UNIVERSITY OF CALIFORNIA

Los Angeles

Deep Scene Understanding
using RF and
its Fusion with other Modalities

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical and Computer Engineering

by

Akash Deep Singh

2023

ABSTRACT OF THE DISSERTATION

Deep Scene Understanding
using RF and
its Fusion with other Modalities

by

Akash Deep Singh
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2023
Professor Mani B. Srivastava, Chair

Rich scene understanding is a critical first step in creating autonomous systems with situational awareness – i.e. systems that can not only perceive and comprehend their environments but also project what the future states are going to be. Current vision-based methods of tackling this problem are inadequate as cameras are restricted to the visible spectrum. While they can detect objects, track movements, and make inferences about human expressions, they suffer from several challenges such as lack of depth information and weakness to bad weather conditions. Moreover, there are many other modalities in which information is present around us, and relying solely on one makes it susceptible to a higher chance of failure.

Through my thesis, I aim to include RF (radio-frequency) modality in scene understanding since RF has both complementary and supplementary properties to vision. My hypothesis is that by fusing RF with vision, one can create a richer understanding of their scene which I call 'deep scene understanding'. There are four key enablers to deep scene understanding

– (1) Detection of objects' states and activities, (2) Localization of objects in a scene and tracking them, (3) Developing methods to train machine learning models over RF data, and (4) Understanding privacy and societal impacts of instrumenting spaces with sensors.

RF comes with its own set of challenges that make this sort of integration hard. Additionally, instrumenting spaces with sensors such as RF sensors itself can lead to privacy concerns. In solving these challenges, we present – (1) a framework to detect human activities using a mmWave radar that can ingest sparse and noisy radar point clouds and output what activity is being performed in the scene. (2) a framework to detect, identify and localize hidden objects such as cameras in a scene that may be monitoring a user but are not visible to the naked eye. (3) a radar-camera fusion framework that can estimate dense depth in a scene from a sparse radar point cloud and an image. (4) A self-supervised learning approach that can leverage mutual information between a camera and a radar to train the radar. (5) A user study to understand the privacy perceptions of users when spaces are equipped with sensors.

The dissertation of Akash Deep Singh is approved.

Yuan Tian

Omid Salehi-Abari

Danijela Cabric

Mani B. Srivastava, Committee Chair

University of California, Los Angeles

2023

To my grandmother, my parents and my sister ...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to thank my Ph.D. advisor Professor Mani Srivastava for his continued support, patience, and encouragement throughout the course of my thesis. Through our discussions, I have learned many invaluable lessons that I am sure will guide me throughout my career going forward. Professor Srivastava gave me the freedom to explore whatever projects I wanted to work on which helped me expand my horizons.

I want to thank my committee members, Professor Danijela Cabric, Professor Omid Abari, Professor Yuan Tian, and Professor Xiang 'Anthony' Chen whose valuable inputs helped shape my thesis. I also thank them for accommodating my schedule.

I want to thank Luis Garcia and Alex Wong – two postdoctoral scholars who mentored me during various stages of my Ph.D. I learned a great deal from them about how to be patient, accept failures and use them as opportunities for growth.

I want to thank my collaborators Joseph Noor, Yunhao Ba, Ziqi Wang, Julian de Gortari, Ankur Sarker, Brian Wang, Gaofeng Dong, and Bryan Bednarski. I enjoyed working with them and learning from them. I also want to thank all the members of the Networked and Embedded Systems Laboratory (NESL) at UCLA who have made my experience highly enriching.

I also thank Nokia Bell Labs and Amazon Research for allowing me to spend my summers there as an intern. I thoroughly enjoyed my time there and learned how research is conducted in the industry.

I am grateful to the ECE Departmental Fellowship and the Amazon-UCLA Science Hub Fellowship that supported me during various parts of my degree.

Finally, I would like to thank my parents who have always supported and encouraged me to pursue my dreams. My parents have worked very hard to ensure a great life for me and my sister. The atmosphere of learning and curiosity that they created growing up, helped shape my interest in academics which eventually led to me pursuing a Ph.D. degree. I also

thank my sister for her constant backing of whatever I am doing.

Chapter 4 is a version of “Self-Supervised Radio-Visual Representation Learning for 6G Sensing” published at ICC 2022 - IEEE International Conference on Communications. This work was done as a part of my internship at Nokia Bell Labs. I thank my co-authors Mohammad Alloulah and Maximilian Arnold who were instrumental during the course of this project. Nokia and its subsidiaries can reproduce and distribute reprints of Chapter 4 for any purposes notwithstanding any copyright notation here on.

The research reported in this thesis was sponsored in part by the Army Research Laboratory (ARL) under Cooperative Agreement W911NF2020158; the IoBT REIGN Collaborative Research Alliance funded by the Army Research Laboratory (ARL) under Cooperative Agreement W911NF1720196; and, the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARL, DARPA, SRC, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

VITA

2014–2018	B.Tech (Electronics and Communication Engineering), IIIT-D
2018-2020	M.S. (Electrical and Computer Engineering), UCLA
2021	Research Intern (Radio Systems Research Group), Nokia Bell Labs
2022	Applied Scientist Intern, Amazon.com
2018-2023	Ph.D. Student (Electrical and Computer Engineering), UCLA

PUBLICATIONS

CVPR'23: Depth Estimation from Camera Image and mmWave Radar Point Cloud

Usenix Security'21: I Always Feel Like Somebody's Sensing Me! A Framework to Detect, Identify, and Localize Clandestine Wireless Sensors

mmNets'19: Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar

Sensys'20: UWHear: through-wall extraction and separation of audio vibrations using wireless signals

ICC'22: Self-Supervised Radio-Visual Representation Learning for 6G Sensing

Sensys'22: Capricorn: Towards Real-time Rich Scene Analysis Using RF-Vision Sensor Fusion

JAMIA'21: On collaborative reinforcement learning to optimize the redistribution of critical medical supplies throughout the COVID-19 pandemic

ACM TOPS'22: InkFiltration: Using Inkjet Printers for Acoustic Data Exfiltration from Air-Gapped Networks

Nature Scientific Reports'22: Temporal convolutional networks and data rebalancing for clinical length of stay and mortality prediction

CHAPTER 1

Introduction

Smart and autonomous agents are entering all aspects of our lives. Self-driving cars are poised to take over our roads soon, companies have started testing drone delivery systems, and smart-home assistants along with their full suite of smart appliances have become an increasingly integral part of our homes. If one were to replace human beings with these so-called "smart" agents, these agents would first need to develop situational awareness which is at par or even better than humans. Endsley's model [Endsley, 2015] defines situational awareness as not only the ability of an agent to perceive and comprehend its environment, but also project what the next states are going to be. The first step to developing this kind of awareness is to be able to understand the environment that an agent is in. This is where scene understanding comes into play.

Visual systems are the current state-of-the-art when it comes to understanding environments. Scene understanding in the literature surrounding these works has been defined loosely as – "interpretation of videos from pixels to events" [INRIA,]. This involves tasks such as segmentation, classification, tracking, and scenario recognition. While vision as a modality provides us with a rich and dense (in the sense of color) understanding of our environments, it is restricted to the spectrum of visible light only. With the proliferation of wireless devices and wireless standards of communication, a lot of information around us flows in this radio-frequency (RF) spectrum which is outside the visible spectrum. In addition, a lot of devices have electromagnetic (EM) emanations that can potentially convey information about their states and type. As a result, while the current methods of scene understanding are able to provide some preliminary ideas about the scene an agent is in, they are inadequate as they miss out on using a lot of information in the RF spectrum opportunistically to boost their inferences.

Additionally, vision is not robust to poor lighting conditions and bad weather – something very common in a large part of our planet. Hence, one can not rely on just vision for safety-critical applications. RF sensors such as radars on the other hand are robust to environmental changes and have been used for a variety of applications in the past.

RF and vision share complementary and supplementary properties:

- RF can see in the dark, vision cannot.
- RF can capture depth, while vision can capture color
- RF can 'see' in the radio frequency spectrum, vision can capture the visible spectrum
- RF and vision can both do tracking and object detection

Research Challenges. RF comes with its own fair share of challenges which makes it hard to use for sensing or integrate with other modalities. Some of these key challenges are:

- Sparsity: Unlike cameras, radars suffer from specular reflections – the amount of information about the scene captured by radar is very sparse. When this information is used to generate a representation such as a point cloud, they are very sparse and often vary in size based on scene dynamics. This makes it difficult to feed this data directly into a neural network.
- Noise: Radar point clouds are noisy. Not only are the depth (z) and azimuth (x) components noisy, but the elevation (y) is completely off for mmWave radars that do not have an antenna element along the elevation axis.
- Encryption: Most of the data in the RF spectrum is either encrypted or is difficult to attribute to a particular device. In addition, it is difficult to figure out which frequency

bands are of use and which are not because of the excessive amounts of wireless traffic in these regions. As a result, it is challenging to develop frameworks that can use this information for meaningful inferences without breaking the encryption.

- Lack of Labeled Data: Radar data is difficult to label and the human annotation process does not scale well with data collection. Hence, unlike vision and language datasets, radar datasets have limited availability.

In solving the challenges above and combining RF with vision and other modalities for scene understanding, I present my thesis titled Deep Scene Understanding.

Terminology The word 'deep' in Deep Scene Understanding has two meanings, firstly, rather than just looking at the scene visually and making inferences, we want to achieve a deeper understanding of what is happening – i.e. a richer understanding or one can say, we aim to go beyond what cameras can see. Secondly, 'deep' also alludes to the set of techniques – deep learning that I have made use of to realize my vision. There are three key enablers (described below) that together make our understanding of any scene "deep". These are:

1. Detection of objects states and activities

2. Localization of objects in a scene and tracking them

3. Developing methods to train machine learning models over RF data

4. Measuring privacy impacts of instrumenting spaces with sensors that enable Deep Scene Understanding.

Now, let us discuss each facet of 'Deep Scene Understanding' in more detail.

1.1 Detection of objects states and activities

Detecting activities in a space can be important for a variety of reasons. In smart buildings, detecting activities can help optimize the use of space and resources such as lighting and heating. In healthcare, activity detection can ensure the well-being of patients. It can also help with ensuring that safety policies are being followed in factory settings. Similarly, detecting object states (such as when a device is on/off) can help with figuring out when a machine needs maintenance, or whether a camera installed in a space is recording or not. In the following subsections, we discuss some contributions that this thesis makes in this space.

1.1.1 Human Activity Recognition via sparse point clouds obtained through a mmWave radar

Unlike cameras, radars suffer from specular reflections – the amount of information about the scene captured by radar is very sparse. This means that point clouds generated through a (mmWave) radar are of non-uniform size. This makes it difficult to feed this data directly into a neural network. We use voxelization to show that it is possible to not only use non-uniform-sized point clouds as an input to neural networks but also use this approach to classify activities being performed in a space with an RF sensor.

1.1.2 Detecting and identifying hidden sensors snooping on a user in a given scene

The increasing ubiquity of low-cost wireless sensors has enabled users to easily deploy systems to remotely monitor and control their environments. However, this raises privacy concerns for third-party occupants, such as hotel room guests who may be unaware of deployed clandestine sensors. Previous methods focused on specific modalities such as detecting cameras but do not provide a generalized and comprehensive method to capture arbitrary sensors which may be “spying” on a user. In this thesis, we propose SnoopDog, a framework to not only detect common Wi-Fi-based wireless sensors that are actively monitoring a user, but also classify and localize each device.

SnoopDog works by establishing causality between patterns in observable wireless traffic and a trusted sensor in the same space, e.g., an inertial measurement unit (IMU) that captures a user's movement. Once causality is established, SnoopDog performs packet inspection to inform the user about the monitoring device. We evaluated SnoopDog across several devices and various modalities, and were able to detect causality for snooping devices 95.2% of the time.

1.2 Localization of objects in a scene and tracking them

The knowledge of the accurate location of an object in a space can be used for various tasks such as discovering hidden sensors that may be spying on a user or trying to locate missing objects. It is also useful for robot navigation and autonomous driving. In the following subsections, we discuss some contributions that this thesis makes in this space.

1.2.1 Going from 2D to 3D – how to recover dense depth in the scene with images and radar

We present a method for inferring dense depth from a camera image and sparse noisy radar point cloud. Unlike existing works that densify the scene by learning a direct map from image and sparse point cloud to dense geometry, we decompose the problem into two parts: (1) Given the radar point cloud which has ambiguous elevation and noisy azimuth components, our method maps a single point to the possible surfaces that may correspond to that point in the image. By querying the set of radar points, our method produces a quasi-dense depth map by associating radar points with likely image pixels. (2) We fuse the quasi-dense depth map with the camera image to learn a dense depth representation of the scene.

1.2.2 Localizing Hidden Snooping Sensors

We propose SnoopDog that can localize clandestine devices in a 2D plane using a novel trial-based localization technique. We evaluated SnoopDog across several devices and various modalities and were able to localize devices to a sufficiently reduced sub-space.

1.3 Developing methods to train machine learning models over RF data

RF data is hard to label and hence it is difficult to train machine learning models with it for a variety of tasks. As RF sensing proliferates, there is a need to create techniques that can learn from this RF data without labels. In this thesis, we combine radio and vision to *automatically learn* a radio-only sensing model with minimal human intervention. We want to build a radio sensing model that can feed on millions of *uncurated* data points. To this end, we leverage recent advances in self-supervised learning and formulate a new *label-free* radio-visual co-learning scheme, whereby vision trains radio via cross-modal *mutual information*. We implement and evaluate our scheme according to the common linear classification benchmark, and report qualitative and quantitative performance metrics. In our evaluation, the representation learned by radio-visual self-supervision works well for a downstream sensing demonstrator and outperforms its fully-supervised counterpart when less labeled data is used. This indicates that self-supervised learning could be an important enabler for future *scalable* radio sensing systems.

1.4 Measuring Privacy Impacts of Instrumenting Spaces with Sensors

Privacy allows individuals to control how their personal information is collected, used, and stored. In realizing the vision that my thesis presents, we will need to instrument spaces such as homes, offices, roads, and plazas with a multitude of sensors. However, indiscriminately installing sensors can lead to several privacy concerns such as what data is collected, how it is collected, and how it will be used. Often, sensing in public spaces is done without user consent which leads to further violations of their privacy expectations.

The existing notion behind privacy is that the sensors whose data can easily be understood and interpreted by humans (such as cameras) are more privacy-invasive than sensors that are not human-understandable, such as RF (radio-frequency) sensors. However, given recent advancements in machine learning, we can not only make sensitive inferences on RF data but also translate between modalities. Thus, the existing notions of privacy for IoT sensors need to be revisited. In this thesis, we conduct an online study of 162 participants from the USA to find out what factors affect the privacy perception of a user regarding a device or a sensor. Our findings show that a user's perception of privacy not only depends upon the data collected by the sensor but also on the inferences that can be made on that data, familiarity with the device and its form factor as well as the control a user has over the device design and its data policies. When the data collected by the sensor is not human-interpretable, it is the inferences that can be made on the data and not the data itself that users care about when making informed decisions regarding device privacy.

CHAPTER 2

Deep Learning for Non-uniform mmWave Radar Point Cloud Data

2.1 Introduction

Although the recognition and monitoring of human activities can enable safety critical applications–e.g., monitoring disabled and or elderly people living-alone that may need medical attention [Attal et al., 2015,Ni et al., 2011]–the emergence of low cost sensing capabilities have enabled ubiquitous possibilities of human activity recognition for everyday applications. Several context aware applications have recently emerged such as workout tracking and efficacy [Shen et al., 2017], and factory floor monitoring [Hu et al., 2014]. Traditionally, human activity has been inferred either through ambient sensors (e.g., cameras) and/or wearable sensors (e.g., smartwatches with IMUs). Although wearables have proven to be effective approaches for human activity recognition, it is not practical to assume that all of the subjects in a space will use wearables that are compatible with the inference model. Ambient sensor approaches are robust to heterogeneous environments since they do not rely on users having a particular device. However, the sensor data from cameras carry a significant amount of ambient information that may be of concern for privacy-sensitive applications. For instance, there have been cases where cameras in the maternity ward of a hospital were used to spy upon female patients [Bonifield, 2019]. In this context, cameras can be replaced by sensors whose data can provide a sufficient amount of ambient information to realize the same utility, e.g., if we only care about how many patients are in a room or whether there are a certain number of nurses in the ward.

Prior works have shown that less information-rich ambient sensors can effectively infer human activities while not exposing subjects to privacy risks using radio frequency signals. For instance, WiFall [Wang et al., 2016a] showed how WiFi routers can be used to detect whether a human has fallen or not. However, the work is not robust beyond binary classification of two classes that are significantly different from each other. Generally, WiFi has a narrow band (when compared to the high bandwidth of a mmWave radar) and does not have sufficient range resolution to perform robust classification. Radars have begun to emerge as a popular modality for activity recognition [Çağlıyan and Gürbüz, 2015] as they provide the advantage of operating in any lighting condition and work through a multitude of environmental conditions, e.g., fog and rain. Further, the emergence of millimeter-wave technology has enabled cost-effective distributed sensing applications.

Millimeter-wave (mmWave) technology operates in the frequency range of 30GHz and 300GHz. Since, antenna size is inversely proportional to frequency, the higher you go up in the frequency spectrum, the lower the size of antenna becomes. As a result, mmWave radars are compact in size. Also, we can pack a large number of antennas into a very small space which enables highly directional beam-forming ($\approx 1^\circ$ angular accuracy). Since these radars have a large bandwidth, they have a superior range resolution. Further, new low cost, off-the-shelf radars have led to an increase in the popularity of mmWave based sensing solutions. However, these devices are resource-constrained and, instead of providing raw data, their output is in the form of point clouds[1]. The number of points in each frame captured by the mmWave radar varies, increasing the complexity of constructing a neural network architecture that can process this data as is. Hence, several feature extraction and data pre-processing techniques have been proposed in previous works which convert this data into a format which is constant in size and can be given as input to a neural network [Zhang and Cao, 2018a, Zhao et al., 2019].

[1] To get raw ADC data from these devices, you need to connect them with expensive hardware

In this chapter, we propose RadHAR, a framework for human activity recognition that utilizes point clouds generated from a mmWave radar. To account for the sparsity of the mmWave radar point clouds, RadHAR leverages the notion that human activities typically last over a few seconds and accumulates point clouds over a sliding time window. Each point cloud is voxelized to overcome the non-uniformity of the data and is then fed into a set of classifiers. We collected a new HAR dataset consisting of point clouds using mmWave radar for 5 different classes of activities. We evaluated RadHAR and compared the accuracy of various classifiers on the collected dataset. In our evaluation, the best performing deep learning classifier composed of a set of convolutional layers and long-short term memory layers can achieve an average test accuracy of 90.47%.

Contributions. Our contributions are summarized as follows.

- We propose RadHAR, a framework that performs human activity recognition using a pre-processing pipeline for point clouds generated by mmWave radar.
- We evaluate different machine learning approaches for human activity detection using point cloud.
- We generate a new point cloud dataset for human activity detection and make it available open-source along with the data processing, classifier training and evaluation code, and pre-trained classifiers.

The rest of this chapter is arranged as follows. In Section 5.2, we present the preliminary information required to understand the RadHAR framework. We provide an overview of RadHAR and its implementation in Section 2.3, and evaluate the classification approaches in Section 5.7. The related work is presented in Section 5.13, and we discuss in Section 5.12, respectively. The source code and datasets of RadHAR are available online at https://github.com/nesl/RadHAR.

2.2 Background

We provide the preliminary information necessary to understand the RadHAR framework. We discuss the basics of mmWave radar physics.

2.2.1 Millimeter-wave Radar

Over the last several years, there has been a growth in low cost single chip radars that work in the mmWave range. One family of such popular devices are Texas Instruments' mmWave radar. These sensors output the point clouds that contain information like x,y,z positions of each point among other data.

Bandwidth and range resolution. Range resolution of a radar is its ability to distinguish between 2 targets present very close to each other. The range resolution and bandwidth are related as,

$$d_{res} = \frac{c}{2B} \tag{2.1}$$

where d_{res} is the range resolution in m, c is the speed of light in m/s and B is the bandwidth in Hz swept by the chirp of the radar. Hence, if we want a better range resolution, the bandwidth should be high. The maximum continuous bandwidth for the radar that we used is 4 GHz which corresponds to a range resolution of about 4 cm.

2.3 RadHAR Overview

The full pipeline of the RadHAR framework is depicted in Figure 2.1. The framework first collects data from a mmWave radar that is monitoring a human. The point cloud data is pre-processed before being fed into a HAR classifier. We present an overview of each component in detail.

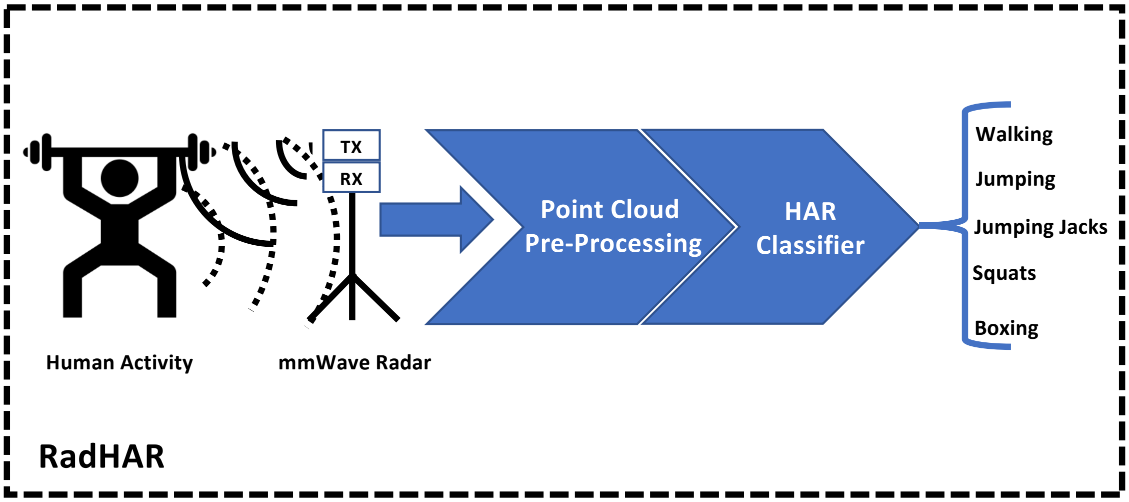


Figure 2.1: RadHAR framework overview.

2.3.1 Data Collection & Pre-processing

We have used TI's IWR1443BOOST [Instruments, 2019] radar to collect the new point cloud dataset called *MMActvity* (millimeter-wave activity) dataset. It is a FMCW (Frequency Modulated Continuous Wave) radar which uses a chirp signal. This radar works in the 76-GHz to 81-GHz frequency range. The radar includes four receiver and three transmitter antennas, which enable tracking multiple objects with their distance and angle information. This antenna design enables estimation of both azimuth and elevation angles, which enables object detection in a 3-D plane [Instruments, 2018].

2.3.2 MMActvity Dataset

For data collection, the radar is mounted on a tripod stand at a height of $1.3m$. The data from the radar is sent to the laptop via ROS (Robot Operating System) messages over USB. The ROS node running on the laptop is developed and described in [Zhang and Cao, 2018a]. To record and store the data, we use rosbag which another ROS package. Finally, we convert these rosbags into .txt files. These files are then used to create voxelized representation of the point clouds. The data collection and pre-processing pipeline is describe in Figure 2.3.

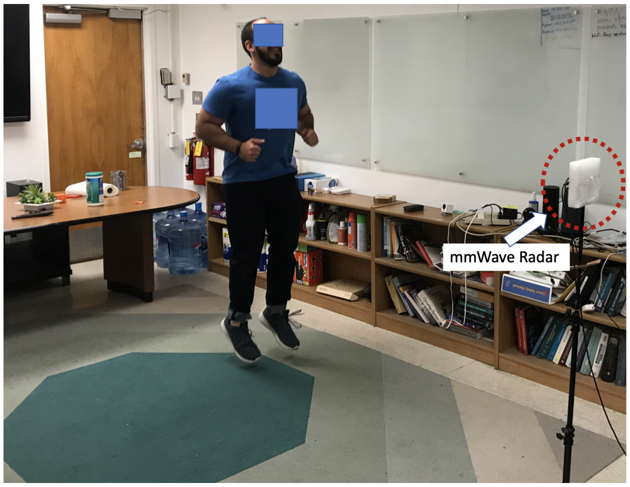


Figure 2.2: Data collection setup.

We have collected the data from two users[2]. The users perform 5 different activities in front of the radar as shown in Figure 2.2. These activities are: Walking, Jumping, Jumping Jacks, Squats and Boxing. The data is collected in a continuous periods of about 20 seconds for a subject performing the same activity. Some of the data files are longer than 20 seconds. In total, we have collected 93 minutes of data. The description of the dataset can be found in Table 2.1.

The captured point clouds contains spatial coordinates (x,y,z in meters) along with velocity in meters/second, range (distance of the point the from radar) in meters, intensity (dB) and bearing angle (degrees). The sampling rate of the radar is 30 frames per second.

2.3.3 Data Pre-processing

We divided collected data files into separate train and test files with 71.6 minutes data in train and 21.4 minutes data in test. To overcome the non-uniformity in number of points in each frame, we converted the point clouds into voxels of dimensions 10x32x32 (depth=10) which makes the input of constant size irrespective points of the number of points in the frame. We decided these dimensions empirically by testing their performance. In our voxel

[2] The data is collected from the authors and thus does not require approval from IRB.

Activity	# of data files	Total duration (seconds)
Boxing	39	1115
Jumping Jacks	38	1062
Jumping	37	1045
Squats	39	1090
Walking	47	1269

Table 2.1: Details of the MMActivity dataset.

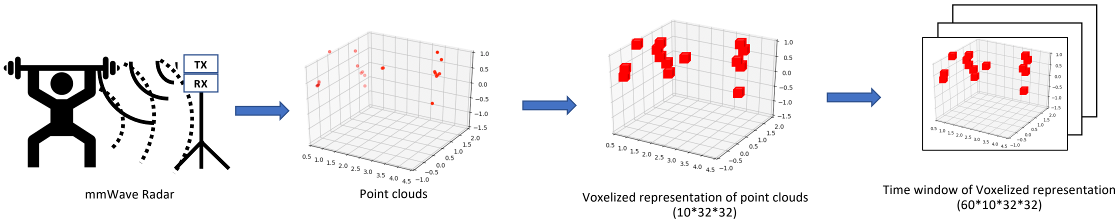

TX
RX
mmWave Radar
Point clouds
Voxelized representation of point clouds
(10*32*32)
Time window of Voxelized representation
(60*10*32*32)

Figure 2.3: Workflow of data preprocessing. The voxel size if 10*32*32. The time windows are generated by grouping 60 frames (2 second) together.

representation, the value of each voxel is the number of data points present within its boundaries. While having large number of voxels may represent underlying information well, it increases the data size by several orders of magnitudes.

Since activities are performed over a period of time, the time window from activities are generated in-order to capture the temporal dependencies. We create windows of 2 seconds (60 frames) having a sliding factor of 0.33 seconds (10 frames). The 2-second window was chosen based on the previous works in human activity recognition from multimodal timeseries datasets [Xing et al., 2018] and human identification using point clouds [Zhao et al., 2019]. Finally, we get 12097 samples in training and 3538 samples in testing. We use 20% of the training samples for validation. In the *time window voxelized representation*, each sample has a shape of $60 * 10 * 32 * 32$.

2.3.4 Classifiers

We evaluate different classifiers on the MMActivity dataset. We train Support Vector Machine (SVM), multi-layer perceptron (MLP), Long Short-term Memory (LSTM) and convolution neural network (CNN) combined with LSTM. We compare the inference capability of these classifiers on the same train and test split of MMActivity dataset. These deep learning classifiers are generally adapted in a wide range of applications. LSTM and CNN combined with LSTM architectures are inspired from [Zhao et al., 2019]. Next, we explain the details of classifiers (data inputs, architectures and training details).

2.3.4.1 SVM Classifier

The input to the Support Vector Machine (SVM) classifier is generated by flattening the time window voxelized representation (60∗10∗32∗32) and then applying the Principal Component Analysis (PCA) for dimensionality reduction. We used PCA to reduce the dimensions of data from 614400 $(60 * 10 * 32 * 32)$ to 6000 which explained 80% of variance in data. SVM with RBF kernel was used.

2.3.4.2 MLP Classifier

It is composed of fully-connected layers and an output layer. We flatten the time window voxel representation $(60 * 10 * 32 * 32)$ of the sample to create input size of 614400 dimensions for the MLP classifier. The MLP classifier has 4 fully connected layers followed by the output layer. We use dropout layers to avoid overfitting. It has 39.35 million trainable parameters.

2.3.4.3 Bi-directional LSTM Classifier

A bi-directional LSTM layer consists of two LSTM layers operating in parallel. The input to the first layer is provided as-is whereas the input is reverse copy of the data for the the second layer. As a result, a bi-directional LSTM layer preserves the information from both the future and the past. This network consists of the Bi-Directional LSTM layer followed by the 2 fully connected layers and an output layer. The input (60*10240) to the network is created by preserving the time dimensions (60) and flattening the spatial dimensions in the samples (10*32*32). We used Bi-Directional LSTM with size of 64 and 64 hidden units. The Bi-directional LSTM classifier has 5.29 million trainable parameters.

2.3.4.4 Time-distributed CNN + Bi-directional LSTM Classifier

Time-distributed CNN applies CNN layers to every temporal slice of the input data. The architecture of Time-distributed CNN + Bi-directional LSTM classifier consists of 3 time distributed convolutional modules (convolution layer + convolution layer + maxpooling layer) followed by the bi-directional LSTM layer and an output layer. Overall the network has 291k trainable parameters. This classifier is directly trained on the the input sample with its time and spatial dimensions.

Training and implementation. The classifiers were implemented using Sklearn and Keras. We use GridSearchCV function from sklearn to optimize the hyperparameters (C and gamma) of the SVM. Adam optimizer with a learning rate of 0.001 was used to train deep learning classifiers. The models with minimum loss on the validation data was saved after training for 30 epochs.

S.No	**Classifier**	**Accuracy**
1	SVM	63.74
2	MLP	80.34
3	Bi-directional LSTM	88.42
4	Time-distributed CNN+ Bi-directional LSTM	90.47

Table 2.2: Test accuracy of different activity recognition classifiers trained on the MMActivity Dataset.

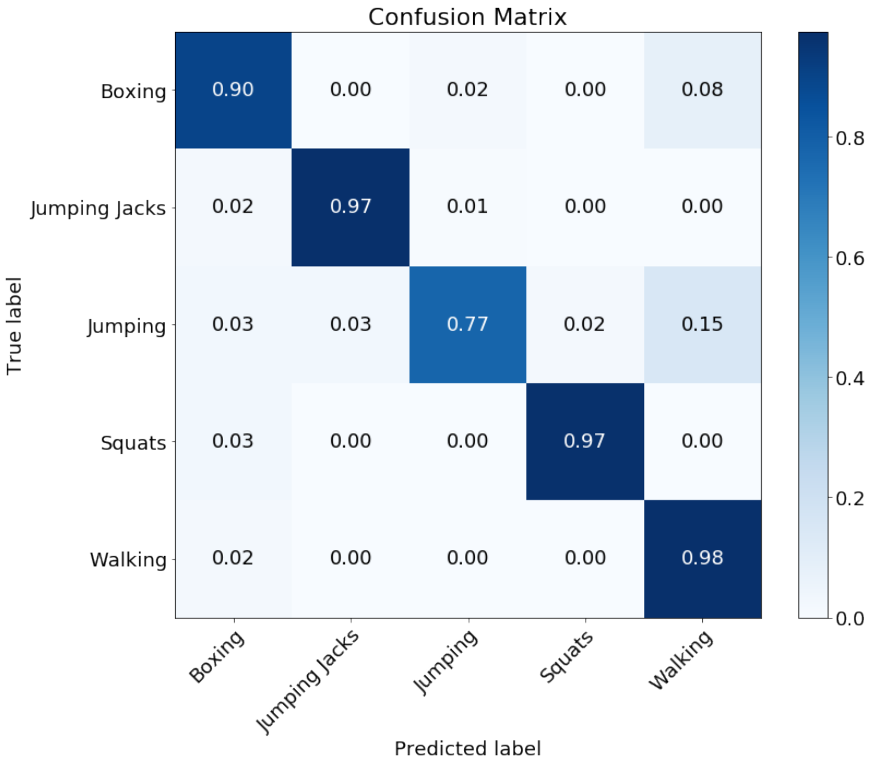


Figure 2.4: Confusion matrix of time-distributed CNN + bi-directional LSTM classifier.

We now evaluate the aforementioned approaches.

2.4 Evaluation

As shown in Table 2.2, the classifiers trained for the human activity recognition have different performance. The table reports average results of 5 different training sessions. The SVM classifier has poor performance with test accuracy of 63.74%. One reason might be the input to the SVM is not using the domain specific feature extraction approaches as used by Kim et al. [Kim and Ling, 2009] where they use a Doppler radar (2.4 GHz) and then convert the output into micro-Doppler signatures. All the three deep learning classifier are working directly on the time window voxel data. MLP classifier consists of the fully connected layers which doesn't assume anything about the input data and has test accuracy of 80.34%. Bi-directional LSTM classifier tries to learn the sequence of input data. The input data

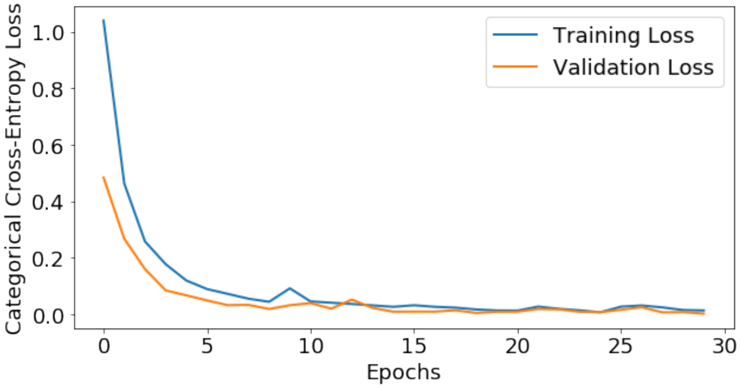


Figure 2.5: Variation of training and validation loss of Time-distributed CNN + Bi-Directional LSTM

to the LSTM preserve the time component. Since human activities are performed over a duration and due to preserving time sequence for input to LSTM, the LSTM performance is significantly better then the MLP with test accuracy of 88.42%. The best performing classifier is Time-distributed CNN + Bi-directional LSTM which has test accuracy of 90.47%. Time-distributed CNN layers learn the spatial features from the data, as the point clouds are spatially distributed and the Bi-directional LSTM layers learn the time dependency for the activity windows. Our evaluation shows specialized spatial and temporal layers in deep learning architectures can result in boost in accuracy. The confusion matrix for one of the trained Time-distributed CNN+ Bi-directional LSTM classifier is shown in Figure 2.4. As seen from the figure, the activity of jumping and boxing is confused with the walking. The reason might be the similarities in the data for these activities. The variation of the training loss and the validation loss for Time-distributed CNN + Bi-directional LSTM classifier with the training epochs is shown in Figure 2.5.

Voxelized representation with velocities. All the evaluation presented above used the voxel representation, where the value of each voxel is the number of data points present within its boundaries. We also evaluated with the voxelized representation where the value of each voxel is the sum of the velocity of all the points present within its boundaries. The Time-distributed CNN + Bi-directional LSTM classifier trained on velocity voxel representation also had the similar performance as shown in Table 2.2. We now discuss the works directly related to the RadHAR framework.

2.5 Related Work

Human activity recognition is widely explored using various sensing modalities. Researchers have used sensors like cameras and inertial measuring units [Chen et al., 2015, Xing et al., 2018, Sun et al., 2017], sound [Zhan and Kuroda, 2014, Xing et al., 2018] and WiFi [Wang et al., 2016a]. However, optical sensors like cameras capture a significantly large amount of information and sensors like inertial measuring units need to be present on the user body.

Micro-Doppler spectrograms using radars for human activity recognition have been studied in detail over the last decade [Çağlıyan and Gürbüz, 2015, Fairchild and Narayanan, 2016, Kim and Ling, 2009]. In [Kim and Ling, 2009], the authors use a Doppler radar to collect data of 12 subjects performing 7 different activities. They create micro-Doppler spectrograms from this data and extract 6 features from it to train an SVM classifier. Unlike our work, the radar used here is in the S-band (2-4 GHz).

Recently, researchers have exploited low-cost single-chip mmWave radar systems for person identification and tracking [Zhao et al., 2019] and human activity recognition [Zhang and Cao, 2018a]. In [Zhang and Cao, 2018a], the authors convert the point cloud data into micro-Doppler spectrograms before using a CNN to classify it. In [Zhao et al., 2019], the authors use voxelized representation of point clouds for human identification using a LSTM and a CNN + LSTM classifier. Our deep learning classifier architectures are inspired from [Zhao et al., 2019], however, we are targeting a different problem of human activity recognition.

In this work, we show that the time window voxel representation of the sparse points clouds can be used for human activity recognition. We evaluate multiple classifiers and achieve test accuracies as high as 90% percent for the deep learning classifier based on the convolutional layers and long-short term memory layer. Our evaluation shows that deep learning approaches can achieve comparable performance to the previous domain specific feature extraction approaches like used by Kim et al. [Kim and Ling, 2009].

2.6 Discussion and Future Research

We now discuss the limitations of our approach and enumerate future research directions.

Spatial and temporal dependencies in point clouds. As shown in Table 2.2, MLP classifier has poor performance. The reason might be due to fact the fully connected layers in MLP classifier makes no spatial and temporal assumption about the data. On the other hand, Time-distributed CNN + Bi-directional LSTM classifier assumes spatial and temporal dependency in the data and hence performs better.

Limitations of voxelized representation. Voxels result in significant increase in the required memory and computation. This can be seen in dimensionality of each input sample (60*10*32*32 = 614400), which has to be processed by the deep learning classifier. This begs the need for neural networks which are trainable directly on point clouds. One such network is the PointNet [Qi et al., 2017] which can be used for applications like object classification, part segmentation and scene semantic parsing.

CHAPTER 3

Fusion of RF and Vision to Exploit Complimentary Properties

3.1 Introduction

Understanding the 3-dimensional (3D) structure of the scene surrounding us can support a variety of spatial tasks such as navigation [Maier et al., 2012] and manipulation [Choi and Christensen, 2010]. To perform these tasks, an agent is generally equipped with multiple sensors, including optical i.e., RGB camera and range i.e., lidar, radar. The images from a camera are "dense" in that they provide an intensity value at each pixel. Yet, they are also sparse in that much of the image does not allow for the establishing of unique correspondence due to occlusions or the aperture problem to recover the 3D structure lost to the image formation process. On the other hand, range sensors are typically sparse in returns, but provide the 3D coordinates for a subset of points in the scene i.e., a point cloud. The goal then is to leverage the complementary properties of both sensor observations – an RGB image and a radar point cloud that is synchronized with the frame – to recover the dense 3D scene i.e., camera-radar depth estimation.

