A Deep Learning Framework for Blockage Mitigation in mmWave Wireless

by

Ahmed Hazaa Almutairi

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Computer Science

Dissertation Committee:
Ehsan Aryafar, Chair
Wu-Chi Feng
Alireza Keshavarz-Haddad
Ameeta Agrawal

Portland State University
2024

Abstract

Millimeter-Wave (mmWave) communication is a key technology to enable next generation wireless systems. However, mmWave systems are highly susceptible to blockages, which can lead to a substantial decrease in signal strength at the receiver. Identifying blockages and mitigating them is thus a key challenge to achieve next generation wireless technology goals, such as enhanced mobile broadband (eMBB) and Ultra-Reliable and Low-Latency Communication (URLLC). This thesis proposes several deep learning (DL) frameworks for mmWave wireless blockage detection, mitigation, and duration prediction. First, we propose a DL framework to address the problem of identifying whether the mmWave wireless channel between two devices (e.g., a base station and a client device) is Line-of-Sight (LoS) or non-Line-of-Sight (nLoS). Specifically, we show that existing beamforming training messages that are exchanged periodically between mmWave wireless devices can also be used in a DL model to solve the channel classification problem with no additional overhead. We extend this DL framework by developing a transfer learning model (t-LNCC) that is trained on simulated data and can successfully solve the channel classification problem on any commercial-off-

the-shelf (COTS) mmWave device with/without any real-world labeled data. The second part of the thesis leverages our channel classification mechanism from the first part and introduces new DL frameworks to mitigate the negative impacts of blockages. Previous research on blockage mitigation has introduced several model and protocol based blockage mitigation solutions that focus on one technique at a time, such as handoff to a different base station or beam adaptation to the same base station. We go beyond those techniques by proposing DL frameworks that address the overarching problem: *what blockage mitigation method should be employed?* and *what is the optimal sub-selection within that method?* To do so, we developed two Gated Recurrent Unit (GRU) models that are trained using periodically exchanged messages in mmWave systems. Specifically, we first developed a GRU model that tackled the blockage mitigation problem in single-antenna clients wireless environment. Then, we proposed another GRU model to expand our investigation to cover more complex scenarios where both base stations and clients are equipped with multiple antennas and collaboratively mitigate blockages. Those two models are trained on datasets that are gathered using a commercially available mmWave simulator. Both models achieve outstanding results in selecting the optimal blockage mitigation method with an accuracy higher than 93% and 91% for single-antenna and multiple-antenna clients, respectively. We also show that the proposed methods significantly increases the amount of transferred data compared to several other blockage mitigation policies.

Dedicated to my mom, dad, and wife - Moneerah, Hazaa and Hanin

If there's one key lesson my PhD journey imparted, it's the art of thoughtful.

Acknowledgments

I would like to express my deepest appreciation to Prof. Ehsan Aryafar, my advisor, whose expertise and guidance were crucial in navigating the complexities of my Ph.D. research. His perspective in tackling significant challenges in our field have profoundly influenced my academic development. Throughout my doctoral studies, Prof. Aryafar's empathy and support during both personal and academic challenges were key to my success. His mentorship has been invaluable in refining my research skills, leading to meaningful outcomes and presentations at renowned conferences. His teachings have become fundamental to my professional path ahead. The journey under Prof. Aryafar's guidance has been both memorable and deeply transformative, playing a significant role in my academic and professional growth.

My sincere thanks go to Prof. Alireza Keshavarz-Haddad for his consistent guidance and support, which significantly contributed to the progress of our research goals. I am also deeply grateful to my Ph.D. committee members, Prof. Wu-Chi Feng and Prof. Ameeta Agrawal, for their insightful feedback and advice. Their availability and valuable counsel throughout my Ph.D. studies were crucial.

I extend my heartfelt thanks to Prof. Mark P. Jones, the department chair, whose support was pivotal in my Ph.D. experience. His backing was essential in realizing my doctoral ambitions.

I owe a tremendous debt of gratitude to my family - my mom, Moneerah, my father, Hazaa, and my wife, Hanin, for their constant and unwavering support. Their encouragement and belief in me were indispensable in completing my Ph.D. journey.

Table of Contents

Abstract **i**

Dedication **iii**

Acknowledgments **iv**

List of Tables **x**

List of Figures **xv**

1 Introduction **1**

1.1 Evolution of Wireless Communication 5

1.2 MmWave and IEEE 802.11ad/ay 7

1.3 Thesis Overview 10

2 Deep Transfer Learning for Cross-Device Channel Classification in mmWave Wireless **11**

2.1 Line of Sight - non-Line of Sight Channel Classification 11

2.1.1 Base-LNCC 11

2.1.2 T-LNCC 13

2.1.3 Applications of MmWave LoS/nLoS Channel Classification 16
2.2 Performance Evaluation 18
2.2.1 Base-LNCC 19
2.2.2 T-LNCC 20
2.2.3 Proximity Evaluation 27

3 Gated Recurrent Units for Blockage Mitigation and Duration Prediction in mmWave Wireless 30
3.1 GRUs for mmWave Blockage Mitigation and Duration Prediction . 30
3.1.1 Gated Recurrent Units (GRUs) 30
3.1.2 System Model 31
3.1.3 Blockage Mitigation and Duration Prediction 33
3.2 Blockage Mitigation Techniques as Policies 40
3.3 Theoretical Model on Average Loss Ratio 44
3.4 Performance Evaluation 46
3.4.1 Measurement Campaign 46
3.4.2 Results 53

4 Related Work 70
4.1 Deep Transfer Learning for Cross-Device Channel Classification in mmWave Wireless 70
4.2 Gated Recurrent Units for Blockage Mitigation and Duration Prediction in mmWave Wireless 72
4.2.1 MmWave Blockages 72

4.2.2 Methods to Mitigate Blockages 73
4.2.3 Blockage Timing Estimation 75

5 Limitations and Future Work **77**

5.1 Deep Transfer Learning for Cross-Device Channel Classification in mmWave Wireless 77
5.2 Gated Recurrent Units for Blockage Mitigation and Duration Prediction in mmWave Wireless 78

6 Conclusion **80**

Bibliography **82**

List of Tables

2.1 Accuracy of source only, unsupervised, and semi-supervised experiments using simulated data as the source domain and Travelmate, AD7200, combined, and ASUS-RoG as the target domains. 25

2.2 Accuracy of TrAdaBoost, TCA, and t-LNCC approaches experiments using simulated data as source domain and Travelmate, AD7200, combined, and ASUS-RoG as target domain. 26

3.1 Blockage Mitigation and Duration Prediction GRU Models Structure and Configurations. 38

3.2 Simulation Setup. 48

3.3 Antenna size configurations of the multiple-antenna client network (MACN) environment. 52

3.4 Average throughput of the single-antenna client across all clients, BSs, and blockage events achieved by different ML models. 58

3.5 Comparison of SACN framework against three alternative policies (policies 2, 3, 4) in terms of average percentage drop in the amount of transferred data. 59

3.6 MACN-based GRU model accuracy for different antennas size configurations. 61
3.7 Comparison of MACN-based GRU model against policies 2, 3, 4 in terms of average percentage drop in the amount of transferred data. 66
3.8 Analysis of Upper Bound Loss, Total Data, and Average Loss Ratio. PUB stands for Predicted Upper Bound based on our model in Section 3.3. 68

List of Figures

1.1 (a): During the beacon transmission interval (BTI), BS sequentially sends sector sweep (SSW) frames on each of its sectors. Each client uses an omni antenna pattern and records the beam ID and signal strength of all the received SSW frames; (b): Association beamforming training (A-BFT) is composed of a few slots. A client randomly chooses an A-BFT slot to conduct its sector sweep. . . . 9

2.1 t-LNCC is a domain-adversarial neural network, which consists of three parts. Blue boxes show the features extractor layers, the orange boxes show the classifier layers, and the green boxes show the domain discriminator layers in addition to the GRL layer which is in gray color. Each of the boxes shows the number of units that layer has along with the activation function. 14

2.2 Measured SNR in azimuth plane of sectors 13 and 22 [39]. 18

2.3 (a): We gather data leveraging Talon AD-7200 router, Acer TravelMate laptop, ASUS-RoG Phone and AD-7200 as client; (b): Sample SNR of RoG phone's LoS and nLoS beams versus sector ID; (c): Sample SNR of Travelmate beams versus sector ID. . . . 21

2.4 Curve fitting for LoS and nLoS condition based on SNR values with respect to the distance. We use curve fitting to create a linear approximation of the path loss formula. Note that the distance (x-axis) uses a logarithmic unit. 28

2.5 Average distance error of the LoS and NLoS channel conditions. . 29

3.1 Single-antenna client network (SACN) setup. (a): There are several beams to choose from at the BS while the client has only a single antenna with omni-directional pattern. Beam selection typically happens through a search process at the beginning of every communication interval; (b): BS has identified a proper beam for communication with the client. Here the two red paths capture the multi-path nature of the communication channel; (c): In beam switching, BS switches to a different beam when blockage happens; (d): In handoff, the network may change the BS serving the client; (e): In beam widening, BS widens its beam. Energy reaches the client through other paths. 35

3.2 Multiple-antenna collaborative network (MACN). (a): There are several beams to choose from at both the BS and the client. Beam selection typically happens through a search process at the beginning of every communication interval; (b): BS and client have identified proper beams for communication with each other. Here the two red paths capture the multi-path nature of the communication channel; (c): In BeSw_BeSw, BS and client switch to different beams when blockage happens; (d): In BeWi_BeSw, BS widens its beam and client switches to a different beam; (e): In BeSw_BeWi, BS switches to a different beam and client widens its beam; (f): In BeWi_BeWi, BS and client widen their beams. Energy reaches the client through other paths; (g): In Ho_BeSw, the network may change the BS serving the client and client switches to a different beam. 36

3.3 GRU model structure with four GRU layers. X_{t1} to X_{tN} are the time steps that are used as the input, while Y_1 to Y_N are the outputs of the model, corresponding to the probability of each class (blockage mitigation strategy). Blockage duration prediction uses the same architecture but with only one scalar value corresponding to the blockage duration. 39

3.4 IIoT scenario with six BSs on the two sides, clients, and different types of blockers. We included four different types of blockers: human, cart, truck, and pickup, which are the most commonly encountered blockers in a factory setting. 48

3.5 Single-antenna client network (SACN). (a): Fraction of samples in the dataset labeled with each action; (b): Fraction of samples labeled with each action for each blockage type; (c): Top-1, Top-2, and Top-3 accuracy results of our GRU-based blockage mitigation framework. The correct label is the predicted label in 93% of instances and is among the top three predictions for 99% of instances. 50

3.6 Multiple-antenna client network (MACN). (a): Fraction of samples in the dataset labeled with each action; (b): Fraction of samples labeled with each action for each blockage type; (c): Average Top-1, Top-2, and Top-3 accuracy results (across the different array sizes) of our GRU-based blockage mitigation framework. The correct label is the predicted label in 91% of instances and is among the top three predictions for 97% of instances. 51

3.7 Single-antenna client network (SACN). (a): Accuracy of different ML models. GRU achieves the highest accuracy; (b): Increase in total transferred data for different actions. Here BeamS, BeamW, and HF denote beam switching, beam widening, and handoff, respectively; (c): The average drop in throughput when selected action is not optimal. Y-axis shows the true label and x-axis shows the incorrect predicted label. When the true and predicted label are the same, the selected sub-action is not optimal. The overall average drop in throughput across all actions is 92%. 54

3.8 The residual results represent the disparity between the actual and predicted blockage duration across the four array configuration scenarios: (a) BS: 8 - Client: 4; (b) BS: 8 - Client: 8; (c) BS: 16 - Client: 4, and (d) BS: 16 - Client: 8. 62

3.9 Multiple-antenna client network (MACN). (a): Accuracy of different ML models. GRU achieves the highest accuracy; (b): Increase in total transferred data for different actions. (c): Average throughput across all clients, BSs, and block- age events achieved by different ML models. 65

Chapter 1

Introduction

MmWave communication is one of the essential components of next-generation wireless networks to support extremely high data rate services. The mmWave frequency bands provide an order of magnitude more spectrum than already congested sub-6 GHz bands and associated wireless technologies (sub-6 GHz WiFi, LTE), which can boost communication capacity. The mmWave frequency is a key technology to provide very high data rates in a variety of applications such as Industrial Internet of Things (IIoT) [1]–[15]. As a result, many cellular and wireless LAN operators (such as AT&T, Verizon, smaller/private WiFi and 5G network providers) have begun deploying mmWave wireless networks.

However, mmWave systems suffer from high path loss and high noise power. To address these challenges, mmWave systems use an array of antennas and form highly directional beams[1] at both the transmitter (Tx) and receiver (Rx) to increase the signal-to-noise ratio (SNR). These directional (narrow) beams also reduce the interference, boost the capacity, and increase the security of communication. However, before a Tx and Rx can communicate with each other and take advantage

[1] **We use the words “beams" and “sectors" interchangeably.**

of the aforementioned benefits, they need to find appropriate beams to communicate with each other. In all existing mmWave standards, e.g., 5G New Radio (NR) or mmWave WiFi (i.e., 802.11 ad/ay) the beam search process happens **periodically** at the beginning of each communication interval (commonly referred to as beam training interval).

The other key challenge associated with mmWave communication is susceptibility to blockages, e.g., human body alone can block the signal and significantly reduce its strength at the receiver [16]–[20]. The existing mmWave standards currently employ a reactive approach when dealing with blockages in the communication path. In these standards, it may take several communication intervals before the selected beams are switched or the client is handed off to another base station (BS) to establish a new connection. This reactive behavior introduces additional latency and can have a detrimental effect on the overall data rates, thereby compromising the reliability of wireless communication.

However, by leveraging knowledge about the channel condition, specifically whether it is Line-of-Sight (LoS) or non-Line-of-Sight (nLoS), it becomes possible to significantly improve the performance of the communication system. Having information about the channel condition enables the implementation of proactive measures to mitigate the adverse effects of blockages. These measures may include actions such as switching beams, performing handoffs to alternative BS, or widening the beam to maintain a stable and reliable connection.

By actively monitoring the channel condition and making informed decisions based on this knowledge, it is possible to minimize the impact of blockages and

optimize the communication performance. For instance, when a blockage is detected or predicted, the system can promptly switch to an alternative beam that avoids the obstruction. Additionally, handoffs to nearby BS with better line-of-sight conditions can be initiated to ensure uninterrupted connectivity. Furthermore, by widening the beam, the system can adapt to changing channel conditions and maintain a reliable link even in the presence of temporary blockages.

The incorporation of these proactive measures based on the knowledge of the channel condition is crucial for improving the overall reliability and data rates of wireless communication in mmWave systems. By reducing latency, optimizing beam selection, and minimizing the effects of blockages, these measures enhance the performance and user experience in scenarios where blockages are encountered. Consequently, it is essential to develop techniques and models that can accurately identify the channel condition and enable proactive actions to mitigate the effects of blockages in mmWave communication systems.

Our goal in this thesis is to use the existing message passing overhead (at the beginning of each beam training interval) to solve the LoS/nLoS channel classification, blockage mitigation, and blockage duration prediction problems. For the LoS/nLoS channel classification, we aim to develop a framework that can solve this channel classification problem for any kind of wireless device or new mmWave devices as they are released. For example, consider an indoor mall scenario, which employs many mmWave (e.g., mmWave WiFi) BSs. As new devices (e.g., new mmWave equipped smartphones or augmented-reality glasses) are developed and released, the network operator can run a software update on all BSs, which allows

them to solve the channel classification problem for all such new client devices. The BSs can then use this information in other applications, e.g., to localize the clients and show them ads, determine the direction of the clients, or better adapt the links for clients in nLoS channel conditions.

Next, we aim to address the negative impact of blockages and predict their duration. For blockage mitigation, we focus on addressing the overarching problem: *from the plurality of blockage mitigation techniques, which one should be employed?* To address this problem, we first focus on three techniques: beam switching on the same BS, handoff, and beam widening in a wireless environment where clients employ single antenna and the blockage mitigation technique is taken by the BSs side only. Then, we explore more complex environment where both BSs and clients are equipped with multiple antennas and collaboratively mitigate blockages with five blockage mitigation techniques: beam switching on both the BS and client (BeSw_BeSw), beam switching on the BS and beam widening by the client (BeSw_BeWi), beam widening by the BS and beam switching by the client (BeWi_BeSw), beam widening on both the BS and client (BeWi_BeWi), and client handoff to a new BS, which results in new beam selections on both sides (Ho_BeSw). We also address the associated sub-problem within each technique, e.g., what new beam should be selected, which BS should the client handoff to, and how much to widen the beam. To achieve these goals, we develop frameworks that proactively take the appropriate action in order to minimize the impact of blockages. At its core, our frameworks use Gated Recurrent Units (GRUs), a newer generation of Recurrent Neural Networks (RNNs) suitable for learning sequential

data, and relies on periodic existing message passing in mmWave standards to decide on the appropriate action.

In addition, we address the blockage duration prediction problem. We aim to estimate the exact time interval of the blockage event. Specifically, we develop a framework that is trained on a sequence of historical dataset that covers various blockage events for different blockage types. The trained framework will efficiently predict the future similar blockage events durations and generalize well to new and unseen environments. The trained framework learns the patterns associated with different blockage types and their event durations. By predicting the blockage duration, we can estimate when the blockage event ends and hence the communication can revert back to the original settings (i.e., before applying the mitigation technique).

1.1 Evolution of Wireless Communication

The advancements of wireless communication have been characterized by a series of generations [21], [22]. Each of those brought new capabilities and addressed limitations of the preceding technologies. Here, we provide a brief overview of the key milestones in wireless technology, spanning from the early analog systems of the first generation (1G) to the cutting-edge capabilities of fifth-generation (5G) networks. We gain insight into the progression towards increasingly efficient, high-speed, and versatile wireless communication systems.

