

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

Ahmed Almutairi¹, Suresh Srinivasan¹, Alireza Keshavarz-Haddad², and Ehsan Aryafar¹

¹Portland State University, Computer Science Department, Portland, OR, USA

²Shiraz University, Electrical and Computer Engineering Department, Shiraz, Iran

Abstract—Identifying whether the 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) has many applications, e.g., it can be used in device localization. Prior works have addressed this problem, but they are primarily limited to sub-6 GHz systems, assume sophisticated radios on the devices, incur additional communication overhead, and/or are specific to a single class of devices (e.g., a specific smartphone). In this paper, we address this channel classification problem for wireless devices with mmWave radios. Specifically, we show that existing beamforming training messages that are exchanged periodically between mmWave wireless devices can also be used in a deep learning model to solve the channel classification problem with no additional overhead. We then extend our work 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 accuracy of t-LNCC is more than 95% across three different COTS wireless devices, when there is a small sample of labeled data for each device. We finally show the application of our classification problem in estimating the distance between two wireless devices, which can be used in localization.

I. 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, which can boost communication capacity. 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 beams¹ 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 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).

Our goal in this paper is to use this existing message passing overhead (at the beginning of each beam training interval)

to solve the LoS/nLoS channel classification problem. Addressing this problem is beneficial in many other applications. For example, it can be used to solve the proximity (distance) estimation between two devices, which can itself be used by a wide variety of localization algorithms [1]. In addition, a base station (BS) with knowledge about LoS/nLoS channel condition can use it to reduce the overhead of beam search (and hence initial access) in mmWave systems [2] or use the information to better adapt the transmission strategy to maintain the link quality of service [3].

Moreover, our goal in this paper is 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.

A. Related Work

A typical approach to solve the LoS/nLoS channel classification 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 [3], [4]. 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) [4], [5]. 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 [6] uses the Time Difference of Arrival (TDOA) and Angle of Arrival (AoA) of the received signals

¹We use the words “beams” and “sectors” interchangeably.

at multiple BSs, (ii) least-squares support vector machine (LSSVM) method [7] employs mobile signal parameters such as time delay, power, and AoA across multiple BSs, and (iii) Gradient Boosting Decision Trees (GBDT) method [3] 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 paper 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).

B. Research Contributions

In this paper, we focus on wireless devices equipped with 802.11 ad/ay chipsets, and use the existing communication messages that are periodically exchanged between such devices to solve the channel classification problem. While we focus on WiFi, we emphasize that the methods proposed in this paper are also applicable to 5G cellular chipsets (devices). In particular, our key contributions can be summarized as follows:

- **Data Gathering:** We gathered empirical data of mmWave sectors' SNR values in both indoor and outdoor environments using three different COTS devices over a two-month period. In addition, we developed a simulator, calibrated with 3GPP specifications [8], and used it to generate the corresponding simulated data in different environments.
- **Baseline LoS/nLoS Channel Classification (base-LNCC):** We design a deep neural network based model for mmWave channel classification and train the model using simulated data. We show that base-LNCC has more than 98% accuracy when tested over unseen simulated data.
- **Transfer Learning for LoS/nLoS Channel Classification (t-LNCC):** We design a transfer learning model (t-LNCC) that is trained on a source domain (simulated data) to learn the features of the target domain (empirical data). We then develop three variations of t-LNCC distinguished by whether they use labeled empirical data as part of model training. We show that using a few samples of labeled empirical data (referred to as semi-supervised t-LNCC) results in more than 93% accuracy across all COTS devices.
- **MmWave Proximity Detection:** We demonstrate the application of LoS/nLoS channel classification problem by using

it to solve the proximity (distance) estimation between two points. The resulting distance estimate can then be used in other applications, e.g., localization through triangulation.

We elaborate on these contributions next, starting with some background on 802.11 ad/ay, followed by LoS-nLoS channel classification through base-LNCC and t-LNCC, applications of LoS/nLoS channel classification, evaluation, and conclusion.

II. BACKGROUND

We train and evaluate our classification models 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 [9]. 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 huge propagation and penetration losses in mmWave bands. An array of antennas is 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(a).

MmWave Initial Access. Initial access in mmWave systems is a procedure that allows a client device to discover a cell, helps the BS and clients to find appropriate beams to communicate with each other, and allows the BS to send management and control information to all the clients.