While sensor platforms that pair lidar with camera have been of recent interest i.e., in autonomous vehicles, they are expensive in cost, heavy in payload, and have high energy and bandwidth consumption [Raj et al., 2020] – limiting their applications at the edge [Shi et al., 2016]. On the other hand, mmWave [Johnston, 1980] radars are orders of magnitude cheaper, light weight, and power efficient. Over the last few years, developments in mmWave radars

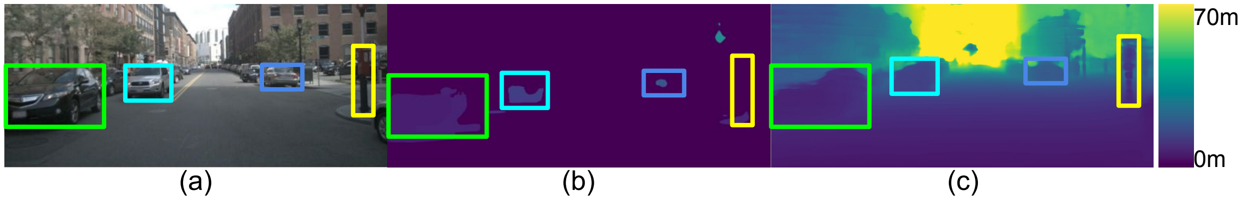


Figure 3.1: Depth estimation using a mmWave radar and a camera. (a) RGB image. (b) Semi-dense depth generated from associating the radar point cloud to probable image pixels. (c) Predicted depth. Boxes highlight mapping of radar points to objects in the scene.

and antenna arrays [IIZUKA et al., 2003] have significantly advanced the performance of these sensors. Radars are already ubiquitous in automotive vehicles as they enable services such as cruise control and collision warning [Eichelberger and McCartt, 2016]. Methods to perform 3D reconstruction with camera and radar are also synergistic with the joint communication and sensing (JCAS) paradigm in 6G cellular communication [Wild et al., 2021, Alloulah et al., 2022, Zhang et al., 2021], where cellular base-stations will not only be the hub of communication, but also act as radars, to sense the environment.

The challenge, however, is that a mmWave radar is a point scatterer and only a very small subset of points (50 to 80 per frame [Caesar et al., 2020]) in the scene, often noisy due to its large beam width, are reflected back into the radar's receiver. Compared to the returns of a lidar, this is 1000x more sparse. Additionally, most radars used in automotive vehicles do not have enough antenna elements to provide the elevation of the points with a high enough resolution to be useful – erroneous at worst (see Sec. 3.3).

As a result, camera-radar depth estimation requires (i) mapping noisy radar points without elevation components to their 3D coordinates (and with calibrating their 2D image coordinates i.e., radar-to-camera correspondence) and (ii) fusing the associated sparse points with images to obtain the dense depth. Existing works have projected the radar points onto the image and "extended" the elevation or y-coordinate in the image space as a vertical line [Lo and Vandewalle, 2021] or relied on multiple camera images to compute the optical-flow which in-turn has been used to learn the radar-to-pixel mapping [Long et al., 2021b].

These approaches overlook that radar returns have *noisy* depth, azimuth and *erroneous* elevation. They also assume access to multiple consecutive image and radar frames, so that they may use the extra points to densify radar returns in both the close (from past frames) and far (from future frames) regions. In the scenario of obtaining instantaneous depth for a given frame, the requirement of future frames makes it infeasible; if delays are permitted, then an order of hundreds of milliseconds in latency is incurred.

Instead, we propose to estimate depth from a single radar and image frame by first learning a one to many mapping of correspondence between each radar point and the probable surfaces in the image that it belongs to – yielding a semi-dense radar depth map. The information in the radar depth map is further modulated by an gated fusion mechanism to learn the error modes in the correspondence (due to possible noisy returns) and adaptively weight its contribution for image-radar fusion. The result of which is used to augment the image information and decoded to a dense depth map.

Our contributions are: (i) to the best of our knowledge, the first approach to learn radar to camera correspondence using a single radar scan and a single camera image for mapping arbitrary number of ambiguous and noisy radar points to the object surfaces in the image, (ii) a method to introduce confidence scores of the mapping for fusing radar and image modalities, and (iii) a learned gated fusion between radar and image to adaptively modulate the trade-off between the noisy radar depth and image information. (iv) We outperform the best method that uses multiple image and radar frames by 10.3% in mean absolute error (MAE) and 9.1% in root-mean-square error (RMSE) to achieve the state of the art on the NuScenes [Caesar et al., 2020] benchmark, despite only using a single image and radar frame.

3.2 Related Work

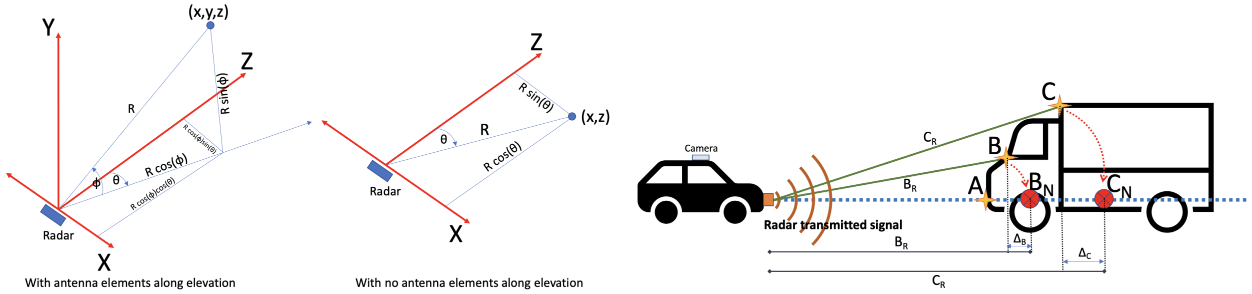
Camera and lidar based depth estimation [Bergman et al., 2020, Fu et al., 2019, Hu et al., 2021, Jaritz et al., 2018, Jaritz et al., 2018, Ma and Karaman, 2018, Qiu et al., 2019, Yang et al., 2019, Van Gansbeke et al., 2019, Wong et al., 2020, Wong et al., 2021, Wong and Soatto, 2021, Xu et al., 2019, Zhao et al., 2021] leverages an RGB image as guidance to densify a sparse lidar point cloud. Most of the works are focus on addressing the sparsity problem. For example, [Chen et al., 2019, Ma et al., 2019, Uhrig et al., 2017, Yang et al., 2019] designed network blocks to effectively deal with the sparse inputs. [Bergman et al., 2020] estimates the lidar sampling location and predicts the depth map more accurately without requiring high sampling rates. [Li et al., 2020a] used a cascade hourglass network, [Hu et al., 2021, Jaritz et al., 2018, Yang et al., 2019] used separate image and depth networks and fused their representations, and [Huang et al., 2019b] proposed an upsampling layer and joint concatenation and convolution. [Van Gansbeke et al., 2019] leveraged confidence maps to fuse predictions from different modalities, [Qiu et al., 2019, Xu et al., 2019, Zhang and Funkhouser, 2018] used surface normals for guidance and [Cheng et al., 2020, Park et al., 2020] use convolutional spatial propagation networks. Another line of work [Wong et al., 2020, Wong et al., 2021, Wong and Soatto, 2021] focus on densifying the inputs through interpolation [Wong et al., 2020] or spatial pyramid pooling [Wong et al., 2021, Wong and Soatto, 2021]. However, lidars are expensive, have high energy consumption and are limited in real world applications; whereas, mmWave radars are cheap to purchase and common in many sensor platform. Adapting these methods to radar point clouds is nontrival since they assume point cloud sizes of ≈30k points that are aligned to the image; in contrast radar point clouds are on orders of 50 points with noisy azimuth and ambiguous elevation.

Camera and radar based depth estimation uses sparse mmWave radar point clouds and camera images [Long et al., 2021b, Lin et al., 2020, Lo and Vandewalle, 2021, Long et al., 2021a, Gasperini et al., 2021]. Unlike camera-lidar depth estimation, it brings new challenges due to the sparsity and noise of the radar point clouds. [Long et al., 2021b] learn a mapping from radar data to image pixels using a radar-to-pixel association and then train a network to using a lidar point cloud is used predict dense depth. To deal with the sparsity, however, [Long et al., 2021b] reproject multiple radar sweeps into the current frame to increase the density of points and use multiple camera images (some from the future) to compute the optical and radar flow – something which is not practical in real world. [Lin et al., 2020] propose a two-stage encoder-decoder architecture to reduce the noise in radar point cloud, and like [Long et al., 2021b], also uses future frames. Similarly, [Lo and Vandewalle, 2021] create a height-extended radar representation and then fuse it with camera images to generate dense depth. [Gasperini et al., 2021] fuses sparse point clouds as a weak supervision signal during training and uses it as an extra input to enhance the estimation robustness at inference time. However, these works either ignore the noise and error in radar points or use multiple (future) images and radar scans to get obtain denser points points with additional points in close and far regions. Unlike them, we only require a single image and radar scan to produce dense depth.

3.3 mmWave Radar Point Cloud Generation

In this section, we describe the geometry that determines the generation of point clouds via a mmWave radar. mmWave radars, like other radars send an electromagnetic (EM) [Vainshtein, 1988] wave through their transmitter. This wave hits the objects in the scene, reflected back and collected at the receiver. Unlike visible light, who's wavelength is in μm, mmWave radars suffer from the challenge of specular reflections, due to larger wavelengths larger than visible light. The lack of diffused reflections (which is the case of visible light) means that only the objects that reflect back into the receiver of the radar are captured. Hence, only a small portion of the scene is visible to the radar. The point clouds generated from a mmWave radar will be sparser than a camera image by several orders of magnitude.

To resolve the location of these reflections, radars use multiple receivers. However, popular mmWave radars used in autonomous driving (such as the one used in nuScenes [Caesar et al., 2020] data collection), lack the ability to resolve height of objects in the scenes – a direct result of either not having antenna elements to capture elevation information or not



(a) mmWave radar point cloud geometry

(b) Elevation ambiguity in radar point clouds

Figure 3.2: Challenges with radar point clouds – an illustration of elevation ambiguity (y) and noise in azimuth (x) and depth (z) components. (a-left) Shows the geometry of how the mmWave radar point clouds are obtained in an ideal setting. (a-right) Shows the geometry of how the mmWave radar point clouds are obtained when the radar lacks information along the elevation axis. As a result, the radar assumes that all the points reflecting back into the receiver are reflecting from the plane perpendicular to the radar. This means, that the value of y obtained will always be 0 – which renders it useless for any task. Due to this assumption, The values of both x and z will also be noisy. $\Delta x = Rsin\theta(1 - cos\phi)$, $\Delta y = Rsin\phi$, $\Delta z = Rcos\theta(1 - cos\phi)$. (b) (Updated version of [Long et al., 2021b]) Shows a real world manifestation of this noise. Due to the ambiguity in height, The points B and C have a difference in their depth while the point A is accurate since it lies in the plane perpendicular to the plane of the radar. Hence, a projection of radar point clouds on to the image plane using camera intrinsics and pose will not work.

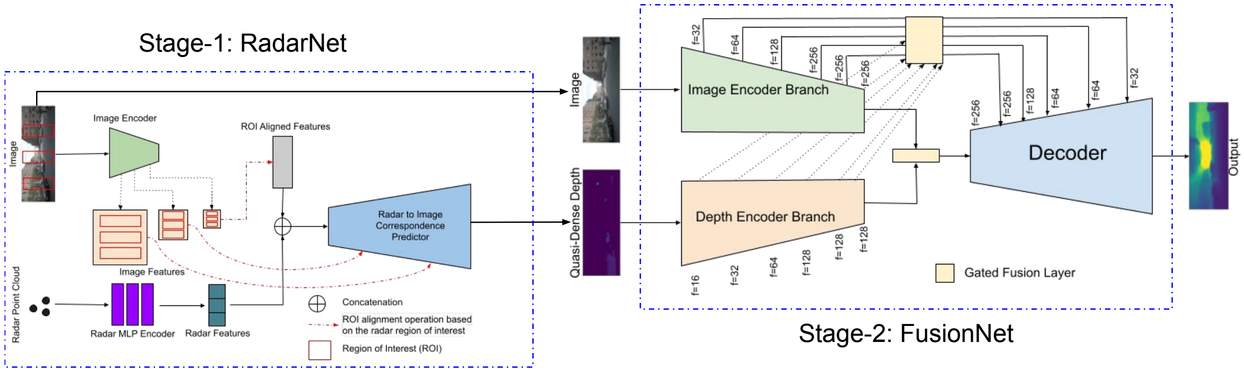


Figure 3.3: System Overview - Our two-stage architecture for estimating dense depth from a mmWave radar point cloud and a camera image.

having sufficient compute to process reflections along the elevation.

As shown in Fig. 3.2, since the radar has no way on knowing where a reflection is coming from along the elevation, it assumes that every reflection is coming from the plane perpendicular to the radar. This causes ambiguity in elevation (y) while also making the azimuth (x) and depth (z) noisy. In addition, the wider beam-width of mmWave radars also leads to some noise along these axes. As a result, it is not possible to directly project the radar point clouds onto the image plane using the pose and camera intrinsics and use them as a means of obtaining depth of the scene. Previous works [Lin et al., 2020, Lo and Vandewalle, 2021, Gasperini et al., 2021] do not account for this and perform adhoc operations such treating the incorrect projections as is or extending each radar point along the y-axis of an image. Unlike them, we learn to map radar points to probable surfaces in the scene to recover denser radar point clouds.

3.4 Our Approach

Formulation. Our goal is to recover the 3D scene from a single RGB image $I \in \mathbb{R}^{3\times H\times W}$ and K points in a point cloud $\mathbf{z}$, where a point $z \in \mathbb{R}^3$, H and W are the height and width of the image – here K (akin to a batch dimension) may vary from point cloud to point cloud, which our method handles through our RadarNet (Sec. 3.4.1). We assume that the point clouds are captured by mmWave radars, which typically have incorrect elevation (y-) along with noisy azimuth (x-) and depth (z-) readings. We propose to learn a function that outputs the dense depth $\hat{d} \in \mathbb{R}_+^{H \times W}$ for every pixel in the image. Rather than directly learning a map from I and z to $\hat{d}$, we divide it into two sub-problems and solve them sequentially – (i) find correspondences between each point in the noisy radar point clouds and its probable projection onto the image plane to yield a semi-dense radar depth map and (ii) fuse information from the semi-dense radar map and the camera images to output $\hat{d}$.

Our approach is realized as two sequential deep neural networks: (i) RadarNet h_θ parameterized by θ takes an RGB image I and a radar point z as input and outputs a confidence map $h_\theta(I, z) \in [0, 1]^{H \times W}$ for the probable surfaces that the point maps to in the image. Alternatively, for K points in the point cloud $\mathbf{z}$, $h_\theta(I, \mathbf{z})$ outputs K confidence maps from which we construct a semi-dense radar depth map by selecting the z-component of the radar point corresponding to the maximum response greater than a threshold $\tau = 0.5$ in the hypothesis $h_\theta(I, \mathbf{z})$ for each pixel to yield radar depth map $\hat{z} \in \mathbb{R}_+^{H \times W}$. This allows us to process point clouds with any arbitrary number of points – for a single point, we naturally default to selecting its z-component for any response greater than τ. (ii) FusionNet f_ω parameterized by ω further fuses together I, $\hat{z}$ and its confidence for each correspondence $\hat{h}_\theta(I, \mathbf{z})$ to yield the dense depth map $\hat{d} = f_\omega(I, \hat{z}, \hat{h}_\theta(I, \mathbf{z})) \in \mathbb{R}_+^{H \times W}$. Fig. 5.1 shows the system overview of our two stage approach.

3.4.1 Learning Radar to Image Correspondence

We assume that we are given a dataset with training samples comprised of an RGB image I, radar point cloud $\mathbf{z}$, ground truth lidar depth map $d_{gt} \in \mathbb{R}_+^{H \times W}$. RadarNet h_θ is comprised of two encoders, one standard ResNet18 backbone [He et al., 2016] with 32, 64, 128, 128, 128 filters in each of its layers, respectively, to process the image and a multi-layer perception (MLP) of 5 fully connected layers with 32, 64, 128, 128, 128 neurons, respectively, to encoder the radar points. The latent of the point cloud is mean-pooled and reshaped to the size of the image latent, then together with skip connections from intermediate layers in the encoder, decoded into response maps or logits. We apply sigmoid activations to the logits to obtain the confidence scores $h_\theta(I, \mathbf{z})$.

To illustrate the challenge of radar to image correspondence, we note that there exists inherent ambiguities in determining radar to image correspondence since the point cloud lacks a viable elevation component. Also due to the noise in radar points, both depth and azimuth can vary between 10cm in the regions near the sensor and up to 40cm in the far regions [Yang et al., 2020]. Thus, unlike previous works [Lo and Vandewalle, 2021] that "extend" the radar point along the elevation (which would yield incorrect correspondences) by copying its z-component along the vertical (y-) direction to create "radar lines", we propose to associate the radar points to the many probable surfaces in the image within a search range of $H \times w$ image crop centered on the position of the point.

ROIAlign for efficient inference. A naive approach to mapping a radar point to probable regions in the image is to simply score the entire image. However, this would require an $H \times W$ search space for each point where most of it will yield low confidence scores – likely regions will be localized to a $H \times w$ crop. Instead, one may observe that the K points in $\mathbf{z}$ maps to the same scene and thus we only need to perform a single forward pass on the image and K forward passes for $\mathbf{z}$. To accelerate the process of finding these correspondences between the radar points and the camera image, we propose to extract regions of interest (ROIs) in the feature maps corresponding to each $H \times w$ crop using a ROI alignment mechanism [He et al., 2017]. Each ROI is the region within which the true position of the radar point lies – anywhere along the vertical axis H and within some region along the horizontal axis w as shown in Sec. 3.3.

Hence, to process an image and its associated point cloud, we extract ROIs from the image features for each point and stack them along the batch dimension. Hence, for K points, we will also have K corresponding ROIs for each encoder scale, which will be passed to the decoder to yield K confidence maps. We predefine a $K \times H \times W$ volume of zeros and transfer the output K number of $H \times w$ confidence scores to their respective ROI locations in the full $H \times W$ image lattice to yield $h_\theta(I, \mathbf{z})$.

We formulate the radar to image correspondence problem as a binary classification of each pixel for a given radar point z, where high responses in $h_\theta(I, z)$ indicate probable surfaces for a given point. As a final step in the forward pass to yield the corresponded radar depth map $\hat{z}$, for each pixel $x \in \Omega \subset \mathbb{R}^2$ i.e., the image spatial domain, we choose the maximum response over the K confidence maps $h_\theta(I, \mathbf{z})$:

$$\hat{z}(x) = \begin{cases} \mathbf{z}[\hat{k}], & \text{if} \quad h_\theta(I, \mathbf{z})(x)_{[\hat{k}]} > \tau \\ 0, & \text{otherwise,} \end{cases} \tag{3.1}$$

where $\hat{k} = \arg\max_k h_\theta(I, \mathbf{z})(x)_{[k]}$ and $\tau = 0.5$ a threshold.

Training RadarNet. we simplify the forward pass to just predicting the $H \times w$ confidence score maps i.e. without the need to choose the maximum response. For supervision, ROIs corresponding to the radar points are extracted from the ground truth d_{gt}. To construct the labels for binary classification, any pixel location in the ground truth d_{gt} that is within 40cm of the depth (z-) component of the radar point is set to belong to the positive class i.e., a correspondence, otherwise the negative class. In practice, the ground truth d_{gt} can be sparse or semi-dense depending on the specification of the lidar so there is a lack of supervision signal in regions where there are no lidar returns.

To address this, we assume that world surfaces are locally connected and piece-wise smooth and build a scaffolding [Wong et al., 2020] over the scene to approximate its dense structure. We then construct labels $y_{gt} \in \{0, 1\}^{H \times w}$ from the scaffolding and minimize a binary cross entropy loss: $\ell_{BCE} = \frac{1}{|\Omega|} \sum_{x \in \Omega} -\big(y_{gt}(x) \log y(x) + (1 - y_{gt}(x)) \log(1 - y(x))\big)$, where $\Omega \subset \mathbb{R}^2$ denotes the spatial image domain, $x \in \Omega$ a pixel coordinate, and $y = h_\theta(I, \mathbf{z})$ the hypothesis of radar to camera image correspondence. By training RadarNet to map radar points to regions in the image space, we are able to query RadarNet with arbitrary number of points to support the varied number of radar returns at each time frame to yield a semi-dense depth map that are several orders of magnitude denser than the radar point cloud.

3.4.2 Radar and Camera Image Fusion

Given the associated radar depth map $\hat{z}$ and its confidence map $\hat{h}(x) = \max_k h_\theta(I, \mathbf{z})(x)_{[k]}$, we propose to learn FusionNet f_ω to fuse $\hat{z} \in \mathbb{R}_+^{H \times W}$ and $\hat{h} \in [0,1]^{H \times W}$ with the RGB image I. FusionNet is comprised of two encoders with ResNet18 backbones, one with 32, 64, 128, 256, 256, 256 filters to encode the image $\phi(I) \in \mathbb{R}^M$ and the other with 16, 32, 64, 128, 128, 128 filters to encode the depth map concatenated with the confidence map $\psi([\hat{z}, \hat{h}]) \in \mathbb{R}^N$. The two branches are processed separately and later fused together via an adaptive weighting layer that learns the contribution of the depth encodings. This is because the radar points are inherently noisy and the depth map putative correspondences and thus we use a learned gating mechanism limit incorrect information flow from the depth branch. The reweighted depth encodings are added to the image encodings and passed as skip connections to the decoder to yield the dense depth map $\hat{d} \in \mathbb{R}_+^{H \times W}$.

Gated Fusion. While $\hat{z}$ is denser than the measured radar returns, it is admittedly still on orders of magnitude sparser than an image; hence, there will be many "empty" regions in an encoding of $\hat{z}$ and thus typical naive concatenation of the image and depth encodings [Fu et al., 2019, Hu et al., 2021, Jaritz et al., 2018, Jaritz et al., 2018, Ma and Karaman, 2018, Qiu et al., 2019, Yang et al., 2019, Van Gansbeke et al., 2019, Wong et al., 2020, Wong et al., 2021, Wong and Soatto, 2021, Xu et al., 2019] would result in convolving over many zero activations. To address this, we propose to augment the image features $\phi(I)$ with depth encodings by learning a set of weights $\alpha = \sigma(p^\top \psi([\hat{z}, \hat{h}])) \in [0,1]^M$ and projecting $\psi([\hat{z}, \hat{h}])$ to match the dimensionality of $\phi(I)$ via $\psi'(z) = q^\top \psi([\hat{z}, \hat{h}]) \in \mathbb{R}^N$, where p and q are trainable linear transformations and $\sigma(\cdot)$ the sigmoid function. The fusion step is given by $\alpha \cdot \psi'(z) + \phi(I)$ to produce the skip connection at each encoder scale and also the latent, which are fed to the decoder to yield $\hat{d} = f_\omega(I, \hat{z}, \hat{h}_\theta(I, \mathbf{z}))$ (see Fig. 5.1). Our gated fusion mechanism modulates the amount of depth information being passed to the decoder based on the training data and in effect learns the error modes of the radar depth map $\hat{z}$ and its confidence scores $\hat{h}$.

Training FusionNet. We assume access to the ground truth lidar depth d_{gt} and minimize the difference between the predictions d and d_{gt} with an L_1 penalty:

$$\ell_{L_1} = \frac{1}{|\Omega_{gt}|} \sum_{x \in \Omega_{gt}} |d_{gt}(x) - \hat{d}(x)|, \tag{3.2}$$

where $\Omega_{gt} \subset \Omega$ denotes the domain where ground truth d_{gt} has a valid depth value.

3.5 Implementation Details

Dataset. We use the nuScenes [Caesar et al., 2020] outdoor driving dataset for our evaluation. The dataset contains 1000 scenes of 20s duration each. A car is fitted with sensors such as a lidar, mmWave radar, camera and IMU and is driven around Boston and Singapore to collect these scenes. Since each sensing modality captures the scene at a different frequency, [Caesar et al., 2020] provides frames where the time-stamps of data from all sensors is very close to each other, called keyframes, which are annotated with object bounding boxes. The dataset contains around 40,000 keyframes (≈40 samples per scene). We use the nuScenes train-test split – 700 scenes for training, 150 for validation and 150 for testing.

Data Preprocessing. Following [Long et al., 2021b, Lin et al., 2020, Lo and Vandewalle, 2021, Long et al., 2021a, Gasperini et al., 2021], ground truth for a given frame is created by accumulating future and past frames by projecting the lidar point cloud at each time step to the frame of reference of the given frame – we use 161 frames in total (80 frames each from the future and past, and the given frame) to yield d_{gt}. Note: dynamic objects given by the bounding boxes are removed from the point clouds from each time step before projecting the points to the given frame. We used d_{gt} to supervise FusionNet. For RadarNet, we perform scaffolding [Wong et al., 2020] on d_{gt} to obtain an interpolated depth map, and use it to create labels y_{gt} for supervision. Note: We only use the accumulated lidar points for training; for evaluation, we use the lidar depth maps provided by [Caesar et al., 2020].

RadarNet (Stage-1). We use ROIs of size $H = 900$ and $w = 288$ for the input image size of 900 × 1600. For constructing y_{gt}, any point in d_{gt} within 0.4m of the z-component of a given radar point is marked as a positive example. We set the weight of positive class to 2 and train using a batch size of 6. We used Adam [Kingma and Ba, 2015] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize RadarNet with a learning rate of $2e^{-4}$ for 75 epochs. We use horizontal flip, saturation, brightness and contrast for data augmentations where each has a 50% probability of occurring. The values of brightness, contrast and saturation adjustment are random uniformly sampled from 0.8 to 1.2. Training takes $\approx$ 36 hours for 75 epochs on a NVIDIA RTX A5000 GPU.

FusionNet (Stage-2). We used Adam [Kingma and Ba, 2015] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize our network with a learning schedule $1e^{-3}$ for 100 epochs, then reduced to $5e^{-4}$ for another 100 epochs, and finally reduced $1e^{-4}$ for 50 epochs. The augmentations used during the training include horizontal flip, and brightness, saturation, and contrast adjustments, each with 50% probability of occurring. Like RadarNet, the values of brightness, contrast and saturation are random uniformly sampled from 0.8 to 1.2. We use a batch size of 16 with random crops of 448 × 448. Training takes $\approx$ 21 hours for 100 epochs on a NVIDIA RTX A5000.

3.6 Experiments and Results

Baselines. We compared our method against different methods [Lo and Vandewalle, 2021, Long et al., 2021b, Lin et al., 2020, Gasperini et al., 2021, Ma and Karaman, 2018, Wang et al., 2018] in Table 3.2 using error metrics in Table 3.1. We downloaded the pre-trained models from the official repositories for [Li et al., 2020a, Lo and Vandewalle, 2021, Long et al., 2021b] and test them on the official nuScenes [Caesar et al., 2020] test set. Results from [Lin et al., 2020, Gasperini et al., 2021, Ma and Karaman, 2018, Wang et al., 2018] were taken from their paper because code was unavailable or did not reproduce their numbers. We note that several baselines utilize either multiple images or multiple radar point clouds or both to estimate depth. For instance, RC-PDA [Long et al., 2021b] uses three camera images and five radar scans to compute "Flow" where the additional frames and scans include those from future timestamps. In real-world, one cannot expect to have access to information from the future, so this is not feasible. Additionally, they project future (for increasing density in far regions) and past (for increasing density in close regions) radar scans onto the current frame to densify the radar returns. RC-PDA with HG is a variant of [Long et al., 2021b] that uses an hourglass (HG) [Li et al., 2020a] network. DORN [Lo and Vandewalle, 2021] combines 5 radar scans from 3 different radars.

We also provide an ablation study in Table 3.2 to gauge the gain from RadarNet. Instead of using the semi-dense dense map concatenated with the confidence map, we simply train our FusionNet to directly estimate dense depth using raw radar points and a camera image. The results of this model as shown in Table 3.2 as Ours (No RadarNet). This also demonstrates the drawback of projecting the raw radar points onto the image plane and treating the sparse depth map (about 50 to 70 points per frame) as input – as customary in existing works [Lo and Vandewalle, 2021, Lin et al., 2020, Gasperini et al., 2021, Ma and Karaman, 2018, Wang et al., 2018].

Depth Considerations. According to the nuScenes [Caesar et al., 2020] documentation, the range of the lidar sensor used is between 80 - 100 meters. However, the usable range is only up to 70 to 80 meters [Weng et al., 2021]. Hence, we test all models with working code between 0-50, 0-70 and 0-80 meters.

Metric	units	Definition
MAE	mm	$\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\hat{d}(x)-d_{gt}(x)\rvert$
RMSE	mm	$\left(\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\hat{d}(x)-d_{gt}(x)\rvert^2\right)^{1/2}$

Table 3.1: Error metrics for evaluating the depth estimation benchmarks, where d_{gt} is the ground truth.

Max Eval Distance	**Method**	**# Radar frames**	**# Images**	**MAE ↓**	**RMSE ↓**
50m	RC-PDA [Long et al., 2021b]	5	3	2225.0	4156.5
	RC-PDA with HG [Long et al., 2021b]	5	3	2315.7	4321.6
	DORN [Lo and Vandewalle, 2021]	5(x3)	1	1926.6	4124.8
	Ours (no RadarNet)	**1**	**1**	1942.5	3986.1
	Ours	**1**	**1**	**1727.7**	**3746.8**
70m	RC-PDA [Long et al., 2021b]	5	3	3326.1	6700.6
	RC-PDA with HG [Long et al., 2021b]	5	3	3485.6	7002.9
	DORN [Lo and Vandewalle, 2021]	5(x3)	1	2380.6	5252.7
	Ours (no RadarNet)	**1**	**1**	2318.2	4825.0
	Ours	**1**	**1**	**2073.2**	**4590.7**
80m	RC-PDA [Long et al., 2021b]	5	3	3713.6	7692.8
	RC-PDA with HG [Long et al., 2021b]	5	3	3884.3	8008.6
	DORN [Lo and Vandewalle, 2021]	5(x3)	1	2467.7	5554.3
	Lin [Lin et al., 2020]	3	1	2371.0	5623.0
	R4Dyn [Gasperini et al., 2021]	4	1	N/A	6434.0
	Sparse-to-dense [Ma and Karaman, 2018]	3	1	2374.0	5628.0
	PnP [Wang et al., 2018]	3	1	2496.0	5578.0
	Ours (no RadarNet)	**1**	**1**	2441.0	5141.4
	Ours	**1**	**1**	**2179.3**	**4898.7**

Table 3.2: We compare our method to the pre-trained baselines that use multiple camera images and radar scans for radar-camera depth estimation. The authors in [Gasperini et al., 2021] do not provide MAE numbers. In DORN [Lo and Vandewalle, 2021], the authors use 5 radar scans from 3 different radars.

Quantitative Results. We compare our methods with the existing methods at 50, 70, and 80 meters depth range in Table 3.2. Compared to baseline RC-PDA [Long et al., 2021b], our method improves MAE by 22.3%, 37.6% and 41.3% and RMSE by 9.8%, 31.4% and 36.3% when the depth is being evaluated up to 50, 70 and 80 meters respectively. Compared to RC-PDA with HG [Long et al., 2021b], our method improves MAE by 25.3%, 40.5%, 43.8% and RMSE by 13.3%, 34.4%, 38.8%. Our method outperformed DORN [Lo and Vandewalle, 2021] by 10.3%, 13%, 11.7% when compared based on MAE and evaluated up to 50, 70 and 80 meters respectively. Similarly, our method improves RMSE by 9.1%, 12.6%, and 11.8% for those ranges. Overall, our method outperformed the best baseline evaluated by 10.3% MAE and 9.1% RMSE. We attribute our success largely to RadarNet being able to correctly correspond radar points to the objects in the scene, which has limited existing methods [Lo and Vandewalle, 2021, Long et al., 2021b, Lin et al., 2020, Gasperini et al., 2021, Ma and Karaman, 2018, Wang et al., 2018] that either directly used the erroneous points or perform adhoc post-processing i.e. vertical extension [Lo and Vandewalle, 2021] on them.

Efficacy of RadarNet Ours (No RadarNet) method performs better than several baselines in Table 3.2. The input to this method is a sparse depth map generated by projecting the raw radar points onto the image plane. Although this depth map contains a somewhat accurate distribution of depths in the scene, the locations are completely erroneous (Sec. 3.3). The difference in performance of our method with and without RadarNet demonstrates the advantage of our RadarNet model which not only helps in correcting the errors in the radar point cloud but also densifies the radar output into a semi-dense depth map. This study additionally confirms the detriment of input noisy radar points as a sparse depth map [Lo and Vandewalle, 2021]. A qualitative comparison of our method with and without RadarNet is shown in Fig. 3.5.

Qualitative Results. In Fig. 3.4, we plot the dense-depth output of our method and the baselines on the nuScenes test dataset. We choose two representative scenes, one of a busy intersection (left column) and one of a pedestrian crossing a road while the traffic is moving

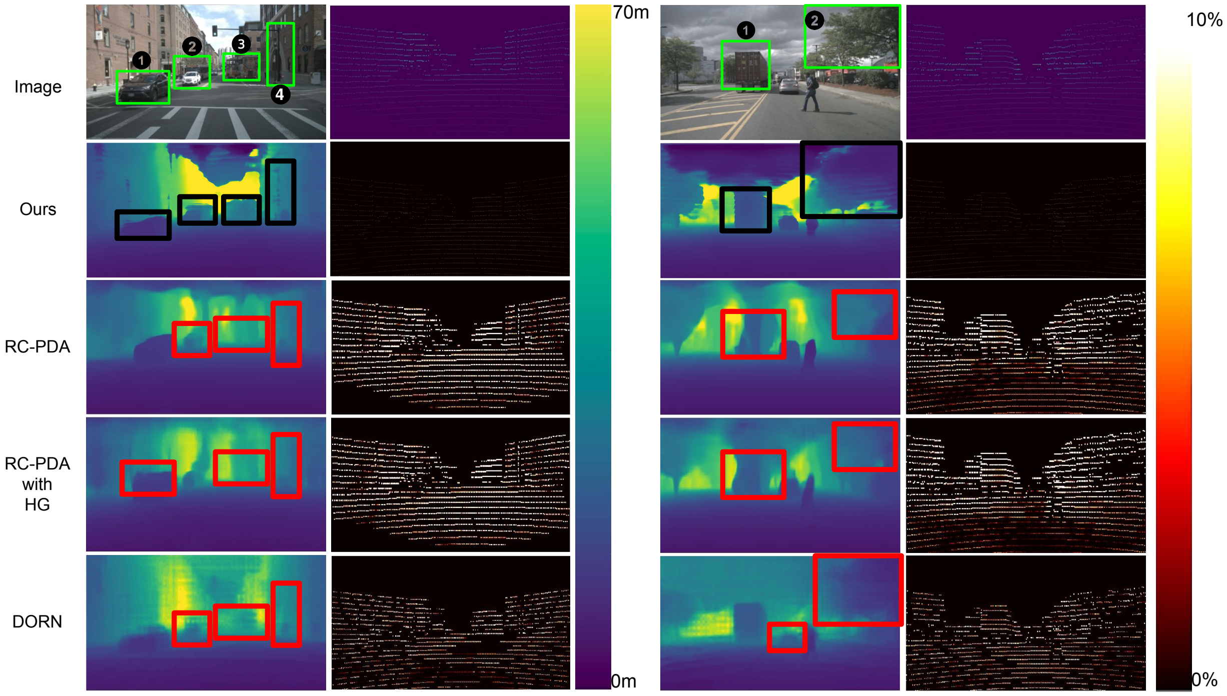


Figure 3.4: Qualitative results for all the methods on two different images from the test set (best viewed in color at 5x). The top row shows the image and the ground truth while the other rows show dense-depth generated by various methods and the error. The range of depth is between $0 - 70$ meter (as shown in the colorbar at the center) while the range of error is between 0% to 10% (as shown in the colorbar on the right). We mark errors in different baselines with a red box and contrast them with our own method.

during overcast weather (right column). We plot the ground truth next to the image on the right. Below the image and the ground truth, we show the output and error for our method as well as the baselines. Our error is much lower (dark) compared to other methods (bright) as evident in our quantitative improvements over them. We note that there is no supervision available in the top part of the scene (the top part generally contains the sky), so all the models hallucinate depth values for those pixels. On the left side, all models think the sky to be a continuation of the buildings along the road. On the right side, since the weather is overcast, all models think the sky to be a part of one of the nearby surfaces (due to the dark color). We use a black box to mark the qualitative advantages of our model when compared to the baselines (whose mistakes are marked with red boxes).

In the left column, our method is the only one which is able to pick up the bus in front (green box, number 3) which is trying to switch lanes. RC-PDA with HG outputs the wrong shape for the black car on the left side (number 1) of the scene where as the output of DORN is full of linear streaks (possibly artifacts from training with sparse supervision). For the building on the left side of the scene, our method shows a smooth increase in depth whereas the other methods have abrupt changes. Box number 2 contains a white car in front of a bus. Only our method picks up this change in depth for the two vehicles while some baselines fail to capture it. Such is also the case for the traffic light post (number 4) that is captured by our method but not by others. For all the baselines, the relative error is very high (all white regions in the error map correspond to the relative error being 10% or more). For our method, high errors are concentrated more along the edge of objects such as cars and poles on the street.

