LOCATION ESTIMATION AND COLLECTIVE INFERENCE IN INDOOR SPACES USING SMARTPHONES

BY

SUBHAM DE

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Hari Sundaram, Chair
Professor Deepak Vasisht, Co-chair
Professor Chengxiang Zhai
Dr. Venkat Padmanabhan, Microsoft Research

ABSTRACT

In the last decade, indoor localization-based smart, innovative services have become very popular in public spaces (retail spaces, malls, museums, and warehouses). We have state-of-art RSSI techniques [1, 2, 3] to more accurate CSI techniques [4, 5, 6] to infer indoor location. Since the past year, the pandemic has raised an important challenge of determining if a pair of individuals are “social-distancing,” separated by more than 6ft. Most solutions have used ‘presence’—if one device can hear another— which is a poor proxy for distance since devices can be heard well beyond 6 ft social distancing radius and across aisles and walls. Here we ask the key question: what needs to be added to our current indoor localization solutions to deploy them towards scenarios like reliable contact tracing solutions easily. And we identified three main limitations—deployability, accuracy, and privacy. Location solutions need to deploy on ubiquitous devices like smartphones. They should be accurate under different environmental conditions. The solutions need to respect a person’s privacy settings. Our main contributions are twofold —a new statistical feature for localization, Packet Reception Probability (PRP) which correlates with distance and is different from other physical measures of distance like CSI or RSSI. PRP can easily deploy on smartphones (unlike CSI) and is more accurate than RSSI. Second, we develop a crowd tool to audit the level of location surveillance in space which is the first step towards achieving privacy.

Specifically, we first solve a location estimation problem in Chapter 3 with the help of infrastructure devices (mainly Bluetooth Low Energy or BLE devices). BLE has turned out to be a key contact tracing technology during the pandemic. We have identified three fundamental limitations with BLE RSSI—biased RSSI Estimates due to packet loss, mean RSSI de-correlated with distance due to high packet loss in BLE, and well-known multipath effects. We built the new localization feature, Packet Reception Probability (PRP), to solve the packet loss problem in RSSI. PRP measures the probability that a receiver successfully receives packets from the transmitter. We have shown through empirical experiments that PRP encodes distance. We also incorporated a new stack-based model of multipath in our framework. We have evaluated B-PRP in two real-world public places, an academic library setting and a real-world retail store. PRP gives significantly lower errors than RSSI. Fusion of PRP and RSSI further improves the overall localization accuracy over PRP.

Next, we solved a peer-to-peer distance estimation problem that uses minimal infrastructure in Chapter 4. Most apps [7, 8, 9] have solved peer-to-peer distances through the presence of Bluetooth Low-Energy (BLE) signals. Apps that rely on pairwise measurements like RSSI suffer from latent factors like device relative positioning on the human body, the orientation of the people carrying the devices, and the environmental multipath effect. We have proposed two solutions in this work—using known distances and collaboration to solve distances more robustly. First, if we have a few infrastructure devices installed at known locations in an environment, we can make more measurements with the devices. We can also use the known distances between the devices to constrain the unknown distances in a triangle inequality framework. Second, in an outdoor environment where we cannot install infrastructure devices, we can collaborate between people to jointly constrain many unknown distances.

Finally, we solve a collaborative tracking estimation problem in Chapter 5 where people audit the properties of localization infrastructure. While people want services, they do not want to be surveilled. Further, people using an indoor location system do not know the current surveillance level. The granularity of the location information that the system collects about people depends on the nature of the infrastructure. Our system, the CrowdEstimator, provides a tool to people to harness their collective power and collect traces for inferring the level of surveillance. We further propose the insight that surveillance is not a single number, instead of a spatial map. We introduce active learning algorithms to infer all parts of the spatial map with uniform accuracy. Auditing the location infrastructure is the first step towards achieving the bigger goal of declarative privacy, where a person can specify their comfortable level of surveillance.

This thesis is dedicated to my mom and dad who inspired me to pursue Ph.D., and to my wife Dipannita without whose unyielding support and encouragement, I would never have been able to complete this research.

ACKNOWLEDGMENTS

I want to thank those who have helped me grow as a researcher and person during my Ph.D. life.

First and foremost, I am indebted to Professor Hari Sundaram, my advisor, for shaping my identity as a researcher. Hari taught me the ABCs of research. He helped me get out of my undergraduate mindset of instantly looking for solutions to a research mindset of understanding the problem first in more detail, laying out all aspects and assumptions of the problem. His guidance during my initial Ph.D. years was invaluable in helping me understand what research is. Hari also taught me the art of presentation. He helped me overcome my fear of writing and guided me on approaching writing in a more structured way. His writing tips and tactics will help me throughout my life to write more fluently.

I am grateful to Professor Deepak Vasisht, my thesis co-advisor, for providing valuable feedback on systems research. He taught me how to focus on the right questions and highlight my contributions to the systems community. He also taught me different skills about writing a systems paper. He helped me to understand the structure of the introduction of a paper. I have been greatly inspired by Deepak's passion and energy towards research. I also appreciate Deepak's help in conducting experiments for contact tracing and location surveillance projects.

I would also like to thank other members of my dissertation committee—Professor Chengxiang Zhai and Doctor Venkat Padmanabhan, for their insightful comments on my research and accommodating a tight defense schedule. Cheng also taught me the course Information Retrieval during my second year of Ph.D., which developed my interest in the text mining area and gained knowledge about toping modeling and generative models in general, which I have applied in my research.

I want to thank my colleagues in the contact tracing project—Jay Shenoy, Bill Tao, Nisha, and Ruihua who helped build an end-to-end contact tracing application. I also appreciate Jay and Bill's extra efforts in collecting data for the projects. I would also like to thank Aniket Shirke, a summer intern in our research group at the time, who helped a lot in the initial round of testbed set-up and data collection for the location project. I would also like to thank my Crowd Dynamics Lab peers.

Finally, and more importantly, I would like to express my gratitude towards my family for their unwavering support and encouragement. My father, Sankar, has been a constant source of inspiration in my life. He never lost faith in me, even during the hard times in my career. I started my journey of mathematics and science under his tutelage, and I hope that I can always retain within me his passion and interest in solving math problems. My mother, Mitra, has encouraged me throughout my career. She always kept me motivated to not get fulfilled with current accomplishments and look towards a bigger goal. My loving wife, Dipannita, has been the constant source of a smile on my face. She has stood by me and been a witness of all the low and high times of my Ph.D. life. This thesis would not have been possible without them.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

A significant portion of the worldwide population uses smartphones for communication and information gathering while on the move. At the same time, lower costs of computing, storage, and communication are transforming physical spaces, with the increasing presence of Internet of Things (IoT). The combination of smartphones and increased IoT density is transforming how we imagine and use public spaces. For example, Amazon is experimenting with retail stores with no checkout lanes [10] so that customers can have an experience similar to online marketing. Stores [11] are using smart technology to better market to customers. Smart Museums [12] provide just-in-time information via audio tours, thereby enabling a more engaging experience with your favorite exhibits. Smart technologies can ensure asset tracking and security in the enterprises [13], thereby reducing lost assets which can be crucial for small businesses. IoT in airports like Miami International Airport can provide passengers with directions based on their location and help them to catch a connecting flight with a short layover or quickly find different services. Railway stations in India [14] and Hong Kong [15] have also deployed IoT systems to push notifications to travelers.

To truly transform the use of physical space through innovative services, accurate identification of the location of an individual is essential. In particular, we are interested in accurately identifying the location of individuals in indoor spaces, where noise in familiar Global Positioning System (GPS) signals complicates localization. Stores can figure out items we browsed based on our location inside the space and offer discount notifications through their app. Museums can use location information to figure out the exhibit we are standing in front of, tell us about its history, and recommend related exhibits. Enterprises can figure out the location of their assets through chips attached to them. Airports and railway stations can use our location information to suggest nearby restaurants, security checkpoints, and the shortest path to a gate.

Many state-of-the-art techniques [1, 4, 5, 6, 16, 17, 18] have been developed in the last two decades to find the location of people in indoor environments. Most works use signals (such as radio-frequency signals or light signals) exchanged with anchor nodes (known location) to infer the location of a target. They measure either distance or angle or both from these known anchor locations and use that to obtain the coordinate point of the target location. These works have used a wide range of technologies or anchor nodes for localization like WiFi access points [1, 16, 17], Bluetooth beacons [19, 20, 21], ultra-wide band (UWB) devices [22], RFID tags [23, 24, 25, 26], light emitters [27, 28, 29, 30] to name a few. The works have also used a wide range of features for localization starting from the most standard Received

Signal Strength Indicator or RSSI [1, 16, 17, 31] techniques to the more accurate Channel State Information or CSI [4, 5, 6, 32, 33] based techniques. While RSSI techniques give errors $2-3m$, CSI gives error in the order of decimeters.