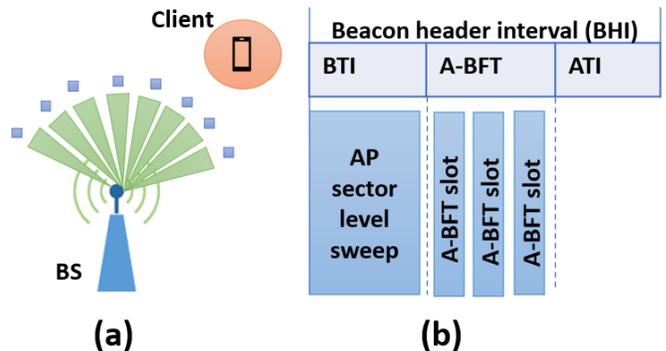


Fig. 1. (a): During the beacon transmission interval (BTI), BS sequentially sends sector sweep (SSW) frames on each of its sectors. Each client uses an omni antenna pattern and records the beam ID and signal strength of all the received SSW frames; (b): Association beamforming training (A-BFT) is composed of a few slots. A client randomly chooses an A-BFT slot to conduct its sector sweep.

In 802.11 ad/ay, this is handled at the beginning of each beacon interval (BI) [10], [11]. 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 than the DTI, e.g., only a few msec.

The BHI, as depicted in Fig. 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 paper, we only consider the BTI sub-interval, which allows a client to use the data exchanged during this interval to identify its channel condition (LoS vs nLoS) with respect to the BS. Similarly, the BS can use the message passing during the A-BFT interval to identify its channel condition with respect to the client. The BTI is described as follows.

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 frame) received from the BS. Fig. 1(a) shows this operation. Our classification model utilizes the measured SNR values of all received beams to identify the channel condition at the client device. So, the input to our classification model is a vector of SNR values and the corresponding beam IDs.

III. LOS - NLOS CHANNEL CLASSIFICATION

In this section, 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.

A. Base-LNCC

System Design. The wireless channel condition is categorized into two classes (LoS or nLoS). Thus, our classification problem is binary. Deep learning (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 [12]. ReLU is defined as follows:

$$S(z) = \max(0, z) \quad (1)$$

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:

$$S(z) = \frac{1}{1 + e^z} \quad (2)$$

For optimization, Adam, a gradient optimization technique widely used in DL [13], is used. Finally, the loss function that we choose is binary cross-entropy which works well with the sigmoid function:

$$L = -(Y \log(p(Y)) + (1 - Y) \log(1 - p(Y))) \quad (3)$$

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.

B. T-LNCC

DL involves complicated processes 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 [14]: (i) instance-based transfer learning, (ii) mapping-based transfer

learning, (iii) network transfer, and (iv) adversarial-based transfer learning.

In this paper, we adopt adversarial learning for domain adaptation that finds the transferable features of the two domains which makes them indistinguishable. Our key idea is to use a simulator calibrated with standardized channel model [8] 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 [15]: (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 paper, we employ a modified model of domain-adversarial neural network (DANN) used in [15] 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 [15].

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 will call t-LNCC, consists of three parts, as depicted in Fig. 2. 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.

C. 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 [16]). 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. 3 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 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

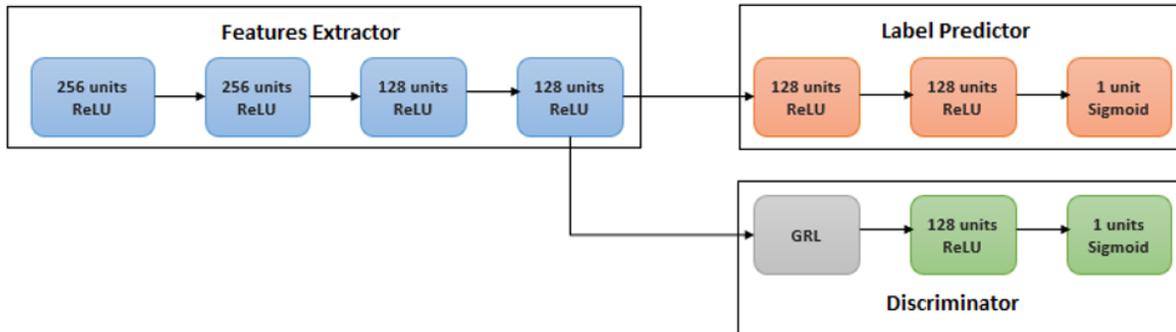


Fig. 2. t-LNCC is a domain-adversarial neural network, which consists of three parts. Blue boxes show the features extractor layers, the orange boxes show the classifier layers, and the green boxes show the domain discriminator layers in addition to the GRL layer which is in gray color. Each of the boxes shows the number of units that layer has along with the activation function.