In the right column, the scene depicts a pedestrian who is trying to cross the street in the middle while a car is in the right lane and a truck (number 1) is in the turning lane. The weather is overcast and the clouds above are dark. There is a tree branch (number 2) followed by a set of trees on the right top corner which is missed by both the RC-PDA baselines. The DORN baseline is able to pick the tree branch but missed all the trees behind

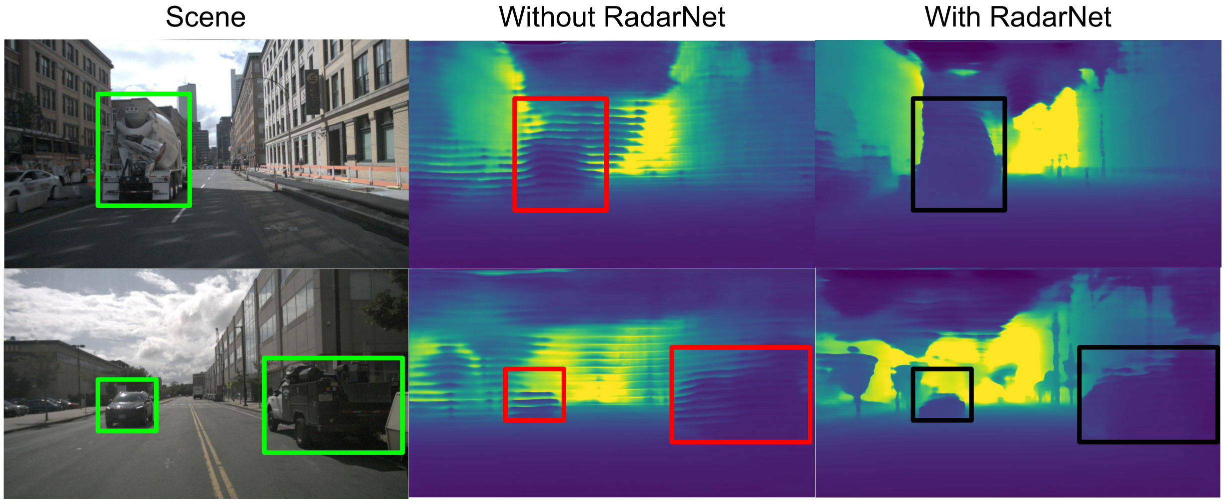


Figure 3.5: Qualitative comparison between our method with and without RadarNet. RadarNet's quasi-dense output enables FusionNet to learn shapes of objects effectively as shown in the bounding boxes above while also improving it's quantitative performance.

it. It also misses a big portion of the car stopped in front. Both the RC-PDA methods map two or three different depth values to the back of the truck. Since the background immediately to the left of the truck is dark, both the methods think that it is a continuation of that background. For all the baselines, the error is high (as shown in the error map) whereas our models suffers high errors along object edges.

In Fig. 3.5 we show the qualitative comparison between our method with and without RadarNet. While the model without RadarNet outperforms the baselines on quantitative metrics, it is unable to effectively learn shapes of the objects in the scene such as cars. This can be attributed to the fact that since metallic surfaces are better reflectors of radars, RadarNet is able to learn the shapes of these surfaces. These learned shapes act as a priors and enables FusionNet to learn scene geometry effectively.

3.7 Discussion

As radar sensors become more ubiquitous in our surroundings, it is essential to develop methods to integrate them with the existing inference pipelines such as the depth estimation frameworks. However, unlike other sensors, the long range, large field of view and high ambiguity of radar point clouds make it challenging for them to be used in the same manner as other sources of point clouds such as lidars.

In this chapter, we focus on a single camera image and a single radar point cloud because with the advent of JCAS with 6G, radars are going to be a part of our infrastructure. Our cellular base-stations will sense their surroundings while also acting as the hub of communication. In such cases, since the sensor itself is not moving, there will be little to no benefit of combining consecutive frames (such is the case with RC-PDA that use multiple camera images) as to the sensor, the environment is stationary (ignoring small movements such as human beings, cars, etc.). Hence, it is imperative that we work towards creating methods that can estimate dense depth from a single vantage point.

The gated fusion mechanism of our FusionNet presents us with some advantages. Firstly, in case of very noisy radar points, the model can learn to rely more on the image branch by assigning a smaller weight to the depth branch. Secondly, in case the radar points are completely erroneous, the model can learn to solely rely on the image input. Thirdly, this mechanism ensures that in situations where the correspondence model is not very good, the performance of the depth completion stage does not significantly suffer since it can correct for error-prone values.

However, such a mechanism does have drawbacks. If the camera-radar setup is mis-calibrated or mis-aligned, the network may assume the radar values to be ‘bad’ and only rely on camera branch to predict depth. Additionally, it is well-known that softmax activations of a deep neural network are neither calibrated nor a substitute for uncertainty. Hence, there can be erroneous over-confident correspondences in the our RadarNet predictions. The goal of our gated fusion layer is to counteract such a case.

CHAPTER 4

Learning over Unlabeled RF Data

4.1 Introduction

State-of-the-art deep learning (DL) sensing models have typically required manual annotation of empirical data. For example, ImageNet, the flagship 14 million images dataset for object recognition, took 22 human years to annotate [NYU, 2021]. Clearly, the progress of radio sensing should not hinge on the availability of *laboriously annotated* data. Alternatively, radio sensing models could be trained on synthetic RF data using ray-trace simulations [Singh et al., 2018, Hsiao et al., 2017, da Silva et al., 2020]. Since the environment is simulated, we can specify the sensing scenario (i.e., label) first and then generate the corresponding data. However, synthetic data would only provide a 1st-order approximation of the real-world that is unlikely to meet the quality, scale, and richness requirements of highly-perfomant deep radio sensing models. Current synthetic data generation techniques for RF work well for simple environments, but their complexity-bounded performance degrades as we move to more realistic scenes which mirror the real-world [Degli-Esposti et al., 2014, Lecci et al., 2020, Fuschini et al., 2017].

Recent advances in self-supervised learning have demonstrated a viable alternative to label-intensive supervised learning [Bengio et al., 2013]. Self-supervised methods provide two key advantages. Firstly, they do not require labelled data. Instead of solving a specific learning task, we solve an auxiliary task that helps the model learn (by *proxy*) the underlying structure of the data. Secondly, self-supervised methods learn more granular and general-

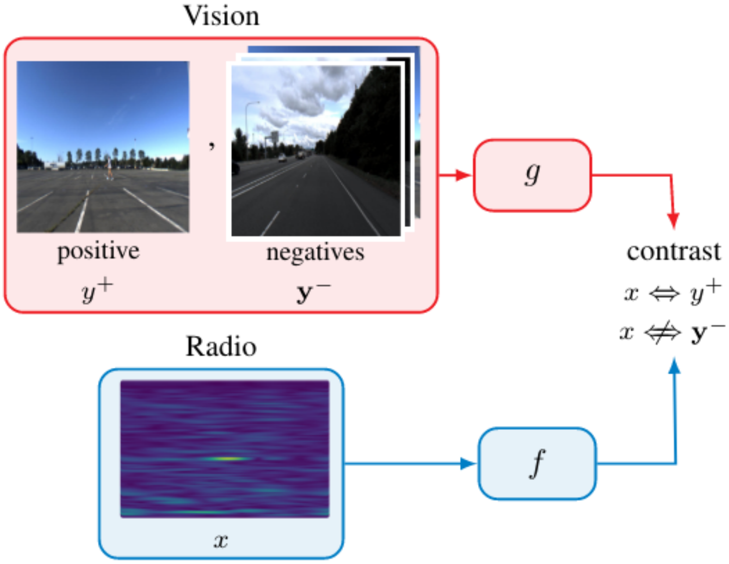


Figure 4.1: Contrastive radio-visual learning. Image y^+ is a positive pair of radar heatmap x because they are recorded simultaneously, while other randomly-sampled images $\mathbf{y}^-$ are negative pairs.

isable discriminative features than task-specific supervised methods [Henaff, 2020]. This is because self-supervision generates higher information rate per sample than supervised learning as captured eloquently in Yann LeCun's cake analogy [LeCun, 2019].

In this chapter, we draw inspirations from recent Natural Language Processing (NLP) breakthroughs [Devlin et al., 2018, T. B. Brown et al., 2020] and Computer Vision (CV) successes [Chen et al., 2020b, Henaff, 2020] and propose a self-supervised method for radio representation learning. Specifically, we formulate an auxiliary prediction task that contrasts RF heatmaps to camera images. We show that such auxiliary *contrastive* prediction promotes learning powerful radio neural networks that can be specialised for specific sensing applications. We argue that self-supervision for radio signals is an important enabler for emergent 6G sensing systems.

The chapter describes the following contributions:

- To the best of our knowledge, this work is the first to demonstrate successful self-supervised representation learning for radio signals using cross-modal mutual information (MI) with camera images. A loss function bounded by MI enables such *label-free* learning.

- We specialise the self-supervised radio model for a downstream task that classifies objects of interest in scenes from empirical measurements.

- We characterise our self-supervision-based model using a number of quantitative and qualitative metrics. When evaluated against a fully-supervised baseline, we find that the self-supervised model outperforms its supervised counterpart when less labelled data is used for training.

4.2 Overview & Intuition

Radio heatmaps are sparse representations of the environment. They are difficult to interpret and label by humans, because the underlying geometry of the physical space they capture is not straightforward to decipher.

Fig. 4.1 depicts our proposed radio-visual co-learning scheme. The idea is to (i) match a radio heatmap with its corresponding groundtruth image and (ii) contrast this true radio-vision pairing against a number of false pairings using images sampled at random. The very act of emphasising the *contrast* between positive and negative radio-visual pairings gives rise to a robust learning signal as will be formalised and explained later. Further, this simple yet powerful *contrastive* learning can be self-administered, assuming only synchronised radio-visual measurements (i.e., true pairings). Hence, the *contrastive* learning scheme is also *self-supervised* and needs no labels.

4.3 Primer

We discuss next the foundational concepts of our self-supervised radio-visual learning.

4.3.1 Self-supervised learning

Unlike supervised learning where we need to manually annotate the data, self-supervised learning leverages intrinsic labels which can be "contrived" using the data itself. The most successful form of self-supervised learning to-date is in NLP, whereby a language model tries to predict randomly masked words in sentences. Surprisingly, this simple contrived prediction acts as a signal that forces the network to learn fundamental aspects that characterise language altogether, e.g., word association and sentence-to-sentence context. In addition to its groundbreaking applications in NLP [Devlin et al., 2018, T. B. Brown et al., 2020], self-supervised learning has also seen numerous successes in domains such as vision [Doersch et al., 2015] and multi-modal learning [Arandjelovic and Zisserman, 2018]. For example, visual self-supervised learning can be made to work by predicting whether randomly sampled image patches are related or not [Doersch et al., 2015].

4.3.2 Contrastive loss

For self-supervision to be effective, the signal that drives learning should also accentuate the difference (i.e. *contrast*) between true and false predictions and their data samples. The contrast between a positive datum and negative data is at the heart of learning good self-supervised models. There are many incarnations to such a contrastive loss [Oord et al., 2018, He et al., 2020, Afouras et al., 2020]. InfoNCE is one contrastive loss designed to drive learning in a way that preserves the mutual information (MI) between inputs [Oord et al., 2018]. Concretely, referring to Fig. 4.2, let x and y_i be two input signals encoded by two neural networks f_θ and g_θ such that $q = f_\theta(x)$ and $k_i = g_\theta(y_i)$, assuming some weights parametrisation θ. With each x, use $K + 1$ samples of y of which one sample y^+ is a true

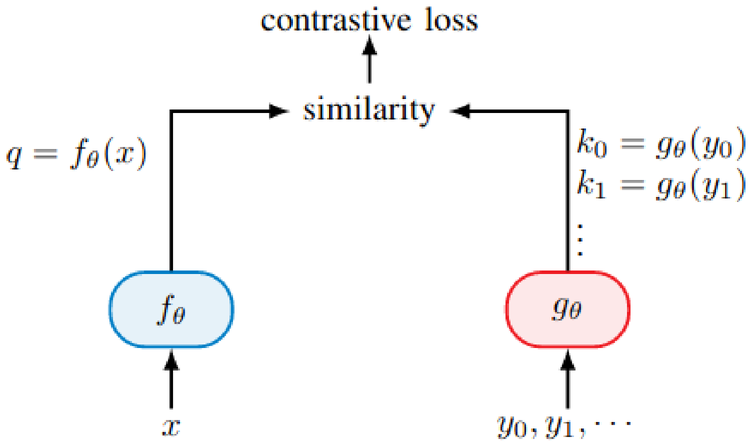


Figure 4.2: Contrastive loss for self-supervised learning.

match to x and K samples $\{y_i^-\}_{i=0}^{K-1}$ are false matches. Then the constrastive loss $\mathcal{L}_c$ is [He et al., 2020]

$$\mathcal{L}_c = -\mathop{\mathbb{E}}_{x,y} \log \left[\frac{\exp\left(q \cdot k^+/\tau\right)}{\exp\left(q \cdot k^+/\tau\right) + \sum_i \exp\left(q \cdot k_i^-/\tau\right)} \right] \tag{4.1}$$

where $\cdot$ is the dot product operator, $k^{+/-} = g_\theta(y^{+/-})$ are encodings that correspond to true and false y signals, vector $\mathbf{k}^- = \{k_i^-\}_{i=0}^{K-1}$ holds K false encodings, and τ is a temperature hyper-parameter.

The contrastive loss of Eq. (4.1) has enabled a wave of progress in self-supervised image classification that inches ever closer and at times exceeds the performance of fully supervised methods. Notably, SimCLR [Chen et al., 2020a] and MoCo [Chen et al., 2020b] are recent examples of bleeding-edge self-supervised vision systems.

The contrastive loss $\mathcal{L}_c$ has information-theoretic connections to the MI between q and k (vis-à-vis x and y). This can be seen noting the following. First, the expression $\exp\left(q \cdot k/\tau\right)$ models a positive real score between q and k, which is proportional to the (unnormalised) density ratio

$$\exp\left(q \cdot k/\tau\right) \propto \frac{p(k|q)}{p(k)} \tag{4.2}$$

Second, Eq. (4.1) has the form of a $(K+1)$-way softmax classifier that matches q to its positive pair k^+. Using Eq. (4.2) in Eq. (4.1), Oord et al. show that $\mathcal{L}_c$ preserves the MI between q and k [Oord et al., 2018]. Specifically, $\mathcal{L}_c$ becomes lower bounded by the MI between the encoded inputs[1] and that using more negative samples K enhances the learnt representation [Oord et al., 2018, Poole et al., 2019]

$$\mathcal{L}_c(q, k^+, \mathbf{k}^-) \geq \log(K) - I(q; k) \tag{4.3}$$

$$\mathcal{L}_c^{\text{optimal}}(q, k^+, \mathbf{k}^-) = \log(K) - I(x; y) \tag{4.4}$$

where $I(q; k)$ & $I(x; y)$ denote respective MI's.

4.3.3 Cross-modal co-supervision

The notion of self-supervision can also be applied across different modalities for data with inherent multi-modal signals, e.g., video. The sheer amount of video freely available on the internet makes this proposition increasingly appealing. For instance, there is a body of literature that studies how representations for audio and vision can be learnt *jointly*, e.g., [Arandjelovic and Zisserman, 2017, Afouras et al., 2020] (and references therein). Besides learning quality audio and vision features that are on par with supervised single-modal approaches, additional cross-modal tasks can be *automatically* instantiated (from the joint embeddings), e.g., synchronisation and object localisation [Afouras et al., 2020, Chen et al., 2021].

Similar to NLP and vision, audio-visual self-supervised systems use a contrived auxiliary task to promote cross-modal representation learning. Earlier works dealt with predicting the *correspondence* of an image and an audio spectrogram snippet using a binary cross-entropy loss, e.g., [Arandjelovic and Zisserman, 2017]. Audio-visual correspondence in this context is static, i.e., from a single measurement snapshot. More powerfully, recent works extend this correspondence to capture shared temporal dynamics between a video snippet and an audio snippet, e.g., [Afouras et al., 2020].

[1] which are in turn lower bounded by the MI between the raw inputs

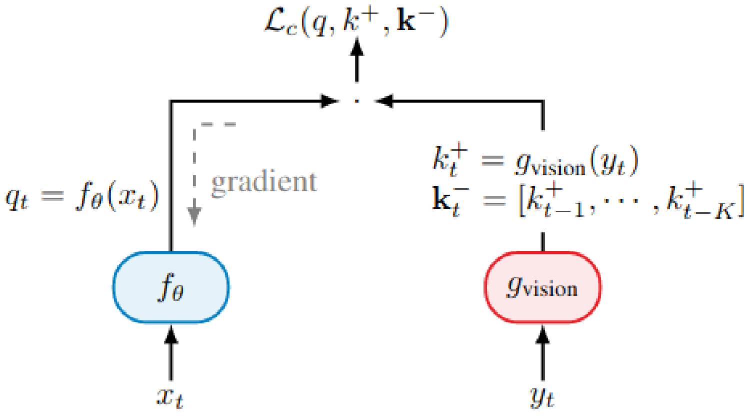

$\mathcal{L}_c(q, k^+, \mathbf{k}^-)$
$q_t = f_\theta(x_t)$
gradient
$k_t^+ = g_{\text{vision}}(y_t)$
$\mathbf{k}_t^- = [k_{t-1}^+, \cdots, k_{t-K}^+]$
f_θ
g_{vision}
x_t
y_t

Figure 4.3: Radio-visual contrastive loss.

4.4 Radio-Visual Contrastive Learning

We next turn to discussing how to adapt the concepts treated in Sec. 4.3 for radio-visual self-supervised learning.

4.4.1 Pre-training

Our objective is to leverage the co-occurrence of events in the radio and vision domains and formulate a learning scheme that requires no labels. That is, the onset of an event of interest across radio and vision will give rise to MI that can be captured and used as an automatic learning signal. The judicious sampling of positive and negative data according to Eq. (4.1) extracts cross-modal MI.

Data collation. In practice, we operate a pair of camera and radar devices that are synchronised to continuously collect measurements. The synchronisation timestamps allow us to construct the positive and negative data sample set $S_d = \{x_t, y_t^+, \mathbf{y}_t^-\}$ and their respective encodings $S_e = \{q_t, k_t^+, \mathbf{k}_t^-\}$, at a given time instance t. The resultant dataset can be further filtered by *light*[2] object detection algorithms in the vision domain to: concentrate on learning good encodings for specific objects of interest (e.g., pedestrians, cyclists, and cars), balance

[2] i.e., without laborious annotations such as bounding boxes

their distributions, etc.

Architecture. Our machine learning model consists of two parts: (i) a radar branch and (ii) a camera branch. During training, the two branches interact through the contrastive loss of Eq. (4.1).

Representation learning. Let $(x_t,\ y_t)$ be a pair of a radio heatmap and a corresponding camera image, as depicted in Fig. 4.3. A neural network with two branches for radio and vision ingests $(x_t,\ y_t)$. For the vision subnetwork, we use a pretrained convolutional model to encode y_t, yielding a positive encoding $k_t^+ = g_{\text{vision}}(y_t)$. The vision subnetwork is frozen and there is no gradient flowing back to it during training. This aspect is different to self-supervised vision systems [Chen et al., 2020a, Chen et al., 2020b]. The idea is to leave the vision encoder maximally consistent during training, which would significantly enhance the contrastive learning of the radar subnetwork. Under our cross-modal settings, it also makes sense to derive learning from the higher-entropy vision modality. We adopt a queue of negative encodings scheme proposed by He et al. in their momentum contrast (MoCo) system [He et al., 2020]. The negative encodings queue $\mathbf{k}_t^-$ is constructed by utilising past image encodings that do not correspond to the current radar heatmap[3], i.e., $\mathbf{k}_t^- = [k_{t-1}^+, \cdots, k_{t-K}^+]$. The queue is a very efficient mechanism to enhance contrastive learning (cf. loss bound in Eq. (4.3)) under practical compute and memory constraints. Throughout training, a radar heatmap x_t is encoded according to $q_t = f_\theta(x_t)$, and a gradient is computed to optimise for q_t's similarity to its positive image pair k_t^+ and dissimilarity to K negative images in $\mathbf{k}_t^-$. This K-way contrastive learning can be trivially extended for the mini-batch settings of stochastic gradient decent (SGD).

[3] by construction of the dataset

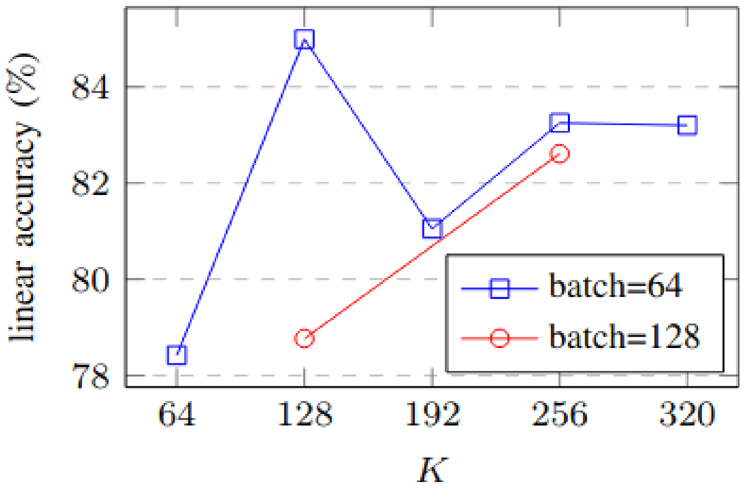


Figure 4.4: Linear classification as a function of number of negatives K, for two batch sizes. Generally, linear classification is enhanced with contrastive learning against more negatives, which agrees with the theoretical MI bound.

4.4.2 Fine-tuning

Contrastive representation learning concludes by arriving at a good f_θ than can used in a variety of sensing applications. A task-specific model is then constructed using the output of f_θ as features combined with a classifier head. Typically, sensing tasks also fine-tune (i.e., bias) these pre-trained representations towards application-specific criteria.

4.4.3 Implementation details

For both the vision and radio branches, we use pre-trained VGG-16 models [Simonyan and Zisserman, 2014]. The VGG-16 model takes a image of $3 \times 224 \times 224$ dimensions as input. For the radar branch, the heatmap is resampled and replicated 3 times for a valid input. The vision branch is frozen and only the radar branch is trained. We slice the VGG-16 model at the 4th layer of its classifier stage to obtain a 4096-dimensional feature vector, i.e., $q, k \in \mathbb{R}^{4096 \times 1}$. We use MoCo's shuffling trick to mitigate against batch normalisation (BN) statistical issues [He et al., 2020]. BN's statistical artefacts have also been observed in other works, e.g., [Labatie et al., 2021, Alloulah et al., 2021].

4.5 Experiments

4.5.1 Dataset

We use the Camera-Radar of the University of Washington (CRUW) dataset [Wang et al., 2021b] for evaluating our radio-visual self-supervised learning. The dataset contains around $47,000$ image-heatmap pairs in a variety of settings such as empty parking lots, driving on roads, and in front of buildings. Objects present in the dataset are: pedestrians, cyclists, and cars. An example pair is depicted in Fig. 4.1.

Downstream task. We balance the class distribution of a subset of the dataset such that there are 4 non-overlapped classes in any given scene: empty, pedestrian, cyclist, and car. The resultant 4-category sensing task maps onto the envisioned 6G use cases, e.g., pedestrian versus car discrimination for safety applications [Wild et al., 2021]. We use a standard 80/20 train/test split.

4.5.2 Training

Similar to MoCo [Chen et al., 2020a, Chen et al., 2020b], we use an stochastic gradient descent (SGD) optimiser for contrastive pre-training. The learning rate starts at 0.03 with an SGD weight decay of 1e-4 and momentum of 0.9. The learning rate decays according to a cosine schedule. We use two mini-batch sizes: 64 and 128. We train in a distributed fashion on 4 GPUs for 400 epochs, which takes $\sim$12 hours for VGG-16. For training the fully-supervised baseline, fine-tuning, and the linear classifier, we use a 1e-4 learning rate. For the fully-supervised baseline, we train for 300 epochs. For fine-tuning and the linear classifier, we train 32 epochs only.

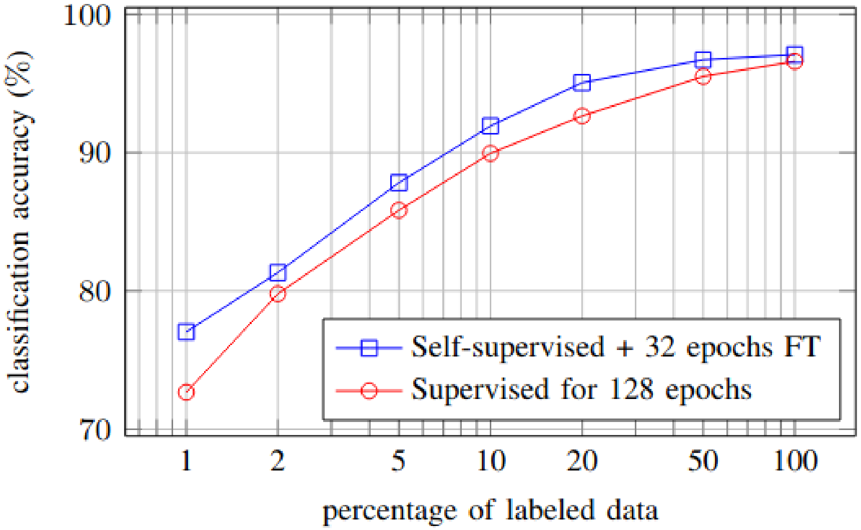


Figure 4.5: Efficiency measured w.r.t. accuracy versus percentage of labels used for fine-tuning (FT). Self-supervised efficiency is compared to its fully-supervised counterpart. Self-supervised contrastive learning needs 32 epochs to *exceed* the level of performance of fully supervised training using 128 epochs: a factor of **4**× reduction in training.

4.5.3 Evaluation protocol

Linear classification. We benchmark the quality of our self-supervised radio representation learning using the common *linear* classification protocol. Under the linear classification protocol, we: (1) pre-train using contrastive learning, (2) freeze the learnt features, and then (3) train a supervised classifier with a *linear* fully-connected (FC) layer whose output is normalised using a softmax.

Efficiency. We investigate the data and training efficiency of the contrastive representation f_θ. Data efficiency refers to the amount of labels required to build a downstream task using f_θ. Training efficiency refers to the number of epochs required to converge to a downstream representation with competitive performance. Both metrics are evaluated against their fully-supervised counterparts.

4.6 Results

4.6.1 Quantitative

We begin by examining the linear classification performance of self-supervised radio features.

Linear classification. Eq. (4.3) tells us that the contrastive loss benefits from increasing the number of negatives. We investigate the effect of queue size K on the linear classification accuracy in Fig. 4.4. For mini-batch size 64, we sweep K from 64 to 320 in 64 increments. For mini-batch size 128, $K \in [128, 256]$ only due to GPU resource constraints. Generally, apart form a blip in mini-batch 64 at $K = 128$, linear accuracy increases steadily with K for both mini-batches.

Efficiency. Fig. 4.5 depicts the label efficiency of self-supervised features and compares it to the fully-supervised efficiency. Using all dataset labels, both achieve a comparable accuracy of 97%. However, self-supervised features beat the supervised ones when we decrease the number of labels used for training. For instance, with 1% labels, we see a gap of about 3.5% in favour of self-supervision. This observation is generally in agreement with recent findings from the CV domain [Henaff, 2020], which report on even more drastic advantages to self-supervision in the "low label density" regime.

In terms of training efficiency, the self-supervised features can reach the fully-supervised performance quite quickly within 32 epochs of fine-tuning. This is corroborated in Fig. 4.6 that analyses the testing loss during fine-tuning against its fully-supervised counterpart. This indicates that the self-supervised representation has indeed learnt good generalisable features. We expect much greater data efficiency gains under more elaborate classification settings, such as a 1000-category classification task [Henaff, 2020]. Note that unlike fully-supervised systems, the initial training of self-supervised features would amortise over many different downstream tasks.

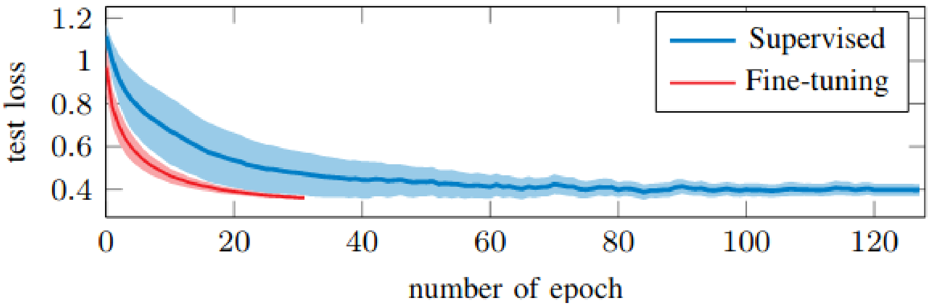


Figure 4.6: Comparing the testing losses of the fully-supervised baseline and the fine-tuned model with early stopping for the downstream classification task.

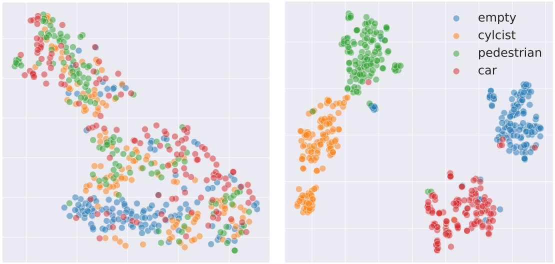
4.6.2 Qualitative

Learnt representation. As discussed in Sec. 4.4.3, we start the self-supervision of the radio branch from a pre-trained vision VGG network. Could this, rather than contrastive training, be responsible for the very good performance on the downstream task? After all, objects showing as blobs in radio heatmaps should in theory also benefit from many learnt vision filters, such as edge and contour detectors. To investigate this, we use t-SNE [Van der Maaten and Hinton, 2008] to visualise the 128-dimensional features of the penultimate layer for (a) a pre-trained VGG combined with randomly initialised classification head, and (b) our fine-tuned contrastive model. Results are shown respectively in Fig. 4.7. Inspecting Fig. 4.7a, no structure in the latent space can be discerned in relation to our 4-category downstream task. In contrast, the latent space of our fine-tuned model exhibits clear clustering around the 4 categories of the downstream task as evident in Fig. 4.7b.

4.7 Related Work

Self-supervised learning, where the model is trained on implicit labels present within the dataset, has gained popularity in the fields of NLP and CV recently. In this section we highlight some key related works.

Computer Vision. Self-supervised CV models trained with simple auxiliary tasks lead to



(a) pre-trained VGG with random head

(b) fine-tuned contrastive radio model

Figure 4.7: t-SNE visualisation of learnt representations.

surprisingly good vision representation learning. Examples include solving puzzles [Noroozi and Favaro, 2016], image colorisation [Zhang et al., 2016], and predicting image rotations [Gidaris et al., 2018]. For the latter for instance, self-supervised learning works by randomly rotating all the images in a dataset and asking the model to predict these rotations. Bleeding-edge self-supervision systems include SimCLR by Google [Chen et al., 2020a] or SwAV by Facebook Research [Caron et al., 2020]. SimCLR generates two augmentations for the same image and tries to maximise the similarities between their latent representations. SwAV builds on SimCLR and adds multi-crop augmentation, i.e., two copies of an image at two different resolutions. After training on auxiliary tasks, the learnt features are then fine-tuned for a given downstream task.

RF Sensing. Self-supervised learning has recently been applied to radar systems research. In [Orr et al., 2021], the authors propose a super-resolution method called Radar signal Reconstruction using Self-supervision (R2-S2) which improves the angular resolution of a given radar array without increasing the number of physical channels. In [Gasperini et al., 2021], the authors propose R4Dyn which uses radars during training as a weak supervision signal, as well as an extra input to enhance the depth estimation robustness at inference time.

CHAPTER 5

Detecting, Identifying and Localizing Devices using RF

5.1 Introduction

The proliferation of low-cost wireless sensors has facilitated increased adoption into smart home, building, and city deployments [Staff, 2018b, Heater, 2019]. Although there are profound positive impacts that ubiquitous sensor-rich environments can have on society, there is an inherent risk in enabling users access to such pervasive sensing, particularly when these environments host occupants oblivious to the presence of these sensors.

An individual's privacy in these contexts is entirely at the discretion of the owner. Regulation is unclear in informal settings, such as a guest residing in a homestay lodging. There have been reported instances where a hosting owner has attempted to spy on homestay occupants [Fussell, 2019], motel lodgings [Jeong and Griffiths, 2019], and rooms aboard cruise ships [Staff, 2018a]. There are even instances in well-established hotel chains and mall restrooms when a malicious employee or customer has bugged several rooms [Press, 2019]. Beyond commercial applications, Southworth *et al.* report that domestic abusers may use such sensors for intimate partner stalking [Southworth et al., 2007]. Thus, potential victims with privacy concerns must take a proactive approach to detect clandestine sensors.

The prevalent method to detect bugs relies on an RF receiver that senses if the received power in a particular frequency range is above a certain threshold. However, as bug detectors work on the principle of sensing surrounding RF signals, they can easily be triggered by legitimate RF devices such as mobile phones, radios, smart TVs, and other smart devices, thus limiting the practicality of these detectors. An alternate method has emerged to detect the presence of IoT devices based on network traffic statistics [Huang et al., 2019a]. However, these methods only ascertain the presence of a device without semantic information regarding device information, location, or whether the device is actually monitoring a user.

More sophisticated solutions have since emerged targeting wireless cameras specifically. Wampler *et al.* [Wampler et al., 2015] showed that changing lighting conditions causes notable variations to appear in a wireless camera's video traffic; that is, video encoding leaks sensitive environmental information. Flickering a light source for a short period of time can then be used in correlation with network traffic changes to identify hidden cameras [Nassi et al., 2019, Liu et al., 2018]. Similarly, an approach has been presented that correlates the Wi-Fi traffic patterns of a trusted camera with Wi-Fi traffic patterns of other hidden cameras on a network to detect whether they are simultaneously observing the same space [Wu and Lagesse, 2019]. Unfortunately, these camera-specific approaches fail to generalize across modalities. For example, varying lighting conditions would be ineffective for detecting a hidden microphone or an RF sensor. In recent work, human motion was used to detect a hidden camera with coarse localization (i.e., indoors or outdoors) [Cheng et al., 2018]. We argue that human motion is an emblematic event to generalize across modalities, as the objective in revealing bugs is typically to determine if the user is being observed.

In this chapter, we propose SnoopDog, a generalized framework to detect clandestine wireless sensors monitoring a user in a private space. SnoopDog leverages the notion of causality to determine if the values of a trusted sensor cause patterns in Wi-Fi traffic stemming from other devices. In particular, SnoopDog works by having the user perturb the trusted sensor values to observe if there is a causal pattern in the Wi-Fi traffic for a different device. For instance, if a wireless camera or a motion detector is monitoring a user who is wearing an inertial measurement unit (IMU), the IMU values will indicate a causal relationship with the camera's Wi-Fi traffic. SnoopDog utilizes encoding scheme models of different wireless sensing modalities to classify the sensor type, and then cross-references packet headers with publicly available information of manufacturers to identify the specific device model. We further introduce a novel fine-grained localization approach that leverages sensor coverage techniques to locate a detected sensor. We implemented SnoopDog using a user's mobile phone for ground truth sensors and a laptop for sniffing Wi-Fi traffic patterns. In the future, we envision SnoopDog to be implemented entirely as an app on either a smartwatch or a smartphone, both of which have sufficient sensing capabilities, but currently require Wi-Fi card improvements to allow for channel hopping in monitor mode, thus making SnoopDog easily accessible to non-technical users.

SnoopDog operates in two stages. SnoopDog begins in a *passive* monitoring phase that searches for suspicious causal patterns between the wireless traffic and the user's normal activity with their smartphone or wearable device. If a device is flagged as potentially monitoring the user, an *active* phase is engaged, and the user is instructed to perform a series of specific actions to detect the sensor with high accuracy. During the active phase, localization can optionally be engaged to find the clandestine sensor. The user can either skip the background or the active phase as per their convenience.

We evaluate SnoopDog over a representative set of wireless sensors following a taxonomy of popular sensing devices that may be used for surveillance. The framework had a detection rate of 96.6% and a device classification rate of 100% when the injected multi-modal event was human motion. We show that the location of the bug can be narrowed down to a sufficiently reduced region that easily facilitates a user's search. This feature is a notable improvement over existing approaches that only localize devices as either indoors or outdoors. While SnoopDog cannot detect *any* wireless sensor monitoring the user (Section 5.11), it can detect a broad set of commonly used wireless sensors [ama, b, ama, a, Ding et al., 2011].

Contributions: Our contributions are summarized as follows:

- We propose SnoopDog, the first generalized framework to detect hidden clandestine sensors, including video, audio, motion, and RF. SnoopDog leverages the cause-effect relationship between a trusted set of sensor values and Wi-Fi traffic patterns when observing a multi-modal injected event.

- We present a novel technique that leverages the notion of directional sensor coverage to provide state-of-the-art localization for clandestine devices.
- We show how SnoopDog can reveal device information by cross-referencing packet inspection with publicly available device manufacturer information.
- We evaluate SnoopDog with a mobile phone and a Wi-Fi packet sniffer on a representative set of clandestine sensors and show a detection rate of 95.2% and device classification rate of 100% when the injected multi-modal event is human motion.

5.2 Background

We provide an overview of that state-of-the-art approaches to detecting the presence of wireless sensors in spaces. We then formalize the notion of detecting whether a sensor is monitoring a particular area.

5.2.1 Detecting Wireless Sensors in Spaces

The general approach to detecting wireless sensors relies on the notion that a device's wireless communication unintentionally leaks information in some out-of-band channel. Recent works exploited these leaks to detect the presence of wireless, transmitting bugs[1] in a space [Sathyamoorthy et al., 2014, Valeros and Garcia, 2017]. The received power threshold and frequency range can be set according to a target set of wireless devices. For instance, to detect sensors that communicate over Wi-Fi, a device would scan frequency ranges around 2.4 GHz or 5 GHz. In tuning the received power threshold, there is a direct trade-off be-

[1] A *bug* in this context refers to a hidden device spying on the user.

tween detection accuracy and false positives [Sathyamoorthy et al., 2014]. If the threshold is too low, one may falsely attribute wireless signals from other devices in the space, like mobile phones, to bugs. On the other hand, a high threshold risks ignoring wireless bugs that are not within close proximity of the detector. As these detectors provide no semantic information about the detected signals, it is difficult to assume whether or not the observed signal is truly originating from a hidden bug [Valeros and Garcia, 2017].

As wireless sensors transmit their information via packets, another technique to detect them uses packet sniffing. Approaches like DewiCam [Cheng et al., 2018] sniff wireless packets and use their characteristics to train a classifier to identify whether or not a particular device is a camera. However, even if the type of device is determined, it may or may not be monitoring the user. If there is a camera monitoring the door of a house, it does not pose the same threat to a user's privacy as a camera that is monitoring the bedroom. Hence, even if we are able to detect what type of device is present in the space, it is difficult to characterize if its intention is adversarial. A direct way to identify whether a device poses a potential privacy threat is to determine whether or not it is actively monitoring the user.

5.2.2 Detecting Sensors Monitoring a Space

If a wireless sensor is monitoring someone in a physical space, the data that it captures is a function of the person's interaction with the space. For example, if someone moves into a space monitored by a motion detector, the sensor's control mechanism may be triggered and begin uploading relevant information to the cloud to be processed and forwarded (e.g., an alert to the device owner or downstream actuation). Similarly, the information recorded by a video camera captures variations due to motion within the captured scene–a characteristic exploited by prior research on detecting hidden cameras [Nassi et al., 2019,Liu et al., 2018,Wu and Lagesse, 2019]. To generalize across sensor modalities, we formalize the notion that if an auxiliary sensor observes and measures a user's interaction with their surroundings, we can identify whether the user's actions indicate a causal relationship with the hidden sensor's wireless traffic. If such a relationship is found, then the sensor must be monitoring the user.