Since the past year, the contact-tracing challenge due to the pandemic has put an urgent, renewed focus on developing robust indoor localization solutions. COVID-19 mainly spreads in indoor spaces due to close contact (closer than 6ft for over 15 minutes as per CDC guidelines [34]) between an infected person and another person. Due to latency in the appearance of symptoms in many infected people and the presence of potentially asymptomatic people, a healthy (but infected) person can end up spreading the virus to many vulnerable people in indoor environments. To stop the rapid spread of the virus, we need automated solutions that can send alerts to individuals who have been exposed to an infected person. The core requirement of such automated solutions is a robust distance estimation technique that can estimate when two people have been closer than 6ft to each other for a significant time.

In the past decade, we have developed a wide array of high-performing indoor localization solutions that range from using the signal strength of received signals [1, 2, 3] to the channel state information of signals [4, 5, 6]. However, it is pretty remarkable that despite two decades of advance in the indoor localization field, most contact tracing apps that were built last year like Aarogya Setu [1] in India and BlueTrace[2] (deployed in Singapore) uses the presence of signals (i.e., a smartphone that can hear another must be in proximity of the other.) to infer distance. Presence is coarse-grained information. Bluetooth Low Energy (BLE) signals transmitted by these apps can be heard long distances and beyond walls and obstacles. As a result, using only the presence of BLE signals as an indicator leads to many false positives, i.e., we detect faraway people to be nearby.

This thesis poses the key question— *what are the limitations in the existing localization solutions that impede their deployment towards scenarios like reliable contact tracing solutions?*. We have identified three main limitations—deployability, accuracy, and privacy.

- **Deployability:** For localization-based digital contact tracing to work reliably, we need all people nearby to have contact-tracing applications. To deploy our applications at scale to a wide range of people, we need to use smart devices that are ubiquitous with many people today. Smartphones are the only such devices today, and they contain sensors like WiFi and BLE, which enable localization. However, our current state-of-the-art accurate location finders [4, 5, 6] uses CSI or channel state information. To measure CSI, we require specialized hardware and firmware support that is not present in today's smartphones. A

[1] `https://www.mygov.in/aarogya-setu-app/`

[2] `https://bluetrace.io`

recent work [35] has enabled CSI for WiFi in some smartphones, but it cannot apply to other technologies like BLE. Smartphones can only receive signals or packets sent over WiFi/BLE and measure coarse aggregated information like signal strength or RSSI. *How can we estimate location without using non-deployable information like CSI and only using readily available aggregate values like RSSI?* Trying to use RSSI for localization leads us to our second limitation.

- **Accuracy:** For localization-based contact tracing to do a good job in identifying people in contact, we need our location estimators to be accurate. State-of-the-art RSSI based location estimators [1, 2, 17] are well known [2] to be less accurate (error over $2m$) and non-robust in their performance in indoor environments. They suffer from a wide range of effects like path-loss, fading, shadowing, and multi-path effects. For example, when a signal travels to a receiver, it does so through multiple paths and merges either in-phase (constructive) or out-of-phase (destructive). The merging that happens controls the received RSSI value, causing a huge variance for the same transmitter and receiver location. *How can we estimate location accurately using RSSI or other readily available information from smartphones?* Note that here we have a trade-off between deployability and accuracy. On one end, we have advanced technologies like CSI, which are very accurate but almost undeployable at scale. On the other end, we have RSSI, which is deployable but highly inaccurate. Can we find a middle ground where the system is still deployable and has better accuracy than RSSI (but lower accuracy than CSI)?

- **Privacy:** For our localization solutions to work reliably, we need more people to trust and adopt such technologies. For example, according to [36], at least more than 15% of a population needs to opt into a localization-based covid tracing system for it to have some decreasing impact on a community's covid case numbers, hospitalizations, and deaths. Other studies [37] claim even more higher numbers close to 60%. As found in [38], trust was an important factor affecting the adaption of contact tracing technologies. People were not confident that their location data would not serve corporations by building detailed behavioral profiles on them. For example, a person opting into a contact tracing system in an airport may not want the airport to learn which vending machine and what items they browsed at a certain point in time. We need to establish people or customers on the same level as these indoor location techniques to guide/provide input on their comfortable level of detection or surveillance. The person in the airport should be able to say—do not locate which item I am browsing in the vending machine; the system can locate the person's gate number at the airport for digital contact tracing.

In this thesis, we first address the problem of deployability and accuracy by designing a new feature—**Packet Reception Probability (PRP)** that can deploy on smartphones. Packet Reception Probability measures the probability that a receiver successfully receives packets from the transmitter. RSSI estimators are deployable on smartphones, but they have low accuracy. A significant challenge with RSSI estimators [1, 39] is the positive bias present in mean RSSI values due to loss of packets. We only include received packets in our aggregate RSSI estimates. We solve positive bias by designing this new feature PRP on negative information—the fraction of packets lost ($1-$ fraction of packets received). We also introduce a new stack-based model of multi-path. PRP only measures the number of packets received and hence can easily deploy on smartphones.

Second, now that we have a deployable and accurate localization system let us see how we can provide a level of anonymity at the same time. We address the question of declarative location privacy by designing a **crowdsourcing system to estimate the level of location tracking or space anonymity** in an environment. In order to achieve a declarative version of privacy, we first need to know the level of privacy exposure or the tracking that is happening in a public space. The tracking performance of a space depends on the nature of the infrastructure installed in that space. For example, an airport that has deployed 1000 devices/anchors in its space will have more data points and better localization than a railway station of a similar floor area that may have installed only 10 devices. We have designed a crowdsourcing system where each person collects raw data while interacting with a public space in their specific way and then contributes that data-trace to a central repository. We will combine and use the crowdsourced traces to infer the privacy exposure or tracking in space.

In this dissertation, we demonstrate the efficiency of our approaches in three different end-to-end systems—a location estimation problem with the help of infrastructure devices (mainly Bluetooth beacons), a peer-to-peer distance estimation problem that uses minimal infrastructure, and a collaborative tracking estimation problem where people estimate properties of localization infrastructure. We deploy these systems in real-world environments like libraries, retail spaces, and academic departments. We will present our deployment experiences and insights.

1.1 SYSTEMS DEVELOPED

In the following sections, we shall present the challenges and systems we developed to solve those challenges towards location estimation, distance estimation, and location surveillance estimation in public indoor spaces.

1.1.1 Location estimation using packet reception probability

During the COVID 19 pandemic, Technological solutions for contact tracing that use smartphones are an essential complement to normative (e.g., wearing a mask) and policy (e.g., stay-at-home) interventions for mitigating effects of the pandemic. Bluetooth Low-Energy (BLE) is key contact tracing technology used by apps like Aarogya Setu [7]. BLE is preferable to WiFi for contact-tracing: BLE uses 10× *less power* than does WiFi; BLE can be easily used to infer the presence of nearby peers without the presence of WiFi infrastructure. The newly proposed Exposure Notification Service by Apple-Google[3] also relies on BLE beacons and signal strength measurements.

Bluetooth Low Energy apps for contact tracing rely on RSSI (Received Signal Strength Indicator) since CSI for BLE is not available on smartphones. BLE RSSI localization suffers from three fundamental limitations. First, we get biased RSSI Estimates due to Packet Loss. Since devices only report RSSI for successfully decoded packets, RSSI-based distance methods suffer from a sampling bias: they only use RSSI from decoded packets. Second, Packet Losses are higher in low-power protocols like BLE, which makes sampling bias in RSSI measurements a more significant challenge for localization. Even at a few meter distances for BLE, the mean RSSI estimate becomes de-correlated with distance and is an unreliable indicator. Third, RSSI suffers from well-known multipath effects [3, 40] where signals travel along different paths and merge in different combinations at the receiver giving high variance in RSSI values for the exact location.

We built a new localization feature, Packet Reception Probability (PRP), which measures the probability that a receiver successfully receives packets from the transmitter. We have shown through empirical experiments that PRP encodes distance. Since PRP measures the fraction of packets received (1− fraction of packets lost), we can extract information from the absence of packets, thereby countering the positive bias present in RSSI values. We also incorporated a new model of multipath in our framework. We observed that many real-world public environments (retail stores, libraries, and warehouses) contain stack-like structures. Hence, we model public spaces, including retail stores, in a modular manner comprising open spaces separated by stacks. We explicitly capture the effect of such stacks by modeling the packet reception in the absence of stacks and the presence of one stack, two stacks, and so on. Our approach, Bayesian Packet Reception Probability (B-PRP), is a novel Bayesian framework that incorporates PRP and the new multipath model to deliver robust and accurate localization.