Wireless communication can be traced back to the first generation (1G) of cellular technology, which emerged in the 1980s. 1G systems primarily employed

analog transmission and provided basic voice services. However, they suffered from poor call quality, limited capacity, and susceptibility to interference. Protocols such as Advanced Mobile Phone System (AMPS) [23] were common during this era. Despite these limitations, 1G laid the foundation for further advancements in wireless communication.

The introduction of second-generation (2G) wireless technologies in the 1990s marked a significant evolution from analog to digital communication. Key protocols such as Global System for Mobile Communications (GSM) [24] and Code Division Multiple Access (CDMA) [25] offered improved voice quality, increased capacity, and enhanced security features. 2G systems also introduced text messaging (SMS) and basic data services. However, data rates were relatively low, making it unsuitable for high-speed internet access.

The transition to third-generation (3G) wireless technologies in the early 2000s aimed to address the growing demand for mobile data services. Protocols like Universal Mobile Telecommunications System (UMTS) and CDMA2000 [26] provided higher data rates, enabling internet access, video streaming, and multimedia applications on mobile devices. 3G networks offered significant improvements in speed and capacity compared to their predecessors. However, coverage gaps and signal penetration issues persisted, particularly in indoor environments and densely populated areas.

The emergence of fourth-generation (4G) wireless technologies, starting around 2010, brought about a paradigm shift in mobile communications [27]. Protocols such as Long-Term Evolution (LTE) and WiMAX delivered unprecedented data

speeds, low latency, and enhanced spectral efficiency [28]. 4G networks enabled seamless connectivity for bandwidth-intensive applications like HD video streaming, online gaming, and cloud computing. Despite these advancements, 4G networks faced challenges related to network congestion, spectrum scarcity, and uneven deployment in rural areas.

The ongoing transition to fifth-generation (5G) wireless technologies represents the latest milestone in the evolution of wireless communication [29]. 5G networks promise to deliver ultra-fast data rates, ultra-low latency, and massive connectivity, paving the way for transformative applications such as autonomous vehicles, remote surgery, and virtual reality. Key features of 5G include mmWave spectrum utilization, advanced antenna technologies, network slicing, and edge computing. While 5G holds immense potential, its deployment faces challenges such as low range of coverage and susceptibility to blockages.

1.2 MmWave and IEEE 802.11ad/ay

We train and evaluate our channel condition classification and blockages mitigation frameworks using beam training data (i.e., beams vs SNR values) gathered during the Initial Access (IA), in which the BS and client establish a connection. During the IA, the BS and clients perform a beam search to identify the best beam pair (i.e., highest SNR) that should be used for communication between each client and the BS [30]. This beam search process is conducted periodically in order to better accommodate variations in the environment and mobility of the clients. In this section, we focus on the basics of the beam search process in 802.11 ad/ay (i.e.,

mmWave WiFi). A similar process is used in 5G NR.

MmWave Beamforming. Beamforming focuses the wireless signal in a specific direction rather than spreading it in all directions, which helps combat large propagation and penetration losses in mmWave bands. An array of antennas are adopted for beamforming in mmWave systems to increase the link capacity and transmission coverage. Adjusting the phase of each signal transmitted from each antenna in the array of antennas steers the beam to a desired direction. In mmWave systems, the BS and client need to find the beam pair that results in the highest SNR for fastest communication. This procedure is shown for the BS in Fig. 1.1(a).

MmWave Initial Access. In mmWave systems, the initial access procedure serves multiple purposes. First, it enables a client device to discover a cell and establish a connection. Second, it assists BS and clients in identifying suitable beams for communication. Last, it allows the BS to transmit management and control information to all connected clients.

In 802.11 ad/ay, initial access is handled at the beginning of each beacon interval (BI) [31], [32]. The length of a BI is typically 100 ms, i.e., the BI is repeated every 100 ms. Our model can classify the channel condition (LoS vs nLoS) based on the data gathered at the beginning of every BI (referred to as beacon header interval). Therefore, the BS and each client can identify their channel condition every 100 ms. The BI is composed of two parts: (i) beacon header interval (BHI), which helps with BS discovery, beam training, and control and management information exchange, and (ii) the data transmission interval (DTI), which is used for data communication. The duration of BHI is much smaller

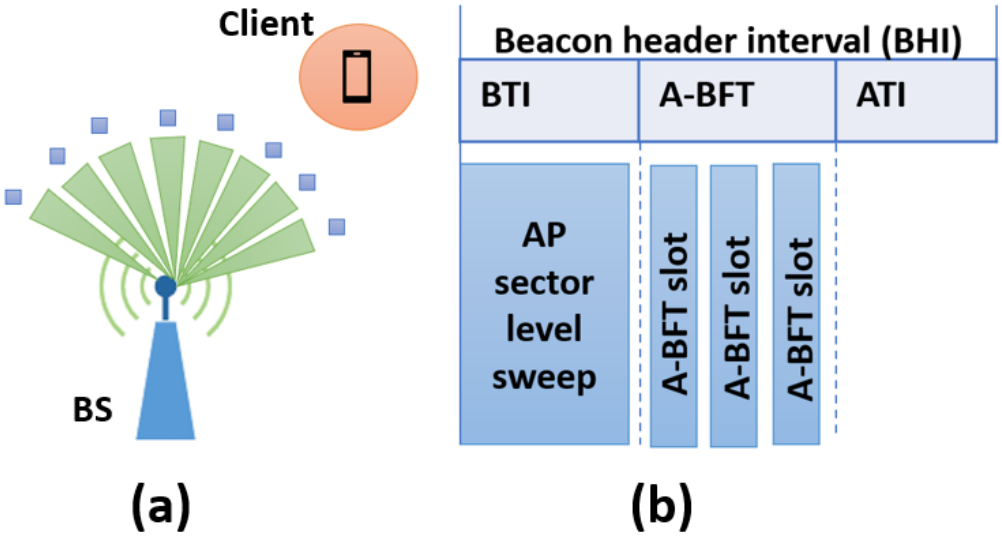


Figure 1.1: (a): During the beacon transmission interval (BTI), BS sequentially sends sector sweep (SSW) frames on each of its sectors. Each client uses an omni antenna pattern and records the beam ID and signal strength of all the received SSW frames; (b): Association beamforming training (A-BFT) is composed of a few slots. A client randomly chooses an A-BFT slot to conduct its sector sweep.

than the DTI, e.g., only a few msec.

The BHI, as depicted in Fig. 1.1(b), consists of three sub-intervals: (i) beacon transmission interval (BTI), in which the BS transmits multiple frames, each of them on a different sector, (ii) association beamforming training (A-BFT) in which the client devices train their sectors for communication with the BS, and (iii) announcement transmission interval (ATI) in which the BS exchanges management information with associated and beam-trained client devices.

In this thesis, we only consider the BTI sub-interval, which allows both frameworks to use the data exchanged during this interval to identify the channel condi-

tion (LoS vs nLoS) and take the right action to mitigate the impact of blockages.

Beacon Transmission Interval (BTI): The BTI comprises multiple beacon frames, each transmitted sequentially by the BS on a different BS sector (beam) to cover the desired directions. This process is referred to as BS sector sweep and is used for network announcement and beamforming training of the BS's sectors. During the BS sector sweep, all clients stay in reception mode using an omni (or quasi-omni) antenna pattern. Each client records the signal strength and beam ID of every sector sweep frame (SSW) received from the BS. Fig. 1.1(a) shows this operation. Our frameworks utilizes the measured SNR values of all received beams to identify the channel condition at the client device and mitigate blockages.

1.3 Thesis Overview

The rest of this thesis is organized as follows. In chapter 2, we present the design and evaluation of the deep transfer learning framework for line of sight and non line of sight channel classification. In chapter 3, we present the design and evaluation of the blockage mitigation and duration prediction frameworks. In chapter 4, related work is discussed on both the topics. In chapter 5, we discuss the limitations and future work. Finally, we conclude the thesis in chapter 6.

Chapter 2

Deep Transfer Learning for Cross-Device Channel Classification in mmWave Wireless

2.1 Line of Sight - non-Line of Sight Channel Classification

In this chapter, we propose a deep neural network-based LoS/nLoS Channel Classification (LNCC) model to classify the channel condition. We build LNCC in two stages. In the first stage, we design the *base-LNCC*, which is trained and evaluated using the simulated data. In the second stage, we extend our base-LNCC model to incorporate transfer learning functionalities leveraging the adversarial domain adaptation approach. We refer to this extended model as *t-LNCC*. We train t-LNCC on data from one domain (e.g., simulated data) and evaluate it using data from another domain (e.g., empirical data). We describe the two models as follows.

2.1.1 Base-LNCC

System Design. The wireless channel condition is categorized into two classes (LoS or nLoS). Thus, our classification problem is binary. DL shows high performance in tackling classification problems in different applications. Therefore, we developed a DL model, base-LNCC, to classify the mmWave wireless channel

condition during the first stage. Base-LNCC consists of six layers, including the visible layer (input layer), four hidden layers, and the output layer. The first two hidden layers consist of 256 neurons each, while the last two hidden layers have 128 neurons each. We use Rectified Linear Unit (ReLU) activation function in each of the hidden layers.

ReLU is a non-linear activation function that allows the model to converge quickly and perform a threshold wherein the case of the input value is less than 0, it is set to 0 (neuron will be deactivated); otherwise, a neuron will be activated [33]. ReLU is defined as follows:

$$S(z) = max(0, z) \tag{2.1}$$

Here, z is the input to the hidden layer. We use the sigmoid activation function for the output layer since our model is a binary classification (LoS and nLoS classes), and the probability is between 0 and 1. Sigmoid function is defined as follows:

$$S(z) = \frac{1}{1 + e^{z}} \tag{2.2}$$

For optimization, Adam, a gradient optimization technique widely used in DL [34], is used. Finally, the loss function that we choose is binary cross-entropy which works well with the sigmoid function:

$$L = -(Ylog(p(Y)) + (1-Y)log(1-p(Y))) \tag{2.3}$$

Here, Y is input's label and $p(Y)$ is the probability of the predicted label. Our model structure and parameters are selected and validated after conducting extensive experiments.

2.1.2 T-LNCC

DL involves complicated processes (e.g., collecting large amount of dataset samples) and requires many decisions that are not theoretically nor mathematically ruled (e.g., number of hidden layers). Also, training a model from scratch is complex and challenging (i.e., due to the lack of availability of a large labeled dataset). Transfer learning techniques allow transferring knowledge between similar domains. The idea is to train a model using the source domain dataset D_s to perform the source task T_s and transfer the pre-trained model knowledge (e.g., weights of hidden layers) to a related domain (target domain D_t) to perform a task (target task T_t). An alternative way for transfer learning is domain adaptation (DA). DA works on extracting similar features between source and target domains, so a classifier trained on the source domain can perform efficiently on the related target domain. Transfer learning approaches can be classified into four categories [35]: (i) instance-based transfer learning,(ii) mapping-based transfer learning, (iii) network transfer, and (iv) adversarial-based transfer learning.

In this thesis, we adopt adversarial learning for domain adaptation that finds

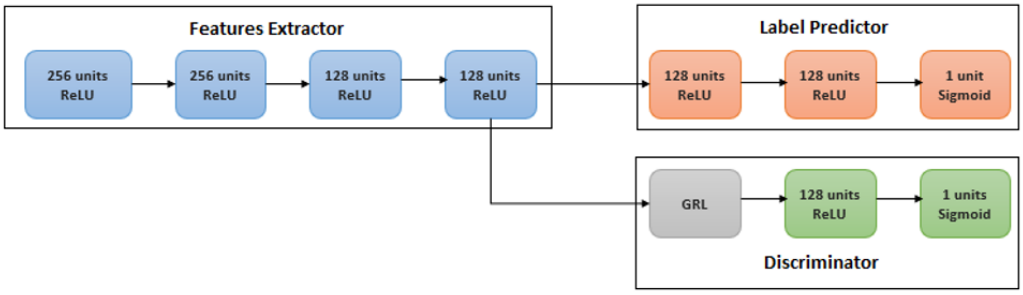


Figure 2.1: t-LNCC is a domain-adversarial neural network, which consists of three parts. Blue boxes show the features extractor layers, the orange boxes show the classifier layers, and the green boxes show the domain discriminator layers in addition to the GRL layer which is in gray color. Each of the boxes shows the number of units that layer has along with the activation function.

the transferable features of the two domains which makes them indistinguishable. Our key idea is to use a simulator calibrated with standardized channel model [36] to obtain a large amount of source data, and then use it (with/without labeled empirical data) to solve the channel classification problem with COTS devices.

In particular, we use an adversarial learning model, which trains a discriminator network to be unable to distinguish between the source and target domains. The model will learn the similarities or identical features of the two domains. This way, the model that works well with the source domain will also effectively work with the target domain. Two different adversarial domain adaptation learning methods could be used with transfer learning [37]: (i) unsupervised adversarial learning is used when labeled samples are available from D_s, but no labeled samples are available from the D_t; (ii) semi-supervised adversarial domain adaptation learning in which a few labeled target samples are provided along with the samples from D_s to train the model.

System Design. In this thesis, we employ a modified model of domain-adversarial neural network (DANN) used in [37] to solve image classification. Most domain adaptation approaches apply the domain adaptation and training processes separately. However, DANN merges domain adaptation into training processes so that the classifier (label predictor) is trained symmetrically with the discriminator.

DANN involves three training processes: (i) feature extractor, which consists of one or more hidden layers and the last hidden layer is the output layer. The output is the extracted features which are considered as an input to both label predictor and discriminator (domain predictor) parts; (ii) label predictor, which could also have one or more hidden layers and one output layer to predict the class label (e.g., LoS/nLoS); and (iii) discriminator, which is used to discriminate between the two domains, and it consists of one or more hidden layers and one output layer (e.g., simulated/empirical data).

DANN aims to extract very similar features between the source and target domains. In order to achieve this, gradient reversal layer (GRL) is inserted between the features extractor and the discriminator to flip the sign of gradient during the backpropagation. Then, we subtract the label predictor and discriminator gradients from each other. After that, we update the weights of feature extractor layers based on the result of gradients subtraction. This will ensure reaching a point where all the extracted features are domain-invariant. For more information, we refer readers to [37].

We extend DANN algorithm to have multiple hidden layers in each part. Also,

we modify the activation functions that shallow DANN uses to use ReLU for all hidden layers of the three parts. Instead of using softmax for the label predictor part, we set sigmoid to be the activation function since it is a binary classification. Finally, unlike DANN, we set binary cross-entropy loss for both label predictor and discriminator.

Our modified DANN model, which we call t-LNCC, consists of three parts, as depicted in Fig. 2.1. For the first part (features extractor), we use our base-LNCC model with minor changes, removing the output layer since we use this part to extract features (not to predict the label) and use SGD optimizer instead of Adam. The second part is the label predictor, which consists of two hidden layers, each with 128 neurons and the ReLU activation function, and the output layer applies the sigmoid activation function. The third part is the discriminator, which involves one hidden layer that has 128 neurons and applies the ReLU activation function, one GRL which is inserted between features extracted and the first hidden layer of the discriminator, and one output layer applies the sigmoid activation function. We set both label predictor (L_y) and discriminator (L_d) to use the binary cross-entropy loss for the loss function. Those parameters and structure are validated after extensive experiments.

2.1.3 Applications of MmWave LoS/nLoS Channel Classification

LoS/nLoS channel identification can be used in a wide variety of applications. In this section, we propose to use it as a solution to solve the proximity (distance) estimation between two nodes with mmWave radios. This information can then

be used by localization algorithms (e.g., triangulation methods) to localize the devices.

We emphasize that localization (particularly in indoor environments with no GPS signals) is a heavily studied topic, particularly with sub-6 GHz RF radios (for an example survey refer to [38]). Many of the techniques that are proposed for mmWave systems, assume access to sophisticated radios, ideal beam shapes, or reflective environments. For example, RF signal-based localization can be classified based on Angle of Arrival (AoA), Angle of Departure (AoD), Time of Arrival (ToA), and Received Signal Strength (RSS). Techniques that require AoA, AoD, or ToA require sophisticated radios (at least with multiple RF chains), which may not be available in COTS mmWave devices. For example, none of the three COTS devices that we had access to have more than a single RF chain. In addition, other works have used the direction of narrow mmWave beams as an approximation of the angular direction of the client. Practical mmWave beams, however, are very different. For example, Fig. 2.2 shows two out of the 32 sectors (beams) that are used by our mmWave radio equipped laptop as it conducts its beam search. Many of the beams have high beamforming gains in multiple directions, which makes it impractical to approximate the client angular direction with the direction of the main beam.

We propose to use a curve fitting method to approximate the proximity (distance) between two nodes, leveraging the path loss formula. Path loss describes the attenuation of the wireless signal strength in (db) with the distance, and it is a linear function of distance (in logarithmic domain), and can therefore, be

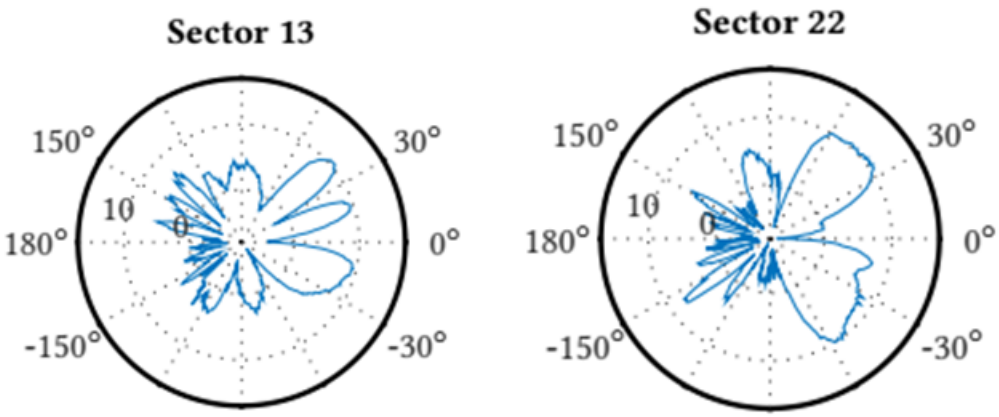


Figure 2.2: Measured SNR in azimuth plane of sectors 13 and 22 [39].

approximated as $aX + b$, where X varies as logarithm of the distance. Here, a is the path loss component, and b is an offset, capturing other components in the path loss formula. The a and b variables are a function of the channel condition (LoS or nLoS). We conduct path loss measurements with varying distance and use curve fitting methods to obtain these two variable for a given environment and for each of the two channel conditions. At run time, a BS or client can use the received signal strength values obtained during the sector sweep to first determine the LoS/nLoS channel condition, and then use the corresponding a and b parameters (along with the maximum beam SNR) to estimate the distance. We discuss this in more detail in Section 2.2.3.