Detecting causality across sensor modalities. Given a target hidden sensor and access to its sensor data, we aim to establish causality between its time-series data and another sensor capturing the private space. A popular method to study causal relationships between two series is Granger Causality [Granger, 1969]. According to Granger Causality, if a series X Granger-causes series Y, then past values of X should contain information that helps predict Y above and beyond the information contained in past values of Y alone. Formally, if we have a series Y as:

$$y_t = a_0 + a_1 * y_{t-1} + a_2 * y_{t-2} + ... + a_n * y_{t-n}, \tag{5.1}$$

and we augment this series with the series X as follows:

$$y_t = a_0 + a_1 * y_{t-1} + ... + a_n * y_{t-n} + b_1 * x_{t-1} + ... + b_m * x_{t-m}, \tag{5.2}$$

then X Granger-causes Y if and only if Equation 5.2 gives a better prediction of y_t than Equation 5.1. Here, y_{t-k} are called lags of y and x_{t-k} are called lags of x where $k \in [1, n]$. However, several design challenges and goals lead up to establishing Granger Causality between a trusted sensor and a remote sensor in a generalized fashion.

In the following section, we discuss the system model and the design of SnoopDog.

5.3 SnoopDog Overview

We present the SnoopDog's threat model assumptions prior to enumerating the system design.

5.3.1 System Model

We consider a system model for SnoopDog where a user has access to a laptop or smartphone device with a network card that can enter monitor mode to sniff wireless packets

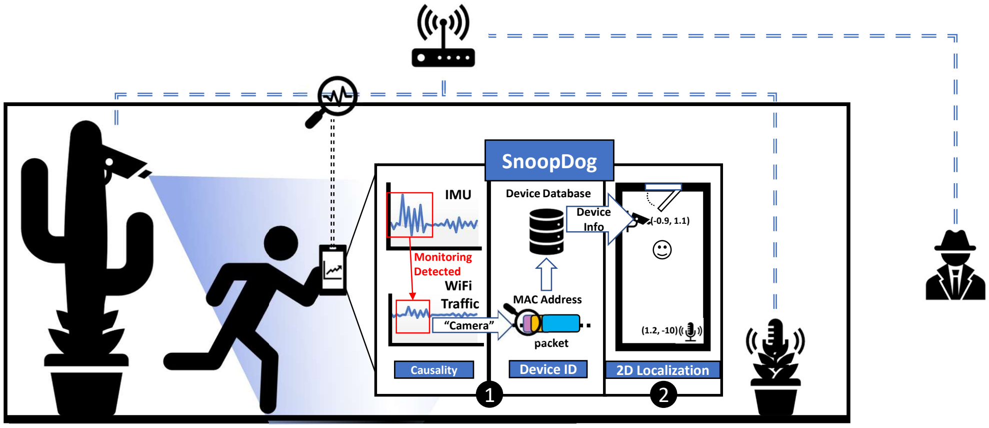


Figure 5.1: Overview of SnoopDog framework. (1) The SnoopDog framework first identifies if a user is being monitored based on the cause-effect relationship between the values of a trusted sensor, e.g., an IMU, and Wi-Fi traffic patterns. It then inspects the associated packets and identifies the possible devices based on the physical (MAC) address. (2) Finally, SnoopDog localizes each device by leveraging directionality and sensor coverage.

over the same channel as one or more clandestine sensors. The system should further be equipped with a trusted set of *ground truth* sensors to establish causality between the sensor values and the associated Wi-Fi patterns from the clandestine wireless sensor(s)[2]. These capabilities entail a set of assumptions.

Wi-Fi sniffing assumptions. We assume that the Wi-Fi sniffer on the user's device can monitor the encrypted traffic streaming from the clandestine device. SnoopDog does not require any form of granted access to a particular network, i.e., SnoopDog should be able to sniff the device regardless of whether or not the network is closed or hidden. Unlike previous solutions, this implies that the user does not need to know the SSID or password of the network.

Causality assumptions. We assume that the user has a sufficient set of trusted ground

[2] We assume there may be additional, non-clandestine sensors that are monitoring the user. Such superfluous information is still informative, as the goal of this work is to detect all wireless sensors monitoring a user.

truth sensors whose modalities are sensing any of the user's activities that would exhibit a causality with the Wi-Fi encoding patterns of any clandestine wireless sensors. The notion of sufficient causality was formalized in Section 5.2.

5.3.2 Adversary Model

We focus on adversaries whose goal is to remotely spy on a third-party occupant of a private space in real-time. This model is consistent with other state-of-the-art methods for detecting hidden cameras [Cheng et al., 2018, Wampler et al., 2015, Nassi et al., 2019, Liu et al., 2018], and is supported anecdotally by several cases where owners were live-streaming guests in private spaces, e.g., [Fussell, 2019, Jeong and Griffiths, 2019]. Further, many commercially available devices do not offer a local storage option for reasons of size, weight, power, and cost – such is the case with six out of the popular thirteen devices we examined. Moreover, live-streaming offers a more practical and scalable solution from a management perspective. Thus, we assume the adversary uses an arbitrary set of wireless, commercial-off-the-shelf (COTS) sensors that are tailored for clandestine placement. The communication between the attacker and sensor may be encrypted and placed on an arbitrary wireless frequency band. We further assume the adversary has deployed these clandestine sensors in a manner that is not apparently visible to the user within the space. We focus on an attacker utilizing devices that communicate over Wi-Fi, as this is the most prevalent method of wireless communication for remote monitoring using commercial and consumer equipment[3]. An adversary may use one of the several techniques mentioned in Section 5.10 to fool SnoopDog, for example with cover traffic or local storage. Implementing these techniques can require modifying the device firmware or physically interfacing with a proxy device (e.g., RPi), thereby increasing the barrier-to-entry for potential attackers. Moreover, techniques such as cover traffic can add significant and undesirable network overhead, particularly for a large number of sensors.

[3] Although SnoopDog focuses on Wi-Fi-connected devices, we discuss in Section 5.11 how such a system could be generalized to other wireless communication standards and protocols.

5.3.3 Design Overview

As depicted in Figure 5.1, SnoopDog detects and localizes a wireless sensor given access to a trusted sensor that can measure and quantify the ground truth in the modality that we are trying to detect. SnoopDog works in two phases. ❶ **Detecting and identifying snooping wireless sensors.** When a user first enters a new space, SnoopDog operates in a background mode to determine whether a user is being monitored based on the cause-effect relationship between the values of a trusted sensor (e.g., an on-body IMU) and Wi-Fi traffic patterns. If the user wants to scan a room immediately, the background phase may be optionally skipped; alternatively, the background phase offers a low-overhead solution to bug detection. If a clandestine sensor is discovered, SnoopDog asks the user to perform a unique perturbation in the space to further ascertain the presence of a snooping sensor. The associated packets are then inspected to identify the possible device type based on the physical (MAC) address. ❷ **Snooping sensor localization.** In the second phase, SnoopDog utilizes a trial-based localization technique to identify the specific placement of the monitoring device. With the appropriate selection of ground truth sensor, that is, a device which can semantically capture at least a subset of the events captured by the snooping device, SnoopDog can detect clandestine wireless sensors of arbitrary modality.

5.4 Detecting and Identifying Snooping Wireless Sensors

This section outlines the ability of SnoopDog to detect whether a clandestine sensor is actively snooping on a user. We describe the search space for wireless sensors, how to establish causality, how to generalize across modalities, and how to understand various sensors' wireless transmission.

5.4.1 Searching for Wireless Sensors

The adversary can create a Wi-Fi network and connect the snooping device to it. As a result, the hidden device can be present in any of the possible Wi-Fi channels. Even though SnoopDog does not need access to these networks, it still needs to scan all Wi-Fi frequencies and look for any devices transmitting on them. 2.4 GHz and 5 GHz are the most popular bands for Wi-Fi networks, and as such, we focus on those particular bands, even though the SnoopDog scan region can be easily extended to include other ranges. During discovery, the Wi-Fi Network Interface Card (NIC) scans through all channels sequentially to find available access points (APs) [Wu et al., 2009, Hu et al., 2015]. Similarly, SnoopDog also scans through all the Wi-Fi channels in monitor mode, but instead of looking for available APs, it looks for transmissions in those channels and creates a list of devices using the MAC address present in packet headers. As a result, SnoopDog does not need to be connected to any specific AP to operate. Even if a network is hidden, its transmissions can still be observed by monitoring the Wi-Fi channel. Thus SnoopDog can detect devices on any Wi-Fi network. Because devices may transmit data intermittently, SnoopDog continuously scans all Wi-Fi channels and actively maintains an aggregate set of traffic data. Once the list of devices has been populated, SnoopDog then seeks to detect causality between user activity and data being transmitted from each device.

5.4.2 Detecting Causality with User Activity

Detecting the cause-effect relationship between the action of a user in a space and the data captured by a clandestine, wireless sensor requires access to two essential components: 1) a ground truth sensor to capture information about the user in the space and 2) a representation of the data collected by the clandestine sensor. While data packets transmitted by wireless sensors may be encrypted, the header information is not. This header information provides us with the MAC address and payload size of each transmitted packet. This data can be grouped and aggregated for all the packets within a time window and provide information as to how much data was transmitted by each device within that period. Given a ground truth sensor, one can then identify causality between the ground truth sensor values and the patterns in the volume of data transmitted by each device in the space. In contrast to machine learning techniques, a causality approach allows SnoopDog to find the cause-effect relationship of arbitrary modality across any device that is transmitting causal data. Because we are interested in the causality between two sensors, SnoopDog will utilize Granger Causality (described in Section 5.2).

5.4.3 Characterizing a Representative Set of Snooping Sensors

In order to choose a set of ground truth sensors that can capture causality across any modality, we focus on generalizing across a representative set, including cameras, RF, and arbitrary sensors that report inferred (as opposed to raw) events.

Visual sensors. Wireless cameras are typically encoded with a codec that recognizes underlying patterns in the frames of the video and utilizes this information for compression. One such codec is H.264 [Wenger, 2003]. An encoder first encodes the video using the standard, and a decoder then reconstructs the original video with minor information loss.

Standard temporal compression algorithms compress the video with 3 key frame-types, denoted I, P, and B frames. I frames (Intra-coded picture) hold complete image information, whereas P and B frames contain fractional image information, i.e., scene differences. As I frames are a complete image, they do not require any other frames to be decoded. P frames (Predicted picture) only contain changes in the image from previous frames. The information in a P frame is combined with the information of the I frame preceding it to obtain the resulting image. B (Bi-directionally predicted pictures) frames can construct the image from either direction using either changes from the I or P frames before them, changes from I and P frames after them, or interpolation between the I/P frames before and after them. B frames are most compressible, followed by P frames, and finally, I frames.

Hence, with increasing motion in the scene recorded by an IP camera, there will be an increase in the data that must be transmitted due to the increase in the number of P and B frames sent. Camera traffic will increase as the number of pixels being perturbed in the scene increases; similarly, traffic will decrease if the scene transitions to a stationary one. As such, if a human subject were to perform some motion in the scene, stop for enough time to let the camera traffic settle down, and then move again, it will result in a unique camera traffic pattern that corresponds to the user's motion. This cause-effect relationship between human motion and camera traffic can then be used to discover if a wireless IP camera is present in an occupied space. If there is no relationship between the camera traffic and user motion, then the camera is not monitoring the user.

RF sensors. Low cost, off-the-shelf millimeter-wave (mmWave) RF sensors are available that record the scene in the form of point-clouds. Recent works [Singh et al., 2019, Zhang and Cao, 2018b] have shown that these point clouds can be used to infer human activity. However, unlike a camera, a radar device is a point scatterer. Thus, at any given time, only certain points in the scene reflect back. Hence, with motion in the scene, the number of points captured in every frame by the sensor (radar) vary considerably. In an empty scene, the number of points captured by these sensors is fairly constant but varies as subjects move about the space. If such a sensor live-streams point-cloud data over Wi-Fi, the payload size will vary over time with changes in the number of points captured in the scene by the sensor. Hence, the network traffic will fluctuate with the number of points that are being captured in the frame. As such, there exists a cause-effect relationship between the subject's motion and the device's traffic.

Acoustic sensors. Another common type of bug used to snoop on people is a microphone. With the growth in personal home assistant devices such as the Google Home or Amazon Echo (Alexa) [Kepuska and Bohouta, 2018], it is trivial for someone to buy and install such listening devices in their homes. Although they are typically triggered by a keyphrase such as "Okay Google" or "Alexa", there are "Drop In" features that facilitate remote snooping. An adversary can also change the wake word of these devices to enable recording conversations of interest. Due to their compact form factor, they can be easily hidden. In such cases, these devices will also work like event-based clandestine sensors. Hence, services like SnoopDog that monitor traffic for change in network patterns and either correlate them with another sensor recording of the same modality or find a cause-effect relationship with the ground-truth can detect their presence using network sniffing [Kennedy et al., 2019, Wright et al., 2008]. Here, instead of the IMU, we use the microphone on the user's smartphone as the trusted ground-truth sensor. In section 5.12-Q4, we discuss why it is challenging to detect and localize acoustic sensors that are continuously streaming.

Wireless sensors that encode inferred events. Motion sensors do not transmit a continuous stream of information. Most off-the-shelf motion sensors are passive infrared (PIR) based. They measure the infrared (IR) light from objects in their field of view. Any change in this incoming IR light is inferred as motion. Instead of continuously transmitting, they send data to their cloud service for processing once triggered by motion. Thus, if a user moves around the room, stops, and moves again, there will be a unique cause-effect relationship between user motion and device traffic. Additionally, a camera can be programmed to continuously record video but only upload when a certain event occurs in the scene. These cameras behave like motion sensors and hence can be treated similarly. Virtual assistants also wait for trigger words to transmit a request to the associated cloud service, e.g., a user uttering the device name to activate it [Kepuska and Bohouta, 2018].

5.4.4 Device Identification via MAC Address

A MAC address is a universally unique ID assigned to the Network Interface Controller (NIC) for every networked device. It consists of 48 bits which are typically represented as 12 hexadecimal characters, i.e., `xx:xx:xx:xx:xx:xx`. The first 24 bits are the OUI (Organizationally Unique Identifier), which can uniquely identify a manufacturer or a vendor.

The MAC address of the sender and the receiver are contained within each exchanged Wi-

Fi packet. More importantly, this information is not encrypted. As a result, SnoopDog can obtain the MAC address to look up the device vendor. While we acknowledge that the MAC address can be spoofed, this technique can still prove useful in the many cases where the adversary is a non-expert and thus has not spoofed the MAC address. Traffic fingerprinting techniques [Gao et al., 2010, Crotti et al., 2007, Apthorpe et al., 2018, Zuo et al., 2019, Ortiz et al., 2019, Meidan et al., 2017, Miettinen et al., 2017] can also be used to overcome the shortcomings of MAC-based identification. Additionally, in case of MAC randomization or MAC spoofing, techniques such as the ones described in [Vanhoef et al., 2016] can be used to first track the traffic from a particular device and then perform cause-effect analysis on it.

SnoopDog contains a database with names and MAC addresses of known vendors that manufacture surveillance devices. As SnoopDog detects more sensors, we add them to the database.

5.5 Snooping Sensor Localization

Algorithm 1: Localize identifies the location of a particular snooping sensor in a defined region-of-interest

```
Input: The sensor's $MAC$ address
       The $region$ of interest
Output: The sensor's location within the region
$BBox \leftarrow \emptyset$
$traversing \leftarrow$ BeginTraversingRegion($region$)
while $traversing$ do
    $userloc \leftarrow$ DeadReckoningLocation()
    $inView \leftarrow$ GrangerCausality($MAC$)
    if $inView$ then
        $BBox \leftarrow BBox \cup \{userloc\}$
    $traversing \leftarrow$ SparseBBox($BBox$)
Loop
    $MLE \leftarrow$ MostLikelySensorLocation($region, BBox$)
    if SufficientBBox(region, BBox) then
        return $(BBox, MLE)$
    $trialRegion =$ GenerateTrial($MLE, BBox$)
    $inView =$ PerformTrial($trialRegion$)
    if $inView$ then
        $BBox \leftarrow trialRegion$
    else
        $BBox \leftarrow BBox \setminus trialRegion$
```

Algorithm 1 details the **trial-based localization** used by SnoopDog to infer sensor location. In the case of multiple active sensors, this process can be repeated for each device.

Setup. Localization requires two input parameters: a region-of-interest to search over, and the snooping sensor's MAC address. To define the region-of-interest, we leverage Dead Reckoning [Patel, ,Levi and Judd, 1996,Beauregard and Haas, 2006] for indoor user localization. A dead reckoning mobile application [Patel,] on a user's phone instructs the user to walk the perimeter and capture the region boundary. Aside from identifying Granger causality in traffic patterns, the MAC address is also used to ensure an appropriate trial method for localization (e.g., via techniques discussed in Section 5.4.4 and [Huang et al., 2019a]).

5.5.1 Identifying Sensor Coverage

Although the malicious sensor is known to monitor somewhere within the region-of-interest, it is unlikely to cover the entire region. Lines (1)-(8) narrow down the full search space into a bounding box *BBox* of the sensor's field-of-view. To begin, a user is instructed to traverse the region (line 2). At regular time intervals, the user's location is captured, and the snooping sensor's traffic is monitored for causality. Using the Granger Causality technique described in Section 5.4, a particular location is identified as either within or outside sensor coverage. This process continues until the bounding box is determined to have sufficient density for performing trial-based localization, depending on the coverage area size.

The remainder of Algorithm 1 (lines 9-18) reduces the *BBox* scope of sensor coverage via directional elimination. Repeated trials are performed to specifically target high-probability origins in order to either identify or eliminate likely sensor locations. Each round begins by solving for the most likely origin *MLE* for the sensor (line 10). While this process could be performed randomly, utilizing physical information about the current bounding box can significantly reduce the number of necessary trial rounds. For example, if the bounding box shape can be reasonably fitted to a triangle, then the sensor is likely horizontal-facing and placed on a wall. On the other hand, an ellipsoid coverage area likely indicates a sensor placed on the ceiling or floor.

An iterative process then proceeds to reduce the area of possible sensor locations to a pre-defined threshold (e.g., 10% of the region), upon which the bounding box and MLE are returned (line 11). In each iteration, a *directional* trial is conducted. **GenerateTrial** identifies a suitable position and heading for the trial by selecting a point near the center of the bounding box and facing the MPE (line 12). In our evaluation, we found distances of approximately 3 meters to be the maximum applicable distance for a trial. The trial takes one of many forms; for an inertial sensor, a user faces the designated direction and waves an object (e.g., hand or shoe) closely in front of their chest while shielding this activity with their body from any sensor present behind them. To trigger a camera sensor, a laptop plays a video clip that randomly flashes the screen with different colors. For audio, a trigger sound is played, and so on. If the trial results increased the device traffic, the bounding box is reduced to areas within visible range (line 16); otherwise, those areas are removed (line 18), and the next iteration begins.

5.5.2 Ensuring Sufficiently Reduced Region

In order to provide a guarantee that this localization method will always result in a minimal bounding box that is sufficiently small (e.g., 10% of the search region), a key assumption must be made: for any arbitrary bounding box, a trial can be identified which will eliminate a proper subset of the bounding box. In the case of Algorithm 1, this assumption can be reformed such that one can always construct a trial that eliminates at least a *single* point contained within the bounding box set. Due to the directional nature of each trial, this can be achieved simply by conducting a trial that is positioned directly between two points within the bounding box, and facing directly towards one of the two points such that the other is obstructed. In the case of two points with large intermediate distances, a two-phase trial must be performed facing towards (and away from) each point, respectively.

Given the assumption that every trial can eliminate at least a single point from the bounding box set, guaranteeing that Algorithm 1 will always reduce the region to a certain size is trivial. In the worst case, for a bounding box of *n* points, *n-1* trials must be performed. In practice, each trial can eliminate many points contained within the bounding box. Furthermore, by leveraging the most likely sensor location, one can reduce the search space significantly and with relatively few trials.

5.6 Implementation

This section presents the implementation details of SnoopDog. We use readily available tools that are likely to be in a user's possession.

5.6.1 Experimental Setup

Wi-Fi Packet Sniffing: The laptop's (Lenovo Thinkpad) network card enters monitor mode and uses Wireshark to capture all transmitted packets in the Wi-Fi frequency band to aggregate traffic statistics for analysis. As it is not necessary to connect to a specific Wi-Fi network to monitor traffic, SnoopDog can capture and identify clandestine wireless sensors across all Wi-Fi traffic, even if they reside on a closed or hidden network. A smartphone can also be used instead of a laptop, but requires a rooted [Sun et al., 2015] phone.

Collecting User's Motion Data: User's motion data is collected via the IMU present on the smartphone (Google Pixel 3). The smartphone is placed either in the user's hand or inside the user's pocket. 50 Hz accelerometer data is collected and used to study the cause-effect relationship between motion and sensor traffic. We collect data along each of the 3 axes and use them separately as if motion is present in only one direction, the other 2 axes contribute minimally to the analysis, and may instead serve as noise. The smartphone is also used to collect audio and localize the user in his/her surroundings.

5.6.2 Detecting the Cause-Effect Relationship between User Motion and Hidden Devices

While sniffing the network, SnoopDog classifies the networked devices present into two categories: devices that transmit data continuously, and devices that have periodic or event-based transmission.

5.6.2.1 Wireless Sensors that Encode Raw Data

Some representative sensors that continuously transmit variably encoded raw data include camera and RF sensors.

Camera: When a camera is monitoring a static scene, its traffic is fairly constant, as shown in Figure 5.2. As the scene is perturbed by human motion, the traffic changes rapidly. However, it is yet unclear whether human motion causes this variation. As soon as the user enters a new space, he or she can turn on SnoopDog, which works in the background to correlate IMU data with Wi-Fi traffic of the transmitting devices. As users walk in a space, the starting and stopping patterns of their motion are unique. This unique pattern creates a fingerprint on the camera traffic. Once SnoopDog is able to determine a cause-effect relationship between device traffic and user's motion, it alerts the user. To definitively ascertain the presence of a camera, SnoopDog asks the user to perform a stop-start-stop-start-stop (**S5**) motion as follows: 1) the user stays stationary for some time to allow the device traffic to stabilize. 2) The user performs jumping jacks at the current position. 3) The user stops again and waits for the device traffic to settle. 4) The user performs jumping jacks. 5) The user stops. The S5 motion causes a unique pattern to appear in the Wi-Fi traffic as shown in Figure 5.3 (Cam. 2).

The entire detection phase requires $35 - 45$ seconds. While the user is performing the above **S5** motion, SnoopDog sniffs the Wi-Fi packets on the network and records the user's IMU acceleration. Figure 5.3 plots the camera traffic after I-frame suppression and user accelerometer data while performing the **S5** motion. We observe that camera traffic is a function of human motion. When the human is static, the traffic is small, but when the human begins performing jumping jacks, the traffic rate increases. To prove that the accelerometer series indeed has an effect on the camera traffic, we leverage Granger Causality using the `statsmodel` package in Python. The null hypothesis of the Granger Causality Test is that the IMU series does not granger-causes the camera traffic series. Hence, if the p-value of our test is below the threshold of 0.08, we can reject the null hypothesis and claim that the IMU series granger-causes the camera traffic series. We selected this p-value using the results obtained from the first camera. However, we evaluate our detection for all the other cameras and show that this p-value threshold is optimal for all the cameras.

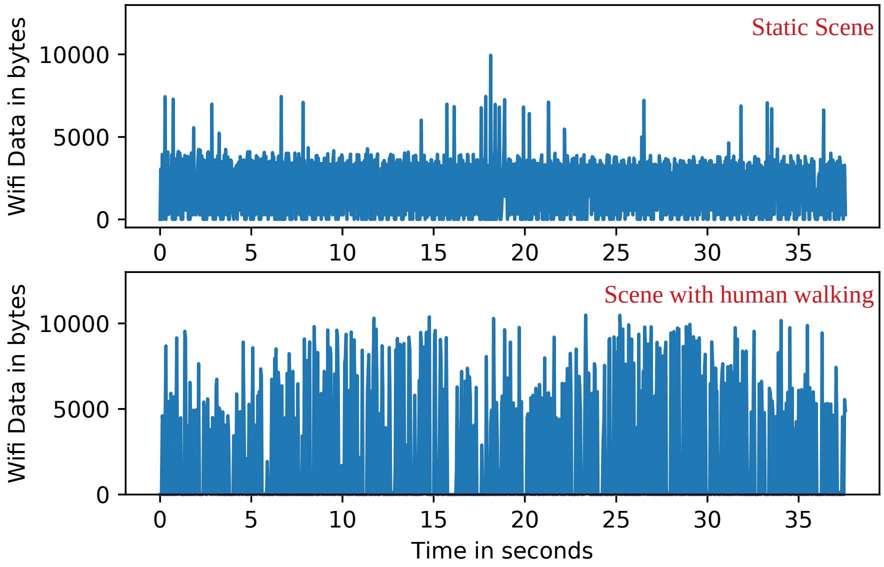


Figure 5.2: Wi-Fi traffic captured from a camera for a static scene and a scene where a human is walking around.

RF sensor: The detection process remains the same for RF as that of a camera. We use an off-the-shelf mmWave RF sensor from Texas Instruments, as shown in [Singh et al., 2019]. We model the information obtained from the sensor as Wi-Fi traffic. The modeled Wi-Fi traffic from the RF sensor due to human motion is shown in Figure 5.4. Unlike a camera, RF sensors respond to a change in RF reflections from the scene.

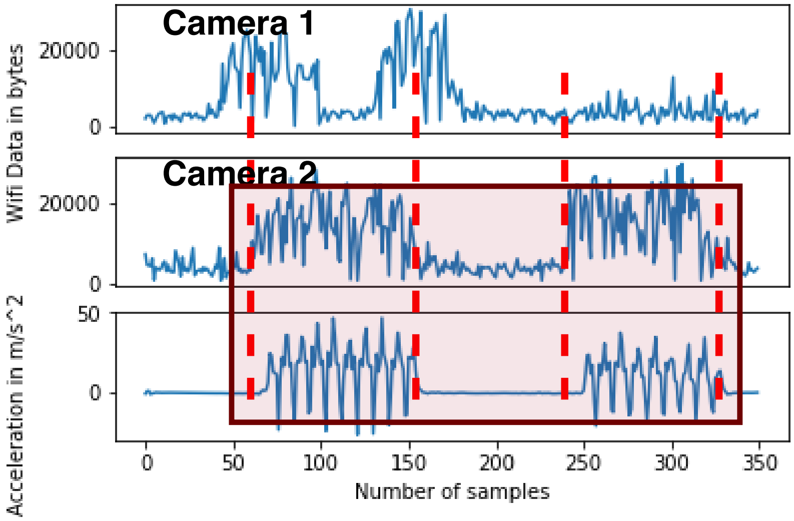


Figure 5.3: Wi-Fi traffic of a snooping camera placed in the same space as the user (Cam. 2) and a non-snooping camera placed in a different space (Cam. 1) and its comparison with IMU data of the user being monitored in the scene.

As soon as motion occurs within the space, the traffic changes rapidly in response. This is because the points captured by the RF sensor vary with motion. If the traffic of some device which was static when there was no motion but changes rapidly when there is motion and goes back to being static when motion stops, it is an indicator that the device is monitoring user movement. To detect such devices, SnoopDog first monitors the traffic when the scene is static. It then asks the user to perform the **S5** motion in the space while SnoopDog monitors the traffic. As soon as the user is finished, the user should leave the space so that SnoopDog can monitor the traffic again and conclude the presence or absence of an RF sensor.

5.6.2.2 Wireless Sensors that Encode Inferred Events

Sensors that encode inferred events transmit upon event detection. By examining network traffic, it is difficult to ascertain if the device is transmitting periodic data, like a temperature sensor, or transmitting inferred events like a motion sensor.

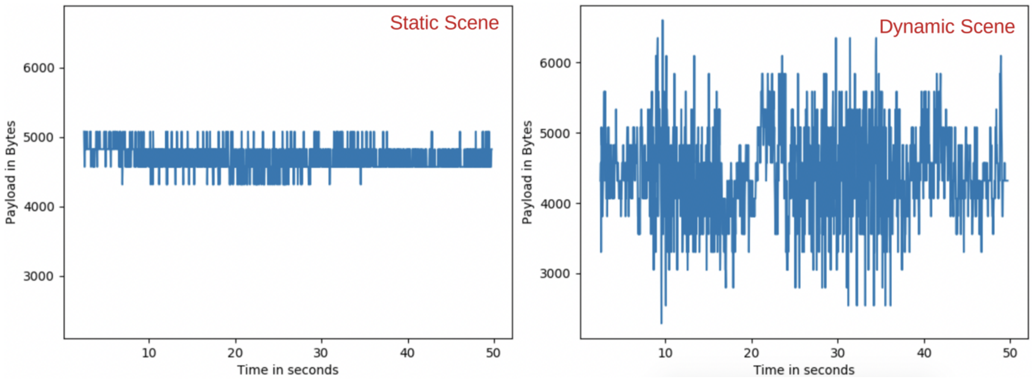
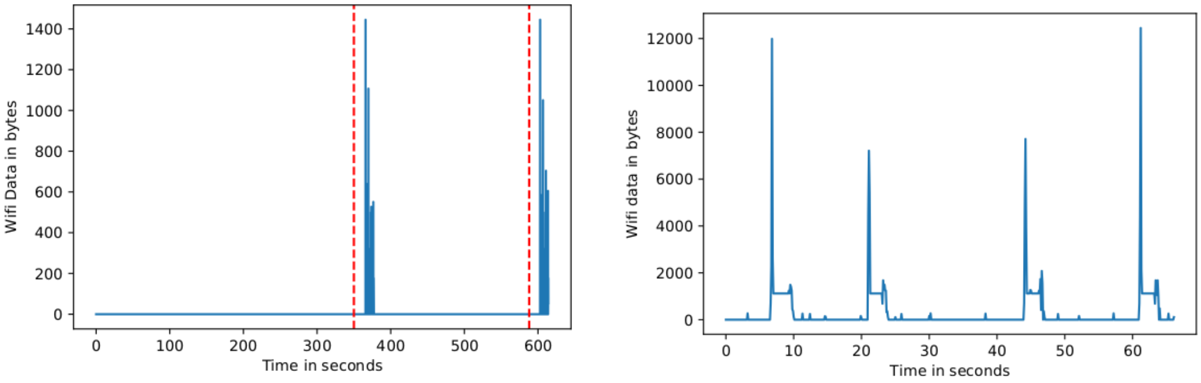


Figure 5.4: Modeled Wi-Fi traffic for an RF sensor in a static scene and one where a user performs our detection trial.

Motion Sensor: Typical off-the-shelf motion sensors have a timeout to prevent continuous alerts. The motion sensor sends motion information to a cloud server, which in turn sends an alert to the snooping user's smartphone or performs an action like turning on lights. After sending an alert, the sensor waits for the timeout period before it looks for more events. This period is between 30 seconds and 3 minutes for most motion sensors. Similarly, there can be other sensors in the scene that have a timeout period between uploading events. To discover a device's timeout period, SnoopDog correlates user movements with device traffic. If two events are detected in the traffic of a device and the user was in motion during the time between the two events, this time is noted as the timeout period. SnoopDog uses its active phase to further improve the timeout estimation by asking the user to move around the space until two events are detected in the device's network traffic. SnoopDog asks the user to move around the space, leave the space for the timeout period, and then move around the space again. After that, the user moves out from the space and then waits for the timeout period to end. If SnoopDog detects traffic by the device around the same time the user moved and none when the user is not moving, it concludes that the traffic of the device is caused by user movement. This process can be repeated to increase the confidence of detection. In Figure 5.5, we move around the room and notice that the Wi-Fi



(a) (b)

Figure 5.5: (a) Wi-Fi traffic of a motion sensor. The red-dotted line represents a motion event. (b) Wi-Fi traffic of an Alexa device for the user repeating the same phrase 4 times.

traffic from the motion sensor responds to these motion events. Since this traffic is discrete, we cannot perform time-series Granger causality analysis. Instead, we perform an activity and track network response. To detect the presence of a motion sensor, we ask the user to move around the room, wait for the timeout period, and move around again. SnoopDog scans all device traffic within a period of 5 seconds after the motion to determine which device responds to user motion. If the device has traffic activity after the user moved, then the device is inferring events from the user motion.

Audio snooping: SnoopDog records user conversations in the background and monitors the network traffic. If the occurrence of a certain phrase or a word causes the traffic of a device to change, SnoopDog asks the user to repeat it until it can establish a causality between the occurrence of that phrase and the traffic of the device. Once SnoopDog knows the "wake word" for the acoustic home-assistant device, it repeats the recording several times while monitoring the device traffic to increase the confidence level of detection.

In our implementation, we used an Amazon Echo and Echo Dot whose wake word was "Alexa" and "Computer" and a Google Home Mini with the wake phrase "Hey Google". In Figure 5.5, we say the phrase "Alexa, what's the time right now?" four times and observe four distinct events in the device traffic. In 35 trials with different phrases, SnoopDog was able to detect causality 100% of the time. Additionally, we show the variations in device traffic for Alexa Drop-In features when the Drop-In is active and when it is not active in Section 5.9.

5.6.2.3 Device ID via MAC Address Lookup

SnoopDog checks its database for a match of OUI in the device's MAC address. If present, SnoopDog can inform the user with higher confidence that the device is indeed a surveillance device. Otherwise, it is added to the database and identified as a clandestine sensor.

5.6.3 Device Localization

SnoopDog uses dead reckoning [Ojeda and Borenstein, 2007] and asks the user to walk around the perimeter of the room to create rough map of the room. Next, the user performs a detection trial at various locations in the room. More trials lead to better localization. At every location, SnoopDog tries to establish a cause-effect relationship with the device traffic. Regions with no cause-effect relationship are eliminated.

IP Camera: The traffic generated by a camera monitoring a scene will increase when the scene is dynamic. To exploit this, we first monitor the traffic of the device identified as a camera for 30 seconds over a static scene. Each trial consists of standing in a particular location (e.g., the middle of the scene), pointing a laptop in a particular direction, and playing a video that rapidly changes the colors on the screen of the laptop for 30 seconds. This process is then repeated in different directions. If the camera is able to monitor the laptop screen, its data rate during that period will be higher. On the other hand, if the laptop screen is not visible, the camera's traffic rate will be similar to the static scene. We can eliminate a fraction of the space where no activity is detected and repeat the process for the remaining region. In this way, we narrow down the possible region where a camera is located. We give a step by step walk-through of this process in section 5.7.

RF sensor: RF sensor localization is similar to that of a camera. However, since RF sensors cannot detect the flickering screen of the laptop, we use human movement. SnoopDog asks the user to stand in the middle of the space and wave their arm up and down rapidly in front of them while shielding this motion from the other side of the space with their back. If the RF device traffic does not respond to these stimuli when performed on one side but responds to it on the other side, we can eliminate that space.

Motion Sensor: Motion sensors are triggered by motion in front of them. SnoopDog first identifies the motion detector timeout (refer section 5.6.2.2), and then asks the user to stand in the middle of the room before the timeout expires. After timeout expiry, they are asked to move their hand in front of them while shielding it from the other side with their body.

Acoustic (Audio) sensors: SnoopDog records the wake word of the device and asks the user to move around the room while this sound is repeatedly played from the smartphone app. If the user walks around the room but does not find any place where there the traffic of the device changes, we increase the volume and repeat the experiment. On the other hand, if the sound played at every point in the room causes the traffic of the device to vary, we decrease the volume and repeat the experiment. Finally, we identify areas where the sound causes network response and areas where it does not. We continue to reduce the volume of the device until the search space has been sufficiently reduced[4].

5.7 Evaluation

We evaluated SnoopDog on a set of sensors from well-known brands as well as best-selling sensors on Amazon. These are listed below in Table 5.1.

[4] A walk-through of this process is provided in Section 5.8

Name	**Type**	**Cost**
Kamtron	Camera	$39.99
Panasonic (HomeHawk)	Camera	$77.64
Wansview	Camera	$29.99
Arlo (NetGear)	Camera	$107.50
Victure	Camera	$35.99
Foscam	Camera	$49.99
Ring (Amazon)	Camera	$59.99
Amazon Echo Dot	Home Assistant	$29.99
Amazon Echo	Home Assistant	$99.99
Google Home Mini	Home Assistant	$39.99
Kangaroo Home	Motion Sensor	$12.95
Samsung Smart Things	Motion Sensor	$24.99
TI IWR1443	RF Sensor	$299.99

Table 5.1: List of snooping sensors evaluated

5.7.1 Sensors that Encode Raw Data

Wireless IP Cameras. For Granger causality analysis, we lag the first series by one element at a time and observe what value of the lag results in the lowest p-value. Cameras have a delay between when the scene changes and when the data is visible to the adversary. We found that this delay can vary between a few milliseconds to up to 4 seconds. If the adversary is using a tape delay in transmission, we can perform this analysis over a longer delay period. Assuming symmetrical delay, SnoopDog sniffs the packets during the first half of the transmission; we choose a lag value of 2 seconds.

We evaluated our detection on 7 cameras. All of them use H.264/MPEG-4 codecs which are the most popular codecs used for IP cameras. We performed 131 trials on 2 different users[5] to evaluate the detection accuracy. The results of our experiments are presented in table 5.2. To improve the detection accuracy and confidence of detection, a user can perform the detection trial several times and take a majority vote. The detection works well even when a portion of the human body is occluded by objects such as a table.

Camera	**Trials**	**Successful**	**Accuracy**
Panasonic	15	14	93.33%
Arlo (Netgear)	10	10	100%
Ring (Amazon)	10	9	90%
Foscam	15	15	100%
Wansview	30	29	96.6%
Kamtron	25	21	84%
Victure	26	26	100%
Total	**131**	**124**	**94.65%**

Table 5.2: Evaluation results for camera detection

RF sensors. We use a TI mmWave IWR1443 to evaluate the performance of SnoopDog. In 20 experiments, SnoopDog was able to detect RF sensor's presence every time.

5.7.2 Sensors Encoding Inferred Events

Motion Sensors. We evaluated on an off-the-shelf motion sensor from Kangaroo Security and a smart-things motion sensor from Samsung. The smart-things sensors are a special case as these sensors use Z-Wave and ZigBee to communicate with a smart-things hub which in

[5] The data is collected from the authors and hence does not require IRB approval.

turn sends the information over Wi-Fi. As a result, SnoopDog can sniff the traffic of this hub and establish causality. However, if there are multiple devices connected to the same hub, SnoopDog will not be able to detect them. We performed 25 trials, and SnoopDog was able to detect the motion sensors every time except for 3 trials. We suspect that this was caused because the devices send some sort of "status" messages to their respective cloud service which result in events in the sniffed traffic that throw the detection off.

Smart-home Assistants (Audio Sensors). In Figure 5.5, we say the phrase "Alexa, what's the time right now?" four times and observe four distinct events in the device traffic. In 35 trials with different phrases, SnoopDog was able to detect causality 100% of the time. Additionally, we show the variations in device traffic for Alexa Drop-In features when the Drop-In is active and when it is not active in Section 5.9.