We evaluated B-PRP in two real-world public places, an academic library setting, and a

[3] `https://www.apple.com/covid19/contacttracing/`

real-world retail store. B-PRP achieves a median localization error of 1.03 m (library) and 1.45 m (retail store). The state of the art Bayesian RSSI system [39] has corresponding errors of 1.30 m (library, 26.2% more error) and 2.05 m (retail store, 41.3% more error). Fusion of B-PRP and RSSI further improves the overall localization accuracy over B-PRP.

1.1.2 Distance estimation using minimal infrastructure

During the COVID 19 pandemic, a significant problem is finding peer-to-peer distances between people in indoor public spaces. These distance estimations can alert people who came within 6ft (social distancing threshold) of an infected person. Most applications [7, 8, 9] have solved peer-to-peer distances through Bluetooth Low-Energy (BLE). The apps continuously broadcast BLE packets, while the receiving apps in the vicinity receive these packets and use specific properties of the received packets (RSSI in [7], presence in [8]) to measure distance.

BLE apps that rely on pairwise measurements like received signal strength to measure contact between two people have a well-known shortcoming. Latent factors like device relative positioning on the human body, the orientation of the people carrying the devices, and environmental multi-path effect can impact the measured power besides distance. We can receive the same signal strength value for two people standing 3ft apart facing away from each other and two people facing each other but standing more than 6ft apart. BLE apps that use presence suffer higher errors. Presence is a poor proxy for distance since devices can hear Bluetooth beacons beyond 6 ft social distancing radius and hear them across aisles and walls.

We have proposed two solutions in this work—using known distances and collaboration to solve the challenge of estimating distance solely from peer-to-peer measurements. First, if we have a few infrastructure devices (e.g., Bluetooth beacons) installed at known locations in an environment, we can make more measurements between a person's phone and the BLE beacons. Also, since we install the beacons at known locations, we know the distances between these beacons. We have used these known distances in a triangle inequality framework to estimate the unknown distance between two persons. Second, in an outdoor environment where we cannot install infrastructure devices, we can collaborate between people to solve distances more accurately. We can impose triangle inequalities between each triplet of people. More people help us form more of these triplets, leading to more constraints and, hence, a better localization solution.

We experimented with an infrastructure-based contact tracing solution in the library and retail store. We get median error of 0.89 m (library) and 1.07 m (retail store) with PRP values. The corresponding errors with RSSI are 1.36 m (library, 52.8% more error) and 1.34 m (retail store, 25.2% more error). Using the covid risk metric proposed in [41], we see that PRP does 1000X better than RSSI in the library. We have designed the architecture, protocol, packet structure, iOS, and Android apps to deploy the collaborative contact tracing solution. However, we learned from empirical experiments that there is high intra-device variance in RSSI/PRP values due to different smartphone antenna designs. Hence, we need future research to learn RSSI and PRP models that adapt/generalize to different devices.

1.1.3 Location surveillance estimation using crowdsourcing

Location services raise a fundamental problem—*Location surveillance or tracking.* While people want services, they do not want to be surveilled. Tracking can reveal much sensitive information about a person without taking their explicit consent. A location tracker in an airport can reveal products that we browsed in a vending machine. It can reveal people we interacted with within a mall using our location and the location of our acquaintances. The possibility of such finer level tracking raises doubts in the minds of people about adapting location services [38, 42].

People using an indoor location system do not know the granularity of the location information that the system collects about them. The performance of a location system depends on the nature of the infrastructure. For example, an airport may have better localization with a dense deployment of devices than a railway station with only a few devices. One may argue that an indoor location system can reveal the level of tracking that it does in a particular space to increase goodwill and trust. However, to solidify the claims of tracking level made by a system, we need to verify this information externally or using a third party. Here we ask the question—*Can we detect the level of location tracking by consuming only the raw data that the system collects from us?*

In this work, we build CrowdEstimator, a crowdsourcing system that harnesses the power of the crowd to audit the surveillance or level of tracking in space. Each person collects raw wireless data while interacting with a public space in their specific way and then contributes that trace to a central repository. We will use the crowdsourced traces to infer the surveillance in space. The system works under an assumption of data symmetry, i.e., our crowd system and the infrastructure can access the same wireless raw data on a person's phone. The infrastructure has access to additional information (such as the location of devices and fingerprinting maps) that our crowd system does not. The infrastructure uses this additional information for location finding.

This work also introduces the insight of location tracking or surveillance as an error radius that varies across space. This insight is a novel contribution to the best of our knowledge since previous works have only thought about tracking accuracy and tried to measure a single number for the entire space. The reality is that tracking granularity is spatial, and it depends on the nature of infrastructure in space. For example, we will have finer levels of granularity in places with a dense deployment of devices than places with lower device density. We claim that finding a single-valued metric of surveillance for a public space is insufficient. We propose finding a surveillance map or distribution over the space.

We need to estimate all portions of the surveillance map with a certain accuracy threshold. The challenge is that we need to collect a sufficient amount of data in all regions of space to ensure those accuracy thresholds. Some portions of public space are always more frequented than others, leading to increased accuracy in few regions followed by a high error in others. *Are there spots or regions inside a public space where data collection will lead to a maximal improvement of our overall estimate of the surveillance map?* We have designed active learning strategies to find such locations. We can then provide cues to people requesting them to collect data near that region. Our active learning strategies find regions of maximum uncertainty to collect data based on current surveillance maps and infrastructure properties estimates.

We did synthetic studies based on models trained by actual data collected from the library environment. Our studies showed that active learning strategies could more uniformly determine the location surveillance map over the entire space. With the best performing strategy, we were able to determine the tracking map with an accuracy of $0.4m$ and all the infrastructure device locations with an accuracy of less than $1m$.

1.2 CONTRIBUTIONS

In building these systems, this thesis makes the following core contributions:

- Packet Reception Probability (PRP) is a novel contribution in terms of signal measurement. It is a statistical measurement compared to CSI or RSSI, which are physical measurements of the signal. PRP uses negative information as a feature to solve the positive bias present in RSSI values and can achieve accurate localization using low-power BLE on easily deployable smartphones. PRP also incorporates a new model of multi-path for public indoor spaces.

- ContactTracer-our distance estimation system uses minimal infrastructure to solve peer-to-peer distances accurately in the presence of latent factors like orientation and phone location on the body. We use the known distances between infrastructure devices to constrain the unknown distances. Due to lack of infrastructure or known distances in outdoor environments, we propose using collaboration between people to solve distances robustly. We constrain many unknown distances at the same time through triangle inequalities.

- CrowdEstimator presents the first crowd tool to estimate the level of surveillance or tracking by infrastructure in a public indoor space. We also introduce the insight that tracking is not a single number, rather a spatial distribution. We combine wireless and inertial sensor data streams to audit the surveillance by a location infrastructure. We introduce active learning algorithms to estimate all parts of the spatial surveillance map with uniform accuracy.

1.3 THESIS OUTLINE

The rest of the thesis is organized as follows. In Chapter 2, we shall discuss the prior work in localization and privacy. In Chapter 3, we will describe our PRP-based localization work. Chapter 4 describes our contact tracing efforts. Chapter 5 gives details of the crowdsourcing efforts to find the surveillance or tracking map of a space. We conclude with a discussion of the lessons learned and a list of possible future directions in Chapter 6.

CHAPTER 2: LITERATURE REVIEW

We will discuss prior work in the indoor location estimation, contact tracing, and the privacy landscape. We will first lay out the different state-of-art features and technologies for indoor location finding. Then we will discuss the contact tracing applications that tried to solve the indoor proximity estimation problem during the COVID-19 pandemic and contrast those with the localization works. Next, we will talk about the importance of location privacy in localization and contact tracing works, focusing on declarative privacy. Finally, we will talk about crowdsourcing works that audit infrastructure properties. Auditing infrastructure properties is an important stepping stone towards achieving declarative privacy.

2.1 INDOOR LOCALIZATION

Indoor localization is a well-known problem that has attracted much work in the last two decades. Most works use signals (radio-frequency signals or light signals) exchanged with anchor nodes (known location) to infer the location of a target. We can classify indoor location works using two verticals—technology (anchor nodes or signals) used for localization and the features of signal communication used for localization. We will compare how the different technologies and features perform in the context of two essential localization properties that we outlined in Chapter 1—deployability and accuracy.

The most commonly used features for indoor localization are—RSSI or received signal strength indicator [1, 16, 31], AoA or angle of arrival [5, 32, 43], ToF or time-of-flight [33, 44, 45].

2.1.1 RSSI

RSSI stands for received signal strength indicator. We measure signal strength at the receiver and map that to distance through physical models of wireless transmission. If we know the distance from multiple devices, we can use trilateration to find the location. These are mainly two genre of RSSI techniques—model-based [16, 39, 46] and fingerprint-based [1], [17]. We discuss some state-of-the-art techniques below.

