distance function.

Besides, CAUTION uses an intruder threshold for unknown illegal intruders detection. Inspired by the research works on the open-set challenge [110], we enable CAUTION with the ability to detect unknown intruders whose data are not involved during the training phase. The intruder threshold is used to measure the proper range where the users' data belong to.

5.3.1 Gait Representation Learning

Firstly, CAUTION uses a convolutional neural network F with learning parameters θ as a feature extractor to convert large dimensional CSI samples into low dimensional representations. The network F consists of three convolutional layers $C(n_k \times n_k; n_{fm})$, where n_k is the kernel size and n_{fm} is the number of feature maps, three max pooling layers P. The model architecture is represented by the shorthand notation: $C(5 \times 5; 32) \rightarrow P \rightarrow C(5 \times 5; 128) \rightarrow P \rightarrow C(5 \times 5; 128) \rightarrow P$. Leaky rectified linear units (leaky ReLUs) are used in each layer with the negative slope of 0.01.

In our system, the dimensions of the received gait CSI data from the transmitter are very large which is not suitable for further computation and model construction. We use the CNN F to process these samples. Through each convolution layer, the CSI samples are downsampled to a lower dimension. Then the pooling layers further reduce the size of data by pooling the important features of the downsampled data. This process reduces the computational effort for model construction as CNN can effectively learn relevant features for high-dimensional gait CSI data and convert them into low-dimensional representations.

5.3.2 Selecting Prototypical Feature

In the few-shot learning for image classification [111], an M dimensional representation for each class is calculated as the class prototype. The prototype is seen as the cluster center for each class. Inspired by this idea, in our work, CAUTION aims to select prototypical features by computing the central points in the feature space. CAUTION computes the central points for each user with these low dimensional representations processed from CNN on a low-dimensional feature plane. It is essential to have these central points in our system. They can be computed with

limited amounts of CSI data and further used as users' CSI profiles. A distribution over different users for a new CSI sample can be given based on its Euclidean distances to these central points.

Given that we have CSI data $\{X, Y\} = \{(x_1, y_1), ..., (x_N, y_N)\}$ where x_i is the CSI samples of different users' gaits and $y_i \in \{1, ..., K\}$ is the labels of users. $\{X, Y\}_k$ is used to represent the set of CSI data with user k. In order to train the system, $\{X, Y\}$ are separated into two sets. Half samples from each user are grouped as the support set $\{X, Y\}_{sup}$ and the rest are grouped as the query set $\{X, Y\}_{que}$.

In order to train CAUTION, we firstly input the support set into the system. Data inside the support set is processed by the CNN and corresponding low dimensional representations are computed. Having these representations on a low-dimensional feature plane, CAUTION is able to compute the central points as prototypical features for each user which is essential for building our few-shot learning model.

With CSI samples of K different users, CAUTION calculates central points for different users. The central point c_k for user k is the mean vector of CSI samples in the support sets for the user k.

$$c_k = \frac{1}{|\{X, Y\}_k|} \sum_{(x_i, y_i) \in \{X, Y\}_k} F_\theta(x_i). \tag{5.2}$$

For CSI samples in the support set for user k, after being converted to low dimensional representations, their mean vectors of each user in the feature plane are computed, which is referred to as the central points. They are treated as the CSI profile for corresponding users. With those central points computed using the support set, we then input the query set. CSI data from the query set are also processed by the convolutional neural network and converted into low dimensional representations. For each representation embedded on the feature plane, CAUTION calculates distances between the representation and every central point. Furthermore, a distribution over different users for this CSI sample can be obtained based on a softmax over distances.

$$p_\theta(y = k|x) = \frac{exp(-d(F_\theta(x), c_k))}{\sum_{k'} exp(-d(F\theta(x), c_{k'}))}, \tag{5.3}$$

where d is a distance function, and we use the Euclidean distance in CAUTION. Then CAUTION classifies CSI data from the query set based on their distributions calculated over different users. To optimize the network, we minimize the negative log probability with the ground truth.

$$L(\theta) = -log p_\theta(y = k|x), \tag{5.4}$$

where L is the cross entropy loss. We have also tested with other loss functions, such as Mean Square Error and Mean Absolute Error. The cross entropy loss function performs better than others. The loss is used to optimize the neural network, therefore the prototypical features and low dimensional representations are also optimized accordingly. Finally, CAUTION is able to classify different users based on new CSI samples received.

5.3.3 Intruder Threshold Optimization

Besides user recognition, user authentication system should also have the ability to detect intruders. In practice, CSI samples from intruders are difficult to obtain in advance. Most of the time, intruders will be strangers whose samples are never seen by these systems. In order to detect those intruders, systems must have the ability to classify input data from unknown classes. Inspired by research works on the open-set problem [110], without any prior knowledge of intruders, we enable CAUTION to recognize strangers and classify them as illegal intruders using an intruder threshold.

Given a new CSI sample x_n, CAUTION processes it with its feature extractor and generates the corresponding low dimensional representation on the feature plane as shown in Fig. 5.1. CAUTION computes and selects the distance to the nearest central point C_{1st} and the distance to the second nearest point C_{2nd}. With these two distances, CAUTION computes a distance ratio R.

$$R = \frac{d(F_\theta(x_n), C_{1st})}{d(F_\theta(x_n), C_{2nd})}. \tag{5.5}$$

The ratio R is then compared with a threshold T, where $0 < T < 1$. If R is less than or equal to the threshold T, x_n is classified to the class of C_{1st}. Otherwise, it is classified as intruders.

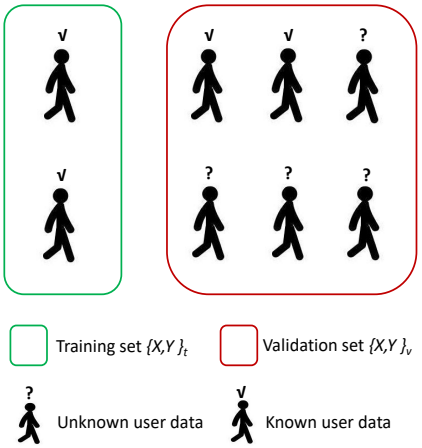


FIGURE 5.2: Data grouping for intruder threshold optimization

In order to detect intruders accurately, it is crucial to optimize the value of the intruder threshold T. In order to optimize T, we separate the previously obtained training data $\{X, Y\}$ from k users into different groups as shown in Fig. 5.2. Half users' data are treated as known users data, and their data are grouped as known CSI samples. The rest users are treated as unknown users data, and their data are grouped as unknown CSI samples. Then we separate $\{X, Y\}$ into two sets: one set is named as the training set $\{X, Y\}_t$ (green rectangle) which contains half of known CSI samples from each known user, and the other set is named as the validation set $\{X, Y\}_v$ (red rectangle) which contains the rest of those known samples and all unknown CSI samples. We train the system on user identification with $\{X, Y\}_t$. After that, we input the validation set $\{X, Y\}_v$. For each CSI sample in the validation set, CAUTION calculates its distance ratio R. The T is chosen from 0 to 1. CAUTION will try out I different values with index i_{th} evenly distributed within this range. We compare the ratio R with these different values and find which value gives the best performance in distinguishing between unknown samples and known samples. The best intruder threshold should provide the smallest loss (error rate) of intruder detection test. Assuming the i_{th} value provides the best result, we further select another I different values from $[(T_{i-1} + T)/2, (T, T_{i+1})/2]$ and repeat the previous procedure for N_{iter} times as shown in Fig. 5.3 until we find the best value for T.

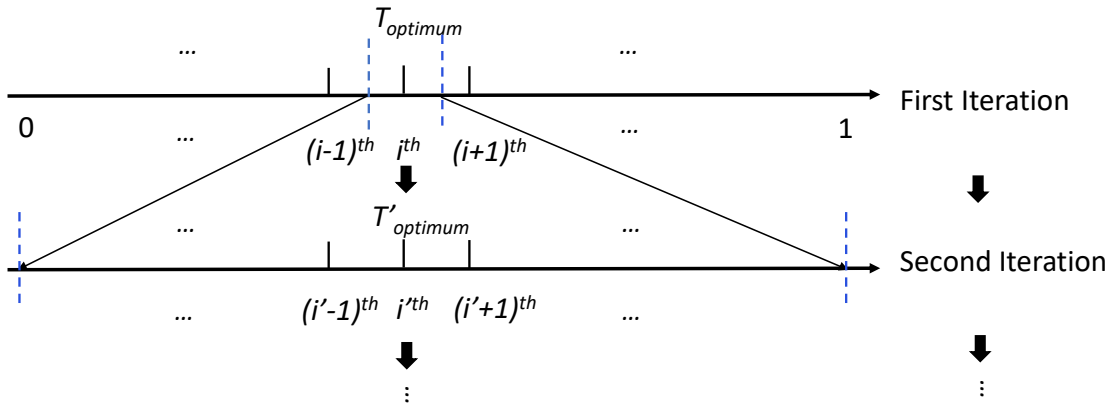
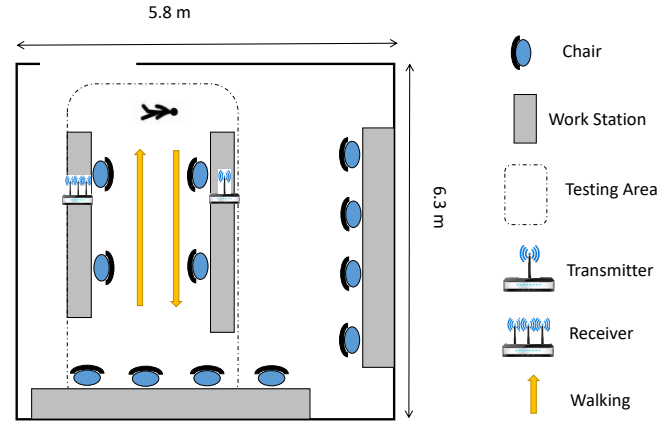


FIGURE 5.3: Threshold optimization iteration

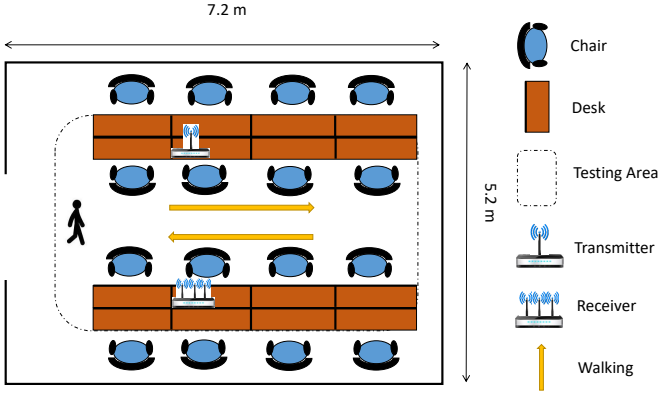
With the set T value, whenever new CSI samples are received by the system, CAUTION calculates their distance ratio R and compares it with the threshold T to detect whether there are intruders. The intruder threshold measures the distances ratio of the low dimensional representations to different central points, which is considered as similarity scores to each users class. The training phase does not involve the CSI samples of intruders. The idea is to train the system so that is is capable of distinguishing known CSI samples from unknown ones. Provided with CSI data only from known users, we simulate the case of intruders and measure the degree of similarity to classify people as either users or intruders. Though CAUTION does not have any features from intruders as no prior knowledge on intruders' data is given, CAUTION is able to tell whether it belongs to any existing classes. Therefore, CAUTION is able to detect unknown CSI samples and classify them as intruders.

5.4 Experiments

We test CAUTION on user authentication in a series of experiments in this section. Besides, we also test its ability on unknown stranger detection. State-of-the-art systems in this field are chosen for comparison.