5.7.3 Quantifying False Positives and Negatives

Overall false positive rate. We perform false positive analysis for two separate cases. In case one, we run cause-effect analysis for every device whose traffic has been captured (a network can have devices such as phones, laptops, and other sensors in addition to snooping devices) and find out if a device is falsely classified to be snooping on the user. In case two, we perform our cause-effect analysis on a camera that is not looking at the scene to see whether it will produce a false positive.

For case one, our experimental test-bed consists of devices monitoring the user such as a camera, motion sensor, home assistant and RF-sensor. We also include innocuous devices usually present in a scene such as smartphones, laptops, Wi-Fi-routers, smart-lights, and embedded devices such as raspberry-pi. We ran this analysis for 50 trials. We found 2 false positives for Active Phase and 9 false positives for Background Phase.

We divide case 2 into two separate parts. In the first part, a camera is placed in another room (where the user who's running SnoopDog is not present). Another person is moving freely in the room. In the second part, the person in the other room is sitting on a desk and working while the camera is monitoring them. We show the results of our experiments in Table 5.3. We perform these trial for both background detection and active detection. The one detection for background phase is because the walking patterns of the user and the person in the other room were similar. In such cases, a user can turn off the light in their room and see if the traffic of the device drops. If it does, it means the camera is in their room.

Case	Total Trials	False Positives	
		Background	Active (S5)
(1) User moving	15	1 (6.66%)	0 (0.00%)
(2) User working on a desk	10	0 (0.00%)	0 (0.00%)

Table 5.3: False-positives for a camera in a separate room

False negatives for background detection. SnoopDog's background detection mechanism is most susceptible to false negatives since it relies on natural human motion rather than active command and response patterns. Figure 5.6 shows camera traffic and IMU data for a user walking randomly in a room. As long as the user stops at some point within the detection period, SnoopDog is able to detect hidden devices. However, if the user is continuously stationary or continuously in motion, SnoopDog fails to detect the presence of clandestine sensors.

5.7.4 Snooping Sensor Localization

We evaluated SnoopDog for 4 different spaces with different sensor placements. The accuracy of localization in all of these cases depends on the user's requirements. The user can perform more trials to reduce the probable region where the sensor is placed. We use an example to demonstrate how the SnoopDog localization algorithm works. To perform our

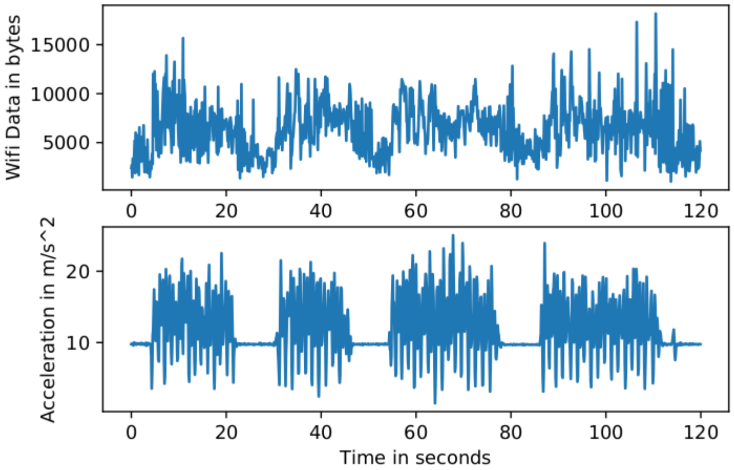


Figure 5.6: Camera traffic and its comparison with IMU data

localization, we chose a room as shown in Figure 5.7. The camera is placed at a corner of the room. We begin by performing our **S5** detection trials in different parts of the room. The location and results of our trials are shown. Based on these observations, we know that the camera is present somewhere in the square region of the room and hence, we eliminate the other part and start our trial-based localization.

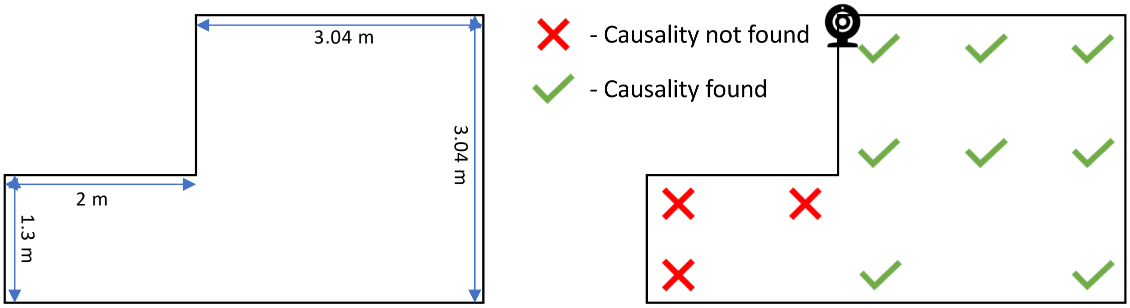


Figure 5.7: Lab dimensions and results of the detection trials.

We stand in the middle of the probable space and hold a laptop such that the screen is pointing in one direction. Then we turn to the other side and repeat the same experiment. We observe that there is a significant (>150%) increase in the camera data rate when the laptop is pointed towards the left side. When pointed to the right, the data rate remains similar to that of an empty room. Thus we eliminate the right portion of the room from the probable area. We again stand in the middle of the leftover space and repeat the experiments until we achieve a sufficiently reduced space.

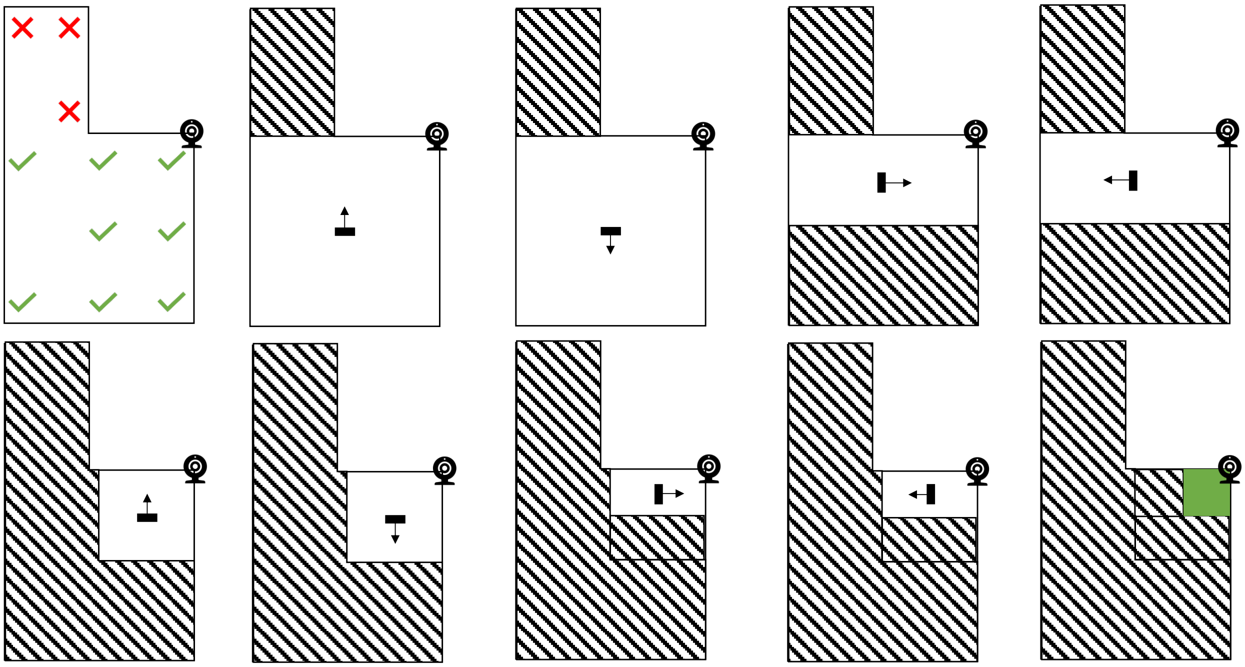


Figure 5.8: A walk-through of the trial-based localization algorithm in the laboratory environment in Figure 5.7. The arrows represent the direction the laptop screen was facing.

5.8 Audio-based Localization for Personal Home Assistants

In this section, we describe the audio localization technique step-by-step. First, we place the source of the sound (smartphone playing a phrase containing the wake word of the device) at different points in the room and see how it affects the device traffic. Then we go around the room while SnoopDog repeats that sound continuously and checks them for causality with device traffic as shown in Figure 5.9. Sound played at the points marked as green produces cause-effect relationship with the device traffic. We eliminate the regions where we detect no causality. Next, we reduce the volume by 1 level and repeat our experiment in the left-over space till we are left with a region of desirable size.

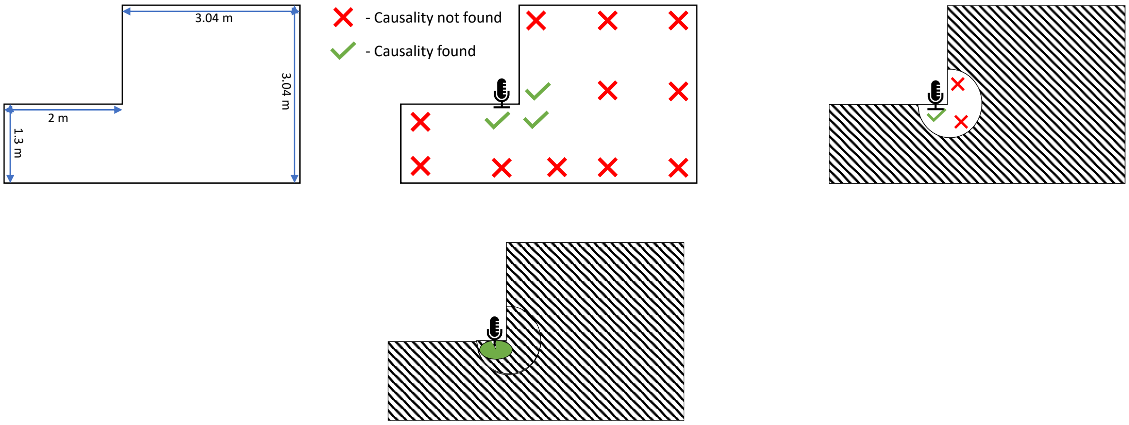


Figure 5.9: Trial-based localization for acoustic sensors.

5.9 Traffic Variation of a Personal Home Assistant During Drop-In

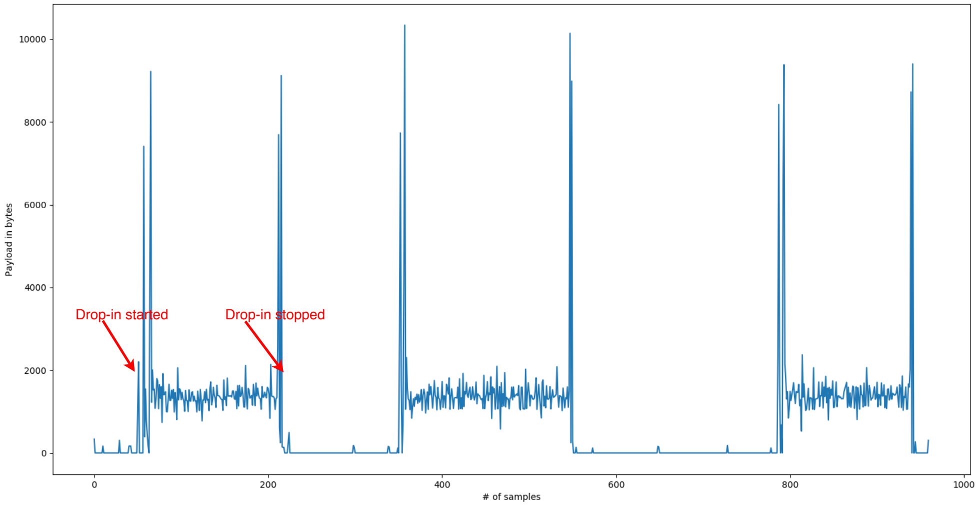


Figure 5.10: Traffic variation of Amazon Echo Dot while dropping-in

As discussed in the previous sections, Amazon Echo devices allow the user to drop into any of their Alexa devices and remotely listen to the audio in the room that they are placed in.

This does not require any authentication on the device side during the drop-in. We perform 3 drop-ins on an Amazon Echo Device and show the traffic variation in Figure 5.10. From the traffic variation, it is clear when the drop-ins start and when they end.

5.9.1 Overhead Analysis

Time: Sensor detection can happen in the background with minimal user intervention. However, this will take some time. In situations where a user wants to immediately know if he/she is being spied on by a sensor (such as when entering a changing room), they can directly begin the active phase where they will perform the **S5** motion. It takes about 40 seconds to perform active detection. For localization, each trial can take 30 seconds. Since the localization space reduction is determined by the user, he/she can perform the trial any number of times. If the total number of trials is n, the overhead will be about $30n$ seconds.

User effort: If the detection occurs in the background, there is no overhead in terms of user involvement. However, both active phase and localization require user effort. In case the user is suffering from physical disabilities, he/she may find it hard to follow through these steps.

Wi-Fi Channel Hopping: In detection mode, SnoopDog must hop channels across all frequencies as an adversary may have configured devices on different Wi-Fi frequency bands. A transmitting device might inadvertently evade detection if it is hopping while SnoopDog is scanning. Hence, we perform multiple scans of the entire Wi-Fi spectrum to reduce misdetections. We would ideally monitor all Wi-Fi channels simultaneously, but this is infeasible with a single WiFi radio. Multiple Wi-Fi radios will reduce the search space per radio at additional cost and degraded portability.

5.9.2 Effects of Range on Detection

Detection depends on the range of a sensor. The camera range is the depth which it can record meaningful information from the scene. If the user stands too far from the camera, their motion may not produce a significant variation in its traffic. For motion sensors and RF sensors, the range is specified by the vendor. The motion sensor that we used has a range of 15 feet (4.6 m) and the RF sensor we used had a range of 4 m. For Amazon Echo, the range is a function of the loudness of sound. For cameras, we perform our detection trials at different distances to find out the range.

We perform the **S5** motion in front of a camera at 1 m and gradually increase the distance. We find that as we move away from the camera, the changes in its traffic decrease. For the Wansview camera, the distance at which traffic changes are not enough to establish causality is the smallest at about 3 m.

5.9.3 Failure Analysis

During our evaluation, SnoopDog's detection failed during some trials for both cameras and motion sensors. In this section, we will describe and discuss those cases.

Camera. SnoopDog failed to detect hidden cameras when the camera video froze (no transmission) for a small duration during our detection trial. Another cause of miss-detection were additional users in the scene since the network traffic due to their motion cannot be filtered out. Also, when the user was present far from the camera (refer Section 5.9.2), the traffic changes due to the motion of the user were not enough to establish Granger causality.

Motion Sensor. Motion sensors sometimes send status messages to the cloud service that they are connected to. These transmissions happen in the form of a short burst. Since there was no motion before these transmissions, they result in SnoopDog failing to establish causality. While we cannot say with certainty what each of these transmissions are, we saw multiple such transmissions when the sensor was started, during update, and sometimes even in between two motion events. A user can also query the sensor status from their phone to which the sensor has to respond. SnoopDog failed to detect causality when these transmissions were present in the network traffic of the motion sensor.

5.10 Techniques to fool SnoopDog

In this section, we discuss how an adversary can fool SnoopDog.

5.10.1 No Encoding or Data Padding

SnoopDog uses the relationship between encoding schemes and ground truth to find out if there is a device which is monitoring the user. Hence, to fool SnoopDog, the sensors can either send un-encoded raw data or they can pad the encoded data to make the data rate constant. Cameras can either pad their traffic or they can send un-encoded images frames. Since sending images will put a large overhead on the network bandwidth, padding the traffic [Apthorpe et al., 2018] is a better idea. We pad the camera traffic with random payload in Figure 5.11. Since SnoopDog cannot see what's inside the payload, it can be anything. The device can even send labels in the payload that help the server decide if this is a valid packet or fake data generated to fool detection. Also in Figure 5.11, we pad the traffic of a motion sensor to make it appear like a constantly transmitting device with no variation in traffic in response to user's motion.

For RF sensors, one can find out the maximum number of points it can output and then always pad the information so that we are transmitting the maximum number of points allowed. These extra points could all be zeros which would make it easier to filter them out on the server side.

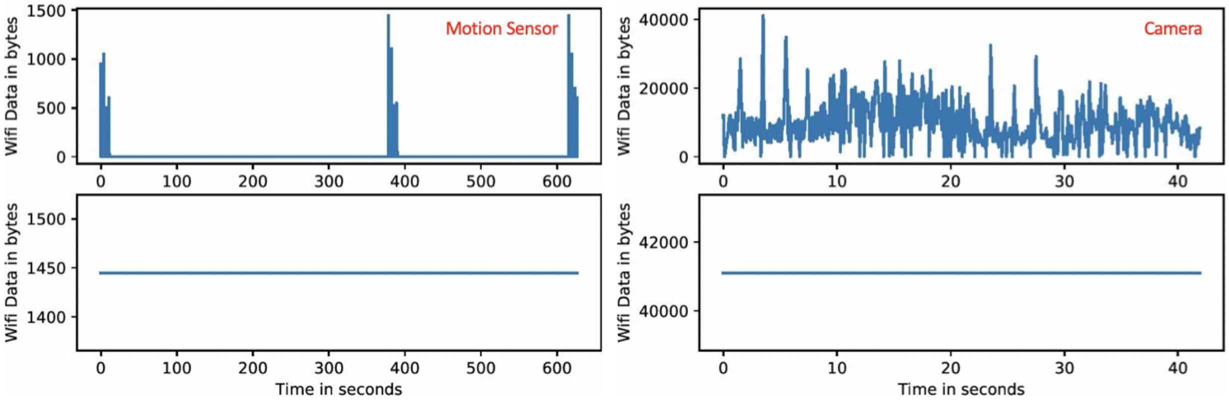


Figure 5.11: Padding the motion sensor and the camera traffic

5.10.2 Adding Random Noise to the Data

Another way to fool SnoopDog is by injecting noise into the device's wireless traffic at random intervals for some time window. Since SnoopDog utilizes the change in device traffic to ascertain a cause-effect relationship, the variations caused by injecting random noise are able to fool the detection.

Devices that do not transmit continuously can randomly send information that creates a pattern similar to their inferred event traffic. This way they can keep sending their information which is hidden within random traffic. We add random noise which appears like regular traffic for a motion sensor in Figure 5.12.

5.10.3 Constantly Vary the Resolution of the Data Being Transmitted

For devices like camera, there are several video resolutions that an adversary can choose. The higher the resolution, the better the video quality is. However, if an adversary chooses a scheme where the video resolution is constantly varying, it will cause random changes in the network traffic. Hence, even if the user's motion is causing changes to the traffic, it is overpowered by the changes in network traffic due to a variation in resolution.

For RF sensors, they can vary the number of maximum points that they transmit con-

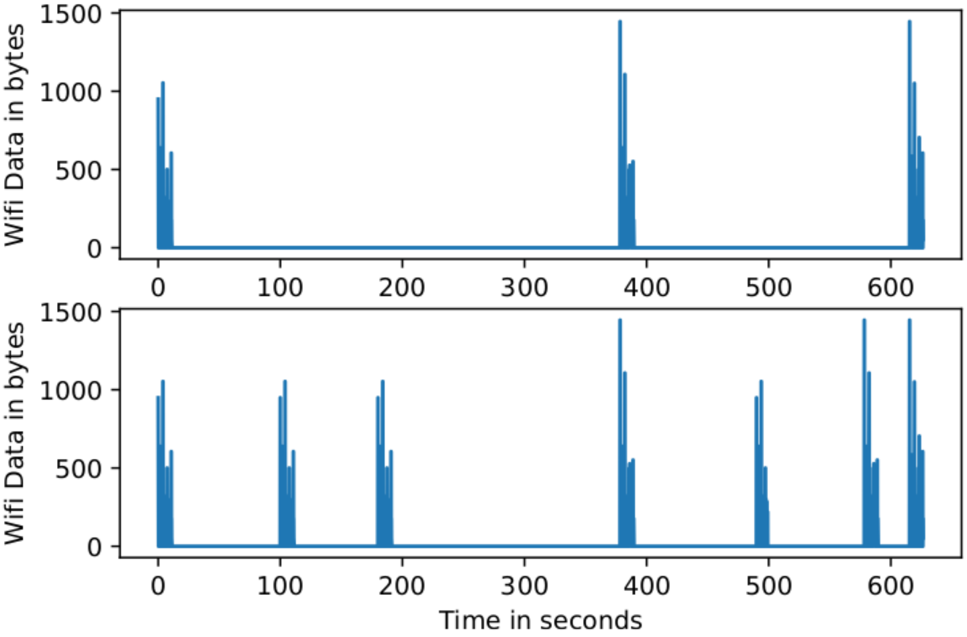


Figure 5.12: Injecting noise in the traffic of a motion sensor to fool SnoopDog

tinuously to achieve a similar effect.

5.10.4 Adding a tape/broadcast delay to the transmissions

An adversary can add a tape delay to the sensor transmissions, i.e. intentionally adding a delay between when something was recorded and when it was transmitted. Since, we are only looking for causality within a small time window, a high tape delay will be able to fool SnoopDog . However, given enough storage capacity and time, it is possible for SnoopDog to scan the entire recording to look for cause-effect relationship with user motion. But for large tape delays, this is not practical.

5.11 Limitations

1: Only limited to VBR devices. Although SnoopDog can detect a wide variety of commonly available sensors, it cannot detect *any* wireless sensor monitoring the user.

For a sensor to be detectable by SnoopDog, the traffic must be encoded with a Variable Bit Rate (VBR) algorithm and the data recorded by the sensor must change in response to user perturbation which can be recorded by a ground truth sensor. That said, most surveillance devices such as cameras, motion sensors and smart-home assistants today fall into this category, and thus we believe SnoopDog can serve as a valid defense.

2: A technically capable adversary can fool* SnoopDog *if they know about its existence. If the adversary suspects SnoopDog is in use, they can use one of the techniques listed in Section 5.10. They can also use channel hopping or MAC randomization. We have not evaluated SnoopDog for any of the above techniques.

3: Evaluation is limited to Wi-Fi devices and devices who route their traffic through a Wi-Fi-hub only. We have evaluated SnoopDog for Wi-Fi-connected devices only. For future work, this framework can be evaluated for other popular wireless communication standards. SnoopDog can be extended to standards like Zigbee [Kinney et al., 2003], Z-Wave [Yassein et al., 2016], and Bluetooth [Muller, 2001, Haartsen, 2003] as long as we have the following: 1) A receiver that can scan their probable frequencies and sniff their packets to find if any devices are transmitting and 2) the ability to find unique device IDs from packet headers and distinguishing header information from payload size. While capturing Zigbee/Z-Wave packets will require additional hardware, recent works have shown that it is possible for a Wi-Fi radio to perform cross-technology communication. [Li and He, 2017, Kim and He, 2015]

5.12 Discussion

Q1: What is the usability of* SnoopDog*? We envision SnoopDog to be implemented as an app on either a smartphone or a smartwatch (or a combination of the two). This means an end-user will not need any prior knowledge about causality and coverage of a device to use it. SnoopDog will continuously work in the background to look for a cause-effect relationship between a user's actions and device traffic. It will then guide a user step-by-step through the entire localization procedure. Since an adversary can place a sensor at any time (e.g.,when a user checks in a room, searches for devices, finds none and then leaves for dinner after which the adversary places the spying device), SnoopDog will still find it because it continuously works in the background. This will not cause any overhead in terms of user involvement.

Q2: How can false positives be reduced? For false positive to occur during active detection, the device's traffic needs to map directly to the **S5** motion during the active phase and user's motion during the background phase, which is unlikely. If there happens to be another camera in an adjacent space monitoring another user who is performing the detection trial within the same time window as the first user, it will trigger a false detection. However, the probability of this happening is low. Nevertheless, it remains a possibility, and mitigating such instances are highly desirable.

Simple strategies can significantly reduce the chances of false positives. First, during the initial monitoring phase for wireless devices, any periodic trends in traffic patterns can be noted; the detector trial should ensure its periods are not synchronous with such periodicity. Furthermore, the detection process can be done multiple times with varying and erratic period lengths. This will drastically decrease the chances of a false positive, as a device would have to coincidentally follow this effectively random traffic pattern. Finally, the entire process itself can be performed repeatedly; each iteration compounds the decrease in false positive rate, such that it eventually reduces to a statistical impossibility.

Q3: Are there alternative approaches to causality? One alternative approach to detecting snooping sensors is correlation. However, correlation does not imply causation. If we have a sensor that measures the ground truth in the modality we want to detect, we need to use causality analysis. For example, it takes the camera some time to process the information and send it over to the server. So if we capture human motion with an IMU, the camera traffic will lag the IMU time series. This is correctly captured by causality analysis but not by correlation. However, if instead of using a sensor to measure the ground truth, we use another sensor that can capture the same modality that we are trying to detect, we can use correlation because if both the devices are capturing the same event, their traffic should show similar trends. Future work can also explore the efficacy of data-driven approaches such as deep learning for time series classification.

Q4: Can we detect continuously streaming audio bugs? There are two ways to encode audio, either constant bit rate (CBR) or variable bit rate (VBR). VBR techniques make use of similarity in sound, such as prolonged silence, to reduce the amount of data required for encoding. In contrast, CBR always encodes with the same number of bits. Many off-the-shelf audio recorders and audio streaming apps use CBR. Since SnoopDog only has access to the payload size of a packet, there must be variation in the payload to determine causality. Hence, SnoopDog cannot detect CBR audio bugs.

Q5: What is the impact of a ground-truth sensor? Qualitatively, the ground-truth sensor enables the detection of causality between human action and hidden sensors. Even if all hidden devices were connected to an accessible Wi-Fi network (which is the same system model used by IoTInspector [Huang et al., 2019a]), one would only be able to detect the presence of a device on the network and not whether it is monitoring a user. To quantitatively demonstrate and evaluate the impact of a ground-truth sensor, Figure 5.3 illustrates an example where an IMU enables SnoopDog to identify between a hidden sensor monitoring a user and disregard a camera in a separate room. Moreover, one may argue that an application can actively instruct the user to move and establish causality between the period of instruction and the Wi-Fi traffic patterns. First, such an approach relies on a general user motion model to establish causality during these time frames. Second, this approach is not capable of background detection as it would rely on active command and response patterns. In Table 5.3 case 1, without a ground truth sensor, the false positive rate is 100%. With a ground truth sensor, this decreases to 6.66%.

5.13 Related Work

This section presents the most relevant and related works.

Detecting hidden devices using RF signals. A popular tool to detect hidden devices is called a bug detector [Nbc, 2019] – an RF receiver that can sense if the received power in a frequency range is above a threshold. The problem with such devices is that they can produce false alarms when used near other RF sources such as mobile phones or laptops [Valeros and Garcia, 2017, Sathyamoorthy et al., 2014]. Also, they give no additional information about the type of device or where it is located. After detection, the onus lies completely on the user to physically find the device and verify if it is a surveillance device or not. The host may have a wireless device to monitor the power consumption of his property, but to the bug detector, it would seem similar to an IP camera.

Classifying devices on the network using wireless traffic sniffing. While services like Princeton IoT Inspector [Huang et al., 2019a] collect traffic statistics to identify the types of devices present on the network, they fail to identify if those devices are indeed spying on the user or not. Just ascertaining the presence of a surveillance device is not enough. The device may be present outside the house or it may be monitoring some part of the house which was already disclosed by the home owner. In cases like this, just identifying such a device exists is not enough, we also need to determine two important facets – is the device spying on the user and is it located in an area of the house that has the potential to violate user privacy. Moreover, tools like this need to have access to the network in order to be effective. If the snooping devices are placed in a hidden network or on a password protected network, the use cases of such a tool are limited.

Other network traffic analysis tools [sol, , Schmitt et al., 2018] utilize traffic data to find which devices are consuming high bandwidth. Such techniques can be used to classify audio and video data streams present in the wireless networks. However, with an increase in streaming services [Steele, 2019, Kumar et al., 2019], it is difficult to distinguish camera video and audio flows with those of streaming services based on just their bandwidth usage.

Detecting cameras on the network using wireless traffic sniffing. Wampler *et al.* [Wampler et al., 2015] and others [Nassi et al., 2019, Liu et al., 2018] show that information leakage occurs in camera traffic due to how videos are encoded. They observe that changing lighting conditions cause noticeable variations in the network traffic. Though these techniques perform well, their performance degrades when the environment lighting changes naturally. Additionally, while these techniques work well for a camera, they do not generalize to other types of snooping devices, like RF sensors or motion detectors. Finally, in order to be able to change the lighting conditions of a space, the user requires either specialized hardware (like an LED board or a bulb) or access to lighting controls, which is not guaranteed.

Approaches like DewiCam [Cheng et al., 2018] exploit the correlation between human motion and camera data flows to determine if the camera is indoors or outdoors.

In [Wu and Lagesse, 2019], Wu *et al.* use their own camera to record a scene while simultaneously sniffing the network traffic. They compare the data rate and pattern of their camera with other devices in the network to look for any similarities. If a similarity exists, there is a high probability that the device is a camera.

Localizing wireless devices using RSSI. Received Signal Strength Indicator (RSSI) is the estimate of the power received at the receiver from the transmitting device. The power received drops with distance, and so does the RSSI. This property is leveraged to localize devices using RSSI [Sun et al., 2014, Luo et al., 2011, Xue et al., 2017, Li et al., 2018]. However, due to phenomenon like multipath and shadowing, the accuracy varies from space to space [Jondhale et al., 2016]. The error is very high (several meters). For small rooms, such a result will be meaningless, as the snooping device can be effectively hidden anywhere.

CHAPTER 6

Understanding Privacy Impact of Instrumenting Spaces with Sensors

6.1 Introduction

With the recent growth in the Internet-of-Things (IoT), sensors and sensing devices have infiltrated all aspects of our daily life – in smart homes, office spaces, and public areas. The data collected by these sensors provide utility to the users whose data is being collected (e.g. fitness activity tracking), or the device owner (e.g. home security). However, the opaque data collection, storage, and processing by manufacturers and device operators has motivated a discussion on privacy - particularly, how to instrument physical spaces with sensors and what to disclose to the sensed users [Naeini et al., 2017, IGP, 2015, Emami-Naeini et al., 2020].

Privacy discussions on *invasiveness* [Lee and Kobsa, 2017, Psychoula et al., 2018, Gochoo et al., 2018] or *sensitivity* [Zhou and Piramuthu, 2015] of sensors informally allude to protecting the information representing a monitored user's actions and behaviors in the physical world. Non-expert users (someone who is not well-versed in various conceptions of privacy (Section 6.3)) rely upon their prior experiences with a device or a class of devices to form their notions of privacy. This notion is also reflected in several research works where in lieu of the knowledge of privacy-conceptions, they use human-understandability as a proxy for privacy. A common notion of privacy invasiveness is that sensors which capture data easily understood by human sensory organs (e.g. visual and audio recordings) should not be placed in physically privacy-sensitive areas (e.g. bedrooms, changing rooms, or washrooms) [Taylor, 2010, Schwartz, 2012]. This notion is the premise for several recent works claiming that lower-dimensional RF sensors are more privacy-sensitive than visual sensors [Singh et al., 2019, Billah et al., 2021, Aziz Shah et al., 2020, Rahaman and Dyo, 2021, Ashleibta et al., 2021, Liu et al., 2019, Zeng et al., 2020, Yin et al., 2022, Feng et al., 2020, Gurbuz et al., 2021, Li et al., 2021, Yang et al., 2021, Wang et al., 2021a, Fan et al., 2020, Saeed et al., 2022, Raeis et al., 2021, Avrahami et al., 2018, Liu et al., 2021]. In particular, some works claim that raw RF sensor data reveals less information than visual data such as camera images [Fan et al., , Raeis et al., 2021, Avrahami et al., 2018] due to poor interpretability by humans. In reality, raw information collected by a visual sensor is not an image but merely radiation on the individual pixel sensors, which is later translated, processed, and presented as human-understandable images [Isola et al., 2011]. Similarly, low-dimensional RF sensor data can be translated, abstracted, or "imaged" into privacy-sensitive information. This begs the question:

RQ1a. *If we take non human-understandable data collected by a sensor and convert it into a more understandable form, will that change the user's privacy perceptions about that sensor?*

While humans cannot understand the information collected by RF sensors (e.g. EM radiation collected at the receiver generally represented in the form of IQ samples [Scott, 2001]), they can certainly understand the inferences that algorithms can make on that data. Importantly, advances in deep learning have enabled various inferences from RF data. These various inferences may be privacy-sensitive, such as human emotion [Zhao et al., 2016] and several forms of human activity recognition [Singh et al., 2019, Billah et al., 2021, Aziz Shah et al., 2020, Rahaman and Dyo, 2021, Ashleibta et al., 2021, Liu et al., 2019, Zeng et al., 2020, Yin et al., 2022, Feng et al., 2020, Gurbuz et al., 2021, Li et al., 2021, Yang et al., 2021, Wang et al., 2021a, Fan et al., 2020, Saeed et al., 2022, Raeis et al., 2021, Avrahami et al., 2018, Liu et al., 2021]. Other works [Zhao et al., 2018, Wang et al., 2020b] showed that RF sensors could surpass a visual sensor's capabilities by detecting human posture and activities through a physical wall. More critically, the advent of generative adversarial networks (GANs) [Goodfellow et al., 2014] has enabled the ability to synthesize images from low-level information, including across modalities [Kezebou et al., 2020]. Deep learning has enabled us to make privacy-sensitive inferences, as well as creating interpretable data representations from low level structures - both of these capabilities require us to reconsider the privacy perceptions surrounding RF sensors. Thus, we can further simplify *RQ1a* to ask a second question:

RQ1b. *What matters more to users when they are forming their privacy perceptions about a device: the data collected by the device (modality and human-interpretability), or the inferences that can be made on that data, or both?*

Theories such as the mere-exposure effect [Bornstein and D'agostino, 1992, Bornstein and Craver-Lemley, 2016] state that human preference for objects also depends upon their familiarity with those objects. Hence, we also need to explore whether it is possible that a user's perceptions of privacy may be affected by how familiar they are with the device. This sense of familiarity may include whether the user has interacted with that particular device, other devices from the same manufacturer, other devices from the same family of the devices or has worked on those devices in some capacity. This leads us to another question:

RQ2. *How does familiarity with a device affect a user's perception of its privacy-sensitivity?*

Human beings create their first-impressions about an object using its physical appearance. This phenomenon is also seen in shopping where the physical appearance of products greatly influences a user's decision to buy them [Creusen and Schoormans, 2005]. The physical appearance of an object also invokes a user's familiarity with that object [Bornstein and Craver-Lemley, 2016]. Since sensing devices instrumented in a physical space do not come with any disclosures or labels, users have to rely upon physical appearance to build privacy perceptions of that device. This leads us to our next question:

RQ3. *How does the physical appearance of a device affect a user's privacy perception of that device?*

For example, if a device that is considered to be innocuous, such as a WiFi router, were to look like a more privacy-invasive device, such as a camera, will that affect a user's privacy perceptions?

Finally, with growing user concern over how data is being collected, stored and shared [Wang, 2016, Wang et al., 2016b], it is important to discuss how the amount of control (or lack thereof) a user has over their data helps shape their privacy perceptions. Recent advancements in machine learning have further exacerbated the rampant unauthorized secondary use of data [Chen and Kim, 2013, Oliveira and Zaiane, 2010]. Furthermore, zero-day vulnerabilities [Bilge and Dumitraş, 2012] and the constant fear of devices being hacked [Rostami12 et al.,] have made users wary of installing sensing devices in their personal space. These implications lead us to our final question:
RQ4. *How does control over the design of a device and its data policies (type of data collected, its storage and transmission) affect a user's perception of privacy-sensitivity of that device?*

By answering questions *RQ1b, RQ2, RQ3, RQ4* we investigate user perceptions of privacy surrounding low-dimensional sensor data and how they compare against existing notions of privacy for such data. We use RF sensors such as mmWave radar and WiFi router as examples of low-dimensional sensors for our study due to their ubiquity and a recent surge in their sensing applications. We aim to understand how factors such as modality of sensing (data collected by the device), inferences that can be made on the data, familiarity, physical appearance, and control help shape a user's privacy perceptions regarding sensor privacy. Intuitively, given that more and more spaces are instrumented with sensors, we aim to validate (or invalidate) prior researchers' notions of privacy given user perceptions. First, we aim to compare the privacy perceptions of data and the inferences that can be made from that data. Second, we study the impact of users' prior knowledge about the properties of the sensor in question, such as the sensor type, form factor, manufacturer, and data-processing algorithm. Finally, we study the perceived privacy impact of a user's control over the device configuration, the data collection, and data storage processes for a given device.

Study Details. To validate our hypothesis, we performed an online survey study[1] on 122 respondents from the United States to measure users' perception of privacy across the four factors described previously: data information, prior device knowledge, physical appearance and device control. Our study seeks to understand the contribution of each factor in helping shape the privacy perceptions of users. We also present the respondents with scenarios and open-ended questions that allow them to express their thoughts. We then conduct both qualitative and quantitative analysis of the responses to answer our research questions and ascertain the validity of our hypothesis.

Study Results. The main findings of our study are as follows:

- In the case of sensors that collect human-interpretable data (such as cameras), data and inferences can be used interchangeably to inform user decisions regarding privacy.
- In the case of sensors that collect non human-interpretable data (such as RF sensors), it is the inferences, rather than data collected, which affects a user's privacy perceptions most.
- When the data and inferences are not available, the physical appearance and familiarity of a user with a device type are the primary factors that dictate how users perceive the privacy-sensitivity of a device.
- If given complete control over the data policies and device design, users are more likely to instrument their personal spaces with sensing devices.

Research Goals. Our goal is to study what factors affect the privacy perceptions of a non-expert user when they encounter an IoT device. The factors discussed in this work can be used in conjunction with other conceptions of privacy (Section 6.3) to better inform future works that aim to instrument user-spaces with sensors.

[1] exempted as part of IRB approval process

Contributions. The main contributions of our work are summarized as follows:

- We rigorously define (Section 6.4) and investigate the factors contributing to a non-expert users' privacy-sensitivity in the context of advances in deep learning for low-dimensional sensor data. To the best of our knowledge, we are the first ones to do so.
- We design and conduct a user study that aims to understand the impact of the defined factors that affect a user's privacy perceptions of devices.
- We discuss how researchers can utilize our findings to improve their disclosures regarding privacy-sensitivity across sensing applications.

6.2 Related Work

We first describe related work on claiming RF to be less privacy sensitive than visual modalities. We then cover research investigating different dimensions of sensor modalities that affect user privacy perceptions. Lastly, we cover related research that seek to inform and measure privacy behaviors of sensors and systems. To the best of our knowledge, our privacy study is the first of its kind where we compare data collected from a device to the inferences that can be made on that data in terms of their effects on a user's privacy perceptions.