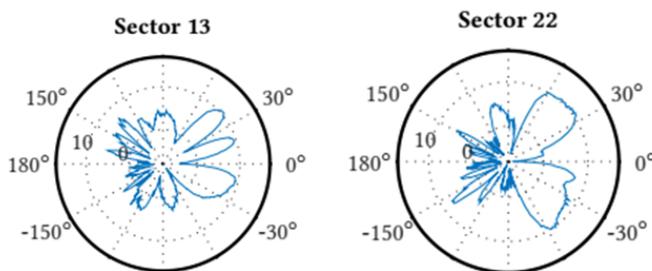


Fig. 3. Measured SNR in azimuth plane of sectors 13 and 22 [17].

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 IV-C.

IV. 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 classification 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.

A. 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 [8]. The channel model is a statistical model. In each simulation run, we have a single BS and 20 clients. We set the carrier frequency to 60 GHz and set the noise figure to 7 dB. For each run, we use different phased antenna array sizes for all devices within

that run. The phased antenna array sizes that we consider are 8-antenna (4x2), 16-antenna (8x2) and (4x4), and 32-antenna (8x4). We set BSs to act as Tx with directional antenna and clients as Rx with an omnidirectional antenna. Each BS uses 36 beams to cover 120° of azimuth and 120° of elevation. We ran the simulator for 4000 times and each time, we collect 20 samples (total 80000 samples). Each sample is a vector of 72 elements which are the SNR values and their IDs. After cleaning (e.g., get rid of 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 dataset, 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 beams' SNR values, which are obtained at the beginning of every beacon interval. We compare Base-LNCC against Gradient Boosting Decision Tree (GBDT) [3], 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%.

B. 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

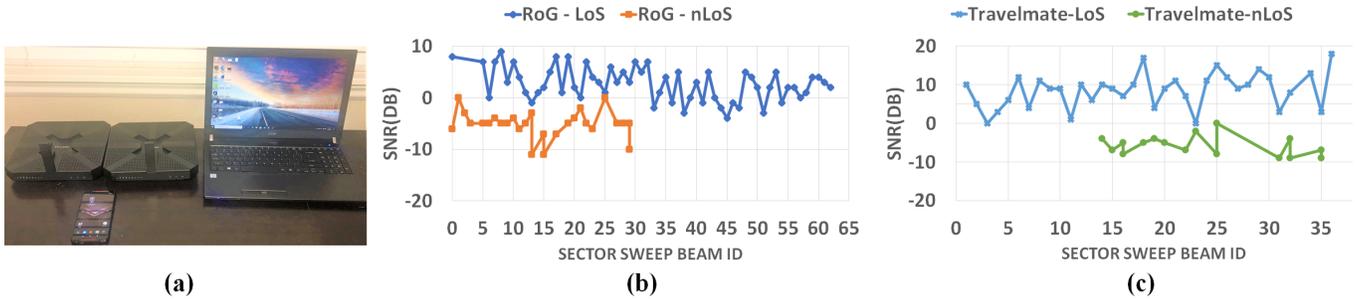


Fig. 4. (a): We gather data leveraging Talon AD-7200 router, Acer TravelMate laptop, ASUS-RoG Phone and AD-7200 as client; (b): Sample SNR of RoG phone’s LoS and nLoS beams versus sector ID; (c): Sample SNR of Travelmate beams versus sector ID.

locations in and around a single family home.

Devices. We use a Talon-AD7200 as 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. 4(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 [18] 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 to amend patches in C language rather than 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 [18]. 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.

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 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 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 III, 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 [19] 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 [20] 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. 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. 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%	82.6%	86.76%	78%
T-LNCC (unsupervised)		93%	88.3%	90.23%	67%
T-LNCC (semi-supervised)		96%	95.89%	95.35%	97%

C. 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 III-C 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 [21]. A typical SNR received at the BS is shown in Fig. 3. 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. 5.

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. 6 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 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.