1. **Horus** [17] is an RSSI fingerprinting-based technique and stores radio maps in a probabilistic distribution fashion to localize. It involves a training phase when it builds the maps and a localization phase when it uses the maps to localize. The map consists of states or locations where we store an RSSI distribution for each access point. Horus uses

the nearest neighbor technique in the signal strength domain to find the nearest state or location during localization. The accuracy of Horus depends on the number of training states used and the distance between them.

2. **Bayesian RSSI** [39] uses the following generative model on RSSI loss to determine location.

$$\begin{aligned}
R\hat{S}SI_i &\sim N(w_{i0} + w_{i1}\log D_i, \tau_i), && (2.1)\\
w_{ij} &\sim N(w_j, \tau_{w_j})\, i \in \{1,\ldots,B\}, j \in \{0,1\}, && (2.2)\\
w_j &\sim N(\mu, \sigma)\, j \in \{0,1\}, && (2.3)\\
\tau_{w_j} &\sim \Gamma(\alpha, \beta)\, j \in \{0,1\} && (2.4)
\end{aligned}$$

   where $RSSI_i$ is the signal strength from the i-th beacon. We have B beacons in total. D_i is the distance of i-th beacon from the receiver. The coefficients for each beacon w_{i0} and w_{i1} are drawn hierarchically from the same normal distribution to capture similarity in beacon transmission properties. [39] set $\mu = 0$, $\sigma = 10$, $\alpha = 0.001$ and $\beta = 0.001$ so that priors allow for a wide range of values. $R\hat{S}SI_i$ is the averaged RSSI value over all $c_{i,t}$ packets received from beacon b_i in the current time interval t.

3. **Bayesian RSSI Fingerprinting** (or Bayesian FP) [47] is a Bayesian Fusion technique applied to a fingerprinting-based method for BLE devices. It stores fingerprints like Horus but employs a fusion technique to combine current RSSI and prior location information. It assumes that both follow a Gaussian distribution. It uses simple additive techniques to fuse these two distributions and gets a joint estimate for the reception location.

RSSI fingerprinting techniques come with a huge training overhead. Other methods [48, 49, 50, 51] reduce the overhead via crowdsourcing. [52] use a graph-based method for easy fingerprinting.

2.1.2 AoA

AoA stands for the angle of arrival information of the received signal. WiFi APs with multiple antennas can support MIMO communications. The basic idea of these systems is to calculate the AoAs of the multipath signals received at each AP, find the AoA of the direct path to the target, and then apply triangulation to localize. [4, 5, 32, 43] are well known AoA techniques which can achieve accuracy in the order of $0.4m$. Systems like ArrayTrack [5] are hard to deploy since they require hardware changes on access points by introducing as high as eight antennas. Ubicarse [32] combines AoA information with inertial sensor information like accelerometer and gyroscopes. However, Ubicarse requires a specific circular motion with the device to enable localization which may not be feasible in many localization scenarios. SpotFi [4] can deploy using commercially available access points that contain only three antennas. However, such AoA techniques require the smartphone to transmit WiFi signals which will consume energy actively.

2.1.3 ToF

ToF stands for time of flight information of the received signal. If we calculate the propagation delay (time) that a signal takes to traverse to the receiver, we can multiply that by the speed of light to obtain distance. [6, 33, 44, 45] are well known ToF techniques. Besides hardware/firmware modifications to overcome coarse ToF estimates, systems like Cupid [44], and Sail [33] can achieve localization error in the range of $2m$. Besides, these systems require users to walk around, perform measurements in multiple locations, or intersect those measurements with accelerometer readings to infer locations. Chronos [6] is a more accurate ToF system that enables a single WiFi access point to localize clients to within tens of centimeters.

2.1.4 RSSI vs CSI

AoA and ToF techniques are more broadly clustered as channel state information or **CSI** based techniques. While RSSI techniques measure coarse-grained higher-level information like signal strength, AoA and ToF based techniques have to look at the physical channel to estimate the direct path of signal transmission. Since a signal can traverse multiple paths to the receiver while measuring the angle or time (a proxy for distance) to the transmitter, we need to find the direct path. A reflected path will give us the angle of the reflector and will include time from the transmitter to the reflector and then the reflector to the receiver. In order to filter out the direct path, CSI-based techniques need to look at the physical channel and hence require specialized hardware.

RSSI-based techniques are less accurate than CSI-based techniques. CSI [4, 6] can give decimeter level accuracy. RSSI give errors over $2m$. RSSI is well-known [2] to suffer from path-loss, fading, shadowing, and multi-path effects. When a signal travels to a receiver, it does so through multiple paths and merges either in-phase (constructive) or out-of-phase (destructive). The merging depends on the environment and not distance. It controls the received RSSI values.

CSI-based techniques are not deployable on most smartphones. To get CSI accuracy, we need specialized hardware to monitor the physical channel, which is not available on most smartphones today. A recent work [35] has enabled CSI for WiFi in some smartphones. However, CSI is still available on limited smartphones today, and we cannot use it for an application like location-based contact tracing, which requires wide-scale adaption.

2.1.5 Technologies for localization

The most commonly use technologies for localization are WiFi access points [1] and Bluetooth beacons [19, 20, 21]. Other technologies or anchor nodes used for localization are—FM radios [53], Zigbee devices [54], ultra-wide band (UWB) devices [22], RFID tags [23, 24, 25, 26], ultra-sound emitters [55], light emitters [27, 28, 29, 30],60GHz devices [56, 57], sub-centimeter sized devices [58]. [22] has shown the promise of low-cost UWB sensing. However, these technologies are currently not as ubiquitous as commodity WiFi AP infrastructure or BLE technology used for peer-to-peerc onnection on smartphones.

WiFi and BLE technology are available on most smartphones today, while UWB, RFID is yet to be adapted on a large scale. We require peer-peer, low power measurements. Can WiFi accommodate? BLE is well documented in literature [59] to consume 10X less power than WiFi. If we want localization solutions to run 24×7 on a person's phone, BLE is a preferred technology. Also, BLE allows for direct peer-to-peer communication between two smartphones without requiring any infrastructure (like WiFi routers). BLE can enable peer-to-peer distance estimation in outdoor environments where we cannot install extra infrastructure. Finally, BLE has been adapted as a key technology by a myriad of contact tracing apps [7, 8] during the COVID-19 pandemic.

In this thesis, we have focused on developing a technique that can accurately localize using features of low power BLE that are available on smartphones. CSI with BLE is not available on smartphones yet. RSSI with BLE is wildly inaccurate and a non-reliable signal for distance estimation. Due to the low power of BLE signals, RSSI values even at smaller distances accumulate noise and become decorrelated with distance. This thesis investigates the challenges with BLE RSSI and proposes a new measurement PRP that covaries with distance and is different from CSI or signal strength.

2.2 CONTACT TRACING INFRASTRUCTURE

Contact tracing is a significant problem to solve during the COVID 19 pandemic. If we can automatically infer distance between two people, we can alert people who came within

6ft (social distancing threshold) of an infected person. Automation can help us solve this problem at scale. [60, 61] shows the applications have helped curb the spread of the disease. We will discuss the different contact tracing efforts [62] and how each of these efforts tried to solve distance. We will compare how the distance solutions position in the context of the indoor localization algorithms developed over two decades.

The contact tracing apps fall into four broad categories based on the technology they use to find the proximity of two persons.

1. **GPS/Cellular**: The apps like ViruSafe [63], Rakning c-19 [64] use GPS or triangulation from nearby cell towers to find the location of a phone on a person. If two phones have been in the exact location simultaneously, the apps infer the two people to be in proximity. Applications relying on GPS or cellular data are less accurate than Bluetooth-based technologies since they can infer far away people to be in proximity. They are more privacy-invasive than Bluetooth since they can find the absolute location of people.

2. **Bluetooth**: The apps like Virusradar [65], Covidradar [66] use Bluetooth in phones to swap encrypted tokens with any other nearby phones. If a phone receives the Bluetooth token from another phone, then the two persons owning the phones are inferred to be close. Apps relying on Bluetooth perform well than GPS on privacy since tokens can be anonymized. However, Bluetooth packets can reach as far as $30ft$ and detect faraway people in proximity.

3. **Apple/Google Exposure**: Apps like Stopp-corona [67] in Austria, Covid-alert [68] in Canada, GuideSafe [69] in Alabama USA, and CovidDefense [70] in Louisana rely on the joint API developed by Apple and Google. The API allows iOS and Android phones to communicate with each other over Bluetooth.

4. **DP-3T**: Apps like SwissCovid [71], Coronalert [72] use decentralized privacy-preserving proximity tracing protocol over Bluetooth. It is an open-source protocol in which an individual phone's contact logs are stored locally, so no central authority can know who has been exposed.

The main limitation of the above apps is that Bluetooth Low Energy (BLE) signals can be heard at large distances and beyond walls. Using only presence as a feature leads to many false positives, i.e., we detect far away persons to be in proximity. It is pretty remarkable that despite the different features like RSSI [1], CSI [4, 6] developed in the last two decades, none of the contact tracing applications incorporated BLE RSSI or CSI.