2.2 Performance Evaluation

We evaluate our base-LNCC model using simulated data to find its channel classification accuracy. Then, we evaluate t-LNCC to evaluate the transfer learning performance with COTS devices. Also, we show the benefit of channel classifica-

tion is estimating proximity (distance). Our experiments show robust results with both base-LNCC when trained only with simulated data and transferring learning (domain adaptation) when trained with simulated data as the source domain and empirical data as the target domain.

2.2.1 Base-LNCC

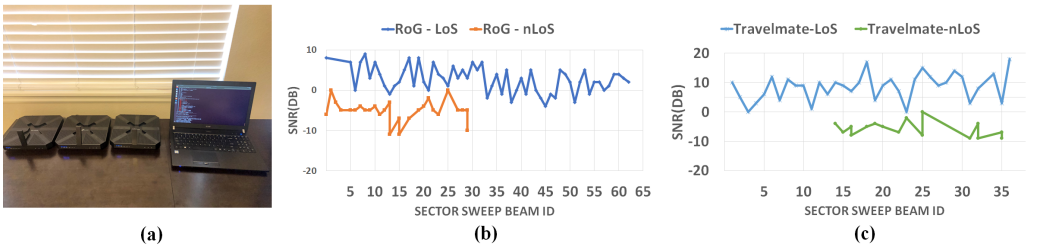
Simulator Setup and Data Gathering. We have developed a standard compatible mmWave simulator that adopts the channel and radio models standardized by the 3GPP [36]. The channel model is a statistical model. In our simulation, we consider a single BS and 20 clients. We set BSs to act as Tx with directional antenna while the clients are set to act as Rx with omnidirectional antenna. We consider three different antenna array sizes: 8, 16, and 32 antennas. The number of beams is assumed to be 36 beams that cover 120°. To gather large amount of dataset, we run our simulator for 4000 times. For each run, the location of both BS and clients, number of antennas, and the channel condition are randomly chosen. Randomization ensures collecting different samples in every run. We collect 20 samples in each run (total 80000 samples for the 4000 runs). Each sample is a vector of 72 elements which are the SNR values and their IDs. After cleaning (e.g., removing corrupted samples) and balancing data, we end up with 50000 samples (25000 LoS and 25000 nLoS). The same process is repeated to gather another 50000 samples where the BS is set to use 62 beams. Note that we will refer to the two datasets as 36 and 62 beams datatset, respectively.

Base-LNCC Evaluation. We train our base-LNCC model using the two simulated dataset (36 and 62 beams) separately. For training, we ran the model for 100 epochs using a batch of size 32. We use 10-fold Cross-Validation to test the performance of our model on simulated data. Our purpose of evaluating the base-LNCC model with simulated data before using it as part of the t-LNCC model is to test how accurate a deep learning baseline model is in classifying wireless channel condition and learning specific features of the simulated data, which will be used as source domain in the t-LNCC model.

Base-LNCC Results. Base-LNCC shows highly accurate results using only the beams' SNR values, which are obtained at the beginning of every beacon interval. We compare Base-LNCC against Gradient Boosting Decision Tree (GBDT) [40], which uses more data (Received signal strength, maximum path power, mean excess delay, RMS delay spread, maximum excess delay, kurtosis, skewness, and rise-time) for training their model. Base-LNCC has an accuracy of 98.5% while the GBDT has an accuracy of 97.9%.

2.2.2 T-LNCC

Experiment Setup and Data Gathering. We setup a single-cell mmWave network in both indoor and outdoor environments using a Talon AD-7200 router. We connect different clients to the BS in both LoS and nLoS channel conditions and gather BS beams' SNR values at different client locations in and around a single family home.



(a) (b) (c)

Figure 2.3: (a): We gather data leveraging Talon AD-7200 router, Acer TravelMate laptop, ASUS-RoG Phone and AD-7200 as client; (b): Sample SNR of RoG phone's LoS and nLoS beams versus sector ID; (c): Sample SNR of Travelmate beams versus sector ID.

Devices. We use a Talon-AD7200 as the BS and different clients such as Acer TravelMate laptop, AD-7200 router configured as client, and an ASUS RoG phone to gather beam SNR' connected to the BS. Fig. 2.3(a) shows a picture of our equipment. Travelmate and TP-Link Talon AD7200 use qualcomm QCA9500 IEEE 802.11 ad chipset, which uses a phase array antenna with 32 elements and 36 sectors (beams). ASUS RoG phone uses qualcomm QCA9500 IEEE802.11 ad chipset as well but uses a phase array antenna with 8 elements and 62 sectors. The default firmware for the AD-7200 router neither supports A-BFT SNR dump nor sniffer mode. To enable these features, we modified the default firmware using Nexmon framework [41] and installed that on the router to gather low-level signal statistics. This framework is a jailbreak into the 802.11 ad default firmware, which allows changing patches in C language instead of assembly, and it also provides new attributes and programs such as a GCC plugin. Talon AD-7200 routers with Nexmon firmware can be configured as either an BS or client [41]. In all of our experiments, we let the clients get connected to the BS with the modified firmware to measure their packets' signal strengths. This is because the default BS firmware

does not provide any signal statistics.

Single Family Home Environment We choose a stand alone single family home to gather the SNR data for each sector sweep beam. This home is a 3500 sq ft area with a master bedroom, a great room, and a backyard.

T-LNCC Configuration. We trained our adversarial learning model (t-LNCC) using both the simulated and empirical data. We used simulated data as the source domain while the empirical data (combined across client devices or separated) as the target domain. We conducted two main experiments: unsupervised adversarial domain adaptation learning and semi-supervised adversarial domain adaptation learning. We also conducted a third experiment in which we disabled the discriminator part, so no domain adaptation processes were involved (referred to as "source only"). To avoid confusion, there are two types of labels associated with each sample of the training dataset. First, the class label tells that the sample belongs to which channel condition class (LoS/nLoS). Second, the domain label tells that the sample is from which domain (e.g., source or target). The class labels are provided to label predictor, while the domain labels are provided to the discriminator during the training process. There are two exceptions for this: (i) During the source-only training, no domain labels are provided because we disable the discriminator part. (ii) During the unsupervised training, samples from the source domain are associated with both types of labels, while the target domain samples are only associated with domain labels. All the three experiments are run for 100 epochs using 32-batch. The three experiments are detailed as follows.

Source-Only. During this experiment, we disabled the discriminator part. So, the domain adaptation is not involved. The experiment is conducted using the whole source domain data for training (36 and 62 beams data) separately. Note that source domain data only includes the simulated data. The trained model is then evaluated on four empirical datasets: the combined (Travelmate and AD7200 data) dataset, only TravelMate dataset, only AD7200 dataset, and only ASUS-RoG. Since ASUS-RoG uses 62 beams, it is evaluated using the model trained on simulated data with 62 beams.

Unsupervised Adversarial Learning. For this experiment, we provide labeled samples (class label) of the source domain (i.e., simulated data) and unlabeled samples (no class label) of the target domain (i.e., empirical data) during the training. We also provide domain labels for both source and target domains' samples. The primary challenge with unsupervised adversarial learning is that the class labels of target domain data are unknown during training. The label predictor is trained on labeled samples from the source domain, and it must be able to predict the class labels of the samples from the target domain correctly. On the other hand, the domain discriminator is provided with domain labels for samples from both source and target domains. For the evaluation, we used unseen samples from the target domain. This experiment is conducted using the whole source domain data and 700 unlabeled samples of the Travelmate data for training. For evaluation, we used the rest (300 samples) of unseen Travelmate data. We repeated the same process for Ad7200 and ASUS-RoG. For the combined data, we used the whole source domain data and 1400 unlabeled samples of the combined (Travelmate and

AD7200) data for training, while for evaluation, the rest (600 samples) of unseen combined data were used. The choice of a smaller sample set from the target domain (e.g., Travelmate and AD7200) was intended to highlight the robustness of the transfer learning techniques employed, showing that even with limited target data, the model could achieve satisfactory performance.

Semi-supervised Adversarial Learning. To conduct this experiment, we used the whole labeled samples (class label) of the source domain data and a few labeled samples of the target domain data during the training. The domain labels of both source and target domains data are provided. For the evaluation, we use unseen labeled target domain data. Particularly, we followed the same process that we used in the unsupervised adversarial learning. However, instead of using unlabeled samples (class label) data from the target domain, we provided few labeled data from the target domain during training. First, we used the whole source domain data and 600 labeled samples of the Travelmate data for training, while for evaluation, the rest (400 samples) of unseen Travelmate data were used. We repeated the same process for Ad7200 and ASUS-RoG. For the combined data, we used the whole source domain data and 1500 labeled samples of the combined (Travelmate and AD7200) data for training, while for evaluation, the rest (500 samples) of unseen combined data were used.

T-LNCC results. Table 2.1 presents a summary of results. With source only learning (i.e., when the discriminator network is not used), we achieve a 82%-84% classification accuracy when we used AD7200, combined, and Travelmate data for evaluation. The unsupervised approach improves the accuracy for all the different

target domain data. This demonstrates the effectiveness of t-LNCC in extracting the similar features of the source and target domains data and efficiently classifies the target domain data even when no labeled target data is available. Travelmate data shows the best accuracy result among other devices' datasets. In both source-only and unsupervised approaches, the RoG phone shows poor performance with only 67% accuracy.

Using the semi-supervised approach, we reached the upper bound accuracy of the domain adaptation learning. Travelmate, AD7200, and the combined data show a noticeable increase in the accuracy. Further, ASUS-RoG achieves the highest accuracy compared to the other datasets with 97% accuracy.

Table 2.1: Accuracy of source only, unsupervised, and semi-supervised experiments using simulated data as the source domain and Travelmate, AD7200, combined, and ASUS-RoG as the target domains.

Method	Source	Simulated Data			
	Target	Travelmate	AD7200	Combined	ASUS-RoG
Source-Only		84.9%	82.53%	82.1%	61.9%
Unsupervised		93%	88.3%	90.23%	67%
Semi-supervised		96%	95.89%	95.35%	97%

T-LNCC Against Other Transfer Learning Models. In Section 2.1.2, we discussed four different transfer learning models, and proposed to adopt the adversarial learning technique. Other works have proposed alternative solutions for domain adaptation. TrAdaBoost [42] explores instance-based transfer learning by proposing a boosting-based learning algorithm. The framework drops out the source domain instances that are dissimilar to the target and re-weights the instances in the source domain so that the two domain distributions become similar.

Unlike unsupervised t-LNCC, TrAdaBoost requires a few labeled target data and a large amount of source domain data. TCA [43] investigate mapping-based transfer learning by bringing the two domains distributions as close to each other as possible by learning transfer components across the two domains using Maximum Mean Discrepancy (MMD). By taking each domain distribution to a new space where the new representations of the two domains are similar, training machine learning models using the source domain will best fit the target domain. This approach is called transfer components analysis (TCA). Note that both TrAdaBoost and TCA require labeled samples from target domain. They cannot be conducted using unsupervised learning whereas t-LNCC can be done without labeled samples from target domain. We compare t-LNCC against those two approaches to measure and validate our selection of adversarial learning. We use the same datasets and techniques that we used for evaluating t-LNCC. The results are shown in Table 2.2. T-LNCC outperforms both approaches with its unsupervised method. TCA accuracy is similar to unsupervised t-LNCC and outperforms unsupervised t-LNCC on the ASUS-RoG data. However, our semi-supervised technique outperforms both TrAdaBoost and TCA for all the four different datasets.

Table 2.2: Accuracy of TrAdaBoost, TCA, and t-LNCC approaches experiments using simulated data as source domain and Travelmate, AD7200, combined, and ASUS-RoG as target domain.

Method	Source	Simulated Data			
	Target	Travelmate	AD7200	Combined	ASUS-RoG
TrAdaBoost		85.1%	81.3%	82.1%	60.8%
TCA		90.24%	82.06%	86.76%	78%
T-LNCC (Unsupervised)		93%	88.3%	90.23%	67%
T-LNCC (Semi-supervised)		96%	95.89%	95.35%	97%

2.2.3 Proximity Evaluation

We next evaluate the effectiveness of our curve fitting solution, which uses the LoS/nLoS channel classification outcome along with a and b variables that were used in Section 2.1.3 to create a linear approximation of the path loss formula.

We setup a Talon-AD7200 as BS and a Travelmate laptop as the client. To find the a and b variables, we recorded the SNR of the client at the BS in an indoor environment by moving the client in the steps of 10 cm up to 16 m in both LoS and nLoS channel conditions. In LoS, the client had a direct line of sight whereas in nLoS we had used a human body to block the mmWave signal [17]. A typical SNR received at the BS is shown in Fig. 2.2. We take an average of the signal strength of the best five received beams at the BS. Once we gathered the SNR at the BS for both LoS and nLoS, we estimated the distance using path loss equation and curve fitting to generate a linear curve as shown in Fig. 2.4.

Note that once a client (BS) identifies its channel condition, it can use the appropriate linear coefficients (a and b values) along with the best received beam signal strength to approximate the distance to BS (client).

Finally, we evaluate the combined accuracy of using a linear approximation and our semi-supervised t-LNCC in determining proximity (distance). Fig. 2.5 shows the average estimated distance error for LoS and nLoS channel conditions for clients that are within 16 meters from the BS. Estimating the distance of clients that are in LoS condition and five meters or less from the BS is accurate with average distance error close to zero. The average distance error increases gradually

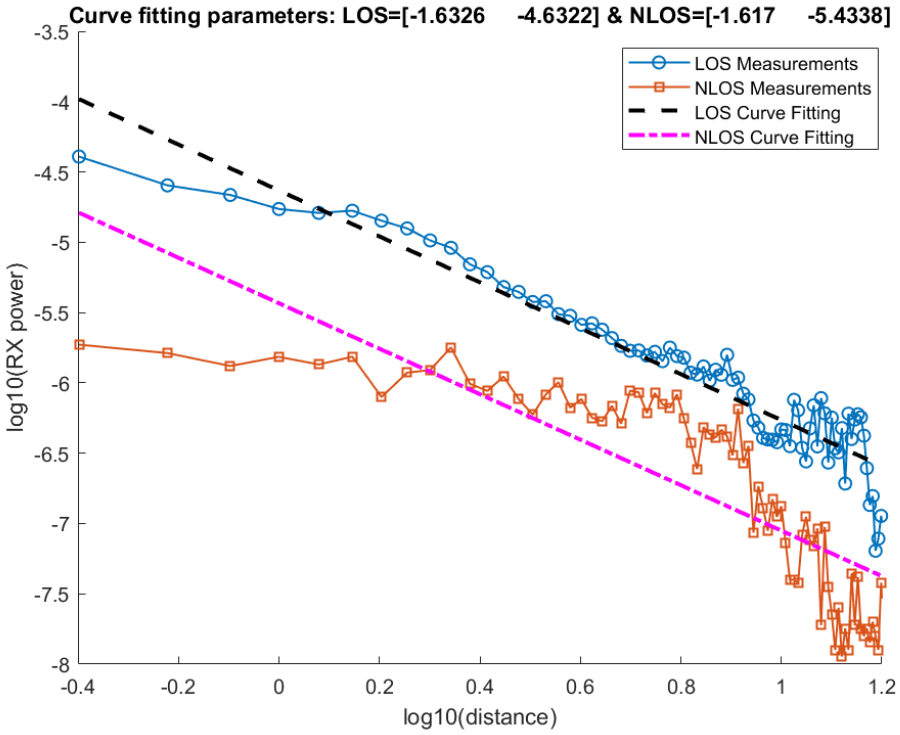


Figure 2.4: Curve fitting for LoS and nLoS condition based on SNR values with respect to the distance. We use curve fitting to create a linear approximation of the path loss formula. Note that the distance (x-axis) uses a logarithmic unit.

as the client gets further away from the BS. Clients that are thirteen meters from the BS have an average distance error of three meters. For the nLoS condition, error increases as a function of distance (similar to the LoS) and the approximated distance is less accurate than LoS. Average distance error for clients that are three meters from the BS is less than one meter. The clients that are eleven meters from the BS have an average distance error of three meters. The big jump in error happens for clients that are thirteen meters or more from the BS with an average error of 10 meters.