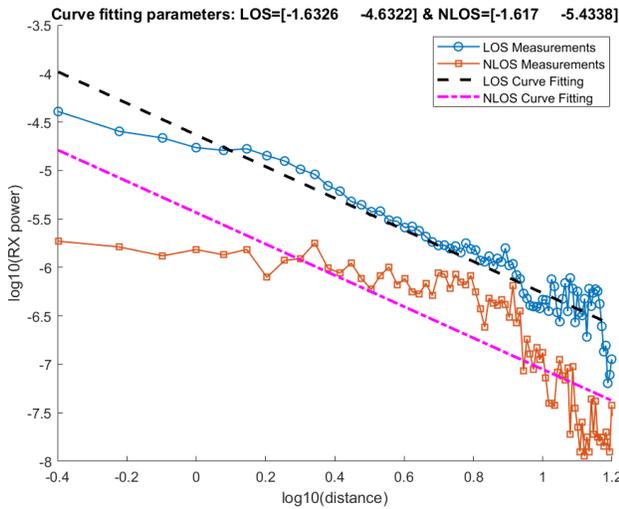


Fig. 5. Curve fitting for LoS and nLoS condition based on SNR values with respect to the distance. We use curve fitting to create a linear approximation of the path loss formula. Note that the distance (x-axis) uses a logarithmic unit.

V. CONCLUSION

We addressed the problem of channel classification in mmWave systems. We investigated the accuracy of DL in solving the problem and showed that the solution provides high accuracy by using only beam SNR values that are readily available at COTS devices as part of the periodic beam search process. We also proposed a transfer learning model that uses simulated data with/without empirical data to solve the classification problem in real COTS devices. We showed that the semi-supervised transfer learning model has more than

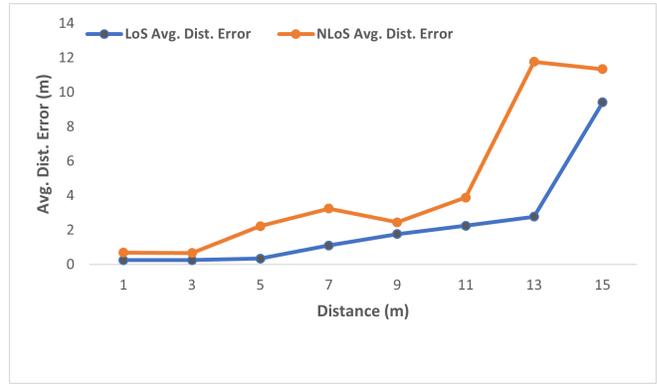


Fig. 6. Average distance error of the LoS and NLoS channel conditions.

95% accuracy in any device. Finally, we showed an application of LoS/nLoS channel classification in solving the proximity problem for mmWave networks.

As part of our future work, we plan to propose methods to increase the accuracy of our proximity estimation algorithm.

VI. ACKNOWLEDGEMENTS

This research was supported in part by an NSF award (CNS-1942305).

REFERENCES

- [1] V. Marco and C. Dajana, "Mmwaves rssi indoor network localization," in *Proceedings of IEEE ICC*, 2014.
- [2] T. S. Cousik, V. K. Shah, J. H. Reed, T. Erpek, and Y. Sagduyu, "Fast initial access with deep learning for beam prediction in 5g mmwave networks," in *arXiv:2006.12653*, 2020.
- [3] A. Huang, L. Tian, T. Jiang, and J. Zhang, "Nlos identification for wideband mmwave systems at 28 ghz," in *Proceedings of IEEE VTC*, 2019.
- [4] 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.
- [5] 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.
- [6] B. Hu, H. Tian, and S. Fan, "Millimeter wave los/nlos identification and localization via mean-shift clustering," in *Proceedings of IEEE PIMRC*, 2019.
- [7] C. Benny, A. Simon, W. Stephen, and B. Mark, "Nlos identification and mitigation for geolocation using least-squares support vector machines," in *Proceedings of IEEE WCNC*, 2017.
- [8] 3GPP, "Study on channel model for frequencies from 0.5 to 100 ghz," in *3GPP Technical Report, 38.901 (Release 15)*, 2018.
- [9] S. Srinivasan, X. Yu, A. Keshavarz-Haddad, and E. Aryafar, "Fair initial access design for mmwave wireless," in *Proceedings of IEEE ICNP*, 2020.
- [10] 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 wi-fi," in *IEEE Communications Magazine*, 2014.
- [11] Y. Ghasempour, C. R. C. M. da Silva, C. Cordeiro, and E. W. Knightly, "Ieee 802.11ay: next-generation 60 ghz communication for 100 gb/s wi-fi," in *IEEE Communications Magazine*, 2017.
- [12] 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 (PCI9)*, 2019.
- [13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations*, 2015.
- [14] 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.

- [15] 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.
- [16] 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.
- [17] 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 *Proceedings of ACM CoNEXT*, 2017.
- [18] D. Steinmetzer and M. Hollick, "Nexmon: The c-based firmware patching framework. <https://nexmon.org>," 2017.
- [19] 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.
- [20] 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.
- [21] 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*, 2019.