BLE CSI is not available on smartphones. BLE RSSI is noisy due to multi-path and other latent factors like person orientation, phone location on a person's body besides distance. In this thesis, we propose two solutions to the problem— the help of some minimal infrastructure in the environment to reliably infer distance or collaboration among people and impose triangle inequalities to solve distances. The main intuition is to use some known distances to constrain the unknown distances or solve many unknown distances together, which can constrain each other through triangle inequalities.

2.3 LOCATION SURVEILLANCE

To enable wide-scale adaption of location technologies, people need to trust that such solutions will not leak their private information. In a Microsoft research study about contact tracing apps [42], privacy concerns were almost as high on people's minds as accuracy. 73% of people said they would only install if the app is private, which is close to the 85% who demanded accuracy. People want the service, but they do not want to be surveilled. Here, we will discuss a few privacy-preserving algorithms in indoor localization space and show how our work is different. Then, we show that our work is similar to auditing the location tracking infrastructure and outline two important previous works. Finally, we will discuss some active learning tools that we have utilized to better inference of location surveillance.

2.3.1 Privacy preserving indoor location tracking

There are many proposed solutions for Privacy-Preserving Indoor Localization (PPIL) [73] using Homomorphic Encryption (HE) such as the Paillier cryptosystem [74] or k-Nearest Neighbors (k-NN) algorithm [75]. [73] analyzed the privacy issues of WiFi fingerprint-based localization system and then used the Paillier cryptosystem to protect both the client's location privacy and the service provider's fingerprint data privacy. [75] enables a user to localize privately through an Indoor Positioning System by making k camouflaged localization requests. They design a Temporal Vector Map (TVM) algorithm, which guarantees that the IPS system cannot find a user's location with a probability higher than a user-defined preference. These works are specific to WiFi fingerprinting systems where the localization space is segmented into N discrete points. In our work, we are looking at a continuous localization space where instead of k-anonymity, a user wants to achieve α-anonymity, i.e., do not localize the user to a circle of radius smaller than α. Besides, we also assume that the user does not have access to any system-side auxiliary information such as fingerprinting maps and access point locations. The current granularity of location inference is unknown on the user side. To achieve anonymity, we first need to know the level of location inference or surveillance that depends on the infrastructure installed in that space. This work focuses on inferring the infrastructure properties from the data collected on the user side, which is the first step towards achieving α-anonymity.

2.3.2 Auditing of location tracking systems

The previous section highlighted that we want to estimate the granularity of location estimation in a place. To estimate location granularity, we need to audit the location infrastructure that currently exists in that place. Specifically for RSSI-based location systems, we need to find information about the number of access points and their locations. For CSI-based systems, we will need to find extra information like the number and orientation of antennas. We looked at previous systems like EZ [16], LocBLE [76] and Zee [48] which inferred properties of the infrastructure.

1. **EZ** [16] finds the location of WiFi routers as an intermediate step to ultimately finding the location of people. It uses WiFi RSSI data collected from people, and a few GPS annotated locations to find the location of WiFi routers. The key idea is that the physics of wireless propagation constrains all data points from many unknown locations. EZ models these constraints and then uses a genetic algorithm to solve the WiFi router locations. EZ then uses the annotated WiFi routers to solve the location of people.

2. **LocBLE** [76] finds and tracks nearby BLE beacons. It combines RSSI datastream from BLE and inertial sensor datastream from the accelerometer and gyroscopes to obtain a more accurate location of BLE beacons in proximity.

3. **Zee** [48] makes indoor localization easy by enabling the automatic collection of training data through crowdsourcing. It leverages the inertial sensors (e.g., accelerometer, compass, gyroscope) present in mobile devices such as smartphones carried by users to track them as they traverse an indoor environment while simultaneously performing WiFi scans. However, it also requires a site-specific map input showing the pathways (e.g., hallways) and barriers (e.g., walls). Zee infers the WiFi fingerprinting of a space.

In this thesis, we are not interested in finding locations like EZ, infrastructure device locations like LocBLE, or wireless fingerprinting maps like Zee. Instead, our ultimate goal is to determine the surveillance level, a function of the infrastructure properties.

2.3.3 Active learning

People have mostly studied active learning in the context of classification problems [77] where labeling a data point comes with the cost of getting a human annotator to label that point. In most cases, we need multiple human annotators on a single data point to avoid personal bias in data labeling. Active learning helps to pick the best data point so that we can learn more with less cost. In our problem setting, since we will rely on people to collect RSSI or PRP data (equivalent to labels) to estimate infrastructure properties, it makes sense to use active learning to collect data to learn more with the same amount of data. Classification setting assumes having a finite unlabeled set from which they need to choose the data point. In contrast, in our problem setting, we are faced with a continuous space. In the current work, we discretize the space into a fine grid and then treat the set of points in the grid as our unlabeled set. There are different kinds of active learning techniques that have been proposed—uncertainty sampling [78], expected error reduction [79], information density framework [80]. In this work, we want to select locations (equivalent to features in the classification scenario) to collect PRP or RSSI data (equivalent to labels) to reduce the uncertainty in some objective distributions. The objective distributions can be PRP or RSSI data distributions for all locations, all infrastructure device locations, or directly the surveillance level distribution.

CHAPTER 3: LOCALIZATION

3.1 INTRODUCTION

Indoor positioning is a widely studied problem in academia and industry [4, 6, 17, 27, 81, 82, 83]. Coupled with the high penetration of consumer radio devices (e.g. smartphones), indoor positioning can re-imagine use of indoor spaces like retail spaces, malls, museums, and warehouses. Today, the contact-tracing challenge due to the pandemic has put an urgent, renewed focus on developing a robust, low-cost, scalable, indoor localization solution. Indoor-localization based contact-tracing[1] that helps us determine if a pair of individuals are "social-distancing," separated by more than 6ft, offers the possibility of safely re-opening the world economy.

Technological solutions for contact tracing that use smartphones are an important complement to normative (e.g., wearing a mask) and policy (e.g. stay-at-home) interventions for mitigating effects of the pandemic. Bluetooth Low-Energy (BLE) is emerging as the key contact-tracing technology and is being used in contact-tracing apps around the world. For example, the Aarogya Setu contact-tracing app[2] in India, uses BLE and has been downloaded 120M times. The open-source, privacy-preserving contact-tracing framework, BlueTrace[3] (deployed in Singapore) uses BLE packets to detect presence (i.e., a smartphone that can hear another must be in proximity of the other.), *not* distance. BLE is preferable to WiFi for contact-tracing: BLE uses 10× *less power* than does WiFi; BLE can be easily used to infer the presence of nearby peers without presence of WiFi infrastructure. The newly proposed Exposure Notification Service by Apple-Google[4] also relies on BLE beacons and signal strength measurements.

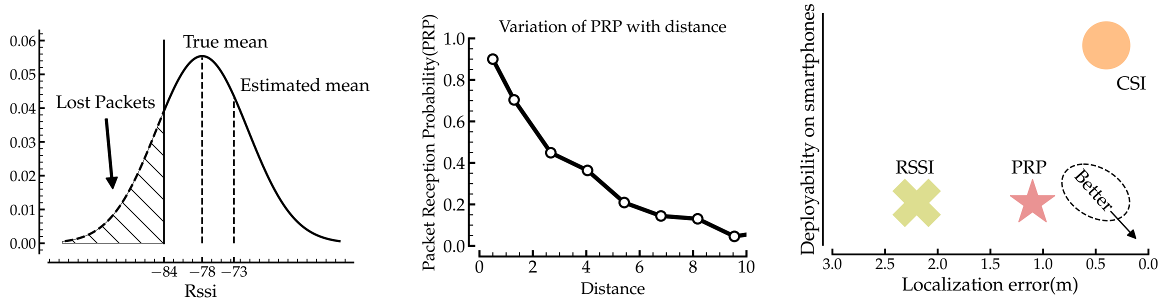
Bluetooth Low-Energy based apps for contact-tracing have two well-known shortcomings. These apps primarily use either RSSI (Received Signal Strength Indicator) or presence to determine risk to COVID exposure . Prior work [1, 2, 3] demonstrates that RSSI-based methods experience large errors (order of several meters) in positioning, especially in the low RSSI-large distance regime. RSSI has an important benefit: it is present on all modern devices. In contrast, we cannot use CSI (Channel State Information) [4, 5, 6], a recent method that enables sub-meter accuracy since off-the-shelf devices typically do not report CSI. A

[1] Contact-tracing requires us to calculate relative distance between individuals. Inferring relative distance from location is straightforward.