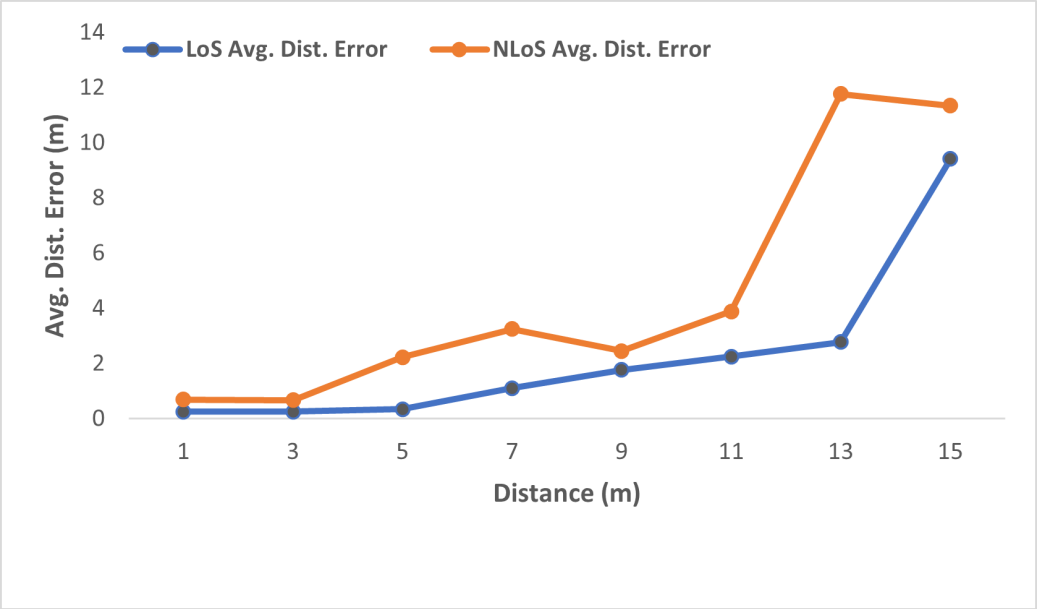


Figure 2.5: Average distance error of the LoS and NLoS channel conditions.

Chapter 3

Gated Recurrent Units for Blockage Mitigation and Duration Prediction in mmWave Wireless

3.1 GRUs for mmWave Blockage Mitigation and Duration Prediction

In this chapter, we present our GRU-based blockage mitigation and duration prediction frameworks. We first briefly discuss GRUs and their distinction from Long short-term memory (LSTM). Then, we introduce our system model for both the single-antenna client network scenario, where each client employs only one antenna with omni beam, and the more complex (Multiple-antenna collaborative network) scenario, where both BSs and clients employ phased array of multiple antennas. Further, we discuss using GRU models to mitigate blockages for both scenarios and predict blockages duration.

3.1.1 Gated Recurrent Units (GRUs)

GRU [44] is a type of RNN architecture that is commonly used for modeling sequential data. It was introduced as an alternative to the more popular LSTM architecture.

Like LSTM, GRU is designed to address the vanishing gradient problem that

can occur in traditional RNNs. The vanishing gradient problem refers to the issue where gradients can become extremely small as they propagate through the network, which can make it difficult for the machine learning (ML) model to learn long-term dependencies.

GRU accomplishes this by using gating mechanisms to selectively update and reset the hidden state of the network. Specifically, GRU has two gates: a reset gate and an update gate. The reset gate helps the network decide how much of the past information should be forgotten, while the update gate helps the network decide how much of the current information should be used to update the hidden state.

One of the advantages of GRU over LSTM is that it has fewer parameters, which makes it faster to train and more efficient to store. Additionally, GRU has been shown to perform comparably to LSTM on a wide range of tasks, including language modeling and speech recognition.

3.1.2 System Model

We consider two IIoT setups [1], [3] with fixed BSs and clients, and mobile blockers. One IIoT setup focuses on the single-antenna client network (SACN) while the other is for the multiple-antenna collaborative network (MACN) environment. In the SACN environment, each BS uses a phased array antenna of size M and has access to a set of B directional beams to cover a horizontal range of x degrees. Client devices are much simpler IoT devices and each client device uses a single omni (or quasi-omni) beam for both transmission and reception. The blockage mitigation action is predicted and taken by the BS. On the other hand, MACN

environment introduces more complex scenario. In this scenario, both BSs and clients use phased array of multiple antennas and have access to a set of directional beams to cover a horizontal range of desired degrees. While the blockage mitigation action is taken by both the BS and client collaboratively, the prediction of those actions is done by the BS/network side. In both setups, the BSs and clients operate at a mmWave band.

Our work builds on prior work [45], which leverages an innovative deep learning technique to proactively determine if a blockage is likely to occur in the future time interval and when is going to happen. Their proposed solution shows high accuracy in the near future time interval and only relies on undergoing communication between the BS and client without incurring additional communication overhead. This is achieved by leveraging a sequence of in-band wireless data measurements and jointly employing recurrent and convolutional neural networks. The work in [45] makes the assumption that the current connection is line-of-sight (LoS), which can be accurately predicted in mmWave systems. For example, the work in [46] can accurately classify LoS and nLoS channel conditions at the beginning of each communication interval by only using information from the messages that are exchanged between BS and client during the beam search interval (thus incurring no additional overhead).

The proposed frameworks in this part of the thesis build upon [45] by using their model to predict if a blockage is going to happen, and when it is going to happen. Then, the blockage mitigation framework decides on the best course of action to eliminate the negative impact of the oncoming blockage while the

blockage duration predictor estimates the period of time it takes for blockage to pass.

3.1.3 Blockage Mitigation and Duration Prediction

Our objectives are to mitigate blockages by proactively minimizing their impact and predict duration of the blockages, which can collaboratively lead to increased throughput and reduced latency of communication. Blockages can exhibit similar shapes, velocities, and patterns of trajectories. To address these issues, we first propose two frameworks (one for SACN environment and the other for MACN) that can determine the best action to mitigate blockages. Then, we propose a framework that can predict the duration of blockages. Those frameworks learn similar features from a sequence of reported signal-to-noise ratio (SNR) values to determine the best action to take and the duration of the blockages. These models are detailed as follows.

SACN Framework. Initially, we propose a framework [47] that can decide the best action to be taken by the BSs side only. For this proposed framework, the BSs employ multiple antennas while each client is equiped with a single antenna. The action space consists of three main actions as depicted in Figs. 3.1(c)-(e): beam switching, beam widening, and handoff. Each main action includes multiple sub-actions (e.g., beam switching includes the selection of new beam from all available beams at the BS, and handoff could be to any of the surrounding BSs). Therefore, the total number of actions include the sum of the total number of beams

at a BS[1], total levels of beam widening, and all surrounding BSs that a client can be handed off to. Let W be the total number beam widening levels and H be the number of surrounding BSs. Therefore, the total number of actions will be equal to $B+W+H$.

MACN Framework. In this framework, we tackle more complex wireless environment where both BSs and clients employ multiple antennas and have access to a set of directional beams. Unlike the SACN framework, MACN allows both BSs and clients to take a joint action. This framework introduces five main joint actions as depicted in Figs. 3.2(c)-(g). The first part of the joint actions is taken by the BS and the latter is the client's action. These joint actions include beam switching_beam switching (BeSw_BeSw), beam widening_beam switching (BeWi_BeSw), beam switching_beam widening (BeSw_BeWi), beam widening_beam widening (BeWi_BeWi) and handoff_beam switching (Ho_BeSw). Each main joint action includes multiple sub-actions, e.g., BeSw_BeSw includes the selection of new beams from all available beams at the BS and the client, and Ho_BeSw includes handing of the client to any of the surrounding BSs and the selection of new beam from all available beams at the client. Therefore, the total number of actions includes sum of all sub-actions of the main joint actions. Let TNA be the total number of actions. Note that BeSw_BeSw involves all combinations between the BS and the client beams, BeWi_BeSw includes all combinations between the total number of beam widening levels at the BS and total number of beams at the client, BeSw_BeWi includes combinations between

[1]Note that we assume each BS has access to B beams.

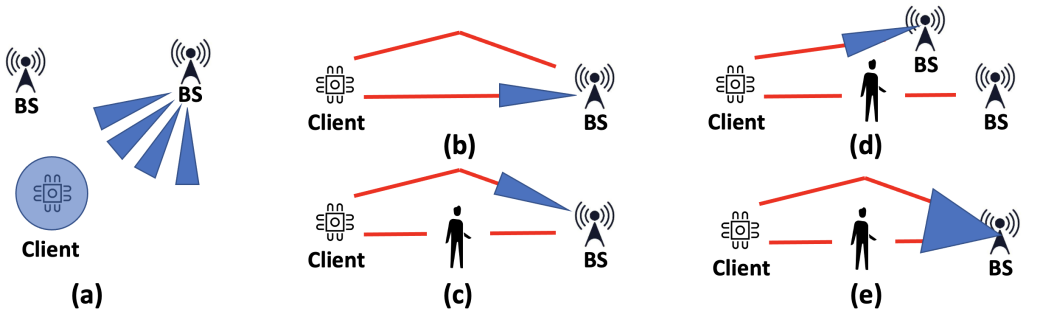


Figure 3.1: Single-antenna client network (SACN) setup. (a): There are several beams to choose from at the BS while the client has only a single antenna with omni-directional pattern. Beam selection typically happens through a search process at the beginning of every communication interval; (b): BS has identified a proper beam for communication with the client. Here the two red paths capture the multi-path nature of the communication channel; (c): In beam switching, BS switches to a different beam when blockage happens; (d): In handoff, the network may change the BS serving the client; (e): In beam widening, BS widens its beam. Energy reaches the client through other paths.

total number of beams at the BS and total number of beam widening levels at the client, BeWi_BeWi includes all combinations between the total number of beam widening at both BS and client, and Ho_BeSw includes all combinations between the total number of surrounding BSs and total number of beams at the client. Let the symbol # denote the number of sub-actions in a joint action, e.g., #BeSw_BeSw shows the total number of BS and client beam combinations. Then, $TNA = \#BeSw\_BeSw + \#BeWi\_BeSw + \#BeSw\_BeWi + \#BeWi\_BeWi + \#Ho\_BeSw$.

In addition to solve blockage mitigation problem, we introduce a framework that can predict the duration of blockages. The blockage duration is a continuous time that depends on type, size, and speed of a blocker.

GRU Model. Our frameworks utilized GRU to develop three models, one dedicated to SACN, the second one focused on tackling more complex environment

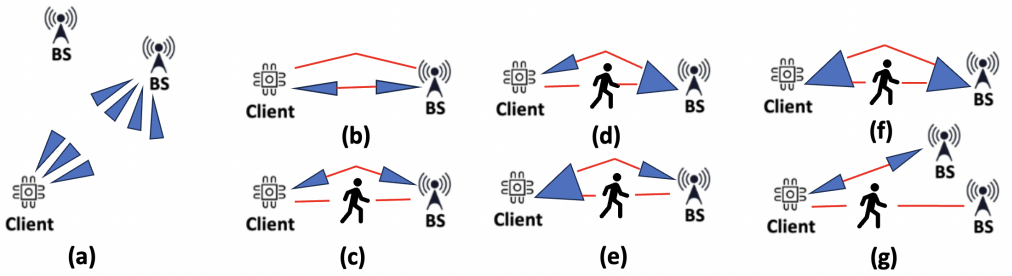


Figure 3.2: Multiple-antenna collaborative network (MACN). (a): There are several beams to choose from at both the BS and the client. Beam selection typically happens through a search process at the beginning of every communication interval; (b): BS and client have identified proper beams for communication with each other. Here the two red paths capture the multi-path nature of the communication channel; (c): In BeSw_BeSw, BS and client switch to different beams when blockage happens; (d): In BeWi_BeSw, BS widens its beam and client switches to a different beam; (e): In BeSw_BeWi, BS switches to a different beam and client widens its beam; (f): In BeWi_BeWi, BS and client widen their beams. Energy reaches the client through other paths; (g): In Ho_BeSw, the network may change the BS serving the client and client switches to a different beam.

with multiple-antenna client (MACN), and the third one is to predict the blockage duration. Each of those models consists of four layers of GRU cells with 128 units for each of the first hidden layers and 64 units for each of the latter two layers. The four GRU layers in each model are followed by a dense output layer with the size equal to the number of actions for the blockage mitigation models and a single unit for the blockage duration prediction.

In the SACN framework, the input to the model is a sequence of length N time steps each of which consists of SNR values of the current BS's beams along with the SNR value of the best beam of each surrounding BS and each BS's ID. In the MACN framework, the input to the model is slightly different. Instead of considering only the SNR values of current BS and surrounding BSs, the input here is taking into account both BSs' and clients' beams. The blockage duration prediction framework uses the same input as the MACN framework.

Formally, let S be a sequence of time steps, and $s \in S$ a time step where $s = \{\{b_1, b_2, \ldots, b_B\}, \{BS_1, BS_{SNR_1}, BS_2, BS_{SNR_2}, \ldots, BS_H, BS_{SNR_H}\}\}$. In the SACN framework, b_1 shows the SNR of beam one of the current BS, and BS_1, BS_{SNR_1} shows the ID of BS one and the SNR value of the best beam (measured at the client) of BS one. In the MACN framework, we assume there are B beam pair combinations between the serving BS and client and each b_i shows the client SNR for beam pair combination i. Further, BS_1, BS_{SNR_1} shows the ID of BS one and the SNR value of the best BS-client beams (measured at the client) of BS one. Note, the H here is the maximum number of surrounding BSs. The time steps sequence will be $S = \{s_1, s_2, s_3, \ldots, s_N\}$.

We refer to the time step sequence S as a sample. All SNR values of current and surrounding BSs (stemming from different beams at each BS and client), of each time step, are measured at the client side and reported to the serving BS during the periodic measurement report (MR) interval. We feed all GRU models with these MRs. For both blockage mitigation models, we optimized the training of our neural network model by using the Adam optimizer with a learning rate of 0.001, cross-entropy loss function, a batch size of 32, a dropout rate of 0.2, and $L2$ regularization with a coefficient of 0.01 to prevent overfitting. For the blockage duration prediction model, we used similar configurations of the blockage mitigation models except for using mean squared error (MSE) as the loss function and linear as the activation function of the output layer. Table 3.1 summarizes those GRU model structure and configuration while Fig. 3.3 shows the GRU architectures we used in our models.

Table 3.1: Blockage Mitigation and Duration Prediction GRU Models Structure and Configurations.

Parameter	SACN	MACN	Blockage Duration
GRU layer1	128 Units		
Dropout	0.2		
GRU layer2	128 Units		
Dropout	0.2		
GRU layer3	64 Units		
Dropout	0.2		
GRU layer4	64 Units		
Dropout	0.2		
Output dense	46 Units	TNA Units	1 Unit
GRU activation	ReLU		
Output activation	Softmax		Linear

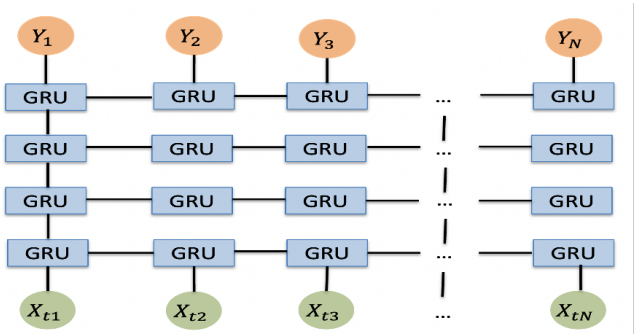


Figure 3.3: GRU model structure with four GRU layers. X_{t1} to X_{tN} are the time steps that are used as the input, while Y_1 to Y_N are the outputs of the model, corresponding to the probability of each class (blockage mitigation strategy). Blockage duration prediction uses the same architecture but with only one scalar value corresponding to the blockage duration.

Action Selection Metric. The dataset samples that are fed to the GRU models must be labeled with the actual blockage duration and the best action. The blockage duration can be achieved through $T = \frac{L}{V}$, where L is the length of the blocker and V is the blocker velocity. However, defining the best action label is more challenging. For example, selecting the best action based on the highest reported SNR value can be inefficient. For example, let the handoff action take 1 second to complete and let the human blockage last for 0.4 seconds. Therefore, if the handoff action is selected because of the the highest expected SNR, there will be 0.6 seconds during which no data is transferred. Thus, we need to choose a metric that not only takes into account the SNR but also both the duration of the blockage event as well as the duration needed to perform the blockage mitigation action. To do this, we first define the average throughput of a client associated with a BS with a given SNR value as:

$$R = \frac{\omega}{U} \times \log_2(1 + SNR) \tag{3.1}$$

here, ω is the communication bandwidth and U is the total number of clients associated with the BS.

Next, we choose the amount of transferred data (D) as the metric based on which we select the best blockage mitigation technique. This metric combines the average throughput metric (R) defined in Eq. (3.1) and both the duration of the blockage event (T) and the duration of a given blockage mitigation technique (C) according to the following formula:

$$D = \max\{(T - C), 0\} \times R \tag{3.2}$$

here, T and C are in seconds. The duration of blockage mitigation technique (C) can also be considered as the cost associated with taking that action. We can also estimate T through $T = \frac{L}{V}$, where L is the length of the blocker and V is the blocker velocity. In our simulations, we let the cost associated with handoff, beam switching, and beam widening be equal to 1, 0.01, and 0.015 seconds, respectively.

3.2 Blockage Mitigation Techniques as Policies

In the previous section, we chose and defined the amount of transferred data as the metric to optimize when selecting the best blockage mitigation technique. In this section, we give a formal definition of our action selection mechanism as a policy. We also define alternative policies that model other blockage mitigation

mechanisms from the literature.

Assume that there are K types of blockages. Denote the probability of occurrence of blockage type i by P_i. Further, assume there are U blockage mitigation techniques, and the cost of each technique j is denoted by C_j. Let T_i be the the duration of blockage type i, and let $R_{i,j}$ be the achievable rate of the user when blockage type is i and mitigation technique is j, in the given network. Note that $R_{i,j}$ is a random variable. Then, we define the amount of transferred data, when the blockage type, selected blockage mitigation technique, and rate are known as:

$$D_{i,j} = \max\{(T_i - C_j), 0\} \times R_{i,j} \tag{3.3}$$

Policy 1. Choose the best mitigation technique when the blockage type and the value of $R_{i,j}$ are given, i.e., use the mitigation technique j with the maximum amount of transferred data as estimated by Eq. (3.3). This policy is the one that we implemented in our approach. The expected amount of transferred data when policy 1 is employed is:

$$E[D_{i,j}\ policy\ 1] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,max} \tag{3.4}$$

where

$$D_{i,max} = \max(D_{i,1}, D_{i,2}, \ldots, D_{i,U})\ \&\ \bar{D}_{i,max} = E[D_{i,max}]$$

The expectation in $E[D_{i,max}]$ is to account for user specific $R_{i,j}$, which in addition to blockage type i and mitigation technique j, depends on the client channel.