(A) Lab layout



(B) Cubic Layout

FIGURE 5.4: Layouts

5.4.1 Environment Setup and Data Collection

The experiments are conducted in two places, a lab and a cubic office. We illustrate the location layouts in Fig. 5.4. We put the routers whose firmware has been upgraded to our CSI platform [97]. The router with one antenna is used as the transmitter and the other router with three antennas is used as the receiver. The router in AP mode sends wireless signals to the receiver in client mode at 5 GHz with a bandwidth of 40 MHz. The two routers are placed within the testing area as illustrated in Fig. 5.4.

We select 12 males and 8 females to join our experiments as volunteers for data collection. Their ages are from 20 to 28 years old. They are of different heights and weights for samples' diversity. We choose 15 people of them as the legal users group and the rest 5 people are labelled as illegal users. The CSI data of illegal

users are only used for intruder detection. Their data are absolutely unavailable during the training phase of CAUTION.

Every volunteer is required to walk in the testing area along the indicated direction using yellow arrows in Fig 5.4. At the same time, the transmitter transmits wireless signals to the receiver, and our CSI platform measures and records the CSI data of users' gait information. The CSI sample is in the size of $114 \times 3 \times 500$. Only the amplitude information is used for system training in CAUTION.

CSI data of users' gaits are collected in different locations. Besides, we design different scenarios at each location to test the robustness of CAUTION under different settings. The experimental scenarios are designed as follows:

- Scenario A (S-A): Volunteers walk through the testing area individually, wearing T-shirts and short pants.
- Scenario B (S-B): Volunteers walk through the testing area individually, wear jackets, long pants and a backpack.
- Scenario C (S-C): Volunteers walk through the testing area individually, wearing T-shirts and short pants. Surrounding people are around the testing area, performing daily activities such as chatting, making gestures, jumping and etc.

5.4.2 Overall Evaluation

In this subsection, we test CAUTION on two tasks: user identification and intruder detection.

For the first task, we select CAUTION with GATEID [91], CSIID [93] and Wihi [95]. All these systems are able to perform user identification well after sufficient training. Their system models are constructed based on deep neural networks (DNN). We test the performances of all systems on user identification with the user group size of 2 to 15 users. For each user, only 20 CSI data are available for training.

The results are shown in Fig. 5.5 and Fig. 5.6, which is for lab and cubic office respectively. We indicate the user group size via the horizontal axis while the

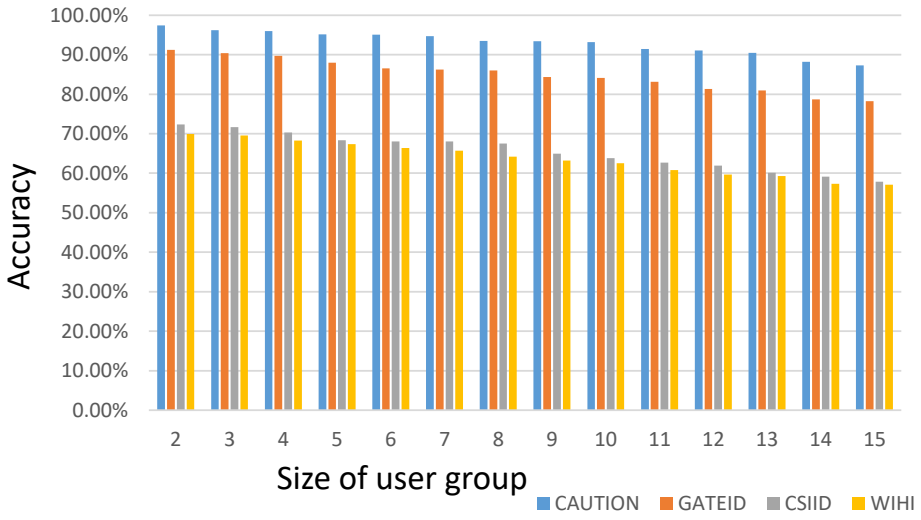


FIGURE 5.5: User identification with 20 CSI samples in lab

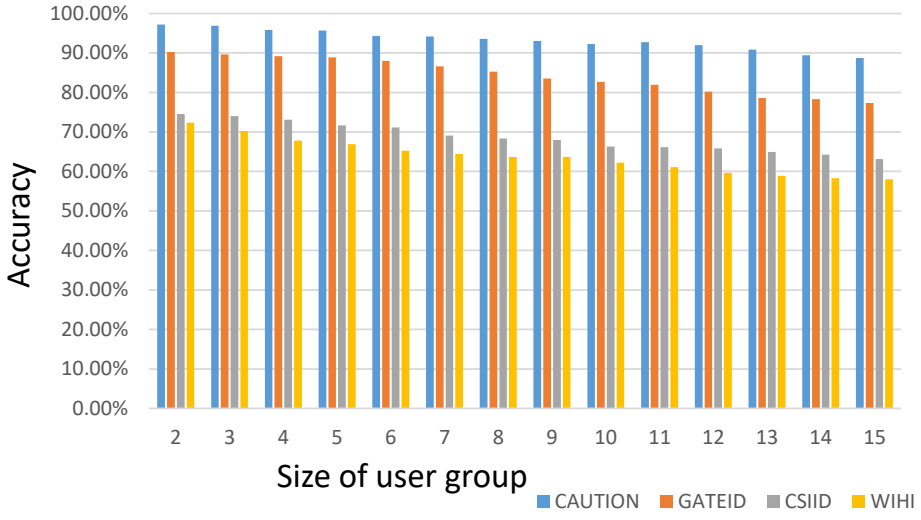


FIGURE 5.6: User identification with 20 CSI samples in cubic office

vertical axis shows the accuracy. With the increasing number of user sizes, it is more difficult to identify users accurately. Trained with only 20 CSI data, CAUTION achieve an average accuracy of 93.06%. It is higher than the second highest system GATEID by 10%. GATEID, CSIID and Wihi are not able to train their model sufficiently with only 20 CSI data of each volunteer. Their DNN models require more CSI data for system training. While CAUTION can train its model well via few-shot learning for user identification. Especially when the user sizes grow, DNN based models require even more data for system training. As a result, their performances drop faster than CAUTION. Similar performances are shown in both locations.

For the second task, intruder detection, WiAu [76], Wihi [95] and CSIDFI [112] are chosen for comparison. As GATEID focuses on user identification and is not designed to detect intruders, we replace it with WiAu in the intruder detection evaluation. The amount of legal users grows from 6 to 15. The more legal users systems need to store, the more difficult the intruder detection task is. Because the probability that intruders' data is very similar to users' data is higher. Only 20 CSI samples are available for each legal user for system training. Then the testing CSI samples include both CSI data from legal users and illegal users. Systems are required to detect unseen strangers from input CSI samples.

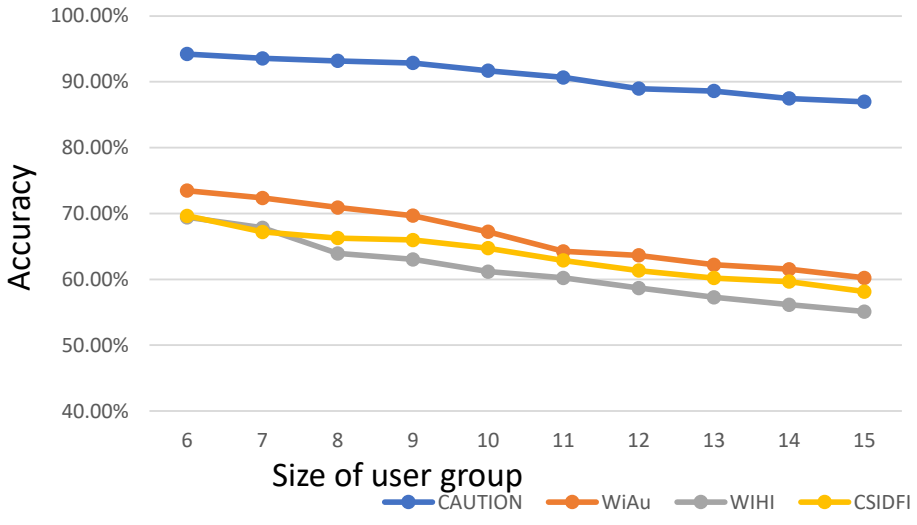


FIGURE 5.7: Intruder detection with 20 CSI samples in lab

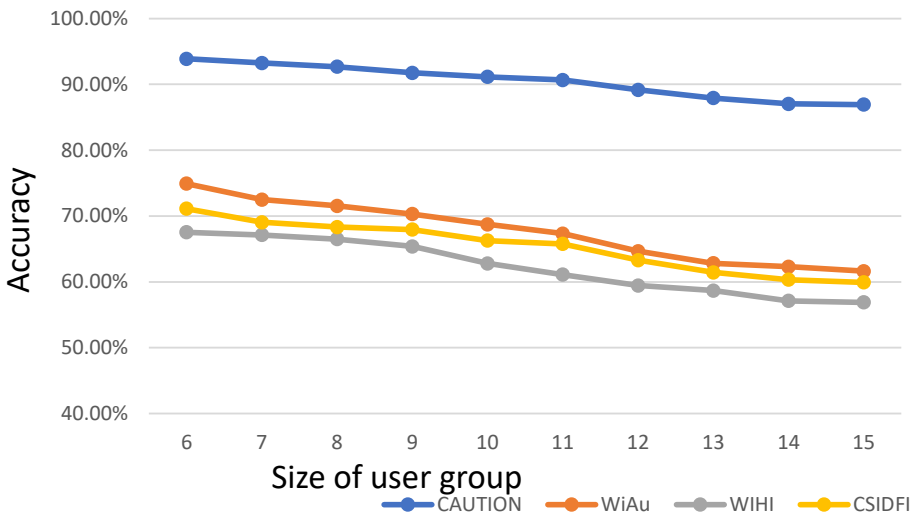


FIGURE 5.8: Intruder detection with 20 CSI samples in cubic office

The results of intruder detection in lab and cubic office are shown in Fig. 5.7 and Fig. 5.8. The horizontal axis indicates the group size of legal users stored in systems and the vertical axis shows the accuracy. CAUTION achieves better performances on intruder detection than the other compared systems. Instead of using the Euclidean distances to define the similarity of the new CSI sample, CAUTION calculates the distance ratio and uses it to measure the relative similarity score to detect strangers. The open-set design allows CAUTION to classify those with low similarity scores as intruders even if their CSI data are not available during the training process. The value of the intruder threshold varies according to the group size, which helps CAUTION to adapt to different group sizes well. For other compared systems, the limited amounts of CSI data provided from each user affect the model construction of legal users. It also impacts the intruder detection performances. Many legal users are detected as intruders due to the insufficiently trained models. When more users are included in the legal user group, CAUTION remains to be robust.