[2] `https://www.mygov.in/aarogya-setu-app/`

[3] `https://bluetrace.io`

[4] `https://www.apple.com/covid19/contacttracing/`



(a) (b) (c)

Figure 3.1: (a) As the mean RSSI decreases, the error in the RSSI estimate increases because of lost packets. (b) Packet reception in Line-of-Sight (LOS) with -20db transmission power decreases with distance. (c) Packet Reception Probability(PRP) technique is more accurate than RSSI [1, 2, 3] and readily deployable on commercial devices as opposed to CSI-based methods [4, 5, 6].

recent work [35] has enabled CSI for WiFi in some smartphones, but cannot be applied for BLE. Contact-tracing apps also use 'presence'—if one device can hear another—to determine if an individuals is close to another infected person. Presence is a poor proxy for distance since devices can hear Bluetooth beacons well beyond 6 ft social distancing radius, and also hear them across aisles and walls.

3.1.1 Overcoming key technological limitations for BLE localization

Establishing the feasibility of a reliable Bluetooth based indoor localization is a key first step for effective Bluetooth-based contact tracing and thus we ask: *Can we develop robust Bluetooth based indoor localization, with existing measurements, deployed on low-cost commodity hardware?* To do so, we need to overcome three fundamental RSSI limitations:

Biased RSSI Estimates due to Packet Loss: We explain with a conceptual example in Figure 3.1(a) that shows a Normally distributed RSSI at the receiver, for a fixed transmitter and receiver. In free space, with increasing distance between the transmitter and the receiver, the RSSI distribution shifts to the left, implying a decreasing RSSI at the receiver. RSSI-based methods [1, 39] empirically measure RSSI and use the mean RSSI estimate to infer distance. However, as the distance between the transmitter and the receiver increases (i.e., the RSSI distribution shifts to the left), packet loss increases with almost certain packet loss at the low-RSSI decoding threshold. Since devices only report RSSI for successfully decoded packets, RSSI-based distance methods suffer from a sampling bias: they use RSSI from decoded packets only. Since they cannot know RSSI values of packets they cannot decode, these methods introduce systematic error in their mean RSSI estimates. This error increases with distance, so much so that at large distances (few meters for BLE), as we shall show in this work, the mean RSSI estimate becomes de-correlated with distance and is an unreliable indicator. The error is different from the typical reduction in SNR due to increase in distance. The error stems from a sampling bias fundamental to RSSI measurements.

Packet Losses are higher in BLE: Packet loss is a fundamental problem in a low-power protocol like Bluetooth Low Energy. Even at distances as small as 1 m, in line of sight, around 10% of the packets get dropped in our empirical evaluation, as shown in Figure 3.1(b). Packet loss rate increases to 50% at 3 m. Thus, the sampling bias in RSSI measurements is a more significant challenge for localization using the BLE protocol compared to high-power WiFi protocol based RSSI methods [1, 16, 17]. As pointed out in [76], BLE limits transmission power to reduce energy consumption. BLE v4.0, v4.1, and v4.2 defined maximum output power is 10mW, which is 10× lower than WiFi. Though BLE v5.0 sets the maximum output power to 100mW, but the high Tx power is designed exclusively for high power devices with Class 1 BLE chip, and not for BLE beacons.

Multipath Effects: Multi-path effects [3, 40] are well-documented and are the second large contributor to RSSI measurement errors. Specifically, the error arises due to reflections of the radio signals by objects in the environment. Thus, the signals from the transmitter travel along multiple paths and combine at the receiver. This combination can be constructive (i.e., in-phase) and increase RSSI or destructive (i.e., out of phase) and reduce RSSI. Since this combination is a function of the environment and *not* the distance between the devices, multipath transmission introduces error in distance measurements.

3.1.2 A counter-intuitive approach: exploit packet loss to infer distance

In this work, we ask a counter-intuitive question: *Could the loss of a packet be a clue to the distance between the transmitter and receiver?* Specifically, we propose a new metric: Packet Reception Probability (PRP), which measures the probability that a receiver successfully receives packets from the transmitter. A simple experiment validates our intuition that PRP can encode distance. We collected packets from BLE beacons transmitting at -20db power at increasing distance values between 1 m to 10 m in a line-of-sight (LOS) scenario. We use maximum likelihood estimates for PRP. We plot the PRP estimate as a function of distance in Figure 3.1(b). Notice that Figure 3.1(b) shows that the probability of receiving a packet *decreases* with distance, implying that PRP encodes distance. We show in this work, that for low energy protocols including BLE, PRP is a good indicator of the distance between communicating devices.

Our approach, Bayesian Packet Reception Probability (B-PRP), is suitable for public spaces including retail stores or libraries, places that are important to current social distancing and contact tracing efforts. B-PRP is a PRP-based approach that develops a novel Bayesian framework to explicitly model multipath reflections in the environment and deliver robust and accurate localization. The Bayesian framework helps to minimize system deployment costs. A public environment like a retail store contains obstructing materials in the form of stacks or shelves. The shelves (including the items placed on them) absorb or reflect the radio signals directed at them. This leads to a lower packet reception probability at the receiver. At a fixed distance, the packet reception probability will vary based on the number and type of obstacles in the signal path. B-PRP must tease apart the effects of distance from the interference effects of the obstacles, when estimating distance. We observe that we can model public spaces including retail stores in a modular manner comprising open spaces separated by stacks. We explicitly capture the effect of such stacks by modeling the packet reception in absence of stacks and in presence of one stack, two stacks and so on. While we use stacks to model retail spaces, we believe that the abstraction of modeling a geometric element is general enough to apply to other large indoor spaces like libraries, warehouses, factories, etc.

We evaluated B-PRP in two real-world public places, an academic library setting and in a real-world retail store, and demonstrate the efficacy of our techniques. In both cases, we did not control for human traffic. Our main results:

Localization Accuracy: B-PRP achieves a median localization error of 1.03 m (library) and 1.45 m (retail store). The state of the art Bayesian RSSI system [39] has corresponding errors of 1.30 m (library, 26.2% more error) and 2.05 m (retail store, 41.3% more error) when trained with the same number of data points and packets per data point.

B-PRP+RSSI Fusion: Fusion of B-PRP and RSSI modestly improves the overall localization accuracy over B-PRP (Table 3.2). We see best fusion results at small distances ($\leq$2 m). At larger distances ($\geq$2 m), errors in RSSI cause fusion results to be significantly worse than B-PRP. PRP+RSSI also improves contact tracing accuracy by 6% for both library and retail store.

Robustness to Multipath: Our multipath model increases the accuracy for PRP from 1.41 m to 1.03 m in the library (a 26.9% improvement) and from 1.60 m to 1.45 m (a 9.3% improvement) in the retail store.

Number of Beacons: As beacon density decreases, B-PRP error is always within $2m$ while RSSI errors are higher than $3m$. With five beacons, B-PRP performs 65% better in library and 50% better in the retail store.

Low Training Overhead: B-PRP can leverage unknown training data to train the B-PRP model, thereby reducing the deployment effort. Specifically, B-PRP can achieve 1.08 m median accuracy with just 8 labelled data points and 4 unlabelled data points.

For completeness, we note that the core limitation of a localization method like B-PRP, a limitation shared with methods including [22, 27, 53, 54] is that it needs deployment of beacons in the public space to locate individuals. However, BLE beacons are inexpensive, and our method, B-PRP, provides meter-level accuracy. A peer-to-peer distance estimation is much more general where we will use devices like smartphones for reception and transmission. We believe that this tradeoff between some upfront infrastructure expense (multiple beacons) and increased localization accuracy is worthwhile in highly frequented public spaces.

3.2 CONTRIBUTIONS

Our work makes the following contributions:

Use of Negative Information: To the best of our knowledge, we are the first to build an indoor positioning system that can extract information from *absence of packets*. In contrast, state of the art RSSI based techniques [1, 16, 17], use observed RSSI to infer distance. We accomplish this through a Bayesian formulation of the packet reception probability, a metric that we show encodes distance. We develop generic stacking models of reception to address multipath effects. While we use PRP as a sole indicator of distance to highlight its benefits, we show that B-PRP when combined with RSSI, improves the performance of the system at shorter distances. Our finding is significant as it shows for the first time, how to use BLE to robustly estimate indoor distances, thus opening the door to reliable BLE based contact-tracing that incorporates distance.

Sampling Bias in RSSI: We show the effect of packet loss on the mean RSSI measurements. Furthermore, we show that with increasing distance, mean RSSI becomes highly unreliable due to sampling bias. Our finding is significant because the state of the art RSSI based techniques [1, 16, 17] when applied to BLE, a low-power protocol, are highly unreliable in the 2 m to 6 m range (Table 3.2). We highlight that $2\,\text{m} \approx 6$ ft, the social distancing range.