Policy 2. Choose a fixed best mitigation technique for each type of blockage. For example, if the mitigation technique j provides on average the maximum amount of transferred data for blockage type i, use this mitigation technique for the specific blockage type i. We use the notation $j = \delta(i)$ to distinguish this mitigation technique j. The expected amount of transferred data when policy 2 is employed is:

$$E[D_{i,j}\ policy\ 2] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,\delta(i)} \tag{3.5}$$

where

$$\begin{aligned}\bar{D}_{i,\delta(i)} = E[D_{i,\delta(i)}] &= \max(E[D_{i,1}], E[D_{i,2}], \ldots, E[D_{i,U}]) \\ &= \max(\bar{D}_{i,1}, \bar{D}_{i,2}, \ldots, \bar{D}_{i,U})\end{aligned}$$

Note, blockage type is assumed to be known.

Policy 3. Unlike policy 2, this policy chooses a single fixed technique for all type of blockages. The chosen technique, which we denote as $j\star$, is the technique that provides on average the maximum amount of transferred data across all types of blockages. We can compute the expected amount of transferred data when policy 3 is employed as follows:

$$E[D_{i,j}\ policy\ 3] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,j\star} \tag{3.6}$$

where

$$\sum_{i=1}^{K} P_i \times \bar{D}_{i,j\star} = \max(\sum_{i=1}^{K} P_i \times \bar{D}_{i,1}, \sum_{i=1}^{K} P_i \times \bar{D}_{i,2}, \ldots, \sum_{i=1}^{K} P_i \times \bar{D}_{i,U})$$

and $\bar{D}_{i,j} = E[D_{i,j}]$

Policy 4. Choose an arbitrary mitigation technique for all type of blockages, i.e., choose and always use a fixed technique j for all of the blockage types. The expected amount of transferred data when policy 4 is employed is:

$$E[D_{i,j}\ policy\ 4] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,j} \tag{3.7}$$

We have the following proposition on the theoretical performance of these four policies. In Section 3.4, we quantify these theoretical results through simulations.

Proposition. The order of performance (in terms of the average amount of transferred data) among the four aforementioned policies is as follows:

$$E[D_{i,j}\ policy\ 1] \geq E[D_{i,j}\ policy\ 2] \geq E[D_{i,j}\ policy\ 3] \geq E[D_{i,j}\ policy\ 4]$$

with policy 1 being the best policy in mitigating the blockages and providing the maximum average amount of transferred data. The degree of difference between these policies measured in terms of the average amount of transferred data depends on the distribution of $R_{i,j}$s for various cases of i and j, the value of T_i, and the value of C_j.

Proof. From the definition, $D_{i,max} \geq D_{i,\delta(i)} \Rightarrow \bar{D}_{i,max} = E[D_{i,max}] \geq E[D_{i,\delta(i)}] = \bar{D}_{i,\delta(i)}$.

$E[D_{i,j}\ policy\ 1] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,max} \geq \sum_{i=1}^{K} P_i \times \bar{D}_{i,\delta(i)} = E[D_{i,j}\ policy\ 2]$.

Next, based on the definition of $\delta(i)$, $\bar{D}_{i,\delta(i)} \geq \bar{D}_{i,j\star}$, hence

$$E[D_{i,j}\ policy\ 2] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,\delta(i)} \geq \sum_{i=1}^{K} P_i \times \bar{D}_{i,j\star} = E[D_{i,j}\ policy\ 3].$$

Finally, from the definition of $j\star$,

$$E[D_{i,j}\ policy\ 3] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,j\star} \geq \sum_{i=1}^{K} P_i \times \bar{D}_{i,j} = E[D_{i,j}\ policy\ 4].$$

3.3 Theoretical Model on Average Loss Ratio

Estimating the blockage duration can be a useful tool in an overall blockage mitigation frameworks. For example, consider a high throughput link that undergoes a blockage event. If the system has an estimate of blockage duration, the BS and client can fall back to the original link once blockage ends. However, over or under estimation of the blockage duration coupled with sub-optimal blockage mitigation technique selection can limit the performance. In this section, we develop a theoretical framework to model the average data loss ratio, a metric that captures these inaccuracies. In short, the metric captures the loss in data rate due to inaccuracies divided by the total amount of data that can be transferred assuming optimal blockage duration estimation and mitigation strategy selection. In Section 3.4, we show through simulations that the loss ratio is indeed very small and the upper bound derived in this section is very tight.

Let T, $\hat{\mathrm{T}}$, C_k, and R_k be the actual blockage duration, predicted blockage duration, the cost of applying the k^{th} mitigation technique, and the data rate (throughput) after applying the k^{th} mitigation technique, respectively. Then, we

define the index of predicted mitigation technique (p), the index of the optimal technique (o), and the index of the maximum achievable rate (q) for each given blockage as follows:

$$\begin{cases} p = \underset{k}{\text{argmax}}(\hat{\text{T}} - C_k) \times R_k \\ o = \underset{k}{\text{argmax}}(T - C_k) \times R_k \\ q = \underset{k}{\text{argmax}}(R_k) \end{cases}$$

Note that we expect $p = o$ for the majority of the cases. If not, we must have:

$$\begin{cases} (\hat{T} - C_p) \times R_p > (\hat{T} - C_o) \times R_o \\ (T - C_o) \times R_o > (T - C_p) R_p \end{cases}$$

To derive an upper bound on the average loss ratio, we first derive an upper bound on the data loss as follows:

$$\begin{aligned} loss\_data &= (T - C_o) \times R_o - (T - C_p) \times R_p \\ &\leq (T - C_o) \times R_o - (T - C_p) \times R_p + (\hat{T} - C_p) \\ &\times R_p - (\hat{T} - C_o) \times R_o = (T - \hat{T}) \times (R_o - R_p) \leq \\ &|T - \hat{T}| \times |R_o - R_p| \leq |T - \hat{T}| \times R_q \end{aligned} \tag{3.8}$$

Next, we derive a lower bound on the total amount of data that can be transferred in the optimal case as follows:

$$total\_data = (T - C_o) \times R_o \geq (T - C_q) \times R_q = T \times R_q - C_q \times R_q \tag{3.9}$$

Note that variables T and R_q are independent. Similarly, variables $T - \hat{T}$ and R_q are independent. Thus, leveraging Eqs. 3.8 and 3.9 we can derive the following

upper bound on the average loss ratio:

$$\overline{Loss-Ratio} = E(\frac{loss\_data}{total\_data}) \leq \frac{E(|T-\hat{T}|) \times E(R_q)}{E(T) \times E(R_q) - E(C_q \times R_q)} = \frac{E(|T-\hat{T}|)}{E(|T|) - \frac{E(C_q \times R_q)}{E(R_q)}} \tag{3.10}$$

3.4 Performance Evaluation

In this section, we discuss our performance evaluation results for SACN, MACN and the duration prediction models. First, we discuss our simulated environment and our methodology to gather and label data. Next, we demonstrate the performance results for the SACN and MACN models considering both ML metrics (e.g., accuracy, F1 score) and networking metrics (e.g., transferred data, throughput). Further, we show the result of blockage duration prediction model.

3.4.1 Measurement Campaign

Simulation Setup. We used a commercially available wireless simulator named Wireless InSite (WI) [48] to simulate an IIoT use case scenarios with a size of $350 \times 150\ m^2$. IIoT is a cutting-edge technology that connects Internet-connected devices, sensors, and machines with industrial processes and systems. It is considered as one of the critical technologies for the development of Industry 4.0 [11], [12]. MmWave wireless technologies play a significant role in IIoT by providing high data rates and assisting with positioning [1], [3]–[5].

Our simulation scenario consists of six BSs and one hundred clients. BSs are

distributed along both sides of the environment with three BSs at each side with 75 m distance between them (see Fig. 3.4). All BSs and clients operate at 28 GHz band and are on a LoS channel condition prior to blockage. BS height is 2.5 m, while the client height is 1.5 m. In the SACN environment, each BS has access to a uniform linear array (ULA) that consists of 16 antennas, which provide 36 beams and cover a horizontal range of 120°. Clients are randomly distributed within the environment. Each client has access to a single beam with an omni-directional radiation pattern. In the MACN environment, the setting of clients differs from the SACN environment in terms of the number of antennas and beams at the clients' side. Specifically, each of the BSs and the clients has access to a ULA that consists of 16, 8, or 4 antennas, which provide 18, 9, or 4 beams, respectively, and cover a horizontal range of 120°. Table 3.2 summarizes the simulation parameters for both environments.

For both environments, we included four different types of blockers: human, cart, truck, and pickup. These particular blockage types are the most commonly encountered blockers in a factory setting. We modeled a human as a cylinder with radius of 30 cm that can move at range from 0.89 to 1.4 m/s speed. The size of carts, pickups, and trucks are $2.70 \times 1.21 \times 1.8$ m^3, $5.4 \times 2.1 \times 1.9$ m^3, and $12.3 \times 2.7 \times 2.3$ m^3, respectively. We assumed the velocity of each of these three blockers can range from one to five miles per hour (mph), which equals to 0.89 to 2.2 m/s.

Table 3.2: Simulation Setup.

Parameter	SACN	MACN
BS antenna array sizes	16 (ULA)	16 and 8 (ULA)
Client antenna array sizes	1	8 and 4 (ULA)
Number of beams at the BS	36	18 and 9
Number of beams at the client	Omni	9 and 4
Frequency	28 GHz	
Bandwidth	1 GHz	

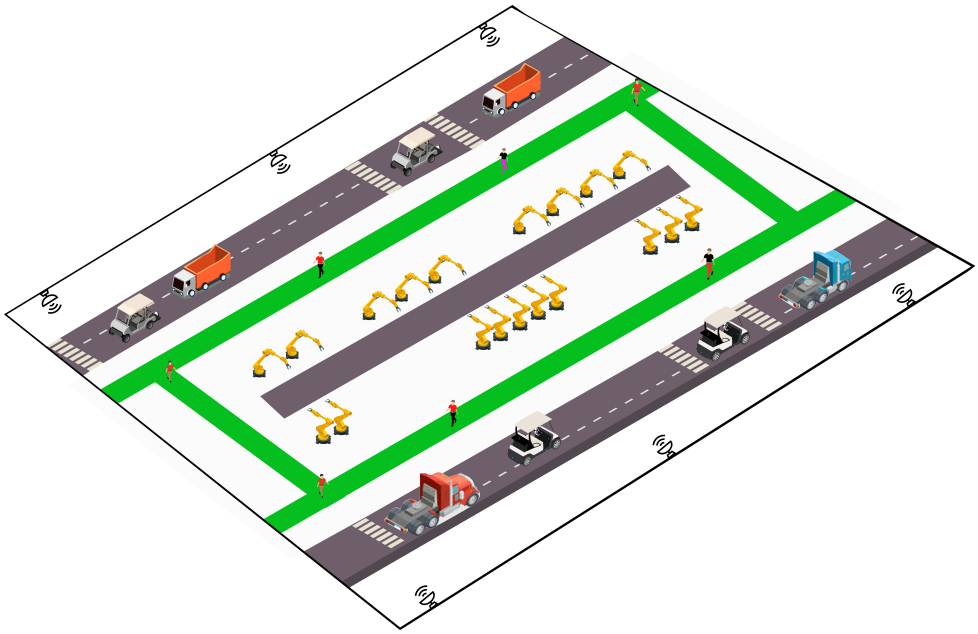


Figure 3.4: IIoT scenario with six BSs on the two sides, clients, and different types of blockers. We included four different types of blockers: human, cart, truck, and pickup, which are the most commonly encountered blockers in a factory setting.

Data Gathering. As discussed in Section 3.1.3, our GRU-based model takes a sequence of data as input to decide on what action to take as an output. This data consists of multiple time steps each of which contains SNR values for all the beams of the current BS along with best SNR (stemming from the best beam) of

each of the surrounding BSs along with BS IDs.

We run our simulation based on the above setup and record the SNR measurements at the clients side every 100 *ms*. In other words, the difference in time between two consecutive time steps is 100 *ms*. The run continues as the blockers move until the current connection is completely blocked. We extract the measured SNR values of the current BS and surrounding BSs' best SNR values and their IDs for every time step, including the time step in which the connection is blocked. The last time step (when the connection is blocked) is repeated with varying number of antennas at the current BS (or client in the case of MACN) to get information about how beam widening performs when the connection is blocked[2].

In the SACN environment, we gathered a dataset comprising 20,000 samples, each consisting of five time steps. Within each time step, there are 36 SNR values that represent the current BS's 36 beams along with the best SNR values for the five surrounding BSs and the BS IDs. The dataset has been evenly distributed across the four different types of blockages, with 5,000 samples collected for each blockage type.

In the MACN environment, we consider four array configurations scenarios: 16 antennas at the BS paired with 8 or 4 antennas at the client, and 8 antennas at the BS paired with 8 or 4 antennas at the client to cover all possible combinations. We gathered a dataset comprising 30,000 samples, each consisting of five time steps. Within each time step, there are 36, 81, 72, or 162 SNR values that represent

[2]In a uniform linear array (ULA) with M antennas and $\frac{\lambda}{2}$ spacing distance (λ is the carrier wavelength), the main lobe beamforming gain is equal to $10\times\log_{10}^{M}$ (in dB) with $\frac{102}{M}$ (in degrees) half power beamwidth.

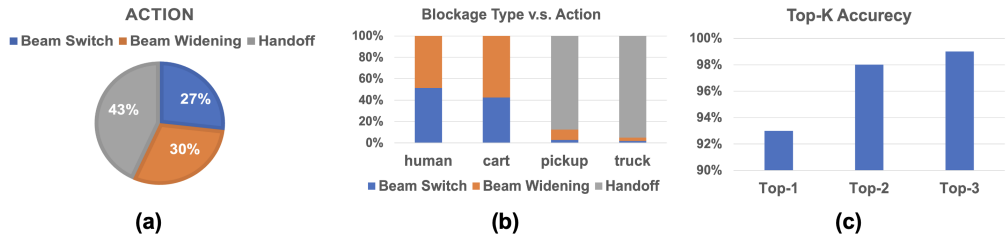


Figure 3.5: Single-antenna client network (SACN). (a): Fraction of samples in the dataset labeled with each action; (b): Fraction of samples labeled with each action for each blockage type; (c): Top-1, Top-2, and Top-3 accuracy results of our GRU-based blockage mitigation framework. The correct label is the predicted label in 93% of instances and is among the top three predictions for 99% of instances.

all combinations of serving BS and client beams, which are 9×4, 9×9, 18×4, or 18×9 beam combinations, respectively, along with the best SNR values for the five surrounding BSs and the BS IDs. The dataset has been evenly distributed across the four different types of blockages, with 7500 samples collected for each blockage type. For each environment dataset, we use 70% of dataset for training and 30% for test.

Data Labeling. Labeling refers to determining the best action and the actual blockage duration. In order to label our dataset samples, we began by determining the number of classes, or actions. Then, as detailed in Section 3.1.3, we labeled each sample in our dataset based on the optimal action that would result in the highest amount of transferred data during a blockage event. To do this, we utilized Eq. (3.3) to label each sample appropriately. We also labeled each sample with actual blockage duration using the formula $T = \frac{L}{V}$.

In the SACN, the total number of actions is 46 actions that are distributed as follows: 36 classes (ranging from 0 to 35) were assigned to the current BS's

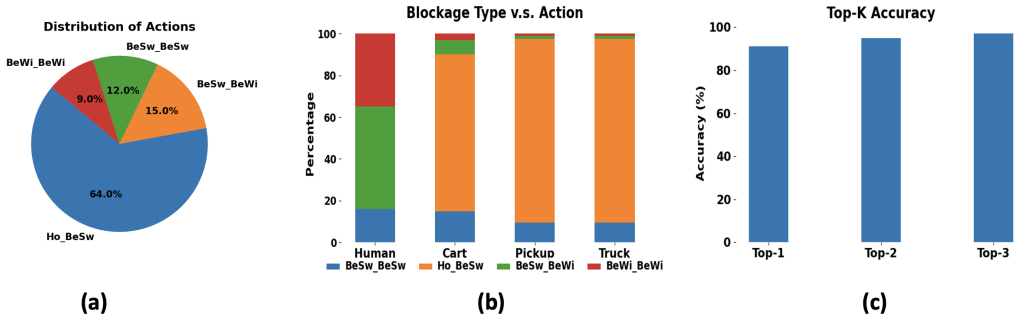


Figure 3.6: Multiple-antenna client network (MACN). (a): Fraction of samples in the dataset labeled with each action; (b): Fraction of samples labeled with each action for each blockage type; (c): Average Top-1, Top-2, and Top-3 accuracy results (across the different array sizes) of our GRU-based blockage mitigation framework. The correct label is the predicted label in 91% of instances and is among the top three predictions for 97% of instances.

36 beams to accommodate the beam switching action, 5 classes (ranging from 36 to 40) were assigned to the number of surrounding BSs to which a client can be handed off to, and 5 classes (ranging from 41 to 45) were assigned to beam widening.

To gain a deeper insight into our labeled dataset, we conducted two distinct analyses. First, we calculated the percentage of samples labeled with each action across the entire dataset, regardless of the type of blockage. Fig. 3.5(a) shows the corresponding result. We observe that handoff was the best action for 43% of our samples, beam widening was the best action for 30% of our samples, and beam switching was the best action for 27% of our samples.