Fig. 5.9 and Fig. 5.10 show the recall of intruder detection. The compared systems have lower recall than CAUTION, as they are only able to detect intruders whose CSI features are not similar to any of these existing users in the database. When the probability distributions of the received CSI samples are approximately uniform among existing user classes, they are classified as intruders. In practice,

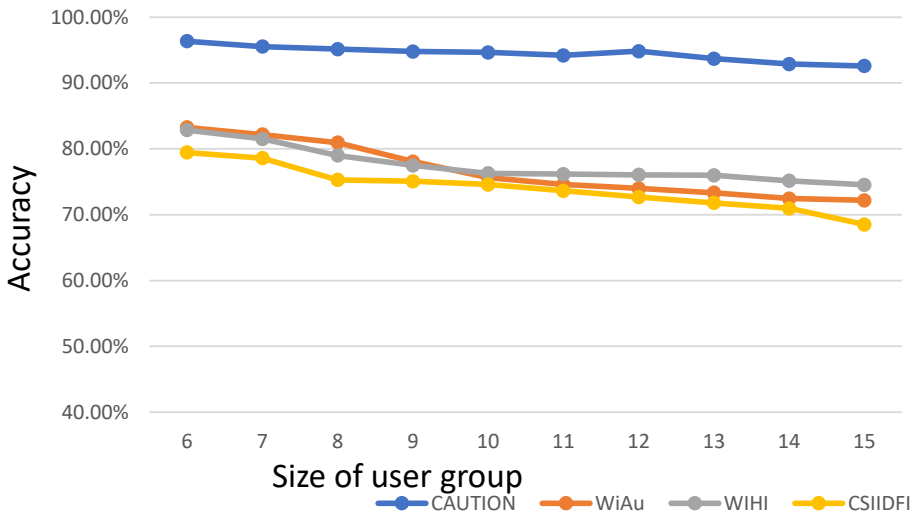


FIGURE 5.9: Recall of intruder detection with 20 CSI samples in lab

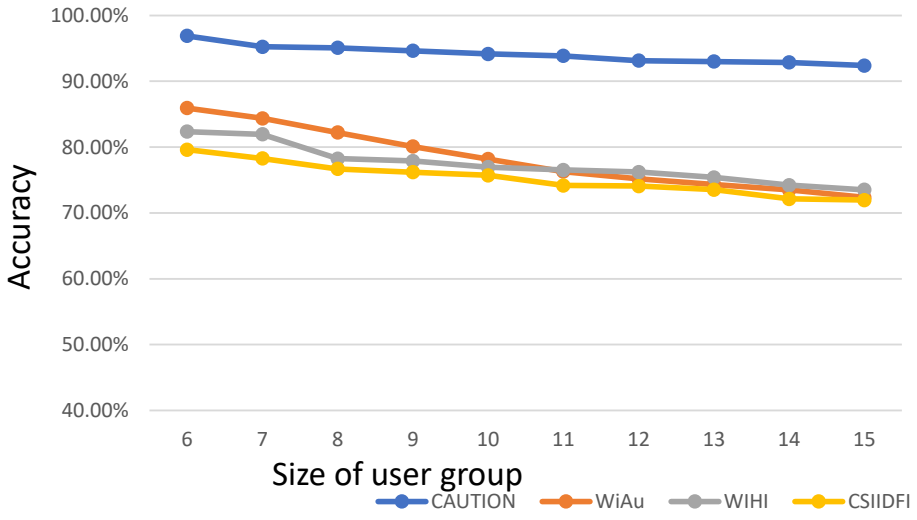


FIGURE 5.10: Recall of intruder detection with 20 CSI samples in cubic office

it is possible that the gait features of intruders are similar to certain stored users. With the optimized intruder threshold T, CAUTION is able to measure whether the similarity is enough to classify them as users or intruders.

5.4.3 Different Sizes of CSI Training Data

We study the impacts of different amounts of training on the user identification accuracy of CAUTION. We increase the amount of training data used for all experimental systems and observe their performance variations.

In the first round of experiments, we set the value of training CSI samples from each user as 40.

TABLE 5.1: User identification using 40 CSI samples

User Size	CAUTION	GATEID	CSIID	WIHI	Loc
2	99.24%	93.41%	86.64%	85.76%	Lab
	99.18%	93.41%	89.15%	90.51%	Cubic
5	97.85%	89.12%	80.35%	81.47%	Lab
	96.28%	89.66%	84.68%	84.71%	Cubic
8	96.37%	86.01%	74.58%	74.92%	Lab
	94.79%	87.03%	75.91%	77.16%	Cubic
11	93.27%	82.61%	72.45%	68.69%	Lab
	91.35%	83.21%	65.15%	65.89%	Cubic
15	88.94%	77.86%	67.00%	65.48%	Lab
	87.69%	76.59%	59.43%	61.65%	Cubic

TABLE 5.2: User identification using 100 CSI samples

User Size	CAUTION	GATEID	CSIID	WIHI	Loc
2	99.67%	99.65%	98.59%	98.37%	Lab
	99.39%	99.14%	99.57%	98.68%	Cubic
5	96.46%	97.72%	97.24%	96.28%	Lab
	96.89%	95.68%	97.05%	96.18%	Cubic
8	95.18%	94.19%	93.95%	93.01%	Lab
	93.91%	93.28%	94.59%	93.07%	Cubic
11	91.88%	93.11%	92.17%	92.05%	Lab
	92.04%	92.25%	92.26%	92.18%	Cubic
15	87.89%	88.91%	88.15%	87.17%	Lab
	88.00%	89.48%	89.64%	88.35%	Cubic

The results are shown in Table 5.1. With more training CSI samples, the performances of all experimental systems are improved. Those DNN based models have more data for system training, so their accuracy increases obviously. CAUTION still performs best among all experimental systems.

In the second round of testing, we use 100 CSI samples from each user for system training. We show the results in Table 5.2. It is observed that with 100 training CSI samples of each user, DNN based systems achieve very good performances. Their models are trained sufficiently. Their performances are sometimes even slightly higher than CAUTION whose performances are not improved greatly. Though CAUTION can construct its model efficiently with a few CSI samples, its potential to grow is limited. In situations where very large amounts of CSI samples are available, CAUTION is not always able to outperform DNN based systems. In our situation, we are only provided with very a limited amount of CSI samples for training, which is more realistic. Thus, CAUTION is able to address this problem and builds its model efficiently.

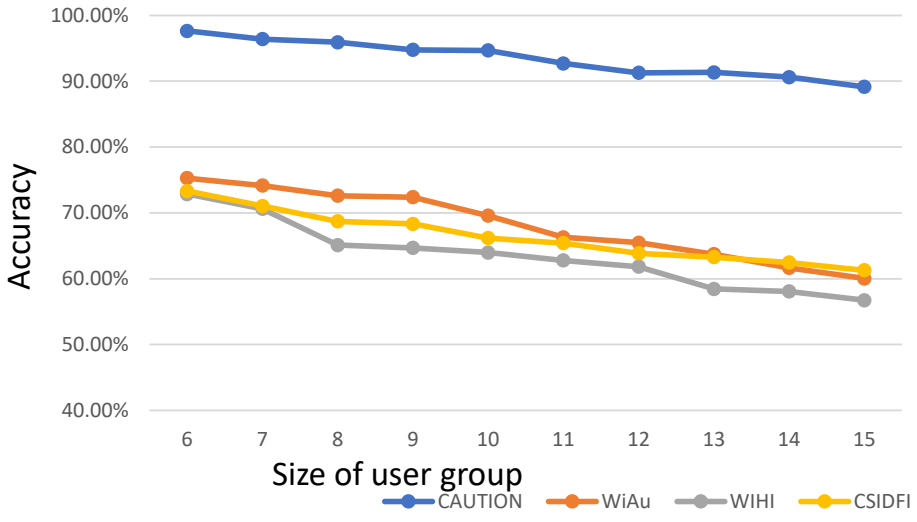


FIGURE 5.11: Intruder detection using 40 CSI samples in lab

The impacts of different amounts of CSI samples on intruder detection are also tested. We increase the amounts of training CSI samples to 40 per user. In Fig. 5.11 and Fig. 5.12, it is shown that performances of CAUTION is improved. More CSI data also help CAUTION to optimize a more precise value of intruder threshold T. The performances of the other systems only increase slightly. For them, more

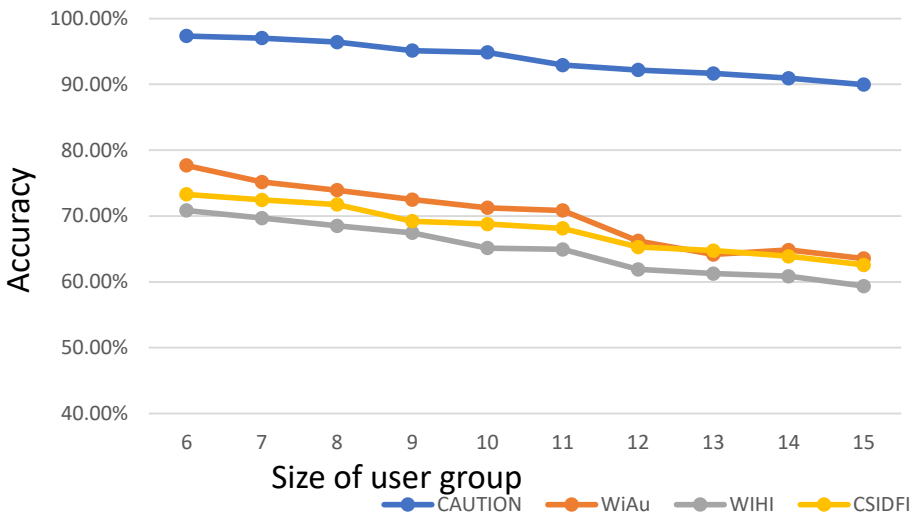


FIGURE 5.12: Intruder detection using 40 CSI samples in cubic

CSI samples provided do not lead to large improvements in intruder detection, as they are not able to optimize their system model effectively for unseen stranger detection.

5.4.4 Impacts of Surrounding Disturbance

In the real world, there are usually unexpected disturbance from the surroundings. We test the pact of surrounding disturbance on CAUTION in this subsection.

TABLE 5.3: User identification under surrounding disturbance using 20 CSI samples

User Size	CAUTION	GATEID	CSIID	WIHI	Loc
2	95.64%	85.72%	70.24%	67.59%	Lab
	96.15%	87.29%	67.64%	67.58%	Cubic
5	95.09%	81.98%	68.11%	64.08%	Lab
	94.37%	85.27%	63.12%	65.66%	Cubic
8	94.75%	76.29%	64.26%	60.64%	Lab
	92.91%	82.11%	61.96%	63.94%	Cubic
11	90.82%	71.54%	61.77%	59.34%	Lab
	89.63%	79.25%	59.58%	60.15%	Cubic
15	85.98%	66.72%	57.39%	57.10%	Lab
	85.06%	77.21%	57.14%	55.97%	Cubic

We evaluate the performance of CAUTION in S-C to test whether the disturbance from surrounding people affects its performance significantly. We also change the layouts around the testing area to create more interference. The selected compared systems are as same as the previous subsection.

We conduct the experiments on user identification with 20 CSI samples per user. The results are shown in Table 5.3. All systems perform worse than the disturbance-free situation. CAUTION performs best among the compared systems after its accuracy decreases by about 3%. The performances drop larger for the other compared systems.

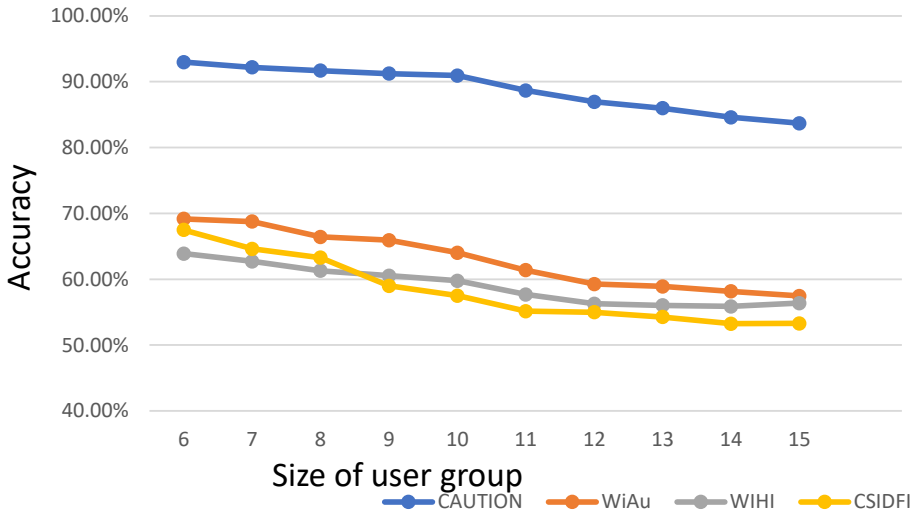


FIGURE 5.13: Intruder detection with 20 CSI samples in lab under surrounding disturbance

The impacts on intruder detection of surrounding disturbance are also tested. The results are illustrated in Fig. 5.13 and Fig. 5.14. The accuracy of all systems becomes lower. CAUTION still remains robust and performs better than the other compared systems.