6.2.1 Claims without validation: RF is more privacy sensitive than camera/vision

There is a growing body of work, particularly in the RF domain, which seeks to replace more 'privacy-invasive' modalities (such as cameras) with RF sensing. The reason for this replacement is at least in part due to concerns of privacy, with some works in this area citing privacy as the primary reason for replacing modalities. Various application scenarios arise where users may have information they wish to keep private - these scenarios involve sensing and monitoring occupancy in buildings [Billah et al., 2021, Aziz Shah et al., 2020], understanding travel patterns in smart homes [Gochoo et al., 2018,Rahaman and Dyo, 2021], inferring breathing patterns [Ashleibta et al., 2021, Liu et al., 2019, Zeng et al., 2020, Yin et al., 2022], detecting posture and gesture at various granularities [Feng et al., 2020,Gurbuz et al., 2021, Li et al., 2021, Yang et al., 2021, Wang et al., 2021a], and general daily life and activity monitoring [Fan et al., 2020, Saeed et al., 2022, Raeis et al., 2021, Avrahami et al., 2018, Liu et al., 2021]. However, these works are motivated by a notion of privacy that has yet to be properly understood - a majority of these works directly claim that cameras are more privacy invasive than the RF modality without specifying what this invasiveness means. In the more extreme cases, some works claim that the RF modality has no issues with privacy at all and is 'non-privacy invasive'. For several of these works, privacy is the primary reason to choose RF modalities, due to an existing notion of privacy where data is less privacy-invasive if it is less interpretable [Fan et al., ,Raeis et al., 2021, Avrahami et al., 2018].

We argue that these claims have not been sufficiently validated, and this notion of privacy itself does not encompass a sufficiently deep understanding of what other modalities are capable of. Time and time again we witness examples of unexpected privacy-invasive inferences made by various modalities, such as reidentifying users based on accelerometery data [Saleheen et al., 2021] or sensing sound from RF sensors [Wang et al., 2020b]. As many of these related works focus on scenarios that monitor a user's daily life and health attributes (which are private from both a personal and legal [Rights (OCR), 2008] perspective), it is imperative that the capabilities of a modality are better understood before such claims are made. This motivates a need to improve our understanding of what truly makes a sensor 'privacy invasive'. In our study we seek to correct an existing notion that the privacy perceptions of a particular sensor can be accurately judged based on data interpretability. Our investigation of existing work has led us to the design of 4 relevant concepts that affect user perception of sensing modalities.

6.2.2 Exploring Privacy Sensitivity of Devices and Sensors

The 'privacy invasiveness' of a sensor is not solely dependent on the sensing modality. There are a number of works investigating different dimensions of sensors (appearance, familiarity, etc.) and how they affect perceptions of privacy. [Ahmad et al., 2020] and [Koelle et al., 2018] explore different physical designs of sensor mechanisms capable of describing sensing state. More specifically, they explore how certain notifiers and actuators, such as lights and shutter mechanisms, are capable of improving perceptions of privacy. [Ahmad et al., 2020] identifies the concept of 'tangible privacy', which describes a set of physical mechanisms on devices to clearly represent privacy states of IoT devices, with the goal of minimizing uncertainty in certain device states (such as being on or off). In a similar vein, [Koelle et al., 2018] focuses on designs for body-worn cameras that clearly describe their sensing state. [Cheng et al., 2019] designs a birdhouse-shaped camera system which grants privacy controls to inhabitants in a home home setting. [Conference and Pierce, 2019] focuses on redesigning smart home security cameras to both improve adoption while enabling privacy controls. In our study, we consider how the concept of physical appearance affects user perception of privacy.

In addition to physical appearance of sensors, privacy perceptions of sensors are also affected by the setting in which they reside, as well as the manufacturer of those sensors. [Haney et al., 2021], [Zheng et al., 2018], and [Malkin et al., 2019] both study perceptions of privacy regarding smart home devices, and how manufacturer trust affect these perceptions. [Haney et al., 2021] focuses on how users believe responsibility should be assigned when it comes to handling privacy concerns of smart home devices - they consider different combinations of personal, manufacturer, and governmental responsibility. [Zheng et al., 2018] studies smart home users and their beliefs in privacy protections of devices - they find that both brand familiarity and reputation in device manufacturers plays a primary role in determining trust-based purchasing decisions. Lastly, [Malkin et al., 2019] describes trust in different smart speaker manufacturers and how they handle data. [Velykoivanenko et al., 2021] investigates the different features of fitness trackers, and how well users understand privacy implications in the fitness-tracker ecosystem. In this study, we also consider how the concept of manufacturer familiarity and reputation influences privacy perceptions of devices.

In addition to manufacturer trust affecting privacy perceptions, [Zheng et al., 2018] also studies how data-sharing behaviors from smart home devices is heavily influenced by perceived benefit of sharing. [Akter et al., 2020] seeks to study image sharing behaviors of visually impaired persons, and how different types of objects present in the images, as well as familiarity with assistants affect sharing decisions. Unlike these works, we focus on specific data-sharing behaviors when their intended inferences are known. We choose inferences that have been shown to be possible on both cameras and RF modalities. [Jin et al., 2022] studies smart home privacy-protective behaviors (SH-PPBs), examining how people currently enforce privacy controls over smart devices, as well as raising ideas for new privacy features - such features come in the form of physical design controls, but also as software features (such as a data management portal). Similarly, our study also aims to understand how varying levels of user control and knowledge of the sensor affects privacy perceptions. [Naeini et al., 2017] identifies a set of factors influencing privacy preferences in IoT scenarios - in particular, they focus on aspects of data collection of different devices. [Zeng et al., 2017] explores different levels of familiarity with IoT technologies, and how it influences knowledge of privacy and security risks arising in smart home scenarios. [Lee et al., 2022] studies different privacy attitudes and privacy concerns regarding data collection via a variety of modalities. [Baron and Musolesi, 2020] studies the privacy sensitivity of different types of information captured in mobile services.

In our study of RF and camera modalities, we explore how familiarity, appearance, and inferences from these particular sensors play a major role in privacy perceptions. Unlike previous works, our goal is not to design new sensors, but rather to explore existing sensor designs and consider the different factors affecting user privacy perceptions.

6.2.3 Structuring and Measuring Privacy Properties

One active area of research relevant to our study is the creation of transparency mechanisms for describing privacy properties of devices and systems. Creating such mechanisms often requires structuring privacy properties into a set of concepts that are critical to privacy perceptions of users. Privacy and Security labels [Emami-Naeini et al., 2020] for IoT devices aim to provide concise insight into a device's data collection behavior, enabling consumers to make better device purchasing decisions. Such methods of improving transparency have already been adopted by companies such as Apple [noa, a] and Google [noa, 2022], which typically focuses on installation of mobile apps. In both cases, they distill a set of concepts that are critical to helping users make better privacy decisions. Concepts include security-update policies, sensor data collection, purpose, manufacturer, and several others. However, these mechanisms often don't include concepts relating to potential privacy-invasive inferences, such as race, occupancy, and activity recognition. As we explore different dimensions of privacy-invasiveness in our study, we believe our findings can be complementary to the design of these transparency mechanisms and introduce new concepts.

In addition to creating a structured set of privacy properties that affect privacy perceptions of devices, various works also seek to measure the privacy properties of systems. Some existing methods focus on mobile applications [noa, b] and websites [noa, c]. For example, PrivacyGrade [noa, b] aims to assign grades (ranging from A to D) to different mobile applications based on discrepancies in behaviors described by the developer, and the underlying application software. In the web domain, PrivacyScore [noa, c] seeks to provide a ranking of websites based on presence of a set of security features. Another broad category of works may also seek to describe formal privacy metrics [Wagner and Eckhoff, 2019] from a technical perspective. Example categories include adversarial success, data similarity, and several others - however, unlike these works, we seek to capture privacy perceptions from a human perspective, with a focus on sensors rather than a system.

Our study is partly inspired by previous works studying various dimensions of sensors. We structure our study around several critical concepts affecting privacy perceptions in order to break existing notions of privacy-invasiveness for RF sensing.

6.3 Privacy Conceptions and Terms in this Chapter

In this chapter, we use the terms privacy notion, privacy perception, and privacy preference interchangeably to describe how a user perceives a device or a sensor from a privacy perspective - does the user think a device is going to protect their privacy or not? A device that is designed with privacy in mind (or is perceived to protect user privacy) is considered less privacy-invasive and more privacy-sensitive.

6.3.1 Popular Conceptions of Privacy

There are several conceptions of privacy [Nissim and Wood, 2018, Solove, 2008, Nissenbaum, 2009, Cranor, 2013]. In this section we discuss some popular conceptions. The definitions for these conceptions have been borrowed from the works cited above.

- **Contextual Integrity**: Contextual integrity assesses how closely the flow of personal information conforms to context-relative informational norms. More precisely, in a context, the flow of information of a certain type about a subject from a sender to a recipient is governed by a particular transmission principle. Contextual integrity is violated when the norms in the relevant context are breached. Intuitively, it recognizes that certain parties may obtain certain types of information about other parties under the right terms and for the right reasons.

- **Anonymization and De-identification**: Many privacy technologies are designed with the goal of de-identifying personal information. This approach equates privacy protection with making personal information anonymous or de-identified, i.e. preventing an individual's information from being linked with him or her. The premise is that it is impossible (or, at least, very difficult) to infer personal information pertaining to an individual from a de-identified dataset or use it to violate an individual's privacy in other ways.

- **Semantic Security**: The definition of semantic security compares what an attacker (without access to the decryption key) can predict about the message m given the ciphertext c with what the attacker can predict about the message m without being given the ciphertext c. The advantage that access to the ciphertext gives to any attacker is quantified. Encryption schemes are designed to make this advantage so negligible that access to the ciphertext does not give the attacker any practical advantage over not getting any information about the message at all.

- **Differential Privacy**: It guarantees mathematically that a person, who is observing the outcome of a differential private analysis, will produce likely the same inference about an individual's private information, whether or not that individual's private information is combined in input for the analysis.

When encountering a new sensor, a non-expert user will have no knowledge of these conceptions and hence will rely more on their experiences with the device or the class of devices. The privacy notions formed during this interaction between a non-expert user and a sensor is what we aim to study in our work.

6.4 Method

In this section, we describe our survey formulation process and data analysis methods.

6.4.1 Hypothesis

In this work, we want to challenge the notion that the most common factor affecting privacy sensitivity of devices is the ability of human beings to understand their data. To evaluate this notion and determine the characteristics affecting privacy perceptions in devices, we came up with a set of 4 factors that affect how a user perceives the privacy-sensitivity of a device presented to them. In the following sub-sections, we describe these factors and our reasoning behind them.

6.4.1.1 Privacy Factor (1): User's Prior Knowledge

Mere-exposure effect [Bornstein and D'agostino, 1992, Bornstein and Craver-Lemley, 2016] states that the more familiar [Raghunathan,] people are with something, the more preference they have for it [Fang et al., 2007]. This effect is seen when people stick to the same brand of cars, devices and even engage with same businesses in their daily lives. Building upon this, we want to examine whether user's prior knowledge of a device, family of devices, or the manufacturer plays a role in their privacy perceptions of that device. Specifically, we want to test the following hypothesis: *A user's familiarity with a device and its meta-properties (e.g. manufacturer, device type) affect their privacy perceptions about that device.*

6.4.1.2 Privacy Factor (2): Physical Appearance of the Device

Human beings tend of build their opinion about objects from the very moment they appear in front of them. Such is the case in shopping where the appearance of products greatly influences a user's decision to buy it or not [Creusen and Schoormans, 2005]. Similarly, whether at work or at a public space, when we see a sensor we immediately begin to associate it with what we know. This association is often on the basis of its physical appearance only, since no other information about that device is available. This physical appearance affects the user's familiarity with said device, which in turn affects privacy perceptions. At the same time, though the user may be familiar with the device, they may not necessarily be comfortable sharing data from it - one example is a camera, which is familiar but users may not feel comfortable around. In our study, we want to examine how the physical appearance of a device affects a user's privacy perception of it. Specifically, we want to test the following hypothesis: *If we take a device that users are comfortable and familiar with, and change its physical appearance to that of a device that they are still familiar but no longer comfortable with, their privacy perceptions about that device will change to being more negative.* Since familiarity also affects our perception based on physical appearances, this hypothesis is an extension of the previous hypothesis.

6.4.1.3 Privacy Factor (3): Interpretable Data Representations

Several research works in the sensing community [Li et al., 2019b] equate privacy with the ability for a human to make inferences on the raw data collected by that sensor. In [Li et al., 2019b], the authors claim that RF sensors preserve visual privacy since it does not capture the visual shape of the target - implying that since humans have vision as a sensory modality, devices that collect vision data are more privacy-invasive. In other words, this notion of privacy is based on using data representations which are less interpretable to humans. Other works describe a similar notion, where RF data is more privacy preserving than visual sensors due to having less-interpretable data representations [Fan et al., , Raeis et al., 2021, Avrahami et al., 2018]. However, in a world where the machine learning is used to make inferences of the collected data, this notion is not sufficient to meaningfully ensure privacy. In fact, works such as [Zhao et al., 2018] show that it is not only possible to extract shape, but also posture of human targets using a radar.

In this study, we want to find out what causes users to perceive a device negatively in terms of it's privacy – the data representation it collects or the inferences that can be made on that data. More specifically, we want to test the following hypothesis: *When shown the inferences that can be made on the (non-human interpretable) data collected by a sensor,* *the user's privacy perceptions change about that device.* Importantly, this allows us to study changes in privacy perceptions when users are made aware of possible inferences from sensor data.

6.4.1.4 Privacy Factor (4): Control over Device Design and Data Policies

As the world moves more and more towards data-driven methods for tasks such as learning user preferences, modeling user behavior and selling ads, users have become wary of what data they share and how it is being used [Wang, 2016,Hochheiser, 2015,Wang et al., 2016b]. Often times, even if the data is being used for one purpose, there is no way to tell if the company collecting that data will start using it for another purpose [Brown and Muchira, 2004]. Such collected data can be used to make various inferences about users. As a result, we investigate the effects of granting users varying degrees of control over collected data, and how such control affects willingness to adopt sensing devices into their environment. More specifically, we want to test the following hypothesis: *If given more control over how data is collected, transmitted and stored, users are more likely to instrument their spaces with sensing devices.*

6.4.2 Survey Study

To test our hypothesis we conducted an online survey on the privacy preferences of users from diverse backgrounds. Participants took approximately 12-18 minutes to complete the survey. In the survey, we showed participants images of sensing devices, the data collected by those devices, and the inferences that can be made on that data. In addition, we also posed some situational questions that allowed us to test our hypothesis. In the following subsections, we dive deeper into our survey methodology.

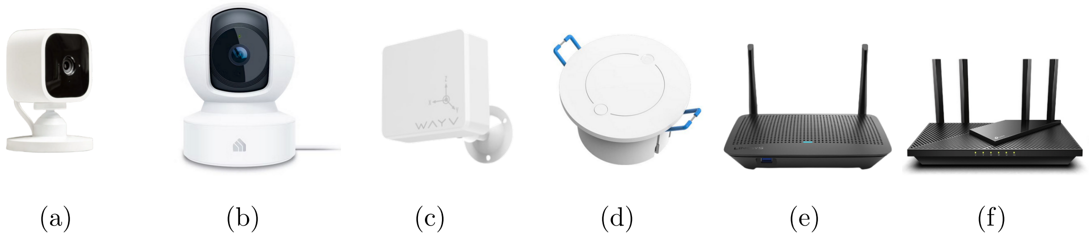


Figure 6.1: We use 3 main classes of devices in our study. Each class has 2 different devices (in order to ascertain that the user responses are dependent on the class of device and not on the device itself.) These devices are **a)** An Amazon Blink home camera, **b)** Kasa Inoor Pan/Tilt Smart Security Camera **c)** A Wayv mmWave radar from Aienstein AI, **d)** LifeSmart mmWave Human Presence Sensor, **e)** LinkSys MAX-STREAM AC1300 WiFi router, and **f)** TP-Link AX1800 WiFi 6 router. In this chapter, we use RF devices to learn more about factors that govern user privacy perceptions.

6.4.2.1 Selection of Scenarios

In our survey, we investigate whether it is the physical appearance, data representation, or the high-level inferences that users most cared about when it came to privacy perceptions. Additionally, we also investigate how familiarity with a sensor or family of sensors affects its privacy perceptions. We followed a three-step approach to decouple various properties from each other. We started by showing users an image of a device and asked them how familiar they are with said device. Then we asked them to write what they believed the device was. Finally we asked them whether they would be comfortable with installing said device in their bedroom. Since we wanted to keep the study in context with radio-frequency (RF) sensing, we chose to compare user opinions about privacy-sensitivity of cameras with mmWave sensors, WiFi routers, and audio sensors – all of which are commonly used in sensing tasks. Finally, the respondents were presented with several scenarios that measure perceptions of privacy in the face of varying degrees of control over shared data from a device.

6.4.2.2 Selection of Devices

In our survey, we mainly focus on 3 classes of devices – cameras, RF sensors (such as mmWave radars) and WiFi routers. The goal of our survey is to highlight how users perceive the privacy of RF sensors when compared to cameras (since cameras are considered to be more privacy invasive by existing notions of sensor privacy). For our survey, we selected three devices – a smart-home camera, a mmWave sensor and a WiFi router. While the WiFi router has become ubiquitous in households, both cameras and mmWave sensors are steadily gaining inroads into the smart-home ecosystem. mmWave based RF sensors have been used for various applications such as human activity recognition [Li et al., 2019b, Ding et al., 2021, Singh et al., 2019] over the past several years. Similarly, WiFi routers are also being leveraged to make a variety of inferences about users in a space [Wang et al., 2017, Li et al., 2019a, Chen et al., 2018]. Hence, we chose these devices to compare and contrast with a camera.

To avoid a particular brand name or a form factor affecting a respondent's decision (ascertain that it is the class of device and not the device itself that is being shown that affects a user's response), we select two sets of images for all the three types of devices as shown in Fig. 6.1. We presented some respondents with first set of images and other respondents with second set of images. Our results show that both sets of images lead to similar trends in user responses. For cameras and WiFi routers, we select best sellers on Amazon.com while for a mmWave sensor we select commercially available off-the-shelf devices (the choice for mmWave sensors is limited). Our two cameras include an Amazon Blink Mini 1080p camera and a Kasa Indoor Pan/Tilt Smart Security Camera. Both of these are best sellers on Amazon.com. For mmWave devices we selected an indoor mmWave radar called Wayv Air which is manufactured by Ainstein AI and another device called the LifeSmart mmWave Human Presence Sensor. Both of these devices are some of the only commercially available off-the-shelf mmWave devices. For WiFi routers, we choose a TP-Link AX1800 WiFi 6 Router (best seller on Amazon.com) and a Linksys MAX-STREAM

AC1300 Dual-Band Mesh WiFi 5 Router. All the six devices are commercially available and can be purchased online.

6.4.2.3 Selection of Sample Data Representations for each Device

In our survey, we show users a snapshot of the data collected by the sensing devices listed above. We ask respondents about their privacy comfort levels with a device/sensor being placed in their bedroom. To help them visualize this scenario, we chose a generic camera image of two people sleeping on their bed as this best describes the scenario that our survey mentions. For the mmWave radar, we took a data snapshot from the company's website. Finally, for the WiFi router, we chose the output of Wireshark packet sniffing to show the sample data collected by it. Wireshark [Banerjee et al., 2010] is one of the most popular open-source network analyzer that is widely used to analyze WiFi traffic.

6.4.2.4 Selection of Possible Inferences for each Device

For each sensing device listed above, we showed the respondents a list of potential inferences that can be made on their data. These inferences and the research works that they are derived from are listed below:

- **Camera:** (i) The number of people in the room [Teixeira and Savvides, 2007], their clothing [Stearns et al., 2018, Huang et al., 2021], race [Layne et al., 2012], ethnicity [Layne et al., 2012], body shape [Guan et al., 2009], height [Guan et al., 2009], and posture [Guan et al., 2009]; (ii) Certain activities, behaviors, and health conditions of the people [Ann and Theng, 2014, Demrozi et al., 2020]: having meals, drinking, smoking, walking, praying, watching TV, using a smartphone, intimate moments (hugging, kissing), and breathing rate [Massaroni et al., 2018].
- **mmWave Radar:** (i) The number of people in the room [Choi et al., 2017, Weiß et al., 2020], body shape, height, and posture [Li et al., 2020b, Sengupta et al., 2020]; (ii)

Certain activities, behaviors, and health conditions of the people [Li et al., 2019b, Ding et al., 2021, Singh et al., 2019]: having meals, drinking, smoking, walking, praying, watching TV, using a smartphone, intimate moments (hugging, kissing), and breathing rate [Wang et al., 2020a, Chauhan et al., 2020].

- **WiFi Router:** (i) Websites that have been visited, total time spent on a smartphone, and sleep schedule [Ohm, 2009, Hernandez-Quintanilla et al., 2021]; (ii) The number of people in the room [Yang et al., 2018, Cheng and Chang, 2017], body shape, height, and posture [Jiang et al., 2020, Ren et al., 2022]; (iii) Certain activities, behaviors, and health conditions of the people [Wang et al., 2017, Li et al., 2019a, Chen et al., 2018]: having meals, walking, praying, watching TV, using a smartphone, intimate moments (hugging, kissing), and breathing rate [Abdelnasser et al., 2015, Gao et al., 2020].

6.4.2.5 Measuring Privacy Perceptions

To gauge how concerned users were regarding the privacy aspects of a sensor, we asked them the following questions:

Q1. *How comfortable are you with this device being placed in your bedroom and streaming/recording data? (an image of the device is shown to the respondents)* We start out with this question since the the form factor of the device and its appearance are the first things that users notice. Even before they are able to read more about the device, examine the data it collects or look at the inferences that it makes, they are already forming privacy perceptions of a device.

Q2. *The data collected by the device shown below looks like the image on the right. How comfortable are you with this device being placed in your bedroom given the data that it collects? (an image of the device and the representation of its collected data is shown to the respondents)* This question helps us measure how much the human ability to interpret data plays into the notions behind privacy. This also helps us evaluate the claims about privacy made by research works that use the human ability to understand raw data as a measure of privacy.

Q3. *The device shown below is from a company that promises that no humans look at the raw sensor data captured by these devices - instead, only specific information (as a result of processing) is shared. The processing algorithms can make the following inferences about your activities, behaviors, and surroundings: [List of inferences]. How comfortable are you with such a device being placed in your bedroom and sharing the inferences shown above? (an image of the device is shown to the respondents)* In this question, we explicitly tell the respondents that the data collected by the devices are not shared with any human beings - instead, only the inferences, as a result of processing, are shared. This helps us decouple the notion of data representations from inferences and to see which of the two matters more to the user in terms of privacy notions.

6.4.2.6 Equating Privacy Sensitivity with Comfort

Privacy-sensitivity and *privacy-invasiveness* are vaguely defined terms for describing privacy perceptions, and it is difficult to convey their meaning to a non-technical user. Existing research works that aim to measure privacy perceptions use questions like *how comfortable?* to understand a user's privacy expectations [Akter et al., 2020]. In our survey, we also use this as a proxy for understanding privacy perceptions. The respondents answers to subjective questions in our survey show that our question formation was able to convey the meaning properly to them.

During the survey, the respondents were able to respond to questions on a 5-point Likert scale [Joshi et al., 2015]. For familiarity, the scale was: (1) Very unfamiliar, (2) Unfamiliar, (3) Have some idea, (4) Familiar, and (5) Very familiar. For comfort, the scale was: (1) Very uncomfortable, (2) Uncomfortable, (3) Neutral - Neither comfortable nor uncomfortable, (4) Comfortable, and (5) Very comfortable.

6.4.3 Choosing a Common Private Space

In the US, a house is considered a private space [law,]. Out of all parts of this private space, bedrooms are considered to be the most private [Berry, 2020]. People use various parts of their homes in different ways and may not always have a high expectation of privacy there. Such is the case in home rentals where the host may place cameras and other sensors in common areas of the house. However, instrumenting spaces like bedrooms with sensors (such as cameras) is not allowed. Since people consider their bedrooms to be a sacred and private space [Berry, 2020], we used the scenario of bedrooms as a tool to help us learn about a user's privacy perceptions.

6.4.4 Organization of the Survey

The survey contains 51 single-choice and subjective (open-ended) questions. The survey was organized as follow:

- Consent form that describes who is collecting the data, why is it being collected and how it will be shared with other researchers.
- Demographic questions about age, gender, annual income, educational background, and familiarity with technology to learn how these factors play into privacy perceptions.
- We show users an image of a device, ask them what they think it is, how familiar they are with it. Then we ask them how comfortable they are if said device is installed in their bedrooms.
- In the next section, we show the users an image of a device and a snapshot of the data that it collects. We then ask them how comfortable they are if said device is installed in their bedrooms.
- In the next section, we show users images of various devices, and a set of inferences that can be made on the data collected by the device. We also describe a scneario where the

manufacturer of the device has promised not to share the raw data with any human, and only share certain inferences. We ask the users again about how comfortable they are if said device is installed in their bedrooms.

- We present the users with a hypothetical scenario in which we ask them to assume that they create their own camera from scratch and then ask them how comfortable they are if this camera is installed in their bedrooms.

6.4.5 Data Analysis

In this section, we describe our data analysis procedures, both quantitative and qualitative.

6.4.5.1 Quantitative Analysis

The data collected in the survey does not meet the assumptions of parametric tests – normal distribution and equal variance of errors. This is because the users have responded with extreme opinions (very uncomfortable or very comfortable) for several questions in the survey. We have one dependent variable – comfort level and several independent variables (physical appearance, data, inference, control, and familiarity).

To analyze our data, we choose non-parametric tests such as the Wilcoxon signed-rank test, Kolmogorov–Smirnov test and the Spearman correlation coefficient test with associated p-value. We use the Wilcoxon test because our study has a repeated-measure design. When having such a design, there might be dependencies among participants' responses to these repeated questions. hence we use the Wilcoxon signed-rank test which is a repeated measures test of dependency. It is performed between two paired groups and used to measure the whether the two related paired samples come from the same distribution. The null hypothesis for the Wilcoxon signed-rank test is *that two related paired samples come from the same distribution.* The Kolmogorov–Smirnov test is used to find similarity between two distributions. The null hypothesis of Kolmogorov–Smirnov test is *that the two distributions are* *similar*. Additionally, the statistic D of the Kolmogorov–Smirnov test is small (closer to 0) when the two distributions are similar and close to 1 when the two distributions are different. Suppose we have two distributions F and G. The Spearman correlation coefficient test with associated p-value is used to measure monotonicity of the relationship between two datasets. The p-value indicates *the probability of an uncorrelated system producing datasets that have a Spearman correlation at least as extreme as the one computed from these datasets.*[2]

In addition, we also use the mean μ, standard deviation σ and 95% confidence interval to describe the overall distribution of user opinions. The user opinions can range from a value of 1 to 5 where 1 means are very negative opinion and 5 means a very positive opinion.

6.4.5.2 Qualitative Analysis

The survey contains several questions where respondents were allowed to give open-ended answers. For each device shown to the respondents, we asked them to name what they think the device was. Additionally, we also asked the users two scenario based open-ended questions. These questions were:

- *OE1. Suppose you create your own camera from scratch. You control what the camera looks like, what parts go inside it, what data it collects, and where the data is stored. How comfortable would you be with that camera being placed in your bedroom given that no one else can see the data except you? Please explain the choice you made in the previous question briefly.*

- *OE2. Which of the following cameras will you prefer if you have to place one inside your home? [The choices were: (i) A camera from a well-known brand that does not sell ads or ad revenue is not a major revenue source for them, (ii) A camera from a company whose main source of income is ad revenue and sells your data to advertisers, (iii) A camera from a company based in the USA, (iv) A camera from a company based*

[2] *The definitions of the tests have been taken from their respective pages in the Scipy documentation.*

in a country that you do not trust]Please explain your answer in the previous question briefly.

In order to better understand how a user's privacy perceptions change when give more control over the data policies and the design of the device, we perform qualitative analysis on *OE1*. All the open-ended answers were coded in a bottom-up approach using inductive coding by two researchers. The researchers met frequently to discuss and iteratively code the answers.

6.4.6 Recruitment

We conducted our survey on Prolific [Palan and Schitter, 2018] [Now prolific.co] during June and July 2022. During the survey, we set a filter to only allow respondents in the US and 18 years or older to participate in the survey. The respondent matching was done by Prolific. Each user was able to participate in the survey once.

6.4.7 Compensation and Ethical Considerations

We recruited the participants through Prolific. The participants were free to choose to participate in the survey. Once the participants selected the survey, they were redirected to a Google form where they were shown a consent form describing the purpose of the study, as well as how the data will be collected and stored. After reading the consent form, the participants could back out of the survey if they chose to do so, without any penalty. After completing the survey, the participants downloaded a 'completion code' and entered it into their Prolific account to mark themselves as complete. Each participant was paid $2.75 for completing the survey. Based on the time taken by each respondent, that amounted to a rate of around $16.75/hour (based on calculations done by Prolific). The study was exempted by our institutions review board (IRB) as a part of the IRB approval process.

6.4.8 Data Release

The responses collected during the survey have been anonymized (the Prolific ID associated with each response has been deleted) and the responses are available publicly at: Link Hidden for Anonymity Purposes.

6.5 Findings: Quantitative Analysis

In this section, we describe the quantitative findings of our survey.

6.5.1 Age and Gender

A total of 162 respondents completed the survey on Prolific. Out of the 162 participants, 94 (58%) identified as male, 64 (39.5%) identified as female, 3 (1.85%) neither identified as male nor as female and 1 chose to not disclose their gender.

Amongst the 162 participants, 5 (3.1%) were less than 20 years of age (and 18 years or older), 110 (67.9%) were between 21 and 40 years of age, 39 (24.1%) were between 41 and 60 years of age and 8 (4.9%) were above 60 years of age.

6.5.2 Education and Technical Level

During the survey, we asked the respondents to self-identify their education level. We asked them to select the highest level of education that they have achieved. 52 (32.1%) of the total respondents' reported their highest level of education as high-school, 23 (14.2%) as Associates, 51 (31.5%) as Bachelors, 32 (19.8%) as Graduate and 4 (2.5%) as Professional.

In order to assess the technical savviness and familiarity of the respondents with sensing devices and the internet-of-things (IoT) in general, we used modified versions of some questions from the Mozilla's 'How connected are you?' survey [Mozilla,]. We asked the respondents: *How would you describe yourself when it comes to your knowledge of information* *technology?* The options were:

- *I am an expert: I build my own technical systems (e.g., computers), run my own servers, and code my own apps.*
- *I am technically savvy: I know my way around a computer pretty well. When anyone in my family needs technical help, I'm the one they call.*
- *I am an average user: I know enough to get by.*
- *I am a novice: Technology scares me! I only use it when I have to.*

Out of the total 122 respondents, 17 (10.5%) considered themselves experts, 94 (58%) considered themselves technically-savvy, 48 (29.6%) considered themselves average and 3 (1.9%) considered themselves as novice.

6.5.3 Data vs Inferences

In this section we explore what matters more to users regarding the privacy-invasiveness of a device/sensor – is it the intepretability of data representations, the possible inferences, or both? We ask the users three questions (listed in Section 6.4.2.5) to decouple the contribution of data and inferences in forming notions around privacy. In the first question, we show a picture of the device and ask the respondents how comfortable are they with the said device being placed in their bedrooms. This helps us determine any preconceived notions that the users may have about the privacy-sensitivity of a device based on its physical appearance. In the second question, along with the image of the device, we also show a sample snapshot of the data collected by the device. This helps us gauge how the intepretability of data representations affects a user's perception of privacy. Finally, in the third question, we show the respondents a set of inferences that can be derived from the data collected by the sensor. We also tell them that the data collected by the sensor will not be seen by another human being and instead be processed (via some algorithms) and then only the inferences will be shared. This helps us understand how the knowledge of inferences affects a user's perception of privacy. In the following subsections, we show the quantitative results for the three classes of sensing devices on the three questions listed above in Fig. 6.1.

6.5.3.1 Camera

We showed the users pictures of a camera (Fig. 6.1) and asked about their comfort level if it were to be placed in their bedroom. Since cameras are ubiquitous today, only 22 out of the 162 respondents said that they were either unfamiliar or very unfamiliar with it. Additionally, since cameras are considered to be privacy invasive, we found that people were not comfortable with installing it in their bedrooms Fig. 6.2. 124 out of the 162 total respondents ($\mu = 1.85$, $\sigma = 1.08$, 95% CI = [1.69, 2.03]) were either very uncomfortable or uncomfortable when just shown the image of the camera. Next, we showed the respondents an image of the same camera, but with a snapshot of a sample data (image) captured by it. We asked the users, the same question regarding their comfort level with installing this device in their bedrooms. 141 out of the 162 total respondents ($\mu = 1.51$, $\sigma = 0.90$, 95% CI = [1.37, 1.65]) were either very uncomfortable or uncomfortable, as shown in Fig. 6.2. We performed a Wilcoxon signed-rank test between the responses after looking at just the physical appearance and the responses after looking at both the physical appearance and the data collected by the camera and found that the difference was statistically significant ($V = 306.5$, $p < 0.0000054$) meaning that showing the collected data representation from the sensor caused a significant shift in their privacy perception. Next, we showed the users an image of the same camera with a list of inferences that can be made on the data collected by it (with the condition that humans won't be able to see the raw data). 137 out of the 162 total respondents ($\mu = 1.60$, $\sigma = 0.95$, 95% CI = [1.46, 1.75]) were either very uncomfortable or uncomfortable, as shown in Fig. 6.2. We performed a Wilcoxon signed-rank test between the responses after looking at just the physical appearance and the responses after looking at both the physical appearance and the list of inferences that can be made by the camera and found that the difference was statistically significant ($V = 700.5$, $p < 0.00082$). Additionally, we did not find a statistically significant difference between the responses after looking at the physical appearance of the camera with the data that it collects and the responses after looking at both the physical appearance and the list of inferences that can be made by the camera ($V = 270.5$, $p = 0.07$). To further analyze the relationship between the data and the inferences, we perform a Kolmogorov–Smirnov test between the two and find that the two distributions are similar ($D = 0.049$, $p = 0.9895$). This means that **for a camera, both the data representations and the inferences lead to similar privacy perceptions.** This can be attributed to the fact that since users can look at camera images and make their own inferences, both data and inferences contains similar amounts of human understandable information.

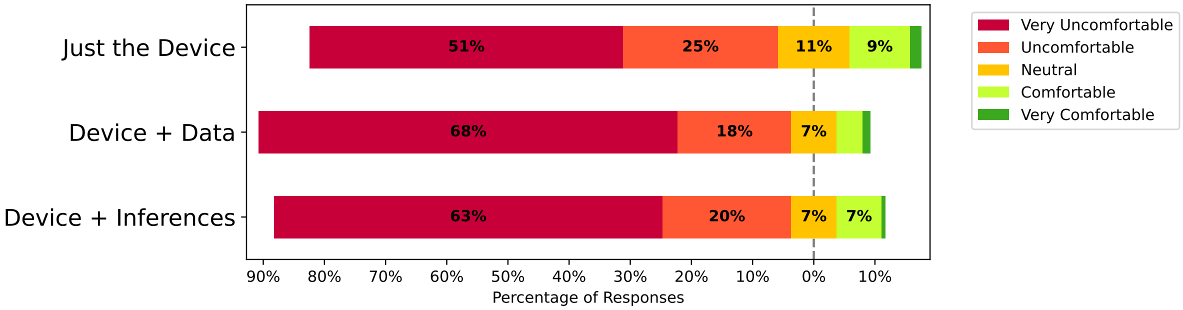


Figure 6.2: User comfort level when shown (a) an image of the camera, (b) an image of the camera and a snapshot of the data that it collects, (c) an image of the camera and list of inferences that can be made on the data that it collects (assuming that only inferences and not data are being shared)

6.5.3.2 mmWave Radar

We showed the users picture of an off-the-shelf mmWave radar (Fig. 6.1) and asked about their comfort level with placing it in their bedroom. mmWave radars are relatively new in the sensing world and hence, only 3 out of the 162 respondents said that they were familiar with it. We found that when shown an image of just the radar, users were not comfortable with installing it in their bedrooms Fig. 6.3. 88 out of the 162 total respondents ($\mu = 2.28$, $\sigma = 1.08$, 95% CI = [2.12, 2.45]) were either very uncomfortable or uncomfortable. Next, we showed the respondents an image of the same radar but with a snapshot of a sample data captured by it. We asked the users the same question regarding their comfort level with installing this device in their bedrooms. Since the data captured by the radar is not human interpretable, the number of users that were either very uncomfortable or uncomfortable decreased to only 71 out of the 162 total respondents (shown in Fig. 6.3) ($\mu = 2.67$, $\sigma = 1.09$, 95% CI = [2.50, 2.84]). We performed a Wilcoxon signed-rank test between the responses after looking at just the physical appearance and the responses after looking at both the physical appearance and the data collected by the mmWave radar and found that the difference was statistically significant ($V = 1018.5$, $p < 2.95 \times 10^{-5}$) meaning that showing users the data collected by the sensor caused a significant shift in their privacy perception. Next, we showed the users an image of the same mmWave radar with a list of inferences that can be made on the data collected by it (with the condition that humans won't be able to see the raw data). 117 out of the 162 total respondents ($\mu = 1.99$, $\sigma = 1.11$, 95% CI = [1.82, 2.16]) were either very uncomfortable or uncomfortable (shown in Fig. 6.3). We performed a Wilcoxon signed-rank test between the responses after looking at just the physical appearance and the responses after looking at both the physical appearance and the list of inferences that can be made by the mmWave radar and found that the difference was statistically significant ($V = 1589.5$, $p = 0.0023$). We also found a statistically significant difference between the responses after looking at the physical appearance of the mmWave radar with the data that it collects and the responses after looking at both the physical appearance and the list of inferences that can be made by the mmWave radar ($V = 596.0$, $p < 4.86 \times 10^{-11}$). To further analyze the relationship between the data and the inferences, we perform a Kolmogorov–Smirnov test between the two and find that their is no similarity between the two distributions ($D = 0.284$, $p < 3.70 \times 10^{-6}$). We conclude that **for a mmWave radar, since the data is not human understandable but inferences are,**

the data and the inferences lead to different privacy notions. In fact, when shown the inferences, the users' privacy perceptions become more negative. This shows that inferences lead to more informed privacy decisions by the users since mmWave radar data is not human interpretable, whereas inferences are. Additionally, since people are not familiar with mmWave sensors, their unfamiliarity initially dominates their decision making about placing this sensors in their bedrooms.