Next, we conducted an analysis to examine the relationship between the best action and the type of blockage. Fig. 3.5(b) demonstrates the corresponding result. Our results indicate that beam switching and beam widening are commonly best

actions for smaller blockers (e.g., human or cart), whereas handoff is a better action for larger blockers (e.g., pickup or truck). Additionally, we discovered that beam switching and beam widening could sometimes better handle large blockages compared to handoff. However, handoff was never a good solution to handle small blockers due to its cost exceeding the blockage event time.

In the MACN, the number of actions depend on the number of antennas and beams at both BSs and clients. Table 3.3 summarizes the four configurations scenarios resulting from different number of antennas at the BS/client along with the total number of available actions.

Table 3.3: Antenna size configurations of the multiple-antenna client network (MACN) environment.

BS Antenna	BS Beams	Client Antenna	Client Beams	No. of Actions
8	9	4	4	194
8	9	8	9	384
16	18	4	4	536
16	18	8	9	870

We conducted the same two distinct analyses that we did for the SACN dataset. In the first analyses, Fig. 3.6(a) shows Ho_BeSw was the best action for 64% of our samples, BeSw_BeWi was the best action for 15% of our samples, BeSw_BeSw was the best action for 12% of our samples, and BeWi_BeWi was the best action for 9% of our samples.

In the second analyses, Fig. 3.6(b). indicates that beam switch and widening combinations (BeSw_BeSw, BeSw_BeWi and BeWi_BeWi) are commonly best actions for smaller blockers (e.g., human), whereas handoff (Ho_BeSw) is a better

action for larger blockers (e.g., pickup or truck). Additionally, we discovered that BeSw_BeSw, BeSw_BeWi and BeWi_BeWi could sometimes better handle large blockages compared to Ho_BeSw. However, Ho_BeSw was never a good solution to handle small blockers due to its cost exceeding the blockage event time.

3.4.2 Results

In the first part of this subsection, we discuss the performance of SACN framework while in the second part we investigate the performance of the MACN framework and duration prediction. For both frameworks, we consider both ML and networking metrics for evaluation.

3.4.2.1 Single-antenna Client Network (SACN)

GRU Evaluation. Accuracy is a popular metric to evaluate the performance of a machine learning model. It is defined as the ratio of correct predictions made by the model to the total number of predictions made. Since our model takes a sequence of data as input to make a decision, we evaluate SACN-based GRU model over five time steps.

Fig. 3.5(c) shows the accuracy of our SACN model across all samples, which is 93% and demonstrates that the model has learned to make accurate predictions.

Next, to obtain a more complete picture of the model's accuracy, we consider Top-K accuracy. Top-K accuracy is a more informative metric that measures the proportion of instances in which the correct label is among the top K predicted labels. For instance, Top-1 accuracy measures the proportion of instances in which

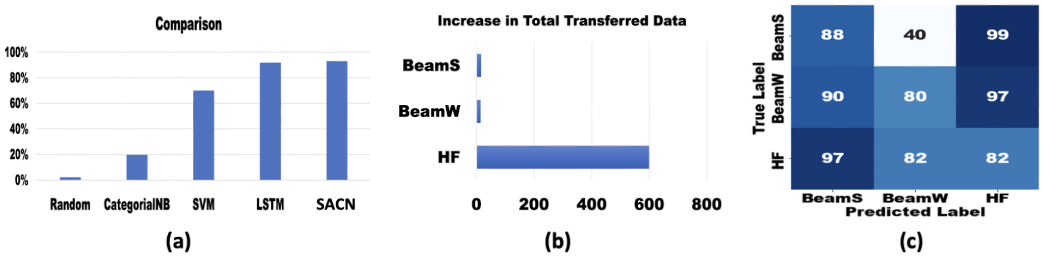


Figure 3.7: Single-antenna client network (SACN). (a): Accuracy of different ML models. GRU achieves the highest accuracy; (b): Increase in total transferred data for different actions. Here BeamS, BeamW, and HF denote beam switching, beam widening, and handoff, respectively; (c): The average drop in throughput when selected action is not optimal. Y-axis shows the true label and x-axis shows the incorrect predicted label. When the true and predicted label are the same, the selected sub-action is not optimal. The overall average drop in throughput across all actions is 92%.

the correct label is the top prediction made by the model. In our case, the Top-1 accuracy of our model is 93%. This result is a testament to the effectiveness of the model, as it is making accurate predictions in majority of cases.

Moreover, our model has achieved a Top-2 accuracy of 98% (Fig. 3.5(c)), meaning that the correct label is among the top two predictions for 98% of instances. Furthermore, our model has achieved a Top-3 accuracy of 99%, which indicates that the correct label is among the top three predictions for 99% of instances. Note that to be counted as a correct action, details of the action must be correct too. For example, when beam switch to a particular beam is the correct action, the model not only has to select beam switching as the appropriate blockage mitigation technique but it must also correctly select the beam to switch to (out of 36 available beams) to match the label.

Comparison with Other ML Models. To demonstrate the effectiveness of

SACN-based GRUs, we compared it with four baseline techniques: random selection, Categorical Naive Bayes (CategoricalNB), support vector machine (SVM), and LSTM.

Random selection is a simple technique that randomly assigns class labels to the data, making it a useful baseline for determining the performance of a model by chance, while CategoricalNB is a popular ML algorithm that works well for text classification tasks and assumes that the features are categorical. SVM works by finding the best possible boundary that separates data points into different classes. In other words, it tries to find the hyperplane that maximizes the margin between the different classes in the dataset. We evaluated each technique on the same dataset and considered accuracy as the performance evaluation metric.

We used scikit-learn, the open-source ML library, to implement CategoricalBN and set the hyperparameters *alpha* and *fit_prior* to 1 and true, respectively. We used the same scikit-learn library to implement an SVM model with “radial basis function (rbf)" *kernel* and parameter *gamma* set to “scale". We implemented the LSTM NN with the same structure and hyperparameters of GRU as described in Section 3.1.1.

Fig. 3.7(a) shows the accuracy results. The results show that the SACN-based GRU model outperforms all baseline techniques, achieving an accuracy of 93%. In comparison, random selection achieves an accuracy of only 2.2%, CategoricalNB 19.7%, and SVM 70%. LSTM, which is very similar to GRU, gets a very close accuracy at 91.8%.

We chose the GRU model to develop SACN over LSTM for several reasons.

First, GRU models require fewer parameters, which reduces the training time and improves computational efficiency. This is especially important when working with large datasets or when there are time constraints for model training. Additionally, we found that the GRU model has a slightly higher accuracy than the LSTM model, further supporting our decision to choose GRU.

Network Evaluation. We next evaluate our model performance in terms of increase in the amount of transferred data when an action is taken compared to the blocked connection. Note that we measure throughput according to Eq. (3.1), which depends on the client SNR. When blockage happens, SNR drops, which reduces the throughput (but is non-zero due to Eq. (3.1)). Fig. 3.7(b) shows the corresponding results where beam switching, beam widening and handoff are denoted by BeamS, BeamW, and HF, respectively. Beam switching increases the total amount of transferred data by x16, while beam widening increases that by x14. Handoff increases the total amount of transferred data by about x600, much higher than beam widening and beam switching. Note, that the results do not mean that beam switching and beam widening are not effective actions. Handoff is commonly used when blocker is large (e.g., with a pickup or truck), which has the most negative impact on the underlying connection. Therefore, the gap between the new throughput (as a result of handoff) and baseline throughput (throughput of the connection blocked by a large blocker) is very large. Additionally, with handoff, the new BS will likely observe no negative impact from the current blocker. Beam switching and beam widening are still efficient actions when the blocker is small (e.g., human or cart as we showed in Fig. 3.5(b)). They also result in more increase

in throughout than handoff when blocker is small, due to the shorter duration of the blockage event and high cost of handoff.

Drop in Throughput when Selected Action is not Optimal. Our results in Fig. 3.5(a) showed that the SACN-based GRU model selects the optimal action in 93% of the times. We next study the drop in performance (in terms of throughput) when the selected action is not optimal, which has a 7% probability. To do so, we calculate the client throughput during the blockage event had we selected the optimal action as well as the action selected by the our model. We then calculate the percentage drop in performance (from the previous two calculations) and find its average value across all clients and blockage events and found that to be equal to 92%. This shows that there is a large average penalty when the optimal action is not taken, or in other words optimal action can result in a very high increase in throughput. Note that while this analysis is specific to our SACN-based GRU model, other methods also have a similar performance, as we will show through average throughput results later in this section.

To gain more insights into this average number, we extend our examination to compare the percentage drop in performance for each specific action. Fig. 3.5(c) shows the corresponding results. Here the y-axis shows the true label and the x-axis shows the incorrect predicted label. For example, when the optimal action is identified as BeamS (top row in Fig. 3.5(c)), we derive the average percentage drop in throughput that would be achieved if the selected action is BeamS to a non-optimal beam ID, BeamW, or HF.

By examining the results across all combinations, we observe that when the true

label is BeamS, handoff (HF) would result in the highest (99%) drop in throughput. Further, if BeamS is the true label, BeamW would result in the lowest (40%) average drop in throughput. This type of a cost analysis could also be used by a network operator to devise more sophisticated blockage mitigation policies by associating different costs to different actions.

Average Throughput Across Different ML Models. Throughput is a critical metric for evaluating overall network performance. To assess the effectiveness of our model, we conducted an evaluation based on the average throughput across all ML models (SACN-based GRU, LSTM, SVM, CategoricalNB, and Random). We measure throughput for the duration of the blockage event taking into account blockage type and velocity, and action delay, among others.

Table 3.4: Average throughput of the single-antenna client across all clients, BSs, and blockage events achieved by different ML models.

	Random	CatNB	SVM	LSTM	SACN
Throughput (Mbps)	0.52	4.62	16.32	21.8	22.7

Table 3.4 summarizes the average throughput results for each model. Note that these throughput results are averaged across all clients, BSs, and blockage events. Our evaluation shows that the SACN-based GRU model outperforms all the baselines with an average throughput of 22.7 Mbps. The ratio of increase in throughput compared to different schemes is 43.7 with respect to (w.r.t.) Random, 4.91 w.r.t. CatNB, 1.39 w.r.t. SVM, and 1.04 w.r.t. LSTM.

Comparison with Other Policies. SACN framework exhibits a high level of performance in mitigating blockages, as observed in terms of both ML-based

and networking metrics. We next compare the performance of our approach with three alternative policies (policies 2, 3, and 4 from Section 3.1.3) using the average amount of transferred data as metric. For this purpose, we utilized the same dataset that was used for both training and testing the SACN framework. Subsequently, we computed the average amount of transferred data for each of the policies. Table 3.5 summarizes the decline in performance when implementing policies 2, 3, and 4 in comparison to our approach, which employs policy 1.

Table 3.5: Comparison of SACN framework against three alternative policies (policies 2, 3, 4) in terms of average percentage drop in the amount of transferred data.

	Human	Cart	Pickup	Truck
Policy 2	-25%	-5%	-29%	-16%
Policy 3	-27%	-5%	99%	-99%
Policy 4	-100%	-100%	-29%	-16%

In Policy 2, a fixed best technique is employed to maximize the average amount of data for each blockage type. Specifically, we found that the optimal technique for blockage type “human" is beam switching, while the best technique for “cart" is beam widening. For “pickup" and “truck," the ideal approach is handoff. To determine the average amount of data transferred for each blockage type, we applied the corresponding best technique and compared the results with our own approach.

The first row in Table 3.5 depicts the results. Our results show that when employing only beam switching for human blockage, the average amount of transferred data decreases by 25% compared to our approach. Similarly, for cart, the use of beam widening only resulted in a 5% decrease in the average amount

of transferred data. For pickup, using handoff only, resulted in a 29% decrease in the average amount of transferred data. Finally, for a truck blocker, the average decrease is 16%.

In Policy 3, a fixed technique is employed to maximize the average amount of transferred data across all blockage types. To determine the optimal technique, we calculated the average amount of transferred data for the three techniques for each type of blockage and compared the results to identify the technique that provided the maximum performance across all blockage types. Our analysis revealed that beam widening was the optimal technique that provided the maximum average amount of transferred data across all blockage types.

To further evaluate the effectiveness of Policy 3, we compared its performance with our approach in the same manner as we did for Policy 2. The results of our analysis (depicted in second row of Table 3.5) indicates a substantial reduction in the average amount of transferred data for larger blockers, i.e., pickups and trucks. Specifically, the average amount of transferred data for pickups and trucks decreased by approximately 99%, while the performance drop for the cart blocker remained at the same level as in Policy 2. Finally, for the human blockage type, the average amount of transferred data decreased by 27%, which was 2% less than the decrease observed in Policy 2.

For policy 4, where an arbitrary technique is chosen for all types of blockages, we found that the results of the chosen technique might do well with some blockage types and poorly with other. For example, handoff works well with larger blockages, but decreased the performance by 100% for smaller blockages since the cost of

taking the handoff exceeds the small blockage duration. Therefore, if all of the blockages in an environment are small (e.g., carts or humans), handoff provides no gains. Hence, handoff decreased the performance for carts and human blockages by 100% and stayed at the same level of policy 2 for the other blockages. The last row in Table 3.5 captures these results.

3.4.2.2 Multiple-antenna Client Network (MACN)

GRU for Blockage Mitigation Evaluation. Similar to the SACN framework, we consider accuracy to evaluate MACN. Therefore, we evaluate our model over the last 5 time steps. Since we have four different array configurations, we train and evaluate our model on the four different array configurations datasets. Our MACN-baes GRU model performed well in learning the four different datasets with over 90%. Our model achieved the highest accuracy with 91.7% when BSs have 16 antennas and clients have 4 antennas. Table 3.6 shows the accuracy results for the four array configuration scenarios.

Table 3.6: MACN-based GRU model accuracy for different antennas size configurations.

Array Size	Accuracy
BS: 8, Client: 4	90.2%
BS: 8, Client: 8	91.4%
BS: 16, Client: 4	91.7%
BS: 16, Client: 8	91.5%

Next, to obtain a more complete picture of the model's accuracy, we consider Top-K accuracy as depicted in Fig. 3.6(c). To investigate the top K accuracy, we consider the average of the four array configuration scenarios. In our case, Top-1

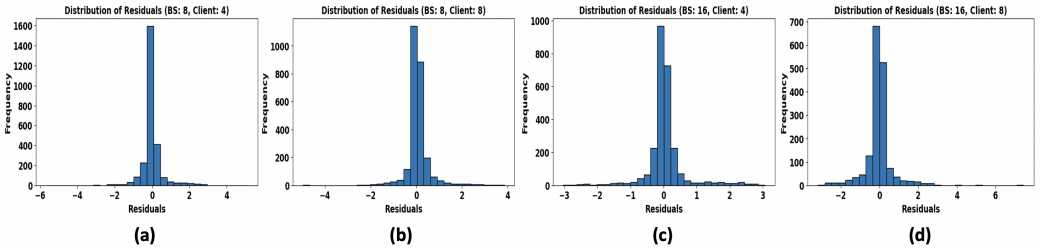


Figure 3.8: The residual results represent the disparity between the actual and predicted blockage duration across the four array configuration scenarios: (a) BS: 8 - Client: 4; (b) BS: 8 - Client: 8; (c) BS: 16 - Client: 4, and (d) BS: 16 - Client: 8.

accuracy of our GRU model is 91%. This result is a testament to the effectiveness of our model, as it is making accurate predictions in majority of cases.

Moreover, our model has achieved a Top-2 accuracy of 95% (Fig. 3.6(c)), meaning that the correct label is among the top two predictions for 95% of instances. Furthermore, our model has achieved a Top-3 accuracy of 97%, which indicates that the correct label is among the top three predictions for 97% of instances. Note that to be counted as a correct action, details of the action must be correct too. For example, when BeSw_BeSw to a particular beams is the correct action, the model not only has to select beam switching for the BS and the client as the appropriate blockage mitigation technique, but it must also correctly select the beam to switch to (out of all the available beams at the BS and the client) to match the label.

GRUs for Blockage Duration Prediction. Since we are dealing with a regression problem here, and the aim is to predict continuous values, we need to use a metric that gives a clearer view by measuring the difference between the actual and predicted values. Thus, we use residual errors, which are the differences between the actual and predicted values. The use of residuals as a metric for performance

evaluation involves assessing how well a model's predictions align with the actual data points. A residual plot or analysis helps to identify patterns or trends in the errors made by the model. Fig. 3.8 shows the residual plots for the four array configuration scenarios. In examining the residual plots of our model, it's evident that, on the whole, our predictions align well with the actual values. The majority of data points cluster around zero, indicating that, on average, our model is providing accurate estimates. The mean of the overall residual across all the four different antennas configuration scenarios is 0.04 second. While assessing the distribution of residuals, we observe a slight right-skewness, suggesting that there are several instances where our model tends to underestimate the target variable. However, it's important to note that our model deviations are generally modest.

Comparison with Other ML Models. Similar to the SACN framework, we compare our MACN-based GRUs with three baseline techniques: Categorical Naive Bayes (CategoricalNB), Transformer, and LSTM. CategoricalNB is a popular ML algorithm that works well for text classification tasks and assumes that the features are categorical. Transformers [49] are a type of deep learning model architecture. Unlike traditional sequential models, transformers utilize attention mechanisms to process input data in parallel, making them highly efficient for tasks such as natural language processing. We evaluated each technique on the same dataset and considered accuracy as the performance evaluation metric.

We used scikit-learn, the open-source ML library, to implement CategoricalBN and set the hyperparameters *alpha* and *fit_prior* to 1 and true, respectively. We used TensorFlow and Keras to implement Transformer model. The architecture includes

four multi-head self-attention and four position-wise feedforward layers, with global average pooling for sequence aggregation. We implemented the LSTM NN with the same structure and hyperparameters of GRU as described in Section 3.1.1.

Fig. 3.9(a) shows the accuracy results. The results show that the MACN-based GRU model outperforms all baseline techniques, achieving an accuracy of 91% on average across all the array configuration scenarios. In comparison, CategoricalNB and Transformers achieve 19.7% and 39% accuracy, respectively. LSTM, which is very similar to GRU, gets an accuracy around 89%.