5.4.5 Impacts of Users' Dressing

In the real world, users are not likely to be with the same clothes every day. It is normal that they change their dressing and carry a bag with them if needed. We evaluate the influences of different dressings on user identification.

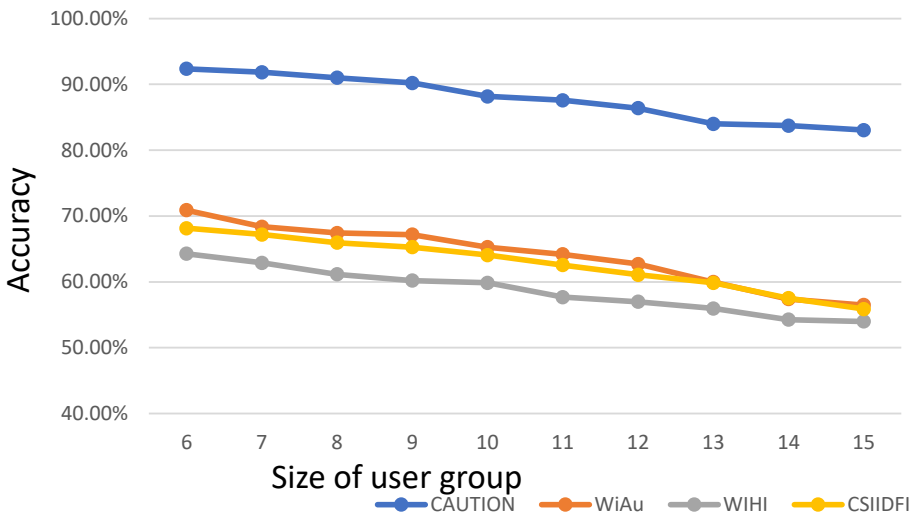


FIGURE 5.14: Intruder detection with 20 CSI samples in cubic office under surrounding disturbance

TABLE 5.4: User identification with different dressing using 20 CSI samples

User Size	CAUTION	GATEID	CSIID	WIHI	Loc
2	98.34%	89.82%	75.53%	69.21%	Lab
	98.11%	91.26%	70.42%	68.25%	Cubic
5	94.28%	87.51%	71.12%	65.48%	Lab
	95.33%	87.94%	67.29%	66.98%	Cubic
8	94.18%	85.62%	67.10%	60.69%	Lab
	93.71%	85.97%	65.16%	66.02%	Cubic
11	90.42%	79.36%	63.21%	57.58%	Lab
	91.07%	83.55%	63.24%	61.49%	Cubic
15	86.29%	74.93%	60.34%	55.67%	Lab
	87.36%	78.68%	60.98%	56.49%	Cubic

The experimental systems are trained using 20 CSI data from each user collected in scenario A and then they are tested using CSI samples collected from scenario B. We show the results in Table 5.4.

It is shown that different dressings do not affect systems' performances obviously. CAUTION can still perform accurate user identification and outperform other compared systems.

5.5 Conclusion

We investigate the issue that current CSI-based authentication systems have to be trained with lots of CSI data, which is difficult to acquire in real life situations. We propose a novel CSI-based user authentication system CAUTION. Leveraging the few-shot learning technology, its system model can be trained with only a few CSI samples. Besides, it is able to perform intruder detection with no prior knowledge of intruders' CSI samples. We conduct a series of experiments to test CAUTION. The results show that CAUTION outperforms the other compared systems with a limited amount of CSI data given and is able to remain robust under surrounding disturbance.

Chapter 6

CSI-based Life-long Smart Sensing System

6.1 Introduction

In previous chapters, we study the problem that how a CSI-based smart sensing system can adapt or generalize to new environment settings. The experimental results in previous chapters show that our proposed techniques can help the CSI-based sensing systems adapt or generalize to a new environment setting effectively.

After a CSI-based smart sensing system is deployed in the new environment settings, it is able to perform its designed functions such as activity recognition, gesture recognition, user authentication and etc for some time. It can give very decent performances as long as the target environment settings remain static. However, in real life, the surrounding environments are usually dynamic. The ever-changing environments affect the performances of the deployed system greatly, as the CSI samples of the same activities or users are influenced by the surrounding environments as well. The deployed systems are required to mitigate the frequent dynamic changes, which may either come from users side or surrounding furniture layouts.

We have addressed this problem in chapter 3. MCBAR retrains its system with all store CSI data to adapt to the environment changes [101]. In fact, most existing systems [63, 79] rely on system retraining to overcome this problem. They usually collect new CSI data samples under the new environment settings and use them

to retrain the systems. However, it leads to another problem. If the system model is retrained with CSI data collected from the new environment setting only, it will forget about the trained knowledge of previous environment settings. This is known as catastrophic forgetting. Because during the model optimization process, model parameters are only optimized for the current tasks. It greatly affects the system performance on previous tasks.

Normally, in order for a CSI-based sensing system to adapt to a new environment setting. It has to collect large amounts of CSI data for all involved classes, which incurs high costs and is not user-friendly. To improve this, MCBAR reduces the amount of CSI data collected from the new environment by generating fake CSI data from the new environment settings through model translation. While in order to prevent catastrophic forgetting, they all have to retrain themselves with CSI data from all environment settings [63, 79, 101]. But this increases the training time and cost significantly. Besides, the hardware storage limitation does not allow large amounts of CSI data stored as well.

In order to address this issue, our proposed system LICAR is equipped with the ability of life-long learning. It can learn about the new environment settings using new CSI samples while still storing the knowledge of the previous environment settings. It uses a CSI data augmentation generator to simulate meta-CSI datasets. It helps to reduce the distribution differences between the training CSI data of different tasks. Then most importantly, when LICAR is trained for the current tasks, the parameters in its system model are updated selectively using a parameter updating buffer. It measures the importance of each parameter for previous tasks and updates them accordingly. The experimental results show that LICAR can perform life-long learning effectively and maintain robust for all trained tasks. It outperforms the compared systems under different dynamic environment settings.

The contributions of the chapter are summarized as follows:

- We propose a novel lifelong learning CSI based smart sensing system LICAR. Using the parameter updating buffer, it is able to be optimized for the current tasks and meanwhile preserving the knowledge on previous tasks by selectively adjusting model parameters.

- The meta-CSI dataset simulated using the CSI data augmentation generator is able to improve the diversity of training CSI data. It improves the training efficiency and reduces the distribution differences between CSI training dataset for different environment settings.
- Experiments show that our proposed system LICAR can perform effective life-long learning and maintain robustness in different environment settings.

6.2 Problem Formulation

In this chapter, we address the issue that existing methods lack lifelong learning ability. Existing works [63, 79, 101] are able to adapt to different environment settings via domain adaption to overcome environment dynamics. However, after they adapt to a new setting, they are not able to maintain the knowledge of previous settings unless they retrain themselves with all previous CSI data. But this leads to long training time and high costs. Large amounts of CSI data stored require large storage space of hardware devices, which is not always available in the real world.

Considering the previous setting as the source domain, CSI data X_{prev} are collected to train the system with parameters θ. The trained system with parameters θ_{prev} can work well in the source domain. After the environmental dynamics happen, the new environment setting is referred to as the target domain. In order to adapt to the new environment setting, CSI data X_{curr} are collected. The retraining process of most existing methods involves both X_{prev} and X_{curr}. If only X_{curr} are used, the new parameters set θ_{curr} probably forget the trained knowledge from X_{prev}. This increases the training time and costs significantly. Besides, the hardware storage limitation does not allow large amounts of CSI data stored as well. The ultimate goal of our system proposed in this chapter is to perform the retraining process of adaption only using X_{curr} and meanwhile keep the previously trained knowledge from X_{prev}, so that the system can perform robust activity recognition in all environment settings trained.

6.3 System Overview of LICAR

Our proposed system LICAR is composed of three parts: One CSI data augmentation generator, one parameter updating buffer and the classification model. It is shown in Fig 6.1. The CSI data augmentation generator is used to augment the received CSI data. More training CSI data usually can improve the system performances. Collecting CSI data and labeling them induce high costs and are labor-intensive. It is not possible to collect CSI data from all different settings. Thus, we use a CSI data augmentation generator to augment the training CSI dataset using the limited amount of collected CSI data. The parameter updating buffer is used to calculate the updating constraint of the parameters in the system model. As we want to let the system model still perform well for previous tasks, the parameter updating buffer is able to restrain some of the parameters from over updating. This helps to store the knowledge of the previous CSI tasks. Finally, a classification model is constructed for LICAR to recognize different human activities. The classification model is built using convolutional neural networks. It can perform deep representation learning to optimize its parameters to perform accurate activity recognition.

6.3.1 CSI Data Augmentation Generator

As we want our system LICAR to maintain robust under different environment settings, it is very helpful if the system can be trained with CSI data samples from different environment settings. Besides, during the process of life-long learning training, LICAR updates the system parameters by optimizing the system to fit well in both previous tasks and new tasks. It does good to the training process by training the LICAR with CSI data samples from more environment settings.

However, due to limited time and resources, it is not possible to collect CSI data from all different environment settings. In order to improve the situation, we add in a CSI data augmentation generator. The idea of data augmentation has been adapted in [101]. It is utilized to address the issue that the collected data is not enough in terms of its quantity. In LICAR, we apply the CSI data augmentation for one more purpose. When LICAR is in the process of life-long learning training, it is trained to fit the new tasks using new CSI data samples, while we want it to

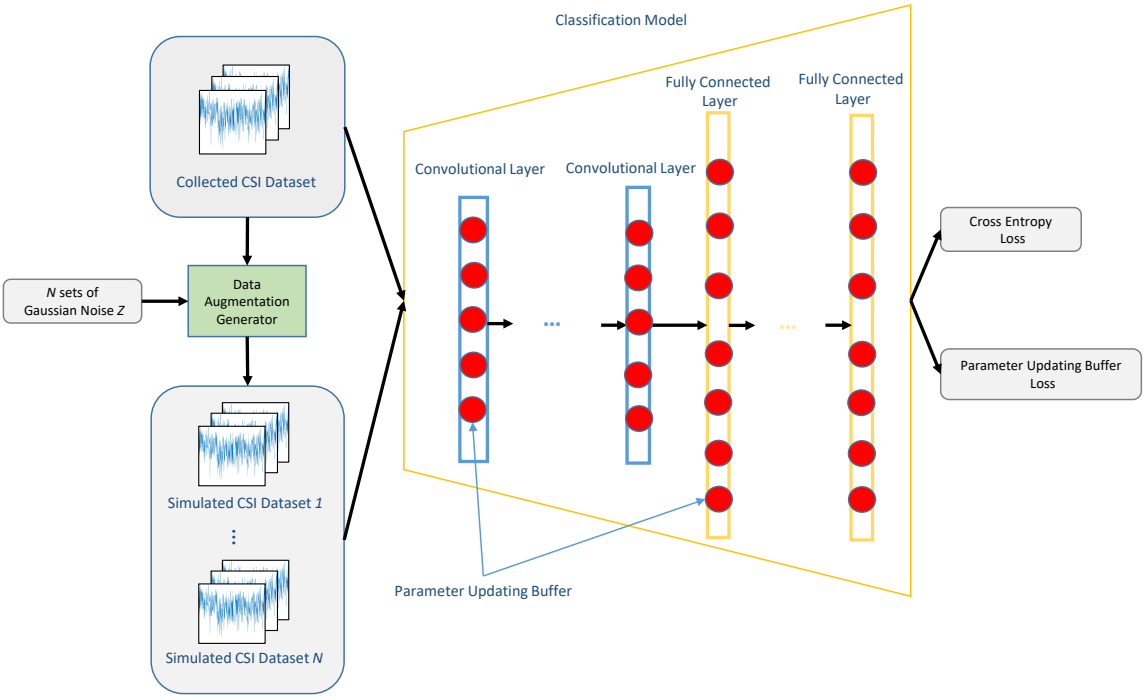


FIGURE 6.1: Training Groups Setup

keep the knowledge trained using the previous CSI data samples. By augmenting the CSI data to generate more CSI data samples under different settings, it may reduce the distribution differences between the previous training CSI data and new CSI training data. It can reduce the training difficulty of the life long learning training and encourages the convergence.