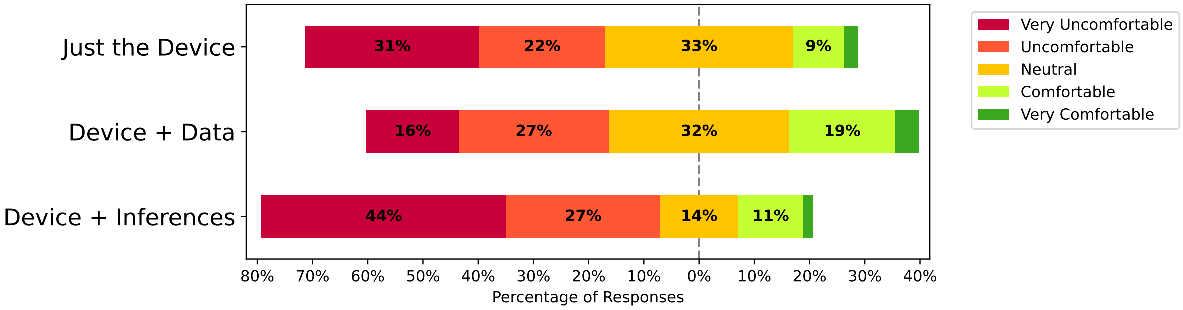


Figure 6.3: User comfort level when shown (a) an image of the mmWave radar, (b) an image of the mmWave radar and a snapshot of the data that it collects, (c) an image of the mmWave radar and list of inferences that can be made on the data that it collects (assuming that only inferences, not data, are being shared)

6.5.3.3 WiFi Router

We showed the users pictures of a generic WiFi router (Fig. 6.1) and asked about their comfort level with placing it in their bedroom. Since WiFi routers are ubiquitous today, only 8 out of the 122 respondents said that they were either unfamiliar or very unfamiliar with it. Additionally, since WiFi routers are considered to be innocuous, we found that people were very comfortable with installing it in their bedrooms (Fig. 6.4). 122 out of the 162 total respondents ($\mu = 3.97$, $\sigma = 1.11$, 95% CI = [3.80, 4.14]) were either very comfortable or comfortable when just shown the image of the WiFi router. Next, we showed the respondents an image of the same WiFi router but with a snapshot of a sample data (Wireshark snapshot) captured by it. We asked the users the same question regarding their comfort level with installing this device in their bedrooms. Since the Wireshark output is not interpretable by regular users, 100 out of the 162 total respondents were either very comfortable or comfortable (Fig. 6.4) ($\mu = 3.67$, $\sigma = 1.08$, 95% CI = [3.51, 3.84]). We performed a Wilcoxon signed-rank test between the responses after looking at just the physical appearance and the responses after looking at both the physical appearance and the data collected by the WiFi router and found that the difference was statistically significant ($V = 647.0$, $p < 3.83 \times 10^{-5}$) meaning that showing users the data collected by the sensor caused a significant shift in their privacy perception. Next, we showed the users an image of the same WiFi router with a list of inferences that can be made on the data collected by it (with the condition that humans won't be able to see the raw data). 89 out of the 162 total respondents were now either very uncomfortable or uncomfortable Fig. 6.4 ($\mu = 2.59$, $\sigma = 1.39$, 95% CI = [2.37, 2.80]). The total number of uncomfortable users went from 13 (when shown just the physical appearance of the WiFi router) initially to 61 (when also shown the list of inferences). We performed a Wilcoxon signed-rank test between the responses after looking at just the physical appearance and the responses after looking at both the physical appearance and the list of inferences that can be made by the WiFi router and found that the difference was statistically significant ($V = 271.0$, $p = 8.63 \times 10^{-19}$). We also found a statistically significant difference between the responses after looking at the physical appearance of the WiFi router with the data that it collects and the responses after looking at both the physical appearance and the list of inferences that can be made by the WiFi router ($V = 237.5$, $p < 2.26 \times 10^{-16}$). To further analyze the relationship between the data and the inferences, we perform a Kolmogorov–Smirnov test between the two and find no similarity between the two ($D = 0.410$, $p < 2.07 \times 10^{-12}$). We conclude that **showing inferences for a common home device like a WiFi router causes a very large negative shift in the privacy perception of that device.** This shift in privacy perception is different from the one which happens when users are shown the data representation, since the data collected is not human interpretable. Hence, **for sensors that collect non-human-interpretable data, it is**

the inferences, and not the data collected, that have a higher impact on how users perceive the privacy-sensitivity of a device.

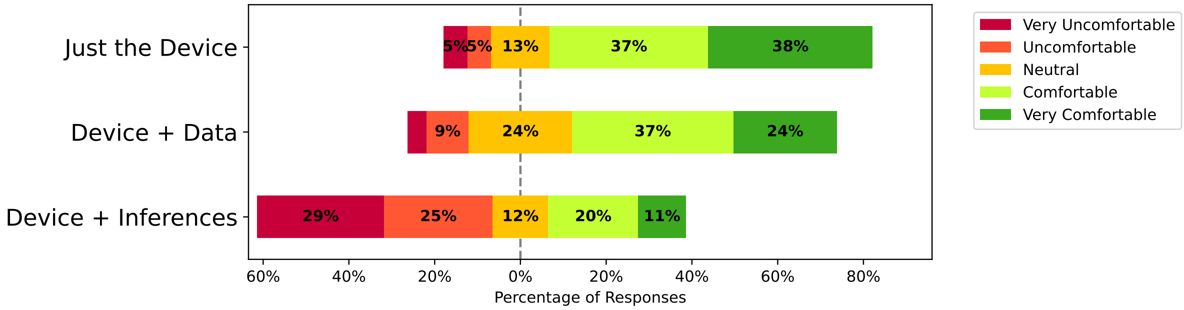


Figure 6.4: User comfort level when shown (a) an image of the WiFi router, (b) an image of the WiFi router and a snapshot of the data that it collects, (c) an image of the WiFi router and list of inferences that can be made on the data that it collects (assuming that only inferences and not data are being shared)

6.5.4 Physical Appearance of the Device and Familiarity

WiFi routers are one of the most common IoT devices in homes in the US [Associates,] – something that is also reflected in the responses to the question – *How familiar are you with the following device/sensor* [with an image of a generic WiFi router]. Hence, it is not surprising to see that most of the users thought of it as a benign device and were comfortable with installing it in their bedrooms. In order to understand how physical appearance affected perceptions of privacy, we found two WiFi router designs that look drastically different. One is an Asus Blue Cave Wifi router that looks like a camera from certain angles, and second is a Maurice Misho Radar Router Design that looks like a microphone. The responses are shown in Fig. 6.5. Initially, when shown the image of a generic WiFi router, only 18 (11.1%) of the total respondents (μ = 3.97, σ = 1.11, 95% CI = [3.80, 4.14]) were uncomfortable with installing it in their bedrooms. However, when shown a WiFi router that looks like a microphone, 83 (51.2%) (μ = 2.48, σ = 1.05, 95% CI = [2.31, 2.64]) were uncomfortable and when shown a WiFi router that looks like a camera, 83 (51.2%) out of the total 162 ($\mu = 2.49$, $\sigma = 1.31$, 95% CI = [2.29, 2.70]) respondents were uncomfortable. We performed a Wilcoxon signed-rank test between responses for a generic WiFi router and a WiFi router that looks like a microphone and found that the difference was statistically significant ($V = 140.5$, $p < 1.981 \times 10^{-21}$). Similarly, the difference between responses for a generic WiFi router and a WiFi router that looks like a camera was also statistically significant ($V = 227.5$, $p < 1.27 \times 10^{-18}$). We also performed a Wilcoxon signed-rank test between responses for a WiFi router that looks like a microphone and a WiFi router that looks like a camera but did not find a statistically significant difference ($V = 1629.5$, $p = 0.73$). Additionally, to further analyze the relationship between the responses for a WiFi router that looks like a microphone and a WiFi router that looks like a camera, we performed a Spearman Correlation Coefficient Test between the two and found a moderate Spearman correlation ($R = 0.47$, $p < 4.385 \times 10^{-10}$). Hence, we conclude that **when the data collected and inferences are not disclosed to users, the physical appearance of a device can be manipulated to influence the users' privacy perception of that device.**

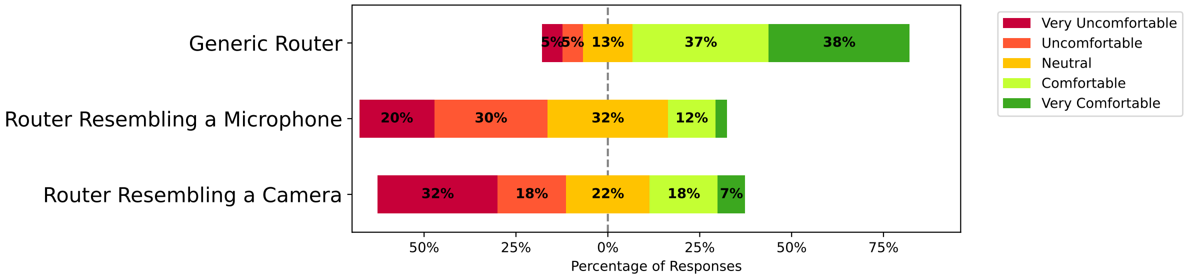


Figure 6.5: User comfort level when shown (a) an image of a generic WiFi router, (b) an image of a WiFi router that looks like a microphone, (c) an image of a WiFi router that looks like a camera.

6.5.5 Control

To find out how control over device configuration and data policies affects a user's perception on privacy, we presented the respondents with a scenario where they get to build a camera from scratch – they can control the physical appearance, hardware components, data collected and how the data is stored. We then asked the same question regarding their comfort levels with placing this camera in their bedrooms. The user responses are shown in Fig. 6.6. We see that initially, only 19 (11.7%) ($\mu = 1.85$, $\sigma = 1.08$, 95% CI = [1.69, 2.03]) out of the total 162 respondents were comfortable with placing a third party camera in their bedrooms, however, when given complete control over the device, 90 (55.6%) out of the total 162 respondents ($\mu = 3.28$, $\sigma = 1.42$, 95% CI = [3.06, 3.51]) are comfortable with placing this camera in their bedrooms. We performed a Wilcoxon signed-rank test between two sets of responses and found that the difference was statistically significant ($V = 284.0$, $p < 1.26 \times 10^{-18}$). Additionally, we also performed the Kolmogorov-Smirnov test between the two sets of responses and found that there indeed was a statistically significant difference between the two ($D = 0.438$, $p < 2.31 \times 10^{-14}$). Hence, we conclude **when given complete control over the device design and data policies, users are more likely to instrument their personal spaces with sensing devices.**

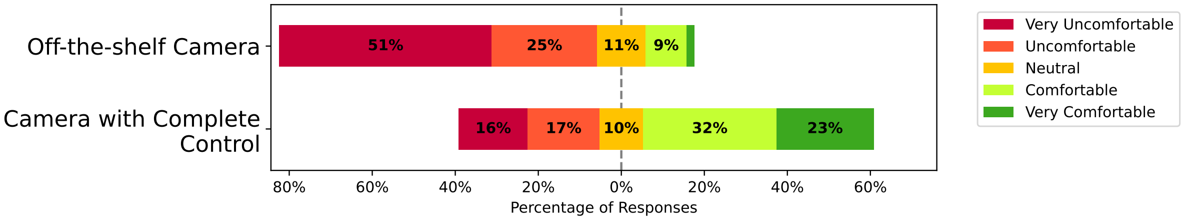


Figure 6.6: User comfort levels with installing a camera in their bedroom when given full control over it's design and data policies. Notice that more users are willing to put a camera in their bedrooms if given more control.

6.6 Findings: Qualitative Analysis

In this section, we analyze and group users' subjective responses to the following question in our survey, in order to better understand the factors that guide a user's privacy decision making process:

OE1. Suppose you create your own camera from scratch. You control what the camera looks like, what parts go inside it, what data it collects, and where the data is stored. How comfortable would you be with that camera being placed in your bedroom given that no one else can see the data except you? Please explain the choice you made in the previous question briefly.

We first categorized the sentiment of user responses as either positive or negative – positive if their response suggested that they were comfortable with the camera being placed in their bedroom and negative is they were not comfortable with the camera being placed in their bedroom. The sentiment was analyzed by the two researchers coding the qualitative responses using the open-ended answers.

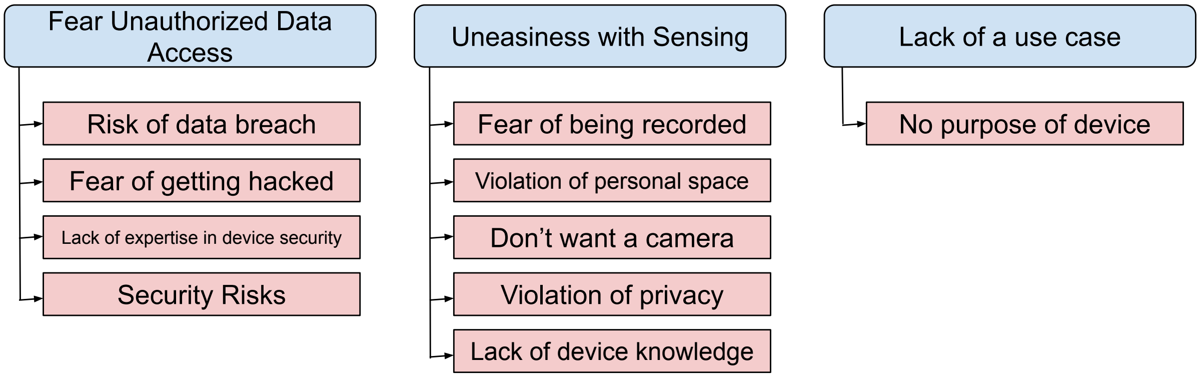


Figure 6.7: We employed inductive coding to code the open-ended responses. For the answers with a negative sentiment, the following themes (in blue) and codes (in red) emerged.

6.6.1 Why users don't want to place a camera in their bedroom even when given complete control over it's design and data policies?

From the responses with a negative sentiment, our inductive qualitative analysis method yielded ten codes. These codes can further be organized into three distinct themes. Fig. 6.7 shows these codes (red) and their associated themes (blue). We describe each theme in more detail below.

6.6.1.1 The Fear of Unauthorized Data Access

In our study, 43 (26.5%) out of the total 162 respondents expressed some concern regarding their data being accessed without their authorization. Not only is this the case when companies are storing their data but also for devices that they design and control themselves. This unauthorized access can include data breach (N=21), device being hacked (N=24), lack of a user's expertise with designing security focused device (N=3) and other security risks that modern internet-connected devices face (N=1)[3]. One respondent mentions: *All systems are hackable. I will never be "the only person" who has access to any digital input/output. Privacy does not exist in this world.* Other users said: *"Cameras are hacked all the time. If someone wants it, they will get it. If I made it myself I would be even more wary"*, *"Even though I know that no one else can see the data, I would irrationally believe that somebody eventually would see the data. So I would continue to be uncomfortable with the camera."*, and *"There is never a guarantee that no one else can see the data but me, that's a wishful fantasy, but if data exists its unlikely to stay with only one person"*. One user mentioned *"There's no such thing as a completely secure piece of technology if it can connect to the internet at all."*. This shows the despite given complete control over the design of the device and its data policies, some users still believe that their data can be accessed by others. In fact, some users are more concerned about privacy if they had created a device, since a manufacturer will be able to do a better job than them when it comes to protecting the data and implementing state-of-the-art security techniques.

6.6.1.2 Uneasiness with Sensing

Despite knowing about the inferences or given full control over the device some respondents were not comfortable with instrumenting their bedrooms with a camera. 30 (18.5%) out of the total 162 respondents were concerned with a sensor such as camera being placed

[3] multiple codes can be assigned to the same response.

in their bedrooms as they either found it “weird” or as something that will cause them paranoia. This uneasiness with sensing can be caused by a fear of being recorded (N=10), a perceived violation of personal space (N=8), a perceived violation of privacy (N=2), not wanting a camera for unspecified reasons (N=9), and a lack of device knowledge (N=2). Several respondents made comments such as: *“I would still feel as if I’m being watched”, “I don’t want a camera in my bedroom at all”*, *“I don’t want a camera in my bedroom even if no one else could see it”*, *“It would still feel like there is something watching me at all times...”*, *“I don’t like cameras in my bedroom it creates the wrong atmosphere and makes me uncomfortable knowing something is always recording”*, *“all this stuff is like spying to me”, “...a camera in general makes me feel like I’m on display”* and *“Even though no one else can access the data, I still feel uncomfortable having everything filmed in my bedroom. I don’t want to record my every movement and activity, even if no one else can view it.”* This shows that despite the best intentions of a manufacturer, there are users who don’t want their private spaces monitored. One needs to be mindful of such preferences when instrumenting spaces with sensors and devices.

6.6.1.3 Lack of a Use Case

In addition to above specified themes, we also found out that 3 (1.85%) users did not want to install any cameras in their bedroom as it served no purpose to them. These users wrote *“This type of camera does not serve a purpose to me”, “..I do not have a necessity..”, and “...I also have no reason to put a camera in my room.”* This shows that manufacturers should try to prove the utility of their devices to the users. Instrumenting a space with sensing devices without consulting the users about their needs and doing a use-case study should be avoided.

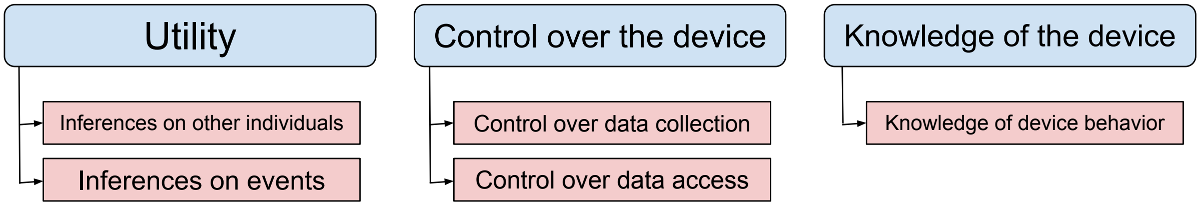


Figure 6.8: We employed inductive coding to code the open-ended responses. For the answers with a positive sentiment, the following themes (in blue) and codes (in red) emerged.

6.6.2 Why users are likely to place a camera in their bedroom even when given complete control over it's design and data policies?

In addition to analyzing why users don't want to instrument their personal spaces with a camera, we also analyzed the positive responses – reasons why users wanted to instrument their spaces with cameras. Our inductive qualitative analysis method yielded five codes. These codes can further be organized into three distinct themes. Fig. 6.8 shows these codes (red) and their associated themes (blue). We describe each theme in more detail below.

6.6.2.1 Utility

Some users are likely to instrument their spaces with cameras is due to the utility that is provides. In our survey, respondents stated *"If I lived with other people I might want to know if other people are going in my bedroom "* and *"i would let it only collect data if there was a break in"*. This shows that demonstrating utility is a possible method that can be used to improve the adoption rate of a particular device.

6.6.2.2 Control over the device

By far, a majority of users wanted to place a camera in their bedroom largely because they can control its data collection (N=19) and the access to collected data (N=40). Several respondents noted *"I control what it sees so it doesn't worry me"*, *"If I could control what it* *saw and collected I might be ok with it", "If I can control the flow of data, I am comfortable with placing the device in my bedroom", "If I had complete autonomy of what the data collects and how it is used, and I only have access to it, I would feel very comfortable because I would design it to suit my privacy needs and protect myself"*, and *"There is no chance that my personal data could be used in the wrong way if I have total control of how it's used."* These comments validate our hypothesis that a user's notions about the privacy sensitivity of a device depend upon the amount of control they have over it.

6.6.2.3 Knowledge of the device

Additionally, we also find that the more knowledge about how a device works that users have, the more likely they are to use it appropriately. In our survey, several (N=11) users said that since they knew how the device works, they are fine with placing it in their bedrooms. Some responses included *"If I know what it is doing at all times I wouldn't mind it being in my room", "I would know exactly what is collected and where the data is going"*, and *"Now that I know what it's used for, I'm not worried about other companies collecting my data"*.

6.7 Discussion

In this section, we discuss key findings of our work, how they can be used by the general community and some limitations of our study.

6.7.1 Key Findings

Our results show that perceptions of privacy do not solely depend upon the intepretability a sensor's data representations, but also on other factors such as possible inferences, physical appearance, familiarity and control. Our main findings are summarized below:

Data Representation vs Inference: When the data collected by the sensor is human interpretable, it does not matter which of the two are shown to the users as users can make inferences themselves by looking at the data. However, when data is not human intrepretable, we found that inferences have more affect on the overall privacy perceptions of the user. In case of both mmWave radar and WiFi router, showing inferences caused a larger negative shift in the overall privacy perception of the device than the data.

Physical Appearance and Familiarity: We found that users were less likely to install sensors in their homes that they are not familiar with. Additionally, when users look at a device, they try to find associations between the appearance of that device and any other device that they may have come across. This is evident when users see a router that looks like a camera and mistake it to be a camera and hence change their perception completely. When data and inferences are not disclosed, people use physical appearance and their sense of familiarity to make any judgements about the device.

Control: When given full control over the device design and data policies, users are more likely to install sensing devices in their private spaces and have an improved privacy perception of the device.

6.7.2 Limitations of our Study

This section discusses some limitations in our study that future research can address. In this study, we showed the users images of a sensor, the data collected, and a list of inferences that can be abstracted from the data. A more thorough in-person study can collect user data using a variety of sensors and show the collected data and the inferences to the users and then ask them about their privacy perception. We expect that when users see their own data and inferences instead of someone else's, it may have a higher impact on their opinions. This approach is analogous to ex-post transparency studies that visualize users' personal data disclosure [Murmann and Fischer-Hübner, 2017]. Moreover, future work can contextualize the inferences from sensor data in current data-sharing practices in commodity markets, e.g., understanding how user opinions change before and after they learn how companies might share these inferences with third parties [Farke et al., 2021]. Finally, our study only focuses on sensors in spaces with a high expectation of privacy, such as bedrooms. Future studies can also learn how users' privacy expectations change with the space where the sensors are present (e.g., bedroom vs. balcony).

6.7.3 Rethinking Privacy in Future IoT/Sensing Privacy Research

In this section, we discuss how researchers, manufacturers and users can use the implications of our study to guide future research, design, and instrumentation of sensing spaces.

Avoiding Assumptions about Sensor Privacy. In light of this study, we hope researchers will avoid using blanket statements about the privacy-sensitivity or privacy-invasiveness of devices such as RF sensors, without a user study. Users are key stakeholders in deciding the privacy viability of sensors and sensing paradigms, and they should have a say in what gets instrumented in their spaces.

Sensor Privacy in the ML world: Relying solely upon human interpretability of sensor data as a predictor for privacy-sensitivity of a device is misguided. We believe that laws and policies need to account for the recent advancements in machine learning in order to determine what type of data to collect from human subjects. Our recommendations are not just useful for RF sensors but can also be applied to any sensing modality.

Today, machine learning algorithms are used in all facets of our lives for various decision making tasks. In future research, instead of only asking *What inferences can a human make from this data?*, we need to ask *What inferences can a machine learning model abstract from this data?* Recent advances in visual-language intelligence that can convert one modality to another such as DALL-E [Ramesh et al., 2021] show that it is now possible to transform text to images. Since our study highlighted the impact of inferences from RF sensor data on user privacy notions, future researchers can extrapolate to scenarios where the embeddings in sensor data-based machine learning models will increasingly carry more semantic information that, in theory, can also be translated to imaging data understandable to humans. Thus, future research can understand the semantic information encoded by sensor data, which may enable future privacy-preserving techniques, e.g., as is done with privacy-preserving techniques for visual images [Kawamura et al., 2020].

Sensor Privacy Disclosure. There have been several attempts such as the nutrition label for privacy [Kelley et al., 2010] that aim to disclose information about what the sensor is, what type of data does it collect, why is the data being collected, or where the data is stored. When implemented, such approaches can help users learn more about a device and make a better decision about their comfort and privacy expectations with installing such a device in their spaces. However, in light of the key findings in our work, we believe that every manufacturer should also disclose three more things about a device to better guide this decision making process – (1) inferences that will be made on the data (2) inferences that can possibly be made on the data collected and (3) an example of some common devices and sensors that are similar to the device in question in terms of functionality. The manufacturers can collect a list of all possible inferences that can be made on the data collected by the sensor from research papers and then divide them into simpler categories. For example, radar sensors can: understand the shape and size of the scene, shape and size of objects and human subjects in the scene, activities being performed, gestures being made, speed of objects, vibrations in the surfaces (which can be used to recreate sound) and vital signs. Disclosures that focus on inferences and use easy to comprehend language for non-technical users need to be used in spaces where privacy needs to be prioritized.

Sensor Privacy Control: In order to make users feel more comfortable, device manufacturers and researchers can work together to create paradigms where users have more control over the design and placement of the device without sacrificing utility. In addition, data policies need to allow users the right to have their data deleted when they want to. Before using the data for training new models or for selling ads, express user consent needs to be sought. Finally, in some cases, control means the right to not be monitored at all. The idea that some users wish to remain anonymous and out of range of any sensing device needs to be respected as well.

CHAPTER 7

Conclusion and Future Work

In this thesis, we looked at how RF can be used independently and with other modalities such as vision and IMU to create a deeper and richer understanding of our surroundings.

In Chapter 2, we presented RadHAR [Singh et al., 2019] framework for HAR using the time window voxel representation of sparse mmWave radar point clouds. Our evaluation of the classifier shows that deep learning classifiers can be directly trained on the time window voxel representation and can achieve test accuracy greater than 90%. The classical machine learning approaches require domain-specific feature extraction and show poor performance on the voxels. Deep learning classifiers are able to learn the feature extraction transformation by directly training on the voxels. The classifiers which are designed to handle the spatial and temporal dependencies in data perform better than the fully connected deep learning classifier.

In Chapter 3, we presented a radar-camera fusion method to estimate dense depth [Singh et al., 2023]. Unlike depth completion with images and lidar points, radar-camera depth completion introduces a series of challenges largely due to the assumptions made while obtaining radar point clouds. These assumptions introduce ambiguity when projecting the point clouds onto the image plane. In this work, we addressed this challenge by proposing a two-step approach for obtaining dense depth via the fusion of a radar point cloud and an image. The method is motivated by our understanding of radar point cloud generation mechanics and is designed to correspond noisy and ambiguous radar points to image regions in a data-driven fashion. While we do not bar the case where correspondences are off i.e.

over-prediction, our experiments show that the proposed method achieves better results compared with other methods (i.e. 10.3% in mean absolute error (MAE) and by 9.1% in root-mean-square error (RMSE) improvement) of obtaining dense depth via radar-camera fusion.

In Chapter 4, we showed that self-supervised learning can be used to train radio models using a radio-visual co-learning approach, and without relying on laboriously labelled data. This technique can help enable next generation of applications that rely on training machine learning models over large-scale unlabeled RF data.

In Chapter 5, we presented SnoopDog [Singh et al., 2021], a framework to detect, identify, and localize Wi-Fi based sensors monitoring a person in an arbitrary space. SnoopDog works by establishing causality between a set of ground truth sensors monitoring a user and the transmitted information of wireless devices on a Wi-Fi network. It then uses this causality to perform trial-based localization. We implement SnoopDog on a set of commonly available devices such as a smartphone and a laptop and evaluate our solution on a set of representative clandestine sensors. The framework had a detection rate of 95.2% and a device classification rate of 100% when the injected multi-modal event was human motion or sound.

In Chapter 6, we conduct a user study to show that the idea that the privacy sensitivity or the privacy invasiveness of a device depends only upon the human interpretability of the data collected by that device is incorrect. Additionally, we show that a user's privacy perceptions regarding a device or a sensor depend upon a combination of multiple different factors such as the data collected by the device, the inferences that can be made of that data, the user's familiarity with the device and the amount of control that the user has over the design and data policies of the device. This is in contrast to existing notions of privacy which assume that human interpretability is primary in determining the privacy-invasiveness of a sensor. We hope that in light of the key findings in this paper, manufacturers will improve their disclosure process by adding the key factors highlighted in this paper to better guide decision-making when it comes to privacy.

Possible extensions of works presented in this thesis are discussed in the following subsections.

7.1 Human Activity Recognition using RF

Our method works over point clouds generated from a mmWave radar which are sparse and noisy. However, over time mmWave radars have improved and can now provide denser and more accurate measurements of the scene. Future research can study new methods that can leverage better-quality data to improve our performance.

Additionally, we studied a simple set of activities such as jumping and walking. In the real world, activities are complex and hence works need to study the performance of mmWave sensors in capturing complex activities in real-world settings.

Several research studies such as [Geng et al., 2022] have already shown superior performance using raw time series signals instead of point clouds. These studies can be built upon in the future for more novel applications.

7.2 Detecting and Localizing Hidden Sensors

While our work can detect and localize hidden wireless sensors, it requires a time-consuming and cumbersome process to do so. The amount of user effort required to search for hidden devices can be a barrier for a lot of people especially the ones with disabilities.

More work needs to be done in order to make these techniques accessible to end users in a user-friendly way.

7.3 Radar-Camera Fusion for Depth Estimation

While our method beats the current state-of-the-art techniques when it comes to estimating depth, its performance is still worse than that of a Lidar. As radars and deep learning techniques to process them improve, new methods can be developed to can achieve Lidar-like performance using a camera and a radar.

7.4 Measuring Privacy Impacts of Instrumenting Spaces with Sensors

During the course of our research, we found an additional factor that affects a user's privacy perception of a device or a sensor – Utility. There is a tradeoff between utility and privacy – when presented with a highly desirable use case, users are more willing to accept sensors that they would have otherwise rejected due to privacy concerns. Future research can study this tradeoff.

Additionally, while we study only the sensing aspect of the problem, real-world systems are often closed-loop – they provide some sort of output after ingesting the sensory data. Future research needs to study privacy perceptions of such closed-loop systems.

REFERENCES

[ama, a] Best sellers in home automation devices.

[ama, b] Best sellers in surveillance and security cameras.

[sol,] Monitor wi-fi traffic - wireless bandwidth monitoring tools.

[noa, a] Privacy - Labels - Apple.

[noa, b] PrivacyGrade.

[law,] Private space definition.

[noa, c] Welcome - PrivacyScore.

[noa, 2022] (2022). Get more information about your apps in Google Play.

[Abdelnasser et al., 2015] Abdelnasser, H., Harras, K. A., and Youssef, M. (2015). Ubibreathe: A ubiquitous non-invasive wifi-based breathing estimator. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 277–286.

[Afouras et al., 2020] Afouras, T., Owens, A., Chung, J. S., and Zisserman, A. (2020). Self-supervised learning of audio-visual objects from video. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 208–224. Springer.

[Ahmad et al., 2020] Ahmad, I., Farzan, R., Kapadia, A., and Lee, A. J. (2020). Tangible Privacy: Towards User-Centric Sensor Designs for Bystander Privacy. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):116:1–116:28.

[Akter et al., 2020] Akter, T., Dosono, B., Ahmed, T., Kapadia, A., and Semaan, B. (2020). "I am uncomfortable sharing what I can't see": Privacy Concerns of the Visually Impaired with Camera Based Assistive Applications. pages 1929–1948.

[Alloulah et al., 2021] Alloulah, M., Arnold, M., and Isopoussu, A. (2021). Deep inertial navigation using continuous domain adaptation and optimal transport. *arXiv:2106.15178.*

[Alloulah et al., 2022] Alloulah, M., Singh, A. D., and Arnold, M. (2022). Self-supervised radio-visual representation learning for 6g sensing. In *ICC 2022-IEEE International Conference on Communications*, pages 1955–1961. IEEE.

[Ann and Theng, 2014] Ann, O. C. and Theng, L. B. (2014). Human activity recognition: a review. In *2014 IEEE international conference on control system, computing and engineering (ICCSCE 2014)*, pages 389–393. IEEE.

[Apthorpe et al., 2018] Apthorpe, N., Huang, D. Y., Reisman, D., Narayanan, A., and Feamster, N. (2018). Keeping the smart home private with smart(er) iot traffic shaping.

[Arandjelovic and Zisserman, 2017] Arandjelovic, R. and Zisserman, A. (2017). Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617.

[Arandjelovic and Zisserman, 2018] Arandjelovic, R. and Zisserman, A. (2018). Objects that sound. In *Proceedings of the European conference on computer vision*, pages 435–451.

[Ashleibta et al., 2021] Ashleibta, A. M., Abbasi, Q. H., Shah, S. A., Khalid, M. A., AbuAli, N. A., and Imran, M. A. (2021). Non-Invasive RF Sensing for Detecting Breathing Abnormalities Using Software Defined Radios. *IEEE Sensors Journal*, 21(4):5111–5118. Conference Name: IEEE Sensors Journal.

[Associates,] Associates, P. Staking a claim in the connected home: Service provider solutions.

[Attal et al., 2015] Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., and Amirat, Y. (2015). Physical human activity recognition using wearable sensors. *Sensors*, 15(12):31314–31338.

[Avrahami et al., 2018] Avrahami, D., Patel, M., Yamaura, Y., and Kratz, S. (2018). Below the Surface: Unobtrusive Activity Recognition for Work Surfaces using RF-radar sensing. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 439–451, New York, NY, USA. Association for Computing Machinery.

[Aziz Shah et al., 2020] Aziz Shah, S., Ahmad, J., Tahir, A., Ahmed, F., Russell, G., Shah, S. Y., Buchanan, W. J., and Abbasi, Q. H. (2020). Privacy-Preserving Non-Wearable Occupancy Monitoring System Exploiting Wi-Fi Imaging for Next-Generation Body Centric Communication. *Micromachines*, 11(4):379.

[Banerjee et al., 2010] Banerjee, U., Vashishtha, A., and Saxena, M. (2010). Evaluation of the capabilities of wireshark as a tool for intrusion detection. *International Journal of computer applications*, 6(7):1–5.

[Baron and Musolesi, 2020] Baron, B. and Musolesi, M. (2020). Where you go matters: A study on the privacy implications of continuous location tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4):1–32.

[Beauregard and Haas, 2006] Beauregard, S. and Haas, H. (2006). Pedestrian dead reckoning: A basis for personal positioning. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, pages 27–35.

[Bengio et al., 2013] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

[Bergman et al., 2020] Bergman, A. W., Lindell, D. B., and Wetzstein, G. (2020). Deep adaptive lidar: End-to-end optimization of sampling and depth completion at low sampling rates. In *2020 IEEE International Conference on Computational Photography (ICCP)*.

[Berry, 2020] Berry, M. (2020). 5 reasons why your bedroom is a sacred place.

[Bilge and Dumitraş, 2012] Bilge, L. and Dumitraş, T. (2012). Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 833–844.

[Billah et al., 2021] Billah, M. F. R. M., Saoda, N., Gao, J., and Campbell, B. (2021). BLE Can See: A Reinforcement Learning Approach for RF-based Indoor Occupancy Detection. In *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021)*, pages 132–147, Nashville TN USA. ACM.

[Bonifield, 2019] Bonifield, J. (2019). Cameras secretly recorded women in california hospital delivery rooms.

[Bornstein and Craver-Lemley, 2016] Bornstein, R. F. and Craver-Lemley, C. (2016). Mere exposure effect. In *Cognitive illusions*, pages 266–285. Psychology Press.

[Bornstein and D'agostino, 1992] Bornstein, R. F. and D'agostino, P. R. (1992). Stimulus recognition and the mere exposure effect. *Journal of personality and social psychology*, 63(4):545.

[Brown and Muchira, 2004] Brown, M. and Muchira, R. (2004). Investigating the relationship between internet privacy concerns and online purchase behavior. *Journal of Electronic Commerce Research*, 5(1):62–70.

[Caesar et al., 2020] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631.

[Çağlıyan and Gürbüz, 2015] Çağlıyan, B. and Gürbüz, S. Z. (2015). Micro-doppler-based human activity classification using the mote-scale bumblebee radar. *IEEE Geoscience and Remote Sensing Letters*, 12(10):2135–2139.

[Caron et al., 2020] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*.

[Chauhan et al., 2020] Chauhan, S. S., Basu, A., Abegaonkar, M. P., Koul, S. K., et al. (2020). Through the wall human subject localization and respiration rate detection using multichannel doppler radar. *IEEE Sensors Journal*, 21(2):1510–1518.

[Chen et al., 2015] Chen, C., Jafari, R., and Kehtarnavaz, N. (2015). Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International conference on image processing (ICIP)*, pages 168–172. IEEE.

[Chen et al., 2021] Chen, H., Xie, W., Afouras, T., Nagrani, A., Vedaldi, A., and Zisserman, A. (2021). Localizing visual sounds the hard way. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16867–16876.

[Chen and Kim, 2013] Chen, H.-T. and Kim, Y. (2013). Problematic use of social network sites: The interactive relationship between gratifications sought and privacy concerns. *Cyberpsychology, Behavior, and Social Networking*, 16(11):806–812.

[Chen et al., 2020a] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

[Chen et al., 2020b] Chen, X., Fan, H., Girshick, R., and He, K. (2020b). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

[Chen et al., 2019] Chen, Y., Yang, B., Liang, M., and Urtasun, R. (2019). Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10023–10032.

[Chen et al., 2018] Chen, Z., Zhang, L., Jiang, C., Cao, Z., and Cui, W. (2018). Wifi csi based passive human activity recognition using attention based blstm. *IEEE Transactions on Mobile Computing*, 18(11):2714–2724.

[Cheng et al., 2020] Cheng, X., Wang, P., Guan, C., and Yang, R. (2020). Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10615–10622.

[Cheng et al., 2018] Cheng, Y., Ji, X., Lu, T., and Xu, W. (2018). Dewicam: Detecting hidden wireless cameras via smartphones. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 1–13. ACM.

[Cheng and Chang, 2017] Cheng, Y.-K. and Chang, R. Y. (2017). Device-free indoor people counting using wi-fi channel state information for internet of things. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE.

[Cheng et al., 2019] Cheng, Y.-T., Funk, M., Tsai, W.-C., and Chen, L.-L. (2019). Peekaboo Cam: Designing an Observational Camera for Home Ecologies Concerning Privacy. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, pages 823–836, San Diego CA USA. ACM.

[Choi and Christensen, 2010] Choi, C. and Christensen, H. I. (2010). Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation. In *2010 IEEE International Conference on Robotics and Automation*, pages 4048–4055. IEEE.

[Choi et al., 2017] Choi, J. W., Yim, D. H., and Cho, S. H. (2017). People counting based on an ir-uwb radar sensor. *IEEE Sensors Journal*, 17(17):5717–5727.

[Conference and Pierce, 2019] Conference, R. and Pierce, J. (2019). Leaky sensor fields: Deviating, accelerating, and restraining the smart home.

[Cranor, 2013] Cranor, L. F. (2013). Conceptions of privacy. In *Privacy Policy, Law, and Technology*. CMU Cylab.

[Creusen and Schoormans, 2005] Creusen, M. E. and Schoormans, J. P. (2005). The different roles of product appearance in consumer choice. *Journal of product innovation management*, 22(1):63–81.

[Crotti et al., 2007] Crotti, M., Dusi, M., Gringoli, F., and Salgarelli, L. (2007). Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Computer Communication Review*, 37(1):5–16.