Network Evaluation. We evaluate our MACN-based GRU model in similar way that we evaluated SACN. Fig. 3.9(b) shows the corresponding results. Note that BeSw_BeWi includes both BeSw_BeWi and BeWi_BeSw. BeSw_BeSw increases the total amount of transferred data by x32, while BeSw_BeWi increases that by x2161. BeWi_BeWi increases the total amount of transferred data by about x971, while Ho_BeSw increases that by x3275, much higher than other mitigation techniques. Note, that the results do not mean that BeSw_BeSw is not an effective action. Ho_BeSw, for example, is commonly used when blocker is large (e.g., with a pickup or truck), which has the most negative impact on the underlying connection. Therefore, the gap between the new throughput (as a result of Ho_BeSw) and baseline throughput (throughput of the connection blocked by a large blocker) is very large. Additionally, with Ho_BeSw, the new BS will likely observe no negative impact from the current blocker. BeSw_BeSw is still efficient actions when the blocker is small (e.g., human as we showed in Fig. 3.6(b)). It also results in more increase in throughout than handoff when blocker is small, due to

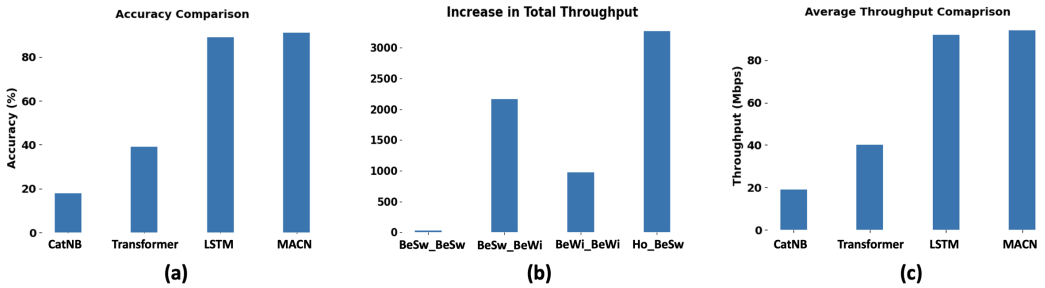


Figure 3.9: Multiple-antenna client network (MACN). (a): Accuracy of different ML models. GRU achieves the highest accuracy; (b): Increase in total transferred data for different actions. (c): Average throughput across all clients, BSs, and blockage events achieved by different ML models.

the shorter duration of the blockage and high cost of handoff.

Throughput Across Different ML Models. To assess the effectiveness of our model, we conducted an evaluation based on the average throughput across all ML models (MACN-based GRU, LSTM, Transfoermer, CategoricalNB (CatNB)). We measure throughput for the duration of the blockage event taking into account blockage type and velocity, and action delay, among others. Fig. 3.9(c) summarizes the average throughput results for each model. Note that these throughput results are averaged across all array configuration scenarios (across all clients and BSs) and blockage events. Our evaluation shows that MACN outperforms all the baselines with an average throughput of 89.9 Mbps. The ratio of increase in throughput is 4.56 with respect to (w.r.t.) CatNB, 2.51 w.r.t. Transformer, and 1.02 w.r.t. LSTM.

Comparison with Other Policies. In SACN evaluation part, we showed that our SACN model outperforms other policies. Here, we conduct similar comparison by comparing MACN with the three alternative policies (policies 2, 3, and 4 from Section 3.1.3) using the average amount of transferred data as metric. For this

purpose, we utilized the same dataset that was used for both training and testing the MACN-based GRU model. Subsequently, we computed the average amount of transferred data for each of the policies. Table 3.7 summarizes the decline in performance when implementing policies 2, 3, and 4 in comparison to our approach, which employs policy 1.

Table 3.7: Comparison of MACN-based GRU model against policies 2, 3, 4 in terms of average percentage drop in the amount of transferred data.

	Human	Cart	Pickup	Truck
Policy 2	-11%	-31%	-7.75%	-2%
Policy 3	-11%	-51.25%	-76%	-67%
Policy 4	-100%	-31%	-7.75%	-2%

In Policy 2, a fixed best technique is employed to maximize the average amount of transferred data for each blockage type. Specifically, we found that the optimal technique for blockage type “human" is BeSw_BeWi, while the best technique for “cart", “pickup" and “truck," is Ho_BeSw. To determine the average amount of data transferred for each blockage type, we applied the corresponding best technique and compared the results with our proposed approach. The first row in Table 3.7 depicts the results. Our results show that when employing only BeSw_BeWi for human blockage, the average amount of transferred data decreases by 11% compared to our approach. Similarly, for cart, the use of Ho_BeSw only, resulted in a 31% decrease in the average amount of transferred data. For pickup, using Ho_BeSw only, resulted in a 7.75% decrease in the average amount of transferred data. Finally, for a truck blocker, the average decrease is 2%.

In Policy 3, a fixed technique is employed to maximize the average amount

of transferred data across all blockage types. To determine the optimal technique, we calculated the average amount of transferred data for the three techniques for each type of blockage and compared the results to identify the technique that provided the maximum performance across all blockage types. Our analysis revealed that BeSw_BeWi was the optimal technique that provided the maximum average amount of transferred data across all blockage types.

To further evaluate the effectiveness of Policy 3, we compared its performance with our approach in the same manner as we did for Policy 2. The results of our analysis (depicted in second row of Table 3.7) indicates a substantial reduction in the average amount of transferred data for larger blockers, i.e., pickups and trucks. Specifically, the average amount of transferred data for pickups and trucks decreased by approximately 76% and 67%, respectively, while the performance drop for the cart blocker decreased by approximately 51.25%. Finally, for the human blockage type, the average amount of transferred data remained at the same level as in Policy 2.

For policy 4, where an arbitrary technique is chosen for all types of blockages, we found that the results of the chosen technique might do well with some blockage types and poorly with other. For example, Ho_BeSw works well with larger blockages, but decreased the performance by 100% for smaller blockages since the cost of taking the Ho_BeSw exceeds the small blockage duration. Therefore, if all of the blockages in an environment are small (e.g., humans), Ho_BeSw provides no gains. Hence, Ho_BeSw decreased the performance for human blockages by 100% and stayed at the same level of policy 2 for the other blockages. The last row

in Table 3.7 captures this.

Empirical Evaluation of Average Loss Ratio. The GRU-based blockage duration prediction framework shows a high level of performance, as observed in terms of residual error values, which exhibit the difference between the actual and predicted blockage duration. To gain a better insight into the impact of model inaccuracies in terms of blockage duration prediction (and the resulting sub-optimal action selection), we simulate the theoretical upper bound on average loss ration introduced in Section 3.3. For this purpose, we utilized the same dataset that was used for both training and testing GRU-based blockage duration prediction model. Subsequently, we computed the upper bound of data loss using Eq. (3.8). We also computed the total amount of data using Eq. (3.9). Lastly, we calculated the actual and predicted average loss ratio by applying Eq. (3.10).

Table 3.8: Analysis of Upper Bound Loss, Total Data, and Average Loss Ratio. PUB stands for Predicted Upper Bound based on our model in Section 3.3.

	Value
Actual Data Loss	22.83 Mb
PUB on Data Loss	25.35 Mb
Total Data	376.5 Mb
Actual Average Loss Ratio	6%
PUB on Avg. Loss Ratio	6.7%

We calculated the upper bound on loss ratio, total data, and actual average loss ratio by taking into account all the four different antenna configurations across the BS and clients. Then, we take the average across all the results. Table 3.8 summarizes our computation results. We observe that the theoretical model indeed provides an upper bound on the actual predicted data loss and average loss ratio. We

further observe that the predicted loss ratio is small and close to the upper bound, indicating a small penalty in terms of transferred data loss due to the inaccuracies.

Chapter 4

Related Work

In this chapter, we delve into the existing research related to the topics that we investigated in this thesis. First, we discuss the previous works of LoS and nLoS channel classification. Then, we explore the prior studies that have investigated the impact of the blockages, the methods to mitigate those blockages' impact, and the prediction of the blockages duration.

4.1 Deep Transfer Learning for Cross-Device Channel Classification in mmWave Wireless

The LoS/nLoS channel classification problem has been extensively studied in previous research [50]–[53]. A typical approach to solve this problem is to rely on statistical parameters of the channel impulse response (CIR), and then conduct a binary hypothesis test to identify the channel condition [40], [54]. Typical parameters that are used include root mean square (RMS) delay spread, mean excess delay, the Rician K-factor, and skewness, among others. More recently, machine learning (ML) based classification approaches are used to solve the LoS/nLoS channel classification problem with higher accuracy, such as support vector machine (SVM)

and relevance vector machine (RVM) [54], [55]. These approaches use the same CIR parameters as the features but don't require any statistical models. All these works are limited to sub-6 GHz systems and moreover require access to CIR data, which may not be easily obtained from existing (standardized) communication packets that are exchanged between COTS wireless devices.

Other recent works have addressed the LoS/nLoS channel classification problem for mmWave systems, e.g., (i) mean shift clustering method [56] uses the Time Difference of Arrival (TDOA) and Angle of Arrival (AoA) of the received signals at multiple BSs, (ii) least-squares support vector machine (LSSVM) method [57] employs mobile signal parameters such as time delay, power, and AoA across multiple BSs, and (iii) Gradient Boosting Decision Trees (GBDT) method [40] gathers mmWave received signal strength, maximum path power, mean/RMS of excess and spread delay, respectively, and kurtosis to solve the channel classification problem. In practice, such data may not be easily available in COTS devices or as part of standardized messages that are exchanged between real devices. For example, TDOA or AoA estimation typically requires sophisticated radios with at least multiple RF chains, whereas none of the COTS that we examined use such radios. Further, channel impulse response data gathering such as mean/RMS path delay estimation may not be easily obtained from standardized communication messages that are exchanged between COTS devices.

In contrast to all prior work, our goal in this part of the thesis is to **only** (i) rely on ordinary COTS mmWave devices, (ii) solve the problem leveraging a single BS, and (iii) use standard compatible communication messages (with no additional

overhead) to solve the channel classification problem. Specifically, we rely on routine periodic sector sweep messages (which are embedded in all mmWave standards), which then allows us to obtain periodic channel classification estimates (e.g., every 100 msec in real mmWave WiFi devices).

4.2 Gated Recurrent Units for Blockage Mitigation and Duration Prediction in mmWave Wireless

Blockage mitigation problem has been intensively investigated in prior works. Here, we discuss the mmWave blockages and their impacts. Then, we discuss the studies that investigate different methods to mitigate blockages. Next, we discuss the previous studies in predicting blockage duration.

4.2.1 MmWave Blockages

MmWave signal can be easily degraded or completely blocked by a variety of blockages [58]. These blockages can be vary depending on the environment. For example, in the urban scenarios, blockages such as buildings and vehicles can be prominent while human and furniture can be prominent for the indoor scenarios. These different type of blockages can be associated with high impact on the mmWave link (e.g., human body alone can highly decrease the overall mmWave communication performance). Many prior works have focused on studying the impact of those blockages on the mmWave link in different environment scenarios. Some of those works have studied impact of human body on the performance of mmWave communication [59]–[61]. Other works have investigated the effects

of different blockages (e.g., road bridge and buildings) in different urban scenarios [62]–[65]. Mmwave communication is not only affected by larger blockages, but it could be also affected by smaller blockages (e.g., road signs) [66].

4.2.2 Methods to Mitigate Blockages

Many of prior works have stated that the negative impact of blockages on mmWave communication can be mitigated with three main techniques (beam switching, handoff, and beam widening). Each of those works has studied one technique at a time as a solution to mitigate the impact of the blockages.

Beam Switching. Beam switching/adaptation is one of the key techniques to mitigate blockages [30]. It refers to switching the beam to another direction that is not blocked (Fig. 3.1(c)). This technique has been studied in several prior works optimized through: (i) leveraging an out-of-band radio [67], (ii) sensing the reflective environment [68], (iii) using model-driven methods [69], and (iv) employing deep learning based on a given client's location and environment [70], [71]. Beam switching can be a reliable blockage mitigation technique in some scenarios and environments. On the other hand, it may fail when the blocker is large (e.g., a truck) or when there are few reflectors in the environment.

Handoff. Handoff (BS adaptation, Fig. 3.1(g)) is another widely used technique to mitigate blockages. It involves handing off the client to another BS when blockage is detected. Several research works (e.g., [72]–[81]) have optimized the handoff decision making process by using Reconfigurable Intelligent Surfaces (RIS) or employing deep learning models based on channel state information, the

client location, and other network parameters. Handoff is an efficient blockage mitigation technique in many scenarios. However, there are many instances in which handoff may fail. For example, when the time to complete handoff is more than the duration of the blockage event, then handoff would be unnecessary. The problem can worsen in higher mobility environments, where the handoff frequency may become too high.

Beam Widening. When a blockage happen, one or both of the two-ends (BS and client) can increase the half-power beamwidth of their current beams (Fig. 3.1(d)-(f)). This technique has been investigated in prior works for mitigating blockages, e.g., the work in [82] has experimentally shown the benefits of beam widening to mitigate blockages in indoor environments, the work in [83] proposes a theoretical model based on multi-armed bandits to adjust the beamwidth on both clients and BSs, and the work in [84] uses partial activation of antenna arrays to widen the beams. However, beam widening reduces the beamforming gain. Additionally, when the blocker is large or located close to the client or the base station, beam widening might still be ineffective.

While previous works have attempted to tackle blockages, they have generally optimized only one technique at a time. Although a technique may work well in certain scenarios, it may fall short in others, suggesting the need to address the problem holistically. Our goal in this part of the thesis is to integrate the above techniques into one system. The resulting framework handles various scenarios and blockage types effectively, providing a more robust approach to mitigating blockages.

4.2.3 Blockage Timing Estimation

Blockage Duration Prediction. Identifying the blockage duration can help in maximizing the overall wireless network performance by incorporating it an overall blockage mitigation framework. For example, the solution can revert back to the original settings (BS associations and beams) once the blockage event ends, increasing the amount of data transferred. This blockage duration prediction problem is less addressed by the research community. The work in [85] uses a Markov chain model to determine the average blockage duration. Similarly, the work in [86] uses a theoretical model to estimate the average fraction of time that the LoS path is blocked. Our work aims to determine the exact blockage duration, which can be very different across different types of blockers. The work in [45] tries to proactively identify the type of incoming blockage based on how severe it is. The severity is quantified using a discrete blockage-severity index, which is derived from the average blockage time interval measured for each object. Next, authors group objects based on each object's average blockage time interval, each of which represents a class label. However, predicting the time interval of blockages in a discrete manner and tied to the blocker type, specifically in a time sensitive system such as mmWave wireless, can have severe downside effects. For example, predicting the blocker as a bus with an average 1300 *msec* duration can result in significant drop in the amount of transferred data when the true blocker is a skateboard with an average 180 *msec* blockage duration. The authors in [87] introduce a meta-learning framework that predict if a blockage is going to happen in

a fixed time window, and then predict if the link is going to be blocked for the entire time window. This approach might carry a high negative impact if the prediction is wrong because the whole time window will be affected. For example, predicting that a blockage is going to happen and blocks the link for 1000 *msec* when there is no blockage can result in significant drop in the amount of transferred data for the whole time window (1000 *mse*). In our work, we consider a continuous time GRU-based framework that solves the blockage duration prediction as a regression problem. We show that our approach results in a very low mean residual error across all different types of blockers, speeds, and antenna array configurations at both the BS and client.

Chapter 5

Limitations and Future Work

In this chapter, we discuss several limitations and challenges encountered during our exploration of the topics presented in this thesis. We also outline potential ideas that could be pursued in future research.

5.1 Deep Transfer Learning for Cross-Device Channel Classification in mmWave Wireless

During our research, we encountered limitations that influenced both the scope and depth of our analysis. A key challenge is that our environment scenarios may not fully capture the complexities and all real-world scenarios. Extending the validation to more dynamic and less predictable environments could help in understanding the adaptability of the model under different conditions and improve its robustness.

Moreover, the study's future work could focus on refining the proximity estimation techniques mentioned. Improving these algorithms by incorporating more detailed environmental variables and advanced modeling techniques could provide more accurate localization and tracking capabilities, which are essential for many

mmWave applications.

5.2 Gated Recurrent Units for Blockage Mitigation and Duration Prediction in mmWave Wireless

Throughout our investigation, we faced several challenges and limitations that stemmed from various sources, including constraints in data availability, scenarios, and clients and BSs setup. These challenges and limitations are detailed as follows.

First, the simulator's ability to mirror the nuances of real-world IIoT environments is limited. Factors such as varying interference patterns, unexpected equipment behavior, and environmental influences can significantly affect signal blockage, which may not be fully captured by the simulator. To mitigate this limitation, we calibrated the simulator using parameters from typical industry-standard models to enhance its realism.

Second, gathering real-world IIoT data to further validate our findings can be limited since real IIoT systems may have proprietary, security, and privacy issues. While simulation is a practical approach, securing partnerships, in future, with IIoT vendors and operators could provide access to real-world data and environments to further validate our findings.

Third, the results obtained may not directly cover all possible IIoT scenarios such as more complex IIoT environment due to the computational limitations that restrict the size of the simulated environment. We recommend future studies to use distributed simulation techniques to overcome these computational challenges and simulate larger networks.

Finally, our work in this part of the thesis assumed static clients and base stations in an IIoT setting, which are initially in LoS condition with respect to one another. We captured network dynamics by introducing mobile blockers of different sizes and speeds. As part of our future work, we plan to investigate the more complex deployments such as urban scenarios with mobile clients that may not initially be in LoS condition with respect to base stations.