To augment the collected CSI data pairs (X, Y), where X denotes the collections of CSI samples and Y is the collection of the corresponding labels, we use a set of arbitrary Gaussian noises. Gaussian is a basic noise model used in information theory to mimic the effect of many random processes that occur in nature. It is widely used in both information data augmentation techniques and computer vision data augmentation techniques to simulate random noise [79, 101, 104], and the introduction of it will not change the label of the CSI data.

We simulate the CSI data sequences in other dynamic settings by combining the introduced noise with the collected CSI samples. We denote all the introduced arbitrary Gaussian noises as Z_n where n is the index of N different Gaussian noise Z. The simulated CSI sequences are referred as X_s. With N different sets

of Gaussian noise Z, we are able to simulate N different sets of simulated CSI sequences X_s^n. As explained above, the introduction of Gaussian noise does not change the label of the CSI data for activity recognition purposes. The simulated CSI sequences can inherit the corresponding labels from the collected CSI data pairs as $(X^s, Y^s)_n$. They improve the diversity of CSI training data and can be used to approximate the distribution of the related classes of CSI data from other environment settings.

6.3.2 Parameter Updating Buffer

A robust CSI based smart sensing system is required to adapt to the ever-changing environment to maintain its performance under the new environment settings. This can be achieved by collecting large amounts of CSI data from the new environment settings and retraining the system model with them.

While retraining the system only with the new CSI samples leads to the situation that the system will forget about its knowledge learned from the last CSI training samples, the performances for previous tasks tend to drop as the parameters inside the model are optimized only for new tasks. Some existing works manage to solve this by retraining the system with all stored CSI data samples, which increases the training cost greatly. Besides, the hardware equipment in the real world can not provide unlimited data storage. It is important to equip the CSI based smart sensing system with efficient life-long learning ability.

To achieve this goal, we add a parameter updating buffer in LICAR to enable life-long learning. When LICAR needs to be retrained with CSI data from the new settings, the parameter updating buffer will measure the importance of each parameter updated in the system model for previous tasks. For those parameters which are very important to maintain good performances in previous tasks, LICAR will update them less aggressively. While for other parameters which are not important for previous tasks, LICAR will update them more aggressively.

To define which parameter is more important for previous tasks is the crucial part of LICAR. Consider a system model with parameter θ. From a probabilistic perspective [69], this problem can be viewed as that given some CSI samples X, we

need to find their most probable values of parameters θ. We manage to calculate the conditional probability $p(\theta|X)$ by Bayes' rules

$$\log p(\theta|X) = \log p(X|\theta) + \log p(\theta) - \log p(X) \tag{6.1}$$

We now consider that we have two groups of CSI training samples, X_{prev} and X_{curr}, which are previous CSI training samples and current CSI training samples correspondingly. X_{prev} is associated with the previous tasks and X_{curr} is associated with the current training task. We take them into the equation above and get

$$\log p(\theta|X) = \log p(X_{curr}|\theta) + \log p(\theta|X_{prev}) - \log p(X_{curr}). \tag{6.2}$$

While the left hand side is the posterior probability of the parameters corresponding to the entire CSI datasets X, the right hand side has nothing to do with previous tasks except for the term $\log p(\theta|X_{prev})$. Therefore, all of the information about previous tasks must be in this posterior distribution. From this posterior distribution, we are able to distinguish which parameters are more important for previous tasks. Though it is not able to calculate the posterior distribution $\log p(\theta|X_{prev})$ directly, by applying the Laplace approximation work by Mackay [113], we approximate the posterior as a Gaussian distribution with mean given by parameter θ_{prev} and computed using the diagonal Fisher Information Matrix F. Then the parameter updating buffer L_{buff} is given as

$$L_{buffer} = \sum_{i}^{I} \frac{F(\theta_i - \theta_{prev,i})^2}{2}, \tag{6.3}$$

where i is the parameter index.

With the parameter updating buffer, LICAR will process the life-long learning retraining with new training CSI datasets as

$$L(\theta) = L_{CE}(X_{curr}; Y) + \lambda_s \frac{1}{N} \sum_{n=1}^{N} L_{CE}(X^s_{curr}; Y^s)_n + \lambda_b L_{buffer}, \tag{6.4}$$

where $L(\theta)$ is the total system loss, $L_{CE}(X_{curr}; Y)$ and $L_{CE}(X^s_{curr}; Y^s)_n$ are the cross entropy losses of current tasks for collected CSI training data and N simulated CSI training data. λ_s and λ_b are the loss weightages. The training of the parameter updating buffer does not involve the previous CSI data. All that the training process needs is the system model and current CSI training datasets. It saves much data storage and improves the training efficiency.

6.4 Classification Model

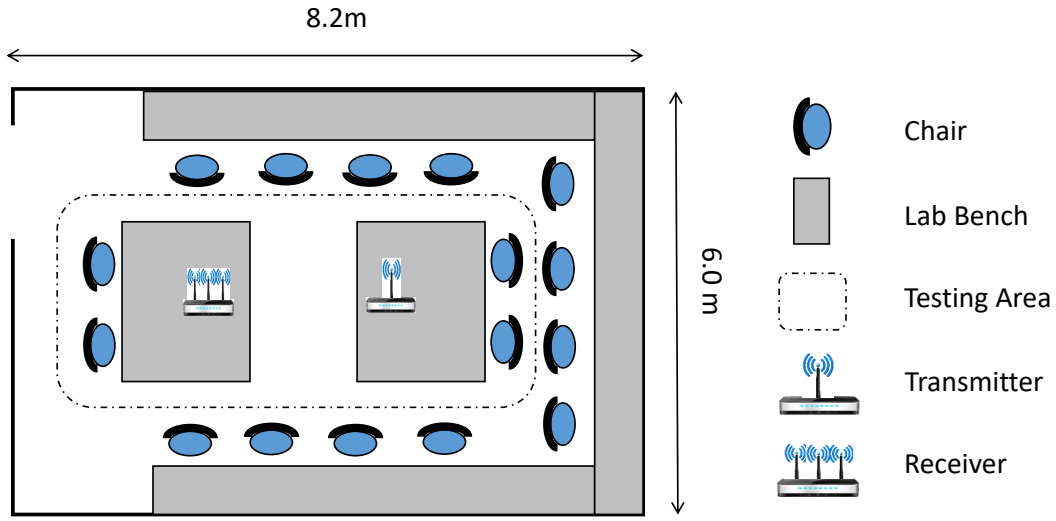
The classification model $\mathcal{C}(\theta)$ is built with three parts: three convolutional layers, three polling layers and three fully connected layers. The convolutional layer is represented as $C(n_k \times n_k; n_{fm})$, where n_k is the kernel size and n_{fm} is the number of feature maps, and pooling layers P work as a feature extractor. They downsample the high dimensional CSI data samples and extract feature codes. The fully connected layers work as a classifier. The model architecture is represented by the shorthand notation: $C(5 \times 5; 32) \rightarrow P \rightarrow C(5 \times 5; 128) \rightarrow P \rightarrow C(5 \times 5; 128) \rightarrow P \rightarrow F \rightarrow F \rightarrow F$. We use leaky rectified linear units (leaky ReLus) in each layer. Then the classification model $\mathcal{C}(\theta)$ is optimized using Eq.6.4.

6.5 Experiments

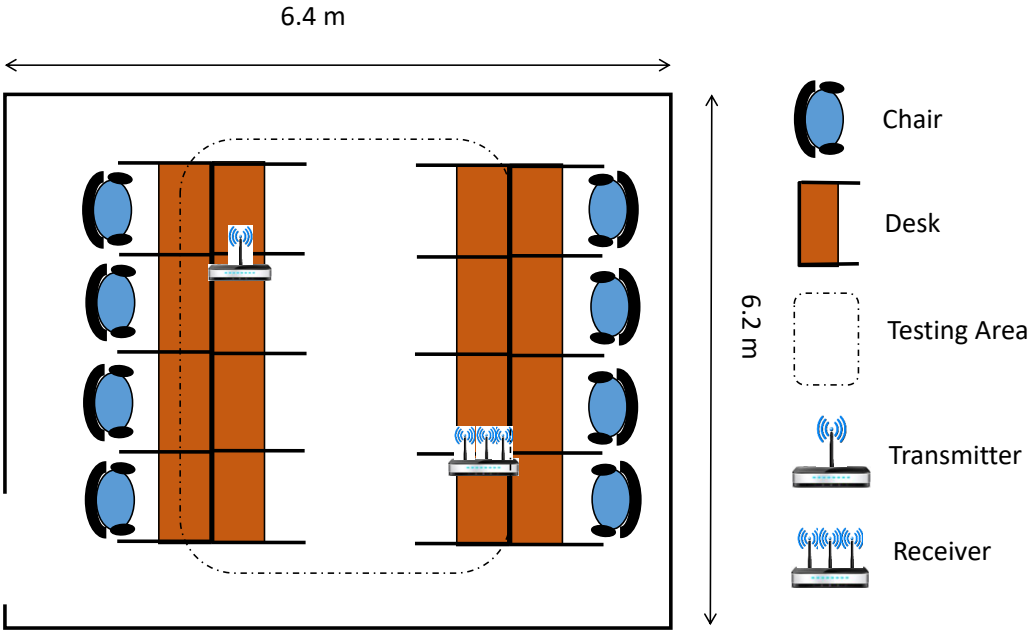
This section introduces about series of experiments for LICAR testing. We test our proposed system LICAR under different environment settings together with other compared systems. The experiments aim to test whether LICAR is able to perform effective life-long learning and maintain robust under the dynamic environment settings.

6.5.1 Environment Setup and Data Collection

We conduct the experiments in two locations, a lab and a cubic office. The layouts are shown in Fig 6.2. Two routers are used, one is the transmitter (one antenna) and the other is the receiver (three antennas). The firmware of both routers has



(A) Lab layout



(B) Cubic Layout

FIGURE 6.2: Experimental Layouts

been upgraded to our CSI enabled platform [97] for data collection. The transmitter is operated in 802.11n AP mode at 5 GHz with a 40 MHz bandwidth and the receiver is connected to the transmitter in client mode. During the experiment the transmitter keeps sending signal packets to the receiver during the experiments. The transmitted signals affected by volunteers and physical environments are received by the receiver, and the CSI enabled platform [97] measures and stores the CSI data. Fifteens volunteers participate in our experiments. Ten of them are males and five are females. We include seven human activities in our experiments including walking, running, waving, boxing, jumping, throwing and cleaning. As illustrated in Fig 6.2. Within the testing area, volunteers perform different activities.

We design four different environment settings within each location. The environment settings are designed as follows

- Environment A (E-A): the original environment setting of the location. Each user perform the designed activities with no other users around.
- Environment B (E-B): compared to the E-A, the layout furniture are different. We randomly move the positions of furniture and add in additional obstacles to change the environment settings. While for users, still only one user stays within the testing area and performs the required activities with no other users around at each time.
- Environment C (E-C): the indoor layouts are kept as same as the original environment setting. Meanwhile, the user dresses differently compared to previous settings. Besides, when one user performs required activities, other users are also around the testing area and do some daily activities.
- Environment D (E-D): dynamics from both layout furniture and users are considered.