[da Silva et al., 2020] da Silva, H. T. P., Duarte, R. M., de Alencar, M. S., and Queiroz, W. J. L. (2020). Cell-free at millimeter wave frequency simulation using the ray tracing method. In *2020 14th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5. IEEE.

[Degli-Esposti et al., 2014] Degli-Esposti, V., Fuschini, F., Vitucci, E. M., Barbiroli, M., Zoli, M., Tian, L., Yin, X., Dupleich, D. A., Müller, R., Schneider, C., et al. (2014). Ray-tracing-based mm-wave beamforming assessment. *IEEE Access*, 2:1314–1325.

[Demrozi et al., 2020] Demrozi, F., Pravadelli, G., Bihorac, A., and Rashidi, P. (2020). Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey. *IEEE Access*, 8:210816–210836.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Ding et al., 2011] Ding, D., Cooper, R. A., Pasquina, P. F., and Fici-Pasquina, L. (2011). Sensor technology for smart homes. *Maturitas*, 69(2):131–136.

[Ding et al., 2021] Ding, W., Guo, X., and Wang, G. (2021). Radar-based human activity recognition using hybrid neural network model with multidomain fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 57(5):2889–2898.

[Doersch et al., 2015] Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430.

[Eichelberger and McCartt, 2016] Eichelberger, A. H. and McCartt, A. T. (2016). Toyota drivers' experiences with dynamic radar cruise control, pre-collision system, and lane-keeping assist. *Journal of safety research*, 56:67–73.

[Emami-Naeini et al., 2020] Emami-Naeini, P., Agarwal, Y., Faith Cranor, L., and Hibshi, H. (2020). Ask the Experts: What Should Be on an IoT Privacy and Security Label? In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 447–464. ISSN: 2375-1207.

[Endsley, 2015] Endsley, M. R. (2015). Situation awareness misconceptions and misunderstandings. *Journal of Cognitive Engineering and Decision Making*, 9(1):4–32.

[Fairchild and Narayanan, 2016] Fairchild, D. P. and Narayanan, R. M. (2016). Multistatic micro-doppler radar for determining target orientation and activity classification. *IEEE Transactions on Aerospace and Electronic Systems*, 52(1):512–521.

[Fan et al.,] Fan, L., Li, T., Yuan, Y., and Katabi, D. In-Home Daily-Life Captioning Using Radio Signals. page 17.

[Fan et al., 2020] Fan, L., Li, T., Yuan, Y., and Katabi, D. (2020). In-Home Daily-Life Captioning Using Radio Signals. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*, pages 105–123, Berlin, Heidelberg. Springer-Verlag.

[Fang et al., 2007] Fang, X., Singh, S., and Ahluwalia, R. (2007). An examination of different explanations for the mere exposure effect. *Journal of consumer research*, 34(1):97–103.

[Farke et al., 2021] Farke, F. M., Balash, D. G., Golla, M., Dürmuth, M., and Aviv, A. J. (2021). Are privacy dashboards good for end users? evaluating user perceptions and reactions to google's my activity. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 483–500.

[Feng et al., 2020] Feng, L., Li, Z., Liu, C., Chen, X., Yin, X., and Fang, D. (2020). SitR: Sitting Posture Recognition Using RF Signals. *IEEE Internet of Things Journal*, 7(12):11492–11504. Conference Name: IEEE Internet of Things Journal.

[Fu et al., 2019] Fu, C., Mertz, C., and Dolan, J. M. (2019). Lidar and monocular camera fusion: On-road depth completion for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*.

[Fuschini et al., 2017] Fuschini, F., Häfner, S., Zoli, M., Müller, R., Vitucci, E., Dupleich, D., Barbiroli, M., Luo, J., Schulz, E., Degli-Esposti, V., et al. (2017). Analysis of in-room mm-wave propagation: Directional channel measurements and ray tracing simulations. *Journal of Infrared, Millimeter, and Terahertz Waves*, 38(6):727–744.

[Fussell, 2019] Fussell, S. (2019). Airbnb has a hidden-camera problem.

[Gao et al., 2010] Gao, K., Corbett, C., and Beyah, R. (2010). A passive approach to wireless device fingerprinting. In *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, pages 383–392. IEEE.

[Gao et al., 2020] Gao, Q., Tong, J., Wang, J., Ran, Z., and Pan, M. (2020). Device-free multi-person respiration monitoring using wifi. *IEEE Transactions on Vehicular Technology*, 69(11):14083–14087.

[Gasperini et al., 2021] Gasperini, S., Koch, P., Dallabetta, V., Navab, N., Busam, B., and Tombari, F. (2021). R4dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes. In *2021 International Conference on 3D Vision (3DV)*, pages 751–760. IEEE.

[Geng et al., 2022] Geng, J., Huang, D., and De la Torre, F. (2022). Densepose from wifi. *arXiv preprint arXiv:2301.00250*.

[Gidaris et al., 2018] Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

[Gochoo et al., 2018] Gochoo, M., Tan, T.-H., Velusamy, V., Liu, S.-H., Bayanduuren, D., and Huang, S.-C. (2018). Device-Free Non-Privacy Invasive Classification of Elderly Travel Patterns in a Smart House Using PIR Sensors and DCNN. *IEEE Sensors Journal*, 18(1):390–400. Conference Name: IEEE Sensors Journal.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

[Granger, 1969] Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438.

[Guan et al., 2009] Guan, P., Weiss, A., Balan, A. O., and Black, M. J. (2009). Estimating human shape and pose from a single image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1381–1388. IEEE.

[Gurbuz et al., 2021] Gurbuz, S. Z., Gurbuz, A. C., Malaia, E. A., Griffin, D. J., Crawford, C. S., Rahman, M. M., Kurtoglu, E., Aksu, R., Macks, T., and Mdrafi, R. (2021). American Sign Language Recognition Using RF Sensing. *IEEE Sensors Journal*, 21(3):3763–3775. Conference Name: IEEE Sensors Journal.

[Haartsen, 2003] Haartsen, J. C. (2003). Bluetooth radio system. *Wiley Encyclopedia of Telecommunications*.

[Haney et al., 2021] Haney, J., Acar, Y., and Furman, S. (2021). "It's the Company, the Government, You and I": User Perceptions of Responsibility for Smart Home Privacy and Security. pages 411–428.

[He et al., 2020] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Heater, 2019] Heater, B. (2019). Amazon upgrades its blink outdoor security camera with better battery, two-way talk – techcrunch.

[Henaff, 2020] Henaff, O. (2020). Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR.

[Hernandez-Quintanilla et al., 2021] Hernandez-Quintanilla, T., Magaña, E., Morató, D., and Izal, M. (2021). On the reduction of authoritative dns cache timeouts: Detection and implications for user privacy. *Journal of Network and Computer Applications*, 176:102941.

[Hochheiser, 2015] Hochheiser, M. (2015). The truth behind data collection and analysis. *J. Marshall J. Info. Tech. & Privacy L.*, 32:32.

[Hsiao et al., 2017] Hsiao, A.-Y., Yang, C.-F., Wang, T.-S., Lin, I., and Liao, W.-J. (2017). Ray tracing simulations for millimeter wave propagation in 5g wireless communications. In *2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting*, pages 1901–1902. IEEE.

[Hu et al., 2014] Hu, J., Lewis, F. L., Gan, O. P., Phua, G. H., and Aw, L. L. (2014). Discrete-event shop-floor monitoring system in rfid-enabled manufacturing. *IEEE Transactions on Industrial Electronics*, 61(12):7083–7091.

[Hu et al., 2021] Hu, M., Wang, S., Li, B., Ning, S., Fan, L., and Gong, X. (2021). Penet: Towards precise and efficient image guided depth completion. *arXiv preprint arXiv:2103.00783*.

[Hu et al., 2015] Hu, X., Song, L., Van Bruggen, D., and Striegel, A. (2015). Is there wifi yet?: How aggressive probe requests deteriorate energy and throughput. In *Proceedings of the 2015 Internet Measurement Conference*, pages 317–323. ACM.

[Huang et al., 2019a] Huang, D. Y., Apthorpe, N., Acar, G., Li, F., and Feamster, N. (2019a). Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale. *arXiv preprint arXiv:1909.09848*.

[Huang et al., 2021] Huang, Y., Wu, Q., Xu, J., Zhong, Y., and Zhang, Z. (2021). Clothing status awareness for long-term person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11895–11904.

[Huang et al., 2019b] Huang, Z., Fan, J., Cheng, S., Yi, S., Wang, X., and Li, H. (2019b). Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *IEEE Transactions on Image Processing*, 29:3429–3441.

[IGP, 2015] IGP, C. (2015). Protecting privacy in an iot-connected world. *Information Management*, 49(6):36.

[IIZUKA et al., 2003] IIZUKA, H., Watanabe, T., Sato, K., and NISHIKAWA, K. (2003). Millimeter-wave microstrip array antenna for automotive radars. *IEICE transactions on communications*, 86(9):2728–2738.

[INRIA,] INRIA. Scene understanding.

[Instruments, 2018] Instruments, T. (2018). Iwr1443boost evaluation module user‘s guide. *http://www.ti.com/lit/ug/swru518c/swru518c.pdf*. Accessed: 2019-07-05.

[Instruments, 2019] Instruments, T. (2019). Iwr1443 single-chip 76-ghz to 81-ghz mmwave sensor evaluation module iwr1443boost (active). Accessed: 2019-07-05.

[Isola et al., 2011] Isola, P., Parikh, D., Torralba, A., and Oliva, A. (2011). Understanding the intrinsic memorability of images. *Advances in neural information processing systems*, 24.

[Jaritz et al., 2018] Jaritz, M., De Charette, R., Wirbel, E., Perrotton, X., and Nashashibi, F. (2018). Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE.

[Jeong and Griffiths, 2019] Jeong, S. and Griffiths, J. (2019). Hundreds of south korean motel guests were secretly filmed and live-streamed online.

[Jiang et al., 2020] Jiang, W., Xue, H., Miao, C., Wang, S., Lin, S., Tian, C., Murali, S., Hu, H., Sun, Z., and Su, L. (2020). Towards 3d human pose construction using wifi. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14.

[Jin et al., 2022] Jin, H., Guo, B., Roychoudhury, R., Yao, Y., Kumar, S., Agarwal, Y., and Hong, J. I. (2022). Exploring the Needs of Users for Supporting Privacy-Protective

Behaviors in Smart Homes. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–19, New York, NY, USA. Association for Computing Machinery.

[Johnston, 1980] Johnston, S. L. (1980). Millimeter wave radar. *Dedham*.

[Jondhale et al., 2016] Jondhale, S., Deshpande, R., Walke, S., and Jondhale, A. (2016). Issues and challenges in rssi based target localization and tracking in wireless sensor networks. In *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pages 594–598. IEEE.

[Joshi et al., 2015] Joshi, A., Kale, S., Chandel, S., and Pal, D. K. (2015). Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396.

[Kawamura et al., 2020] Kawamura, A., Kinoshita, Y., Nakachi, T., Shiota, S., and Kiya, H. (2020). A privacy-preserving machine learning scheme using etc images. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 103(12):1571–1578.

[Kelley et al., 2010] Kelley, P. G., Cesca, L., Bresee, J., and Cranor, L. F. (2010). Standardizing privacy notices: an online study of the nutrition label approach. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, pages 1573–1582.

[Kennedy et al., 2019] Kennedy, S., Li, H., Wang, C., Liu, H., Wang, B., and Sun, W. (2019). I can hear your alexa: Voice command fingerprinting on smart home speakers. In *2019 IEEE Conference on Communications and Network Security (CNS)*, pages 232–240. IEEE.

[Kepuska and Bohouta, 2018] Kepuska, V. and Bohouta, G. (2018). Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 99–103. IEEE.

[Kezebou et al., 2020] Kezebou, L., Oludare, V., Panetta, K., and Agaian, S. (2020). Trgan: thermal to rgb face synthesis with generative adversarial network for cross-modal face recognition. In *Mobile Multimedia/Image Processing, Security, and Applications 2020*, volume 11399, pages 158–168. SPIE.

[Kim and He, 2015] Kim, S. M. and He, T. (2015). Freebee: Cross-technology communication via free side-channel. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 317–330.

[Kim and Ling, 2009] Kim, Y. and Ling, H. (2009). Human activity classification based on micro-doppler signatures using a support vector machine. *IEEE Transactions on Geoscience and Remote Sensing*, 47(5):1328–1337.

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

[Kinney et al., 2003] Kinney, P. et al. (2003). Zigbee technology: Wireless control that simply works. In *Communications design conference*, volume 2, pages 1–7.

[Koelle et al., 2018] Koelle, M., Wolf, K., and Boll, S. (2018). Beyond LED Status Lights - Design Requirements of Privacy Notices for Body-worn Cameras. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '18, pages 177–187, New York, NY, USA. Association for Computing Machinery.

[Kumar et al., 2019] Kumar, D., Shen, K., Case, B., Garg, D., Alperovich, G., Kuznetsov, D., Gupta, R., and Durumeric, Z. (2019). All things considered: an analysis of iot devices on home networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1169–1185.

[Labatie et al., 2021] Labatie, A., Masters, D., Eaton-Rosen, Z., and Luschi, C. (2021). Proxy-normalizing activations to match batch normalization while removing batch dependence. *arXiv:2106.03743.*

[Layne et al., 2012] Layne, R., Hospedales, T. M., Gong, S., and Mary, Q. (2012). Person re-identification by attributes. In *Bmvc*, volume 2, page 8.

[Lecci et al., 2020] Lecci, M., Testolina, P., Giordani, M., Polese, M., Ropitault, T., Gentile, C., Varshney, N., Bodi, A., and Zorzi, M. (2020). Simplified ray tracing for the millimeter wave channel: A performance evaluation. In *2020 Information Theory and Applications Workshop (ITA)*, pages 1–6. IEEE.

[LeCun, 2019] LeCun, Y. (2019). Deep learning hardware: Past, present, and future. IEEE International Solid-State Circuits Conference-(ISSCC).

[Lee et al., 2022] Lee, H., Kang, S., and Lee, U. (2022). Understanding privacy risks and perceived benefits in open dataset collection for mobile affective computing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–26.

[Lee and Kobsa, 2017] Lee, H. and Kobsa, A. (2017). Privacy preference modeling and prediction in a simulated campuswide iot environment. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 276–285. IEEE.

[Levi and Judd, 1996] Levi, R. W. and Judd, T. (1996). Dead reckoning navigational system using accelerometer to measure foot impacts. US Patent 5,583,776.

[Li et al., 2020a] Li, A., Yuan, Z., Ling, Y., Chi, W., Zhang, C., et al. (2020a). A multi-scale guided cascade hourglass network for depth completion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 32–40.

[Li et al., 2020b] Li, G., Zhang, Z., Yang, H., Pan, J., Chen, D., and Zhang, J. (2020b). Capturing human pose using mmwave radar. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–6. IEEE.

[Li et al., 2019a] Li, H., He, X., Chen, X., Fang, Y., and Fang, Q. (2019a). Wi-motion: A robust human activity recognition using wifi signals. *IEEE Access*, 7:153287–153299.

[Li et al., 2021] Li, M., Jiang, Z., Liu, Y., Chen, S., Wozniak, M., Scherer, R., Damasevicius, R., Wei, W., Li, Z., and Li, Z. (2021). Sitsen: Passive sitting posture sensing based on wireless devices. *International Journal of Distributed Sensor Networks*, 17(7):15501477211024846. Publisher: SAGE Publications.

[Li et al., 2019b] Li, X., He, Y., and Jing, X. (2019b). A survey of deep learning-based human activity recognition in radar. *Remote Sensing*, 11(9):1068.

[Li and He, 2017] Li, Z. and He, T. (2017). Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 2–14.

[Li et al., 2018] Li, Z., Xiao, Z., Zhu, Y., Pattarachanyakul, I., Zhao, B. Y., and Zheng, H. (2018). Adversarial localization against wireless cameras. In *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*, pages 87–92. ACM.

[Lin et al., 2020] Lin, J.-T., Dai, D., and Van Gool, L. (2020). Depth estimation from monocular images and sparse radar data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10233–10240. IEEE.

[Liu et al., 2019] Liu, C., Xiong, J., Cai, L., Feng, L., Chen, X., and Fang, D. (2019). Beyond Respiration: Contactless Sleep Sound-Activity Recognition Using RF Signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–22.

[Liu et al., 2018] Liu, T., Liu, Z., Huang, J., Tan, R., and Tan, Z. (2018). Detecting wireless spy cameras via stimulating and probing. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 243–255. ACM.

[Liu et al., 2021] Liu, W., Chang, S., Liu, Y., and Zhang, H. (2021). Wi-PSG: Detecting Rhythmic Movement Disorder Using COTS WiFi. *IEEE Internet of Things Journal*, 8(6):4681–4696. Conference Name: IEEE Internet of Things Journal.

[Lo and Vandewalle, 2021] Lo, C.-C. and Vandewalle, P. (2021). Depth estimation from monocular images and sparse radar using deep ordinal regression network. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3343–3347. IEEE.

[Long et al., 2021a] Long, Y., Morris, D., Liu, X., Castro, M., Chakravarty, P., and Narayanan, P. (2021a). Full-velocity radar returns by radar-camera fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16198–16207.

[Long et al., 2021b] Long, Y., Morris, D., Liu, X., Castro, M., Chakravarty, P., and Narayanan, P. (2021b). Radar-camera pixel depth association for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12507–12516.

[Luo et al., 2011] Luo, X., O'Brien, W. J., and Julien, C. L. (2011). Comparative evaluation of received signal-strength index (rssi) based indoor localization techniques for construction jobsites. *Advanced Engineering Informatics*, 25(2):355–363.

[Ma et al., 2019] Ma, F., Cavalheiro, G. V., and Karaman, S. (2019). Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*.

[Ma and Karaman, 2018] Ma, F. and Karaman, S. (2018). Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE.

[Maier et al., 2012] Maier, D., Hornung, A., and Bennewitz, M. (2012). Real-time navigation in 3d environments based on depth camera data. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 692–697. IEEE.

[Malkin et al., 2019] Malkin, N., Deatrick, J., Tong, A., Wijesekera, P., Egelman, S., and Wagner, D. (2019). Privacy attitudes of smart speaker users. *Proceedings on Privacy Enhancing Technologies*, 2019(4).

[Massaroni et al., 2018] Massaroni, C., Lopes, D. S., Lo Presti, D., Schena, E., and Silvestri, S. (2018). Contactless monitoring of breathing patterns and respiratory rate at the pit of the neck: A single camera approach. *Journal of Sensors*, 2018.

[Meidan et al., 2017] Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J. D., Ochoa, M., Tippenhauer, N. O., and Elovici, Y. (2017). Profiliot: a machine learning approach for iot device identification based on network traffic analysis. In *Proceedings of the symposium on applied computing*, pages 506–509. ACM.

[Miettinen et al., 2017] Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.-R., and Tarkoma, S. (2017). Iot sentinel: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2177–2184. IEEE.

[Mozilla,] Mozilla. Mozilla internet of things survey.

[Muller, 2001] Muller, N. J. (2001). *Bluetooth demystified*, volume 1. McGraw-Hill New York.

[Murmann and Fischer-Hübner, 2017] Murmann, P. and Fischer-Hübner, S. (2017). Tools for achieving usable ex post transparency: a survey. *IEEE Access*, 5:22965–22991.

[Naeini et al., 2017] Naeini, P. E., Bhagavatula, S., Habib, H., Degeling, M., Bauer, L., Cranor, L. F., and Sadeh, N. (2017). Privacy expectations and preferences in an {IoT} world. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 399–412.

[Nassi et al., 2019] Nassi, B., Ben-Netanel, R., Shamir, A., and Elovici, Y. (2019). Drones' cryptanalysis-smashing cryptography with a flicker. In *IEEE Symposium on Security and Privacy (SP), Vol. 00*, pages 833–850.

[Nbc, 2019] Nbc (2019). How to detect hidden cameras.

[Ni et al., 2011] Ni, B., Wang, G., and Moulin, P. (2011). Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1147–1153. IEEE.

[Nissenbaum, 2009] Nissenbaum, H. (2009). Privacy in context. In *Privacy in Context*. Stanford University Press.

[Nissim and Wood, 2018] Nissim, K. and Wood, A. (2018). Is privacy privacy? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2128):20170358.

[Noroozi and Favaro, 2016] Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer.

[NYU, 2021] NYU (2021). Self supervised learning in computer vision. `https://atcold.github.io/NYU-DLSP21/en/week10/10-1/`. Accessed: 24-10-21.

[Ohm, 2009] Ohm, P. (2009). The rise and fall of invasive isp surveillance. *U. Ill. L. Rev.*, page 1417.

[Ojeda and Borenstein, 2007] Ojeda, L. and Borenstein, J. (2007). Personal dead-reckoning system for gps-denied environments. In *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 1–6. IEEE.

[Oliveira and Zaiane, 2010] Oliveira, S. R. and Zaiane, O. R. (2010). Privacy preserving clustering by data transformation. *Journal of Information and Data Management*, 1(1):37–37.

[Oord et al., 2018] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[Orr et al., 2021] Orr, I., Cohen, M., Damari, H., Halachmi, M., and Zalevsky, Z. (2021). Coherent, super resolved radar beamforming using self-supervised learning. *arXiv preprint arXiv:2106.13085*.

[Ortiz et al., 2019] Ortiz, J., Crawford, C., and Le, F. (2019). Devicemien: network device behavior modeling for identifying unknown iot devices. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, pages 106–117. ACM.

[Palan and Schitter, 2018] Palan, S. and Schitter, C. (2018). Prolific. ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27.

[Park et al., 2020] Park, J., Joo, K., Hu, Z., Liu, C.-K., and Kweon, I.-S. (2020). Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision, ECCV 2020*. European Conference on Computer Vision.

[Patel,] Patel, N. Dead reckoning, a location tracking app for android smartphones.

[Poole et al., 2019] Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. (2019). On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR.

[Press, 2019] Press, A. (2019). Cops: Man secretly filmed dozens of women in changing room.

[Psychoula et al., 2018] Psychoula, I., Singh, D., Chen, L., Chen, F., Holzinger, A., and Ning, H. (2018). Users' privacy concerns in iot based applications. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1887–1894. IEEE.

[Qi et al., 2017] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660.

[Qiu et al., 2019] Qiu, J., Cui, Z., Zhang, Y., Zhang, X., Liu, S., Zeng, B., and Pollefeys, M. (2019). Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3313–3322.

[Raeis et al., 2021] Raeis, H., Kazemi, M., and Shirmohammadi, S. (2021). Human Activity Recognition with Device-Free Sensors for Well-Being Assessment in Smart Homes. *IEEE Instrumentation & Measurement Magazine*, 24(6):46–57. Conference Name: IEEE Instrumentation & Measurement Magazine.

[Raghunathan,] Raghunathan, R. Familiarity breeds enjoyment.

[Rahaman and Dyo, 2021] Rahaman, H. and Dyo, V. (2021). Tracking Human Motion Direction With Commodity Wireless Networks. *IEEE Sensors Journal*, 21(20):23344–23351. Conference Name: IEEE Sensors Journal.

[Raj et al., 2020] Raj, T., Hashim, F. H., Huddin, A. B., Ibrahim, M. F., and Hussain, A. (2020). A survey on lidar scanning mechanisms. *Electronics*, 9(5):741.

[Ramesh et al., 2021] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.

[Ren et al., 2022] Ren, Y., Wang, Z., Wang, Y., Tan, S., Chen, Y., and Yang, J. (2022). Gopose: 3d human pose estimation using wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–25.

[Rights (OCR), 2008] Rights (OCR), O. f. C. (2008). Summary of the HIPAA Privacy Rule. Last Modified: 2021-07-27T09:13:59-0400.

[Rostami12 et al.,] Rostami12, A., Vigren, M., Raza, S., and Brown23, B. Being hacked: Understanding victims' experiences of iot hacking.

[Saeed et al., 2022] Saeed, U., Yaseen Shah, S., Aziz Shah, S., Liu, H., Alhumaidi Alotaibi, A., Althobaiti, T., Ramzan, N., Ullah Jan, S., Ahmad, J., and Abbasi, Q. H. (2022). Multiple Participants' Discrete Activity Recognition in a Well-Controlled Environment Using Universal Software Radio Peripheral Wireless Sensing. *Sensors*, 22(3):809. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[Saleheen et al., 2021] Saleheen, N., Ullah, M. A., Chakraborty, S., Ones, D. S., Srivastava, M., and Kumar, S. (2021). Wristprint: Characterizing user re-identification risks from wrist-worn accelerometry data. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2807–2823.

[Sathyamoorthy et al., 2014] Sathyamoorthy, D., Jelas, M. J. M., and Shafii, S. (2014). Wireless spy devices: A review of technologies and detection methods. *EDITORIAL BOARD*, page 130.

[Schmitt et al., 2018] Schmitt, P., Bronzino, F., Teixeira, R., Chattopadhyay, T., and Feamster, N. (2018). Enhancing transparency: Internet video quality inference from network traffic.

[Schwartz, 2012] Schwartz, A. (2012). Chicago's video surveillance cameras: A pervasive and poorly regulated threat to our privacy. *Nw. J. Tech. & Intell. Prop.*, 11:ix.

[Scott, 2001] Scott, I. (2001). Development of a complete radar system model. In *Proceedings of the 2001 IEEE Radar Conference (Cat. No. 01CH37200)*, pages 35–40. IEEE.

[Sengupta et al., 2020] Sengupta, A., Jin, F., Zhang, R., and Cao, S. (2020). mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns. *IEEE Sensors Journal*, 20(17):10032–10044.

[Shen et al., 2017] Shen, C., Ho, B.-J., and Srivastava, M. (2017). Milift: Efficient smartwatch-based workout tracking using automatic segmentation. *IEEE Transactions on Mobile Computing*, 17(7):1609–1622.

[Shi et al., 2016] Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Singh et al., 2023] Singh, A. D., Ba, Y., Sarker, A., Zhang, H. C., Kadambi, A., Soatta, S., Srivastava, M., and Wong, A. (2023). Depth estimation from camera image and mmwave radar point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10.

[Singh et al., 2021] Singh, A. D., Garcia, L., Noor, J., and Srivastava, M. B. (2021). I always feel like somebody's sensing me! a framework to detect, identify, and localize clandestine wireless sensors. In *USENIX Security Symposium*, pages 1829–1846.

[Singh et al., 2018] Singh, A. D., Ram, S. S., and Vishwakarma, S. (2018). Simulation of the radar cross-section of dynamic human motions using virtual reality data and ray tracing. In *2018 IEEE Radar Conference (RadarConf18)*, pages 1555–1560. IEEE.

[Singh et al., 2019] Singh, A. D., Sandha, S. S., Garcia, L., and Srivastava, M. (2019). Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar. In *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*, pages 51–56.

[Solove, 2008] Solove, D. J. (2008). Understanding privacy.

[Southworth et al., 2007] Southworth, C., Finn, J., Dawson, S., Fraser, C., and Tucker, S. (2007). Intimate partner violence, technology, and stalking. *Violence against women*, 13(8):842–856.

[Staff, 2018a] Staff, I. E. (2018a). Couple says they found hidden camera pointing at their bed in carnival cruise room.

[Staff, 2018b] Staff, S. (2018b). Smart home devices market forecast to be growing globally at 31% annual clip.

[Stearns et al., 2018] Stearns, L., Findlater, L., and Froehlich, J. E. (2018). Applying transfer learning to recognize clothing patterns using a finger-mounted camera. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 349–351.

[Steele, 2019] Steele, A. (2019). Music revenue surges on streaming subscription growth.

[Sun et al., 2015] Sun, S.-T., Cuadros, A., and Beznosov, K. (2015). Android rooting: Methods, detection, and evasion. In *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages 3–14. ACM.

[Sun et al., 2017] Sun, X., Qiu, L., Wu, Y., and Cao, G. (2017). Actdetector: Detecting daily activities using smartwatches. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 1–9. IEEE.

[Sun et al., 2014] Sun, Y., Liu, M., and Meng, M. Q.-H. (2014). Wifi signal strength-based robot indoor localization. In *2014 IEEE International Conference on Information and Automation (ICIA)*, pages 250–256. IEEE.

[T. B. Brown et al., 2020] T. B. Brown et al. (2020). Language models are few-shot learners.

[Taylor, 2010] Taylor, E. (2010). I spy with my little eye: The use of cctv in schools and the impact on privacy. *The Sociological Review*, 58(3):381–405.

[Teixeira and Savvides, 2007] Teixeira, T. and Savvides, A. (2007). Lightweight people counting and localizing in indoor spaces using camera sensor nodes. In *2007 First ACM/IEEE International Conference on Distributed Smart Cameras*, pages 36–43. IEEE.

[Uhrig et al., 2017] Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., and Geiger, A. (2017). Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE.

[Vainshtein, 1988] Vainshtein, L. A. (1988). Electromagnetic waves. *Moscow Izdatel Radio Sviaz*.

[Valeros and Garcia, 2017] Valeros, V. and Garcia, S. (2017). Spy vs. spy: A modern study of microphone bugs operation and detection. Chaos Computer Club e.V. https://doi.org/10.5446/34936 $Lastaccessed : 26Nov$2019.

[Van der Maaten and Hinton, 2008] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

[Van Gansbeke et al., 2019] Van Gansbeke, W., Neven, D., De Brabandere, B., and Van Gool, L. (2019). Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th international conference on machine vision applications (MVA)*.

[Vanhoef et al., 2016] Vanhoef, M., Matte, C., Cunche, M., Cardoso, L. S., and Piessens, F. (2016). Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 413–424.

[Velykoivanenko et al., 2021] Velykoivanenko, L., Niksirat, K. S., Zufferey, N., Humbert, M., Huguenin, K., and Cherubini, M. (2021). Are those steps worth your privacy? fitness-tracker users' perceptions of privacy and utility. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–41.

[Wagner and Eckhoff, 2019] Wagner, I. and Eckhoff, D. (2019). Technical Privacy Metrics: a Systematic Survey. *ACM Computing Surveys*, 51(3):1–38. arXiv:1512.00327 [cs, math].

[Wampler et al., 2015] Wampler, C., Uluagac, S., and Beyah, R. (2015). Information leakage in encrypted ip video traffic. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE.

[Wang et al., 2021a] Wang, J., Ran, Z., Gao, Q., Ma, X., Pan, M., and Xue, K. (2021a). Multi-person device-free gesture recognition using mmWave signals. *China Communications*, 18(2):186–199. Conference Name: China Communications.

[Wang et al., 2018] Wang, T.-H., Wang, F.-E., Lin, J.-T., Tsai, Y.-H., Chiu, W.-C., and Sun, M. (2018). Plug-and-play: Improve depth estimation via sparse data propagation. *arXiv preprint arXiv:1812.08350*.

[Wang et al., 2017] Wang, W., Liu, A. X., Shahzad, M., Ling, K., and Lu, S. (2017). Device-free human activity recognition using commercial wifi devices. *IEEE Journal on Selected Areas in Communications*, 35(5):1118–1131.

[Wang, 2016] Wang, Y. (2016). Big opportunities and big concerns of big data in education. *TechTrends*, 60(4):381–384.

[Wang et al., 2021b] Wang, Y., Jiang, Z., Li, Y., Hwang, J.-N., Xing, G., and Liu, H. (2021b). Rodnet: A real-time radar object detection network cross-supervised by camera-radar fused object 3d localization. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):954–967.

[Wang et al., 2020a] Wang, Y., Wang, W., Zhou, M., Ren, A., and Tian, Z. (2020a). Remote monitoring of human vital signs based on 77-ghz mm-wave fmcw radar. *Sensors*, 20(10):2999.

[Wang et al., 2016a] Wang, Y., Wu, K., and Ni, L. M. (2016a). Wifall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing*, 16(2):581–594.

[Wang et al., 2016b] Wang, Y., Wu, X., and Hu, D. (2016b). Using randomized response for differential privacy preserving data collection. In *EDBT/ICDT Workshops*, volume 1558, pages 0090–6778.

[Wang et al., 2020b] Wang, Z., Chen, Z., Singh, A. D., Garcia, L., Luo, J., and Srivastava, M. B. (2020b). Uwhear: through-wall extraction and separation of audio vibrations using wireless signals. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 1–14.

[Weiß et al., 2020] Weiß, J., Pérez, R., and Biebl, E. (2020). Improved people counting algorithm for indoor environments using 60 ghz fmcw radar. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–6. IEEE.

[Weng et al., 2021] Weng, X., Man, Y., Park, J., Yuan, Y., O'Toole, M., and Kitani, K. M. (2021). All-in-one drive: A comprehensive perception dataset with high-density long-range point clouds.

[Wenger, 2003] Wenger, S. (2003). H. 264/avc over ip. *IEEE transactions on circuits and systems for video technology*, 13(7):645–656.

[Wild et al., 2021] Wild, T., Braun, V., and Viswanathan, H. (2021). Joint design of communication and sensing for beyond 5g and 6g systems. *IEEE Access*, 9:30845–30857.

[Wong et al., 2021] Wong, A., Cicek, S., and Soatto, S. (2021). Learning topology from synthetic data for unsupervised depth completion. *IEEE Robotics and Automation Letters*, 6(2).

[Wong et al., 2020] Wong, A., Fei, X., Tsuei, S., and Soatto, S. (2020). Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5(2).

[Wong and Soatto, 2021] Wong, A. and Soatto, S. (2021). Unsupervised depth completion with calibrated backprojection layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12747–12756.

[Wright et al., 2008] Wright, C. V., Ballard, L., Coull, S. E., Monrose, F., and Masson, G. M. (2008). Spot me if you can: Uncovering spoken phrases in encrypted voip conversations. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 35–49. IEEE.

[Wu et al., 2009] Wu, H., Tan, K., Liu, J., and Zhang, Y. (2009). Footprint: cellular assisted wi-fi ap discovery on mobile phones for energy saving. In *Proceedings of the 4th ACM international workshop on Experimental evaluation and characterization*, pages 67–76. ACM.

[Wu and Lagesse, 2019] Wu, K. and Lagesse, B. (2019). Do you see what i see?¡ subtitle¿ detecting hidden streaming cameras through similarity of simultaneous observation. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*, pages 1–10. IEEE.

[Xing et al., 2018] Xing, T., Sandha, S. S., Balaji, B., Chakraborty, S., and Srivastava, M. (2018). Enabling edge devices that learn from each other: Cross modal training for activity recognition. In *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*, pages 37–42. ACM.

[Xu et al., 2019] Xu, Y., Zhu, X., Shi, J., Zhang, G., Bao, H., and Li, H. (2019). Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.*

[Xue et al., 2017] Xue, W., Qiu, W., Hua, X., and Yu, K. (2017). Improved wi-fi rssi measurement for indoor localization. *IEEE Sensors Journal*, 17(7):2224–2230.

[Yang et al., 2020] Yang, B., Guo, R., Liang, M., Casas, S., and Urtasun, R. (2020). Radarnet: Exploiting radar for robust perception of dynamic objects. In *European Conference on Computer Vision*, pages 496–512. Springer.

[Yang et al., 2018] Yang, Y., Cao, J., Liu, X., and Liu, X. (2018). Wi-count: Passing people counting with cots wifi devices. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE.

[Yang et al., 2019] Yang, Y., Wong, A., and Soatto, S. (2019). Dense depth posterior (ddp) from single image and sparse range. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

[Yang et al., 2021] Yang, Z., Liu, X., Li, Z., Yuan, B., and Zhang, Y. (2021). RF-Eletter: A Cross-Domain English Letter Recognition System Based on RFID. *IEEE Access*, 9:155260–155273. Conference Name: IEEE Access.

[Yassein et al., 2016] Yassein, M. B., Mardini, W., and Khalil, A. (2016). Smart homes automation using z-wave protocol. In *2016 International Conference on Engineering & MIS (ICEMIS)*, pages 1–6. IEEE.

[Yin et al., 2022] Yin, Y., Yang, X., Xiong, J., Lee, S. I., Chen, P., and Niu, Q. (2022). Ubiquitous Smartphone-Based Respiration Sensing With Wi-Fi Signal. *IEEE Internet of Things Journal*, 9(2):1479–1490. Conference Name: IEEE Internet of Things Journal.

[Zeng et al., 2017] Zeng, E., Mare, S., and Roesner, F. (2017). End user security and privacy concerns with smart homes. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 65–80.

[Zeng et al., 2020] Zeng, Y., Wu, D., Xiong, J., Liu, J., Liu, Z., and Zhang, D. (2020). MultiSense: Enabling Multi-person Respiration Sensing with Commodity WiFi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):102:1–102:29.

[Zhan and Kuroda, 2014] Zhan, Y. and Kuroda, T. (2014). Wearable sensor-based human activity recognition from environmental background sounds. *Journal of Ambient Intelligence and Humanized Computing*, 5(1):77–89.

[Zhang et al., 2021] Zhang, J. A., Liu, F., Masouros, C., Heath, R. W., Feng, Z., Zheng, L., and Petropulu, A. (2021). An overview of signal processing techniques for joint communication and radar sensing. *IEEE Journal of Selected Topics in Signal Processing.*

[Zhang and Cao, 2018a] Zhang, R. and Cao, S. (2018a). Real-time human motion behavior detection via cnn using mmwave radar. *IEEE Sensors Letters*, 3(2):1–4.

[Zhang and Cao, 2018b] Zhang, R. and Cao, S. (2018b). Real-time human motion behavior detection via cnn using mmwave radar. *IEEE Sensors Letters*, 3(2):1–4.

[Zhang et al., 2016] Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer.

[Zhang and Funkhouser, 2018] Zhang, Y. and Funkhouser, T. (2018). Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185.

[Zhao et al., 2016] Zhao, M., Adib, F., and Katabi, D. (2016). Emotion recognition using wireless signals. In *Proceedings of the 22nd annual international conference on mobile computing and networking*, pages 95–108.

[Zhao et al., 2018] Zhao, M., Tian, Y., Zhao, H., Alsheikh, M. A., Li, T., Hristov, R., Kabelac, Z., Katabi, D., and Torralba, A. (2018). Rf-based 3d skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 267–281.

[Zhao et al., 2019] Zhao, P., Lu, C. X., Wang, J., Chen, C., Wang, W., Trigoni, N., and Markham, A. (2019). mid: Tracking and identifying people with millimeter wave radar. In *International Conference on Distributed Computing in Sensor Systems (DCOSS).*

[Zhao et al., 2021] Zhao, Y., Bai, L., Zhang, Z., and Huang, X. (2021). A surface geometry model for lidar depth completion. *IEEE Robotics and Automation Letters*, 6(3).

[Zheng et al., 2018] Zheng, S., Apthorpe, N., Chetty, M., and Feamster, N. (2018). User Perceptions of Smart Home IoT Privacy. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):200:1–200:20.

[Zhou and Piramuthu, 2015] Zhou, W. and Piramuthu, S. (2015). Information relevance model of customized privacy for iot. *Journal of business ethics*, 131(1):19–30.

[Zuo et al., 2019] Zuo, C., Wen, H., Lin, Z., and Zhang, Y. (2019). Automatic fingerprinting of vulnerable ble iot devices with static uuids from mobile apps. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1469–1483. ACM.