Chapter 6

Conclusion

This thesis makes two key contributions to enable high communication data rate by overcoming the obstacles that degrade the performance of mmWave communication systems. First, we propose a **t**ransfer learning technique for **L**oS/**n**LoS **c**hannel **c**lassification (t-LNCC) framework that accurately predicts the channel condition (LoS or nLoS). This framework is trained using simulated data and validated on real-world data. One benefit of using transfer learning technique is that it solves the issue of low or unavailability of real-world dataset especially for new technologies such as mmWave in wireless systems. Another benefit is that the framework can work well with devices that are yet to be commercially released and there is lack of crowd-sourced data from such devices that can be used for channel classification.

Next, we propose several deep learning frameworks that can predict the proper action for oncoming blockage, taking into account LoS/nLoS channel condition. Our frameworks integrate different actions that can handle different type of blockages. Our frameworks leverages Gated Recurrent Units, which are a gating mechanism in recurrent neural networks and show great promise in determining proper action to mitigate blockages. Unlike traditional techniques, which introduce over-

head and increase latency, and other state-of-the-art techniques that introduce one action for all type of blockages, our frameworks can proactively take the optimal action for each specific blockage case without incurring additional overhead. In addition, we investigate the blockage duration prediction problem and propose another GRU-based framework that performs very well with a small mean residual error in predicting the blockage duration.

Bibliography

[1] J. Yang, B. Ai, I. You, M. Imran, L. Wang, K. Guan, D. He, Z. Zhong, and W. Keusgen, "Ultra-reliable communications for industrial internet of things: Design considerations and channel modeling," in *IEEE Network*, 2019.

[2] M. Alabadi, A. Habbal, and X. Wei, "Industrial internet of things: Requirements, architecture, challenges, and future research directions," in *IEEE Access*, 2022.

[3] Y. Lu, P. Richter, and E. S. Lohan, "Opportunities and challenges in the industrial internet of things based on 5G positioning," in *IEEE 8th International Conference on Localization and GNSS ICL - GNSS*, 2018.

[4] S. Zeb, A. Mahmood, H. Pervaiz, S. A. Hassan, M. I. Ashraf, Z. Li, and M. Gidlund, "On TOA-based ranging over mmwave 5G for indoor industrial iot networks," in *IEEE Globecom Workshops*, 2020.

[5] Q. Chen, X. Xu, H. Jiang, and X. Liu, "An energy-aware approach for industrial internet of things in 5G pervasive edge computing environment," in *IEEE Transactions on Industrial Informatics*, 2021.

[6] Y. Lin, X. Wang, H. Ma, L. Wang, F. Hao, and Z. Cai, "An efficient approach to sharing edge knowledge in 5G-Enabled industrial internet of things," in *IEEE Transactions on Industrial Informatics*, 2023.

[7] A. Mahmood, S. F. Abedin, T. Sauter, M. Gidlund, and K. Landernäs, "Factory 5G: A review of industry-centric features and deployment options," in *IEEE Industrial Electronics Magazine*, 2022.

[8] H. R. Chi, C. K. Wu, N. F. Huang, K. F. Tsang, and A. Radwan, "A survey of network automation for industrial internet-of-things toward industry 5.0," in *IEEE Transactions on Industrial Informatics*, 2023.

[9] B. S. Khan, S. Jangsher, A. Ahmed, and A. Al-Dweik, "URLLC and eMBB in 5G industrial IoT: A survey," in *IEEE Open Journal of the Communications Society*, 2022.

[10] D. Yang, A. Mahmood, S. A. Hassan, and M. Gidlund, "Guest editorial: Industrial IoT and sensor networks in 5G-and-beyond wireless communication," in *IEEE Transactions on Industrial Informatics*, 2022.

[11] S. Li, Q. Ni, Y. Sun, G. Min, and S. Al-Rubaye, "Energy-efficient resource allocation for industrial cyber-physical IoT systems in 5G era," in *IEEE Transactions on Industrial Informatics*, 2018.

[12] H. Ren, C. Pan, T. Deng, M. Elkashlan, and A. Nallanathan, "Joint pilot and payload power allocation for Massive-MIMO-Enabled URLLC IIoT networks," in *IEEE Journal on Selected Areas in Communications*, 2020.

[13] R. W. L. Coutinho and A. Boukerche, "Transfer learning for disruptive 5G-enabled industrial internet of things," in *IEEE Transactions on Industrial Informatics*, 2022.

[14] P. Kortoci, A. Mehrabi, C. Joe-Wong, and M. D. Francesco, "Incentivizing opportunistic data collection for time-sensitive IoT applications," in *Proceedings of IEEE SECON*, 2021.

[15] P. Kortoci, L. Zheng, C. Joe-Wong, M. D. Francesco, and M. Chiang, "Fog-based data offloading in urban IoT scenarios," in *Proceedings of IEEE INFOCOM*, 2019.

[16] M. Gapeyenko, A. Samuylov, M. Gerasimenko, D. Moltchanov, S. Singh, E. Aryafar, S. Yeh, N. Himayat, S. Andreev, and Y. Koucheryavy, "Analysis of human-body blockage in urban millimeter-wave cellular communications," in *Proceedings of IEEE ICC*, 2016.

[17] M. Gapeyenko, A. Samuylov, M. Gerasimenko, D. Moltchanov, S. Singh, M. R. Akdeniz, E. Aryafar, S. Andreev, N. Himayat, and Y. Koucheryavy, "Spatially-consistent human body blockage modeling: A state generation procedure," in *IEEE Transactions on Mobile Computing*, 2020.

[18] A. Alyosef, S. Rizou, Z. D. Zaharis, P. I. Lazaridis, A. M. Nor, O. Fratu, S. Halunga, T. V. Yioultsis, and N. V. Kantartzis, "A survey on the effects of human blockage on the performance of mmWave communication systems," in *IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2022.

[19] C. Gustafson and F. Tufvesson, "Characterization of 60 GHz shadowing by human bodies and simple phantoms," in *European Conference on Antennas and Propagation*, 2012.

[20] V. Raghavan, S. Noimanivone, S. K. Rho, B. Farin, P. Connor, R. A. Motos, Y. Ou, K. Ravid, M. A. Tassoudji, O. H. Koymen, and J. Li, "Hand and body blockage measurements with form-factor user equipment at 28 ghz," in *IEEE Transactions on Antennas and Propagation*, 2022.

[21] T. Mshvidobadze, "Evolution mobile wireless communication and LTE networks," in *IEEE 6th International Conference on Application of Information and Communication Technologies (AICT)*, 2012.

[22] V. Yadav, L. Kumar, and P. Kumar, "Evolution and development of wireless communication system," in *IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, 2019.

[23] F. H. Blecher, "Advanced mobile phone service," in *IEEE Transactions on Vehicular Technology*, 1980.

[24] J. Scourias, "Overview of the global system for mobile Communications," in *University of Waterloo*, 1995.

[25] A. C. Chen, "Overview of code division multiple access technology for wireless communications," in *IEEE 24th Annual Conference of the IEEE Industrial Electronics Society*, 1998.

[26] E. Pittampalli, "Third-generation CDMA wireless standards and harmonization," in *Bell Labs Technical Journal*, 1999.

[27] A. H. Khan, M. A. Qadeer, J. A. Ansari, and S. Waheed, "4G as a next generation wireless network," in *International Conference on Future Computer and Communication*, 2009.

[28] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "LTE-advanced: Next-generation wireless broadband technology," in *IEEE Wireless Communications*, 2010.

[29] S. Mansoor, M. F. Andreas, S. Peter, H. Thomas, Z. Peiying, D. Silva, T. Fredrik, B. Anass, and W. Gerhard, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," in *IEEE Journal on Selected Areas in Communications*, 2017.

[30] S. Srinivasan, X. Yu, A. Keshavarz-Haddad, and E. Aryafar, "Fair initial access design for mmWave wireless," in *IEEE ICNP*, 2020.

[31] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11ad: directional 60 GHz communication for multi-gigabit-per-second WiFi," in *IEEE Communications Magazine*, 2014.

[32] Y. Ghasempour, C. R. D. Silva, C. Cordeiro, and E. W. Knightly, "IEEE 802.11ay: next-generation 60 GHz communcation for 100 Gbps Wi-Fi," in *IEEE Communications Magazine*, 2017.

[33] D. Stursa and P. Dolezel, "Comparison of relu and linear saturated activation functions in neural network for universal approximation," in *22nd International Conference on Process Control (PC19)*, 2019.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations*, 2015.

[35] C. Tan, F. Sun, T. Kong, W. Zhang, and C. Yang, "A survey on deep transfer learning," in *The 27th International Conference on Artificial Neural Networks (ICANN)*, 2018.

[36] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," in *3GPP Technical Report, 38.901 (Release 15)*, 2018.

[37] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," in *The Journal of Machine Learning Research*, 2016.

[38] F. Wen, H. Wymeersch, B. Peng, W. P. Tay, H. C. So, and D. Yang, "A survey on 5G massive MIMO localization," in *Digital Signal Processing*, 2019.

[39] D. Steinmetzer, D. Wegemer, M. Schulz, J. Widmer, and M. Hollick, "Compressive millimeter-wave sector selection in off-the-shelf IEEE 802.11 ad devices," in *the 13th International Conference on emerging Networking EXperiments and Technologies*, 2017, pp. 414–425.

[40] A. Huang, L. Tian, T. Jiang, and J. Zhang, "NLOS identification for wideband mmwave systems at 28 GHz," in *IEEE VTC*, 2019.

[41] "Nexmon: The C-based firmware patching framework," https://nexmon.org.

[42] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.

[43] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *IEEE Transactions on Neural Networks*, 2010.

[44] K. Cho, B.V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *arXiv:1406.1078*, 2014.

[45] S. Wu, M. Alrabeiah, C. Chakrabarti, and A. Alkhateeb, "Blockage prediction using wireless signatures: Deep learning enables real-world demonstration," in *IEEE Open Journal of the Communications Society*, 2022.

[46] A. Almutairi, S. Srinivasan, A. Keshavarz-Haddad, and E. Aryafar, "Deep transfer learning for cross-device channel classification in mmwave wireless," in *IEEE MSN*, 2021.

[47] A. Almutairi, A. Keshavarz-Haddad, and E. Aryafar, "Gated recurrent units for blockage mitigation in mmWave wireless," in *Proceedings of ACM MobiWac*, 2023.

[48] "Wireless insite:," https://www.remcom.com/wireless-insite-propagation-software.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and Ł. Kaiser, "Attention is all you need," in *The 31st International Conference on Neural Information Processing Systems*, 2017.

[50] Q. Zheng, R. He, B. Ai, C. Huang, W. Chen, Z. Zhong, and H. Zhang, "Channel non-line-of-sight identification based on convolutional neural networks," in *IEEE Wireless Communications Letters*, 2020.

[51] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Non-line-of-sight identification and mitigation using received signal strength," in *IEEE Transactions on Wireless Communications*, 2015.

[52] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo, "UWB NLOS/LOS classification using deep learning method," in *IEEE Communications Letters*, 2020.

[53] M. Stahlke, S. Kram, C. Mutschler, and T. Mahr, "NLOS detection using UWB channel impulse responses and convolutional neural networks," in *International Conference on Localization and GNSS (ICL-GNSS)*, 2020.

[54] S. Marano, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on uwb experimental data," in *IEEE Journal on Selected Areas in Communications*, 2010.

[55] N. V. Thang, Y. Jeong, H. Shin, and M. Z. Win, "Machine learning for wideband localization," in *IEEE Journal on Selected Areas in Communications*, 2015.

[56] B. Hu, H. Tian, and S. Fan, "Millimeter wave LOS/NLOS identification and localization via mean-shift clustering," in *IEEE PIMRC*, 2019.

[57] C. Benny, A. Simon, W. Stephen, and B. Mark, "NLOS identification and mitigation for geolocation using least-squares support vector machines," in *IEEE WCNC*, 2017.

[58] G. R. MacCartney, T. S. Rappaport, and S. Rangan, "Rapid fading due to human blockage in pedestrian crowds at 5G millimeter-wave frequencies," in *IEEE GLOBECOM*, 2017.

[59] A. Alyosef, S. Rizou, Z. D. Zaharis, P. I. Lazaridis, A. M. Nor, O. Fratu, S. Halunga, T. V. Yioultsis, and N. V. Kantartzis, "A survey on the effects of human blockage on the performance of mmWave communication systems," in *IEEE BlackSeaCom*, 2021.

[60] A. B. Zekri, R. Ajgou, and M. Hettiri, "Impact of azimuth and elevation half power beam width on human blockage scenarios in mmWave channels," in *IEEE 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)*, 2020.

[61] X. Liu, Y. Zhang, T. Jiang, L. Yu, J. Zhang, and L. Xia, "Multi-person blockage loss modeling at millimeter-wave band," in *IEEE 95th Vehicular Technology Conference*, 2022.

[62] J. Kim, D. Yan, K. Guarr, D. He, G. Noh, S. Choi, and H. Chung, "Effects of signal blockage by a road bridge on mmWave vehicular communications," in *IEEE 13th International Conference on Information and Communication Technology Convergence (ICTC)*, 2022.

[63] I. K. Jain, R. Kumar, and S. S. Panwar, "The impact of mobile blockers on millimeter wave cellular systems," in *IEEE Journal on Selected Areas in Communications*, 2019.

[64] C. Garcia-Ruiz, O. Munoz, and A. Pascual-Iserte, "Effect of correlated building blockages on the ergodic capacity of mmWave systems in urban scenarios," in *IEEE Transactions on Vehicular Technology*, 2022.

[65] C. G. Ruiz, A. Pascual-Iserte, and O. Munoz, "Analysis of blocking in mmWave cellular systems: Application to relay positioning," in *IEEE Transactions on Communications*, 2020.

[66] F. Alsaleem, J. S. Thompson, D. I. Laurenson, S. K. Podilchak, and C. A. Alistarh, "Small-size blockage measurements and modelling for mmWave communications systems," in *IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020.

[67] S. Sur, I. Pefkianakis, X. Zhang, and K. Kim, "Wifi-assisted 60 GHz wireless networks," in *ACM MOBICOM*, 2017.

[68] A. Zhou, X. Zhang, and H. Ma, "Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering," in *IEEE INFOCOM*, 2017.

[69] A. Zhou, L. Wu, S. Xu, H. Ma, T. Wei, and X. Zhang, "Following the shadow: Agile 3-d beam-steering for 60 GHz wireless networks," in *IEEE INFOCOM*, 2018.

[70] S. Rezaie, C. N. Manchón, and E. d. Carvalho, "Location- and orientation-aided millimeter wave beam selection using deep learning," in *IEEE ICC*, 2020.

[71] A. O. Kaya and H. Viswanathan, "Deep learning-based predictive beam management for 5G mmwave systems," in *IEEE Wireless Communication and Networking Conference WCNC*, 2021.

[72] L. Jiao, P. Wang, A. Alipour-Fanid, H. Zeng, and K. Zeng, "Enabling efficient blockage-aware handover in RIS-assisted mmwave cellular networks," in *IEEE International Journal of Advanced Computer Science and Applications*, 2022.

[73] M. S. Mollel, S. Kaijage, and M. Kisangiri, "Deep reinforcement learning based handover management for millimeter wave communication," in *International Journal of Advanced Computer Science and Applications*, 2021.

[74] Y. K. S. and T. Ohtsuki, "Influence and mitigation of pedestrian blockage at mmwave cellular networks," in *IEEE Transactions on Vehicular Technology*, 2020.

[75] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover control in wireless systems via asynchronous multiuser deep reinforcement learning," in *IEEE Internet of Things Journal*, 2018.

[76] Y. Sun, G. Feng, S. Qin, Y. C. Liang, and T. P. Yum, "The smart handoff policy for millimeter wave heterogeneous cellular networks," in *IEEE Transactions on Mobile Computing*, 2018.

[77] C. L. Vielhaus, J. V. S. Busch, P. Geuer, A. Palaios, J. Rischke, D. F. Külzer, V. Latzko, and F. H. P. Fitzek, "Handover predictions as an enabler for anticipatory service adaptations in next-generation cellular networks," in *ACM MobiWac*, 2022.

[78] C. Lee, J. Jung, and J. M. Chung, "Intelligent dual active protocol stack handover based on double DQN deep reinforcement learning for 5G mmWave networks," in *IEEE Transactions on Vehicular Technology*, 2022.

[79] R. Tang, C. Qi, and Y. Sun, "Blockage prediction and fast handover of base station for millimeter wave communications," in *IEEE Communications Letters*, 2023.

[80] S. B. Iqbal, S. Nadaf, A. Awada, U. Karabulut, P. Schulz, and G. P. Fettweis, "On the analysis and optimization of fast conditional handover with hand blockage for mobility," in *IEEE Access*, 2023.

[81] T. Zhang, J. Liu, and F. Gao, "Vision aided beam tracking and frequency handoff for mmwave communications," in *IEEE INFOCOM*, 2022.

[82] O. Bshara, Y. Liu, I. Tekin, B. Taskin, and K.R. Dandekar, "MmWave antenna gain switching to mitigate indoor blockage," in *IEEE USNC/URSI*, 2018.

[83] M. Feng, B. Akgun, I. Aykin, and M. Krunz, "Beamwidth optimization for 5G NR millimeter wave cellular networks: A multi-armed bandit approach," in *Proceedings of IEEE ICC*, 2021.

[84] H. Chung, J. Kang, H. Kim, Y. M. Park, and S. Kim, "Adaptive beamwidth control for mmWave beam tracking," in *IEEE Communications Letters*, 2021.

[85] C. Tunc, M. F. Ozkoc, and S. Panwar, "Millimeter wave coverage and blockage duration analysis for vehicular communications," in *Proceedings of IEEE VTC*, 2019.

[86] D. Moltchanov and A. Ometov, "On the fraction of LoS blockage time in mmWave systems with mobile users and blockers," in *Proceedings of IFIP WWIC*, 2018.

[87] A. E. Kalør, O. Simeone, and P. Popovski, "Prediction of mmWave/THz link blockages through meta-learning and recurrent neural networks," in *IEEE Wireless Communications Letters*, 2021.