The CSI data for each environment setting are collected periodically one week per setting in order to test the life-long robustness of LICAR. For example, during the first week, we collect CSI data for environment setting A at different times. Then in the second week, we collect CSI data for environment setting B. For each activity, 200 CSI samples are recorded under one environment setting at one experimental

location. In total, there are 2800 CSI samples collected in two experimental locations (lab and cubic office). We extract their amplitude information for further system processing. The input size of our CSI data is $114 \times 3 \times 500$.

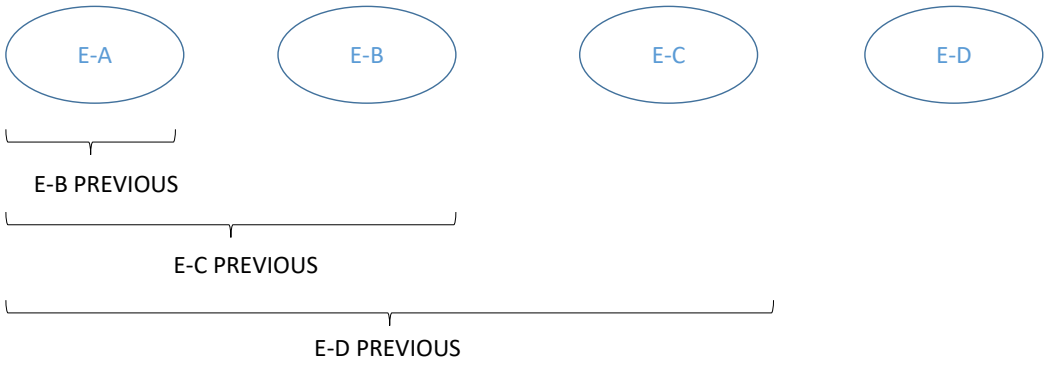


FIGURE 6.3: Training Groups Setup

6.5.2 Overall Evaluation

We compare LICAR with MCBAR [101], CSIGAN [79] and IEMUCS [63]. The compared systems also have the ability to mitigate the dynamics from the environment settings. We train the four systems including LICAR with the same amount of CSI data for each of the designed environment settings. CSI data from four different simulated environment settings are entered into all systems separately. Then we test their performances within each setting. When the systems are optimized in each setting, we also test the systems' performances on previous settings. It is used to test whether the training on current tasks causes catastrophic forgetting on previous tasks. For example, 'E-B previous' in the table means while systems are optimized for the environment setting E-B, we test their performances in previous environment settings which is E-A here. 'E-D previous' in the table means while systems are optimized for the environment setting E-B, we test their performances in previous environment settings which are E-A, E-B and E-C. This is illustrated in Fig 6.3.

The performances of systems are shown in Table 6.1 and Table 6.2. Table 6.1 shows evaluation results in lab and Table 6.2 show results in cubic office. As shown in the tables, for evaluation within one environment setting, LICAR outperforms all other systems as LICAR has a CSI data augmentation generator. It helps LICAR

TABLE 6.1: Overall Performances Evaluation

Sytems	Walking	Running	Waving	Boxing	Jumping	Throwing	Cleaning	Environment Index
LICAR	95.32%	95.15%	96.03%	94.83%	96.72%	96.10%	95.63%	
MCBAR [101]	93.48%	94.82%	95.27%	94.37%	95.21%	94.50%	93.95%	
CSIGAN [79]	94.32%	95.08%	93.18%	93.84%	94.56%	93.97%	94.78%	E-A
IEMUCS[63]	94.65%	95.09%	93.22%	95.92%	94.63%	93.47%	94.71%	
LICAR	96.27%	94.76%	93.44%	95.82%	94.14%	96.63%	95.08%	
MCBAR [101]	93.72%	93.69%	94.58%	96.03%	95.16%	94.11%	92.84%	
CSIGAN [79]	95.77%	94.34%	94.06%	93.88%	94.33%	95.15%	93.92%	E-B
IEMUCS[63]	94.28%	93.67%	95.14%	94.81%	93.69%	94.87%	92.25%	
LICAR	96.19%	97.02%	95.15%	96.64%	95.11%	95.70%	94.42%	
MCBAR [101]	94.51%	93.89%	92.11%	93.34%	95.10%	93.42%	94.29%	
CSIGAN [79]	93.44%	94.28%	94.68%	95.00%	93.74%	92.81%	93.99%	E-C
IEMUCS[63]	93.53%	94.32%	94.74%	95.61%	95.87%	93.79%	92.50%	
LICAR	95.35%	94.61%	94.93%	95.78%	96.71%	95.49%	96.59%	
MCBAR [101]	92.76%	93.82%	93.68%	94.15%	93.81%	94.68%	93.76%	
CSIGAN [79]	94.91%	96.81%	93.63%	94.75%	93.22%	93.94%	94.15%	E-D
IEMUCS[63]	93.92%	92.74%	93.77%	94.68%	94.79%	92.14%	94.86%	
LICAR	94.68%	94.16%	93.10%	94.14%	94.52%	94.98%	94.21%	
MCBAR [101]	90.17%	91.05%	89.24%	89.66%	91.22%	88.97%	90.52%	
CSIGAN [79]	86.14%	88.67%	85.25%	86.11%	87.29%	88.06%	87.49%	E-B PREVIOUS
IEMUCS[63]	89.12%	88.63%	88.72%	87.19%	86.97%	88.74%	87.98%	
LICAR	93.89%	93.56%	93.71%	94.02%	94.66%	93.17%	93.47%	
MCBAR [101]	86.24%	87.39%	88.41%	87.92%	89.27%	89.95%	87.49%	
CSIGAN [79]	85.96%	86.22%	87.84%	87.10%	87.68%	84.09%	83.74%	E-C PREVIOUS
IEMUCS[63]	84.77%	83.68%	84.06%	83.57%	85.34%	83.66%	84.51%	
LICAR	93.02%	93.62%	92.18%	93.82%	92.05%	92.99%	93.75%	
MCBAR [101]	83.54%	86.62%	84.93%	82.71%	83.72%	82.84%	81.09%	
CSIGAN [79]	81.03%	82.67%	80.82%	82.64%	81.62%	80.11%	79.66%	E-D PREVIOUS
IEMUCS[63]	82.10%	80.35%	79.84%	81.18%	82.43%	81.46%	83.69%	

to learn from more dynamic settings with the same amount of CSI data. MCBAR performs second best and is lower than LICAR only.

For the performances on previous tasks, it is shown that LICAR outperforms the compared systems significantly. Though the other three systems are optimized for the current tasks, that leads to catastrophic forgetting of previous tasks. LICAR can be optimized for the current tasks and still maintain its knowledge of previous tasks with the least performances degradation. The differences between LICAR and the other three systems are more obvious when the number of previous tasks increases. For example, when tested with tasks only prior to task two, the performances of compared systems drop less than they are tested with all three tasks prior to task four.

TABLE 6.2: Overall Performances Evaluation

Sytems	Walking	Running	Waving	Boxing	Jumping	Throwing	Cleaning	Environment Index
LICAR	96.28%	95.16%	96.87%	94.08%	95.27%	95.64%	96.14%	
MCBAR [101]	92.35%	93.22%	94.79%	93.92%	94.18%	93.48%	94.31%	
CSIGAN [79]	93.69%	92.04%	94.67%	95.12%	93.87%	93.64%	94.72%	E-A
IEMUCS[63]	93.11%	94.27%	93.54%	93.42%	94.89%	94.96%	93.70%	
LICAR	97.01%	96.39%	95.18%	96.81%	95.03%	94.79%	95.26%	
MCBAR [101]	93.78%	94.81%	93.20%	92.94%	92.08%	93.50%	94.12%	
CSIGAN [79]	93.91%	92.25%	95.62%	93.32%	94.50%	92.45%	93.69%	E-B
IEMUCS[63]	95.62%	94.17%	93.48%	94.78%	95.91%	94.21%	93.48%	
LICAR	96.82%	96.07%	95.16%	94.59%	94.81%	96.06%	95.74%	
MCBAR [101]	94.38%	93.87%	92.58%	93.73%	92.32%	92.47%	93.14%	
CSIGAN [79]	94.86%	92.51%	91.25%	94.26%	93.78%	94.65%	93.52%	E-C
IEMUCS[63]	94.28%	93.41%	92.37%	93.90%	95.89%	94.33%	92.29%	
LICAR	96.89%	94.47%	96.20%	95.31%	94.92%	96.34%	95.26%	
MCBAR [101]	93.81%	94.67%	92.74%	93.49%	94.82%	94.66%	93.64%	
CSIGAN [79]	92.36%	93.58%	94.71%	93.83%	93.11%	92.54%	94.24%	E-D
IEMUCS[63]	92.31%	94.66%	93.05%	94.95%	93.87%	94.60%	93.25%	
LICAR	95.30%	94.76%	94.18%	93.91%	94.84%	95.02%	94.77%	
MCBAR [101]	88.62%	89.70%	89.52%	88.77%	89.94%	87.19%	90.48%	
CSIGAN [79]	87.16%	88.94%	88.67%	89.98%	90.22%	86.61%	88.85%	E-B PREVIOUS
IEMUCS[63]	87.17%	84.98%	87.17%	86.68%	86.76%	86.63%	87.13%	
LICAR	94.15%	93.86%	93.09%	93.43%	94.27%	93.76%	94.32%	
MCBAR [101]	87.95%	86.82%	86.08%	85.11%	86.77%	85.29%	94.29%	
CSIGAN [79]	83.67%	84.60%	85.69%	84.21%	84.94%	84.59%	85.19%	E-C PREVIOUS
IEMUCS[63]	83.22%	83.03%	84.17%	84.97%	82.43%	84.39%	83.98%	
LICAR	93.58%	94.24%	93.30%	92.72%	92.63%	93.82%	92.18%	
MCBAR [101]	82.12%	81.23%	83.58%	82.67%	84.79%	83.41%	83.34%	
CSIGAN [79]	80.22%	81.06%	83.55%	80.20%	81.35%	80.68%	82.95%	E-D PREVIOUS
IEMUCS[63]	80.32%	79.58%	81.44%	78.93%	80.07%	81.65%	80.09%	

6.6 Conclusion

In this chapter, we further investigate on the performance degradation of CSI-based smart sensing systems under different environment settings. Compared to the MCBAR in chapter 3, LICAR does not require the system retraining with the entire CSI dataset stored for all environment settings. Instead, it only needs the CSI data from the new environment settings. It can be optimized for the current tasks and meanwhile maintain robustness for previous tasks. It highly improves the training efficiency and reduces system costs. Large storage space of hardware devices is no longer required as well. The experimental results show that LICAR can adapt to the new environment settings and preserve the knowledge for previous trained environment settings with only CSI data from the new environment settings during the system retraining.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In order to improve existing CSI-based smart sensing systems, there are many challenges that need to be addressed. This thesis focuses on developing robust and training efficient CSI-based smart sensing techniques. We study the problem that existing techniques suffer from severe performance degradation under environment dynamics.

The contributions of the thesis can be summarized as follows:

- We study the issue of performance degradation under environmental dynamics and propose a Multimodal Channel State Information Based Activity Recognition (MCBAR) system. Taking advantages of the domain adaption, it can perform multimodal model translation into different environment settings. With the help of this, MCBAR is able to transfer the labeled CSI data from one environment setting to another environment setting based on small amounts of unlabeled real CSI data collected in the new environment setting. The fake CSI data generated have similar features to the unlabeled data from the new environment setting meanwhile inherit the labels of CSI data from the original environment settings. By model translations, the CSI-based smart sensing systems can be transferred to different environment settings and perform robustly.

- We address the issue that existing CSI-based smart sensing systems must be trained with massive unlabeled high-quality CSI data from the new environment to adapt to the new environment settings, which is usually unavailable in practice. We utilize CSI data collected from several training environment settings where it is easier to collect CSI samples. Then we augment the CSI data collected and extract the critical common features between them. The extracted features are also augmented to improve their diversity. With these augmented features, an augmented environment-invariant robust WiFi gesture recognition system is built. The trained model can be generalized to unseen scenarios, which does not require collecting any data for adaptation to the new environment.

- CSI data can reveal the distinctive gait features which can be used for user authentication. However, existing CSI-based user authentication systems require large amounts of CSI data to train the system model. To address this problem, we leverage the idea of few-shot learning to design the system CAUTION. It utilizes the prototypical features of CSI data to construct its model, which can be achieved using a few CSI samples for each class. Besides, it is equipped with a novel anomaly detection strategy for open-set human authentication problems by comparing Euclidean distance ratios in the feature space. Such strategy can be applied without any prior knowledge of intruders' data.

- Under the ever-changing surroundings, a CSI-based smart sensing system should be able to perform life-long learning to avoid performance degradation when the environment evolves. While simply training the system only with the CSI data from the new environment settings may lead to catastrophic forgetting of previous environment settings. To address this problem, our proposed system LICAR uses the simulated meta CSI training data from the CSI augmentation generator to reduce the distribution difference between each set of CSI training data. Furthermore, it has a parameter updating buffer. During the optimization for the new settings, the buffer keeps the knowledge for the previous CSI training tasks by updating different parameter selectively.

7.2 Future Works

The CSI-based human sensing systems presented in this thesis advance CSI-based smart human sensing technologies. The decent performances of these methods motivate us to step up efforts to conduct in-depth research. Based on the research work in this thesis, there are some possible directions for future works:

- **CSI-based smart sensing via a limited amount of CSI data** Many CSI-based sensing systems construct their system model by taking advantages of deep learning technology. It provides the CSI based sensing system with strong fitting ability and decent performances. However, to train these kinds of system, a large number of CSI data is required, which increase the system cost and training time. In our works, we manage to address this problem using few-shot learning technology. The CSI based user authentication system CAUTION is able to construct its system model using a limited amount of CSI data. Besides, we also address this problem by taking advantages of generative adversarial network in MCBAR. It generate large amounts of fake CSI data to augment the training CSI data. In order to achieve good performances, an adequate amount of CSI data are still required. For future works, we want to explore constructing a well-performed CSI sensing system with very a limited amount of CSI data. We may train a generalized CSI based sensing system with CSI data from different environments, then it can be trained to adapt to the target environment with a limited amount of CSI data.

- **Automatic Evolving CSI based Smart Sensing** When a CSI-based smart sensing system is in the deployed environment, it must have the ability to adapt to the ever-changing environments. We have studied the issue of life-long learning in Chapter 6, however, there are still some other challenges to be addressed. The process of retraining needs to be activated automatically. Systems should be aware of the significant changes in the surroundings and optimize themselves automatically after they are deployed. Most existing methods activate the process based on some empirical threshold that is not advanced and intelligent enough. They measure the difference between the received CSI data and the training CSI data, and set an empirical threshold

for the difference [63]. To set a proper value of the threshold, it requires lots of CSI data. Besides, it can be highly different within different environments.

- **Privacy Protected CSI-based Smart Human Sensing** Privacy preserving is an important issue for CSI based human sensing systems. CSI-based smart sensing techniques have provided better privacy protection than some other sensing techniques. For example, it does not require the installation of cameras compared to vision-based sensing techniques. However, the training of the CSI-based sensing systems normally requires large numbers of CSI data collected from users, which actually raises some privacy threats to system users. The source-free technique which is a new topic in the deep learning research field aims to reduce the probability to track the training data from the system. In [114], they leverage a pretrained model from the source domain and progressively update the target model in a self-learning manner. In [115], a unified adaptation algorithm is proposed. It is capable of operating across a wide range of category-gaps without any access to the previously seen source samples. We can further study to decrease the amount or remove the need for user CSI data by constructing source free CSI-based sensing systems. By this means, we enable the systems with the user privacy protection ability.

List of Author's Publications[1]

Journal Articles

(A) Journal papers that are included in the thesis

- **D. Wang**, J. Yang, W. Cui, L. Xie and S. Sun, "Multimodal CSI-Based Human Activity Recognition Using GANs," in *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17345-17355, 15 Dec.15, 2021, doi: 10.1109/JIOT.2021.3080401.
- **D. Wang**, J. Yang, W. Cui, L. Xie and S. Sun, "CAUTION: A Robust WiFi-based Human Authentication System via Few-shot Open-set Gait Recognition," in *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2022.3156099.
- **D. Wang**, J. Yang, W. Cui, L. Xie and S. Sun, "AirFi: Empowering WiFi-based Passive Human Gesture Recognition to Unseen Environment via Domain Generalization," *IEEE Transactions of Mobile Computing*, under review.

(B) Journal papers that are not included in the thesis

- J. Yang, X. Chen, H. Zou, **D. Wang**, Q. Xu and L. Xie, "EfficientFi: Toward Large-Scale Lightweight WiFi Sensing via CSI Compression," in *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13086-13095, Aug.1, 2022, doi: 10.1109/JIOT.2021.3139958.
- J. Yang, X. Chen, H. Zou, **D. Wang** and Lihua Xie. "AutoFi: Towards Automatic WiFi Human Sensing via Geometric Self-Supervised Learning". *arXiv preprint arXiv:2205.01629.(2022)*

[1]The superscript * indicates joint first authors

- J. Yang, X. Chen, **D. Wang**, H. Zou, Lu. CX, S. Sun and L. Xie. (2022). Deep Learning and Its Applications to WiFi Human Sensing: A Benchmark and A Tutorial. *arXiv preprint arXiv:2207.07859*.

Conference Proceedings

- **D. Wang**, J. Yang, W. Cui, L. Xie and S. Sun, "Robust CSI-based Human Activity Recognition using Roaming Generator," *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2020, pp. 1329-1334, doi: 10.1109/ICARCV50220.2020.9305332.

Bibliography

[1] Junjing Yang, Mattheos Santamouris, and Siew Eang Lee. Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings. *Energy and Buildings*, 121:344–349, 2016. 1

[2] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016. 1

[3] Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S Jaakkola, and Matt T Bianchi. Learning sleep stages from radio signals: A conditional adversarial architecture. In *International Conference on Machine Learning*, pages 4100–4109. PMLR, 2017. 1

[4] Zhenghua Chen, Qingchang Zhu, Yeng Chai Soh, and Le Zhang. Robust human activity recognition using smartphone sensors via ct-pca and online svm. *IEEE transactions on industrial informatics*, 13(6):3070–3080, 2017. 1

[5] Daqing Zhang, Hao Wang, Yasha Wang, and Junyi Ma. Anti-fall: A non-intrusive and real-time fall detector leveraging csi from commodity wifi devices. In *International Conference on Smart Homes and Health Telematics*, pages 181–193. Springer, 2015. 2, 9

[6] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. Csi-based fingerprinting for indoor localization: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 66(1):763–776, 2016. 9

[7] Hao Wang, Daqing Zhang, Yasha Wang, Junyi Ma, Yuxiang Wang, and Shengjie Li. Rt-fall: A real-time and contactless fall detection system with commodity wifi devices. *IEEE Transactions on Mobile Computing*, 16(2): 511–526, 2016. 2, 9

[8] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. *ACM SIGCOMM Computer Communication Review*, 40(4):159–170, 2010. 2, 10, 11

[9] Hefei Hu and Luxi Li. A new method using covariance eigenvalues and time window in passive human motion detection based on csi phases. In *2017 IEEE 5th International Symposium on Electromagnetic Compatibility (EMC-Beijing)*, pages 1–6. IEEE, 2017.

[10] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review*, 41(1):53–53, 2011. 2, 10, 45

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 2

[12] Miho Takayanagi, Osamu Fukuda, Nobuhiko Yamaguchi, Hiroshi Okumura, and Anik Nur Handayani. Vision-based scene recognition for product search. In *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, pages 1–5. IEEE, 2021. 8

[13] K Srinivasan and VR Azhaguramyaa. Internet of things (iot) based object recognition technologies. In *2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 216–220. IEEE, 2019. 8

[14] Ting-Fung Ju, Wei-Min Lu, Kuan-Hung Chen, and Jiun-In Guo. Vision-based moving objects detection for intelligent automobiles and a robustness enhancing method. In *2014 IEEE International Conference on Consumer Electronics-Taiwan*, pages 75–76. IEEE, 2014. 8

[15] Narayana Darapaneni, CM Sunilkumar, Mukul Paroha, Anwesh Reddy Paduri, Rohit George Mathew, Namith Maroli, and Rohit Eknath Sawant. Object detection of furniture and home goods using advanced computer vision. In *2022 Interdisciplinary Research in Technology and Management (IRTM)*, pages 1–5. IEEE, 2022. 8

[16] Pierluigi Casale, Oriol Pujol, and Petia Radeva. Human activity recognition from accelerometer data using a wearable device. In *Iberian conference on pattern recognition and image analysis*, pages 289–296. Springer, 2011. 8

[17] Yunyoung Nam, Seungmin Rho, and Chulung Lee. Physical activity recognition using multiple sensors embedded in a wearable device. *ACM Transactions on Embedded Computing Systems (TECS)*, 12(2):1–14, 2013. 8

[18] Grant Schindler, Christian Metzger, and Thad Starner. A wearable interface for topological mapping and localization in indoor environments. In *International Symposium on Location-and Context-Awareness*, pages 64–73. Springer, 2006. 8

[19] Quanzhe Li, SaeByuk Shin, Chung-Pyo Hong, and Shin-Dug Kim. On-body wearable device localization with a fast and memory efficient svm-knn using gpus. *Pattern Recognition Letters*, 139:128–138, 2020. 9

[20] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 27–38, 2013. 9, 18

[21] Qirong Bu, Gang Yang, Xingxia Ming, Tuo Zhang, Jun Feng, and Jing Zhang. Deep transfer learning for gesture recognition with wifi signals. *Personal and Ubiquitous Computing*, pages 1–12, 2020. 9, 18

[22] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *2015 IEEE conference on computer communications (INFOCOM)*, pages 1472–1480. IEEE, 2015. 9

[23] Xiaolong Zheng, Jiliang Wang, Longfei Shangguan, Zimu Zhou, and Yunhao Liu. Smokey: Ubiquitous smoking detection with commercial wifi infrastructures. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016. 9

[24] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. Keystroke recognition using wifi signals. In *Proceedings of the 21st annual international conference on mobile computing and networking*, pages 90–102, 2015. 9

[25] Yu Gu, Fuji Ren, and Jie Li. Paws: Passive human activity recognition based on wifi ambient signals. *IEEE Internet of Things Journal*, 3(5):796–805, 2015. 10

[26] Zheng Yang, Zimu Zhou, and Yunhao Liu. From rssi to csi: Indoor localization via channel response. *ACM Computing Surveys (CSUR)*, 46(2):1–32, 2013. 10

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998. 12

[28] Wang Zhiqiang and Liu Jun. A review of object detection based on convolutional neural network. In *2017 36th Chinese control conference (CCC)*, pages 11104–11109. IEEE, 2017. 12

[29] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014. 12

[30] Minh Tu Hoang, Brosnan Yuen, Kai Ren, Xiaodai Dong, Tao Lu, Robert Westendorp, and Kishore Reddy. A CNN-LSTM quantifier for single access point csi indoor localization. *arXiv preprint arXiv:2005.06394*, 2020. 12

[31] Yong Zhang, Chen Qu, and Yujie Wang. An indoor positioning method based on csi by using features optimization mechanism with lstm. *IEEE Sensors Journal*, 20(9):4868–4878, 2020. 12

[32] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016. 12

[33] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain Adaptation via Transfer Component Analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010. 12, 13

[34] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial Discriminative Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017. 13, 15

[35] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27, 2014. 13, 15, 52

[36] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 13

[37] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *Advances in neural information processing systems*, 30, 2017.

[38] Hajar Emami, Majid Moradi Aliabadi, Ming Dong, and Ratna Babu Chinnam. Spa-gan: Spatial attention gan for image-to-image translation. *IEEE Transactions on Multimedia*, 23:391–401, 2020.

[39] Feng Xiong, Qianqian Wang, and Quanxue Gao. Consistent embedded gan for image-to-image translation. *IEEE Access*, 7:126651–126661, 2019. 13

[40] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. *Advances in neural information processing systems*, 30, 2017. 13

[41] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018. 15, 28, 51

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[43] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017.

[44] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised Image-to-Image Translation Networks. *Advances in Neural Information Processing Systems*, 30, 2017. 13, 15

[45] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014. 13

[46] Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki, Jingdong Wang, and Xian-Sheng Hua. Global versus localized generative adversarial nets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1517–1525, 2018. 13

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 13

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 39

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 13

[50] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. *Advances in neural information processing systems*, 17:513–520, 2004. 14

[51] Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, pages 412–419. PMLR, 2007. 14

[52] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009. 14

[53] Mei Wang and Weihong Deng. Deep Visual Domain Adaptation: A Survey. *Neurocomputing*, 312:135–153, 2018. 14

[54] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv preprint arXiv:1412.3474*, 2014. 15

[55] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning Transferable Features with Deep Adaptation Networks. In *International Conference on Machine Learning*, pages 97–105. PMLR, 2015. 15

[56] Ming-Yu Liu and Oncel Tuzel. Coupled Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, 29, 2016. 15

[57] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain Generalization with Adversarial Feature Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018. 15, 51, 52, 55

[58] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain Generalization with Mixstyle. *arXiv preprint arXiv:2104.02008*, 2021. 15

[59] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from Several Related Classification Tasks to A New Unlabeled Sample. *Advances in Neural Information Processing Systems*, 24, 2011. 15

[60] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the Damage of Dataset Bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012. 15

[61] Yujia Li, Kevin Swersky, and Rich Zemel. Generative Moment Matching Networks. In *International Conference on Machine Learning*, pages 1718–1727. PMLR, 2015. 15

[62] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep Domain Generalization via Conditional Invariant Adversarial Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018. 15

[63] Qinyi Xu, Yi Han, Beibei Wang, Min Wu, and KJ Ray Liu. Indoor events monitoring using channel state information time series. *IEEE Internet of Things Journal*, 6(3):4977–4990, 2019. 16, 19, 89, 90, 91, 99, 100, 101, 106

[64] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 17

[65] Irene Muñoz-Martín, Stefano Bianchi, Giacomo Pedretti, Octavian Melnic, Stefano Ambrogio, and Daniele Ielmini. Unsupervised learning to overcome catastrophic forgetting in neural networks. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 5(1):58–66, 2019. 17

[66] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pages 3925–3934. PMLR, 2019. 17

[67] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017. 17

[68] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 17

[69] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526, 2017. 17, 94

[70] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3409, 2017. 17

[71] Daqing Zhang, Hao Wang, and Dan Wu. Toward centimeter-scale human activity sensing with wi-fi signals. *Computer*, 50(1):48–57, 2017. 18

[72] Hao Wang, Daqing Zhang, Junyi Ma, Yasha Wang, Yuxiang Wang, Dan Wu, Tao Gu, and Bing Xie. Human respiration detection with commodity wifi devices: do user location and body orientation matter? In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 25–36, 2016. 18

[73] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st annual international conference on mobile computing and networking*, pages 65–76, 2015. 18

[74] Jinyang Huang, Bin Liu, Hongxin Jin, and Zhiqiang Liu. Wianti: an anti-interference activity recognition system based on wifi csi. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 58–65. IEEE, 2018. 18

[75] Yanan Li, Ting Jiang, Xue Ding, and YangYang Wang. Location-free csi based activity recognition with angle difference of arrival. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2020. 18

[76] Chi Lin, Jiaye Hu, Yu Sun, Fenglong Ma, Lei Wang, and Guowei Wu. Wiau: An accurate device-free authentication system with resnet. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2018. 18, 19, 20, 67, 69, 79

[77] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. Crosssense: Towards cross-site and large-scale wifi sensing. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 305–320, 2018. 18, 23, 25, 39, 47, 49

[78] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 617–628, 2014. 19

[79] Chunjing Xiao, Daojun Han, Yongsen Ma, and Zhiguang Qin. Csigan: Robust channel state information-based activity recognition with gans. *IEEE Internet of Things Journal*, 6(6):10191–10204, 2019. 19, 23, 25, 39, 47, 49, 51, 89, 90, 91, 93, 99, 100, 101

[80] Jianfei Yang, Han Zou, Shuxin Cao, Zhenghua Chen, and Lihua Xie. Mobileda: Toward edge-domain adaptation. *IEEE Internet of Things Journal*, 7(8):6909–6918, 2020. 19

[81] Jianfei Yang, Han Zou, Yuxun Zhou, and Lihua Xie. Learning gestures from wifi: A siamese recurrent convolutional architecture. *IEEE Internet of Things Journal*, 6(6):10763–10772, 2019. 19

[82] Chi Su, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Deep attributes driven multi-camera person re-identification. In *European conference on computer vision*, pages 475–491. Springer, 2016. 20

[83] Seokho Chi and Carlos H Caldas. Automated object identification using optical video cameras on construction sites. *Computer-Aided Civil and Infrastructure Engineering*, 26(5):368–380, 2011. 20

[84] Lei Xie, Bo Sheng, Chiu C Tan, Hao Han, Qun Li, and Daoxu Chen. Efficient tag identification in mobile rfid systems. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. IEEE, 2010. 20

[85] Ju Wang, Jie Xiong, Xiaojiang Chen, Hongbo Jiang, Rajesh Krishna Balan, and Dingyi Fang. Tagscan: Simultaneous target imaging and material identification with commodity rfid devices. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 288–300, 2017. 20

[86] Feng Hong, Xiang Wang, Yanni Yang, Yuan Zong, Yuliang Zhang, and Zhongwen Guo. Wfid: Passive device-free human identification using wifi signal. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 47–56, 2016. 20

[87] Tong Xin, Bin Guo, Zhu Wang, Mingyang Li, Zhiwen Yu, and Xingshe Zhou. Freesense: Indoor human identification with wi-fi signals. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2016. 20

[88] Wei Wang, Alex X Liu, and Muhammad Shahzad. Gait recognition using wifi signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 363–373, 2016. 20

[89] Cong Shi, Jian Liu, Hongbo Liu, and Yingying Chen. Smart user authentication through actuation of daily activities leveraging wifi-enabled iot. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 1–10, 2017. 20

[90] Yunze Zeng, Parth H Pathak, and Prasant Mohapatra. Wiwho: Wifi-based person identification in smart spaces. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12. IEEE, 2016. 20

[91] Jin Zhang, Bo Wei, Fuxiang Wu, Limeng Dong, Wen Hu, Salil S Kanhere, Chengwen Luo, Shui Yu, and Jun Cheng. Gate-id: Wifi-based human identification irrespective of walking directions in smart home. *IEEE Internet of Things Journal*, 2020. 20, 77

[92] Vinoj Jayasundara, Hirunima Jayasekara, Tharaka Samarasinghe, and Kasun T Hemachandra. Device-free user authentication, activity classification and tracking using passive wi-fi sensing: A deep learning-based approach. *IEEE Sensors Journal*, 20(16):9329–9338, 2020. 20

[93] Ding Wang, Zhiyi Zhou, Xingda Yu, and Yangjie Cao. Csiid: Wifi-based human identification via deep learning. In *2019 14th International Conference on Computer Science & Education (ICCSE)*, pages 326–330. IEEE, 2019. 20, 67, 69, 77

[94] Xingxia Ming, Hongwei Feng, Qirong Bu, Jing Zhang, Gang Yang, and Tuo Zhang. Humanfi: Wifi-based human identification using recurrent neural network. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 640–647. IEEE, 2019. 20

[95] Jianyang Ding, Yong Wang, and Xiangcong Fu. Wihi: Wifi based human identity identification using deep learning. *IEEE Access*, 8:129246–129262, 2020. 21, 67, 69, 77, 79

[96] Akarsh Pokkunuru, Kalvik Jakkala, Arupjyoti Bhuyan, Pu Wang, and Zhi Sun. Neuralwave: Gait-based user identification through commodity wifi and deep learning. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 758–765. IEEE, 2018. 21

[97] Jianfei Yang, Han Zou, Hao Jiang, and Lihua Xie. Device-free occupant activity sensing using wifi-enabled iot devices for smart homes. *IEEE Internet of Things Journal*, 5(5):3991–4002, 2018. 32, 61, 76, 98

[98] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Device-free human activity recognition using commercial wifi devices. *IEEE Journal on Selected Areas in Communications*, 35(5):1118–1131, 2017. 42

[99] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. Signfi: Sign language recognition using wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):1–21, 2018. 44

[100] Sameera Palipana, David Rojas, Piyush Agrawal, and Dirk Pesch. Falldefi: Ubiquitous fall detection using commodity wi-fi devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4): 1–25, 2018. 44

[101] Dazhuo Wang, Jianfei Yang, Wei Cui, Lihua Xie, and Sumei Sun. Multimodal CSI-based Human Activity Recognition using GANs. *IEEE Internet of Things Journal*, 8(24):17345–17355, 2021. 47, 49, 50, 51, 59, 60, 61, 89, 90, 91, 92, 93, 99, 100, 101

[102] Xie Zhang, Chengpei Tang, Kang Yin, and Qingqian Ni. WiFi-based Cross-Domain Gesture Recognition via Modified Prototypical Networks. *IEEE Internet of Things Journal*, 2021. 47, 49, 59, 60, 61

[103] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial Autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 50

[104] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the Robustness of Deep Neural Networks via Stability Training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480–4488, 2016. 51, 93

[105] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A Simple Feature Augmentation for Domain Generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895, 2021. 53

[106] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert Space Embedding for Distributions. In *International Conference on Algorithmic Learning Theory*, pages 13–31. Springer, 2007. 55

[107] Qirong Bu, Gang Yang, Jun Feng, and Xingxia Ming. Wi-fi based Gesture Recognition using Deep Transfer Learning. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 590–595. IEEE, 2018. 59, 60, 61

[108] Chunhai Feng, Sheheryar Arshad, Siwang Zhou, Dun Cao, and Yonghe Liu. Wi-multi: A Three-phase System for Multiple Human Activity Recognition with Commercial WiFi Devices. *IEEE Internet of Things Journal*, 6(4): 7293–7304, 2019. 59, 60, 61

[109] Laurens Van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008. 65

[110] Pedro R Mendes Júnior, Roberto M De Souza, Rafael de O Werneck, Bernardo V Stein, Daniel V Pazinato, Waldir R de Almeida, Otávio AB Penatti, Ricardo da S Torres, and Anderson Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, 2017. 71, 73

[111] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017. 71

[112] Jie Wang, Yunong Zhao, Xinxin Fan, Qinghua Gao, Xiaorui Ma, and Hongyu Wang. Device-free identification using intrinsic csi features. *IEEE transactions on vehicular technology*, 67(9):8571–8581, 2018. 79

[113] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992. 95

[114] Youngeun Kim, Donghyeon Cho, Kyeongtak Han, Priyadarshini Panda, and Sungeun Hong. Domain adaptation without source data. *IEEE Transactions on Artificial Intelligence*, 2(6):508–518, 2021. 106

[115] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020. 106