Readily Deployable Solution: Our B-PRP framework does not require any hardware, firmware, or driver-level changes in off-the-shelf devices, and requires minimal deployment and re-training costs. In contrast, CSI [4, 5, 6], which can deliver sub-meter accuracy, requires firmware or hardware changes. This is significant: due to the simplicity of the packet reception framework, we can immediately deploy B-PRP as an application on off-the-shelf commodity smartphones.

3.3 RELATED WORK

We can classify localization art on different factors— communication signal used for localization, models to relate distance and signal properties. Most works use signals exchanged with anchor nodes(known location) to infer location of target. Anchor nodes can be —WiFi access points [1], Bluetooth beacons [19, 20, 21], FM radios [53], Zigbee devices[54], ultra-wide band(UWB) devices [22], RFID tags [23, 24, 25, 26], ultra-sound emitters [55], light emitters [27, 28, 29, 30],60GHz devices [56, 57], sub-centimeter sized devices [58]. In contrast, we use BLE beacons which offer advantages over the others. WiFi access points and cameras require continuous power and are more expensive than BLE beacons, which run on long-lasting batteries (lasting 3 to 5 years). A store can deploy hundreds of BLE beacons at a lower cost than WiFi access points or video cameras. We can scale BLE-based systems through past work in opportunistic listening that ensures better channel sharing [84]. WiFi, while widely available in public spaces such as malls and coffee shops, are often absent in large indoor retail stores (e.g., Walmart), in part because the presence of WiFi allows individuals in the store to comparison shop, putting the physical store at a competitive disadvantage. While [22] shows the promise of low-cost UWB sensing, the solution requires the adoption of UWB tags to track objects. With BLE, we can track consumers via their Bluetooth smartphones.

The localization techniques use different signal property — RSS or received signal strength[1, 16, 31], CSI or channel state information [85, 86], AoA or angle of arrival [5, 32, 43] , ToF or time-of-flight [33, 44, 45]. AoA, ToF and CSI systems require hardware level changes on the receiver side and thus cannot be used by a retail store with customers who use commodity smartphones. Range free techniques use less accurate proximity information [87, 88, 89]. We use a new property—packet reception probability which is light weight and can be easily deployed on commercial smartphones.

Received Signal Strength (RSSI) systems are broadly of two types—model-based and fingerprint-based. Model-based techniques [16, 39, 46] represent RSSI loss between anchor and target as a function of distance. Fingerprint-based techniques [1], [17] build a map of probable RSS values from anchor nodes at sampled locations. Building the fingerprint map is a huge training overhead. [48, 49, 50, 51] reduce the overhead via crowdsourcing. [52] use a graph-based method for easy fingerprinting. Here we use a more robust property and design an easy-to-configure framework.

In this work, we study tracking for public spaces like retail stores which have attracted attention due to proximity marketing [90]. [91, 92] look at the problem of inferring item interaction in stores using wearable sensors. iBILL [20] jointly uses iBeacon RSSI model and inertial sensors to localize in supermarkets with 90% error less than $3.5m$. Tagbooth [93] , ShopMiner [94] tracks customer interaction with commodities using RFID tags in retail stores. The closest approach to our work is [95] which counts packets to estimate distance. But here, we estimate using packet reception probability (PRP). We show PRP as a robust estimator of distance, study the impacts of device density on PRP estimation, and reduce beacon set-up and retraining efforts.

3.4 PROBLEM WITH PACKET LOSS

Packet losses can impact the aggregate estimate of a physical signal property that we measure at the receiver end. Many state-of-art RSSI techniques observe the signal strength value of each of the received packets and aggregate them into a measure like mean/median. These aggregated estimates are then used to infer distance. Here we are asking the question—*Does packet loss introduce any bias in the aggregate estimates of RSSI?*

First, we identity that packet losses can be mainly attributed to two reasons—random errors and low signal strength. Errors can occur uniformly at random irrespective of the RSSI of the packet. As a result, such errors do not introduce any bias in the aggregate estimate of RSSI. On the other hand, all packets that are received with a signal strength below a certain decoding threshold get dropped. Since we cannot observe the RSSI values of these low RSSI packets, and hence cannot include them in our aggregate estimates, we should expect to see a positive bias introduced in our RSSI measurements.

Now, lets solidify this hypothesis mathematically by considering the simple case that *packet losses are entirely due to low-signal strength.* Let us assume that the actual RSSI values at a certain location follow the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Lets further assume that the RSSI decoding threshold is α. Since we drop all packets below the threshold, our aggregate RSSI estimates will be based on a Normal distribution truncated at α. The new mean of this truncated normal distribution is given by

$$\hat{\mu} = \mu + \frac{\phi(\alpha)}{1 - \Phi(\alpha)}\sigma, \tag{3.1}$$

where, $\phi(\alpha)$ is the pdf of normal distribution evaluated at $\alpha, \phi(\alpha) \geq 0$. $\Phi(\alpha)$ is the cdf value of the normal distribution at $\alpha, \Phi(\alpha) < 1$. Thus the estimate $\hat{\mu}$ that we obtain by measuring received RSSI values is biased by a positive amount of $\frac{\phi(\alpha)\sigma}{1-\Phi(\alpha)}$. As we move towards the lower RSSI regime, μ becomes closer to α. As a result, both $\phi(\alpha)$ and $\Phi(\alpha)$ increases with lower RSSI values, which leads to a higher bias in the estimated mean RSSI.

Note that we cannot trivially estimate μ from $\hat{\mu}$ in Equation (3.1) since in practice, multi-path effects alter the values of the RSSI in the received packets. Thus recovering μ, σ using say Maximum Likelihood Estimates by assuming a value of α is non-trivial.

In this work, we ask a different question—*Can we use the loss of packets as a signature itself to measure distance?* We define a random variable, packet reception probability, $prp(b)$, for a beacon b whose expected value is defined as:

$$\mathbb{E}(prp(b)) = \frac{\sum_i \mathbf{1}_{i=b}}{R(t_l - t_f)} \tag{3.2}$$

Here, $\mathbf{1}$ is the indicator function that is 1 if and only if packet i is received from beacon b, R is the sending rate of the beacon, and where t_l and t_f are the timestamps of the last and the first packet received from beacon b. Notice that the right hand side of Equation (3.2) is just the frequentist estimate of the probability of packet reception from beacon b: number of packets received divided by the total number of packets sent by beacon b. Notice that while random errors will affect both PRP and RSSI, **packets that are successfully received and influenced by multipath effects, only impact RSSI mean estimate $\hat{\mu}$ not expected PRP value** $\mathbb{E}(prp(b))$.

From the next section, we will focus largely on the evaluation of PRP to measure distance. However, note that we can easily combine PRP with other metrics like CSI or RSSI to measure distance. We evaluate PRP+RSSI in Section 3.9.1.

3.5 PACKET RECEPTION PROBABILITY

In this work, we focus on localizing individuals in indoor public spaces like retail stores, libraries. In these spaces, indoor positioning using BLE beacons can enable traditional applications like capturing behavioral data about shoppers, as well as novel applications like enforcing social distancing and contact-tracing. BLE offers an unique advantage for localization. Due to its low power budget, it can be turned on frequently and hence, enable more frequent location updates as compared to high power protocols like Wi-Fi. Recall that BLE's maximum transmit power (10 dBm) is 10 times lower than that of Wi-Fi (20 dBm). This factor, in addition with its ubiquitous presence on off-the-shelf smartphones, has made

BLE the natural choice for such applications.

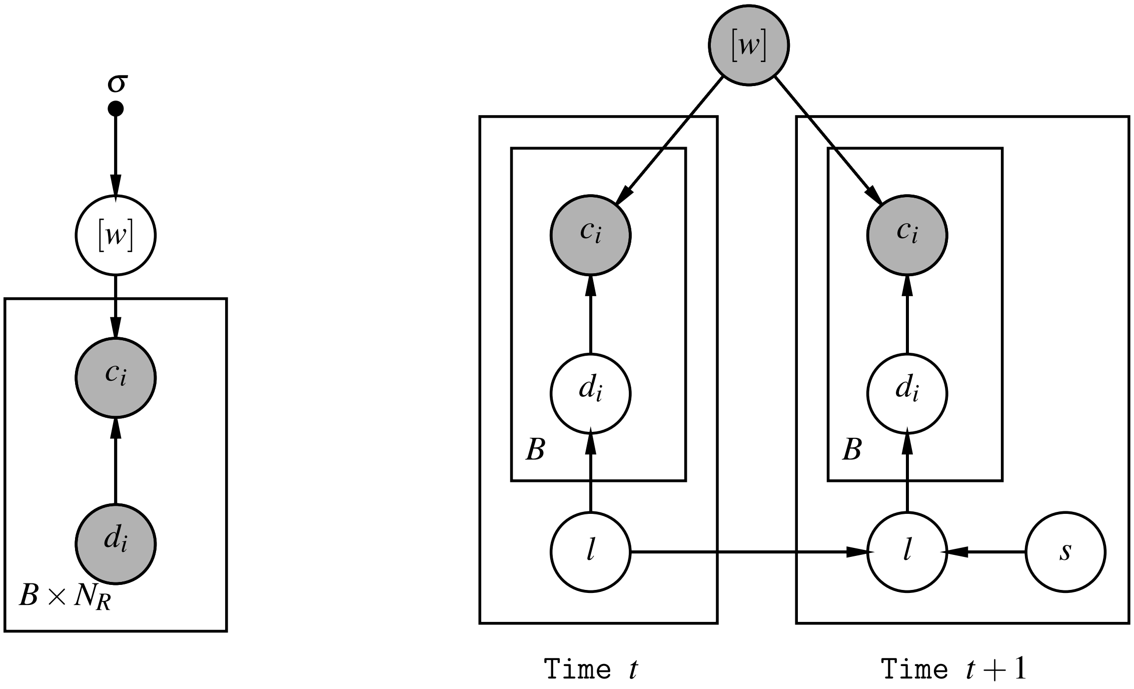
We deploy BLE beacons at fixed locations in the indoor environment to aid localization. We believe that this infrastructure-support is essential for accurate indoor positioning. The beacons are configured to emit Bluetooth packets, power p_t, and a fixed sending rate, R. The smartphone of an individual navigating the public space listens to these packets and logs them in the following form:

$$L = \{(b_1, t_1), (b_2, t_2), \ldots, (b_N, t_N)\} \tag{3.3}$$

where b_i refers to the BLE beacon `id` heard at time t_i. Our goal is to determine a list of locations of an individual walking around in a store at a fixed time resolution δ (i.e., determine location every δ sec.).

We use the packet log, L, to compute packet reception probability, $prp(b)$, for beacon b.

3.5.1 Estimating location using PRP



(a) Training parameters $[w]$

(b) Tracking location l over different time slots

Figure 3.2: **Graphical model**: Shaded nodes are observed, while we need to estimate the unshaded ones. We use the data on number of received packets c_i measured from B beacons at N_R reception locations to train the PRP parameters $[w]$. During tracking, we use the trained parameters $[w]$ and $c_{i,t}$ to estimate location l_t.

Recall, in Figure 3.1(b), PRP degrades with distance. In this section, we discuss how we can model the relationship between PRP and distance, and use this relationship to infer location. Specifically, PRP (*prp*) depends on three factors: (a) distance (d), (b) sending rate (R), and (c) transmission power (p_t). In this subsection, we model the relationship in free space (Figure 3.3(A)). We will incorporate the effect of multipath in subsequent sections.

We use a Bayesian model to model the relationship between PRP estimates from multiple beacons and the underlying physical location. Our choice of the Bayesian approach is motivated by two key design benefits: (a) It allows us to infer not just the location, but also quantifies the uncertainty in the location estimate. Such estimates are very helpful when the location is used for higher-layer applications like customer behavior analytics, contact-tracing, etc. (b) It can be easily extended to scenarios when the beacon loction itself is unknown or the training set is small. As we show in Section 3.6, this reduces the training and deployment costs.

We model *prp* as a function g of the distance d, sending rate R and power p_0 of the beacon. Assume that we receive a packet from beacon (x_b, y_b) at location (x_r, y_r). We calculate the Euclidean distance d between the beacon and receiver. Then, assuming that we know sending rate R and transmission power p_0, we can model the number of packets received c received at (x_r, y_r) as drawn from a binomial distribution with parameter *prp*:

$$c = Bin\left(N, prp\right), \qquad \text{binomial distribution,} \tag{3.4}$$

$$prp = g(d, R, p_0), \qquad \text{PRP link function,} \tag{3.5}$$

$$d = \sqrt{(x_b - x_r)^2 + (y_b - y_r)^2}, \qquad \text{distance to beacon } b. \tag{3.6}$$

N is the total number of packets sent out by the beacon is proportional to the product of the sending rate R, and the time spent T_r at location r. The function $g(d, R, p_0)$ is a link function that connects the underlying infrastructure parameters (R, p_0) and physical distance d, to the packet reception probability.

In identifying the right representation of g, we need to keep two considerations in mind: (a) the value of g has to be between 0 and 1, and (b) g must encapsulate relationship between d, R and p_0, not just their direct effect on *prp*. Therefore, we model $g(d, R, p_0)$ as a logistic function of quadratic interaction between the parameters.

$$\text{logit}\{g(d, R, p_0)\} = w_0 + \sum_i w_i \theta_i + \sum_{i,j} w_{i,j} \theta_i \theta_j \tag{3.7}$$

where, $\text{logit}(p) = \log(p/1 - p)$. And where, $\theta_1, \theta_2, \theta_3$ correspond to the variables of d, R, p_0 respectively. The coefficients $[w] = [w_i, w_{i,j}]$ are drawn from a non-informative prior $N(0, \sigma)$—

a zero mean Normal distribution with variance σ. We choose σ to be large in our system to allow for a large range of values.

Our Bayesian formulation above is shown in Figure 3.2(a). Our framework operates in two phases:

Training Phase: During training, we use a data set D collected in an environment to estimate the underlying parameters. Specifically, we need to estimate the posterior distribution of the unknown parameters $[w]$ given data D i.e. $P([w] \mid D)$. The training set, D, comprises BLE logs. Specifically, to obtain D, we stand at N_R locations in our testing area and listen to the packets from B beacons. Assume further, that we know the B beacon locations (x_b, y_b), $b \in \{1, \ldots, B\}$ and N_R reception locations (x_r, y_r), $r \in \{1, \ldots, N_R\}$. We will relax this assumption in Section 3.6.

Test Phase: During test phase, we do not know the reception locations, (x_r, y_r) $r \in \{1, \ldots, N_R\}$. We use the measured *prp* and the parameters estimated during the training phase to estimate the receiver location. We use PyMC3 [96] framework to do the inference.

3.5.2 Combating multipath effect

We have assumed a free-space propagation model so far, but real-world environments have obstacles. We observe that the main contributor to multipath effect in public spaces like retail stores (or libraries) are the stacks used to list products (or books) and to separate aisles. In such scenarios, the *prp* value depends not just on the distance, but also on the number of stacks the signal has to cross. Crossing one stack is easier than crossing two and will cause fewer packet drops.

To build on this observation, we explicitly model the number of stacks in our framework. This allows us to not just estimate the distance between a beacon and a receiver, but also estimate the number of stacks between them. Estimating this geometric information is useful for both: combating multipath, and exploiting in higher-layer applications. For instance, retail store apps need to estimate what aisle a customer is shopping in, contact-tracing apps want to discount for infection spread if customers are close (but across aisles). To estimate the stack separation, we divide the store layout in Figure 3.3 into five portions based on the given beacon—free space(F-S), one stack (1-S), two stacks away (2-S), corridor (C) and desk (D). In the figure, the packets to receiver 1 in F-S do not have to go through any obstacles. The packets to receiver 2 in 1-S and 3 in 2-S go through one and two interfering stacks respectively. Receiver 4 is in a corridor. We limit ourselves to two stacks away in the model, because we empirically observe that two stacks or more have similar effects on packet reception (high loss).

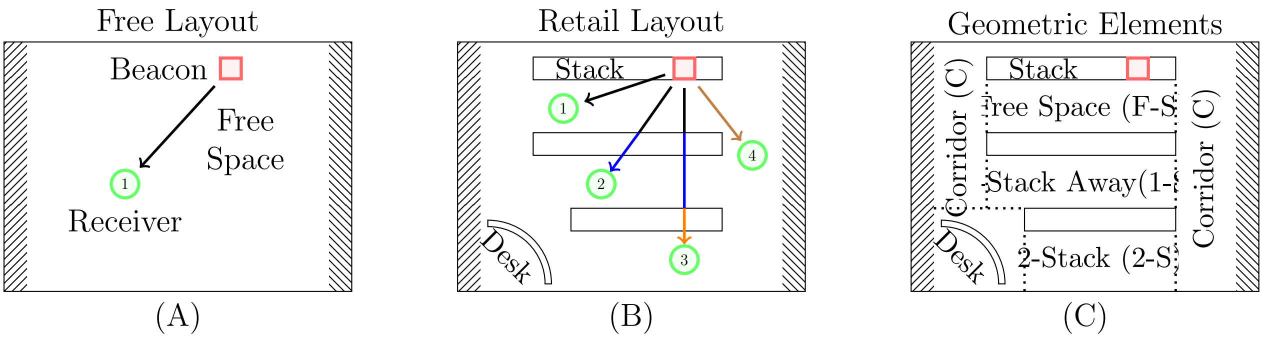


Figure 3.3: **Modelling Obstacles and Multipath:** In (A), there is no obstruction in the path of the receiver. In retail layout (B), receiver 1 is in free space with beacon, 2 is one stack away and 3 is two stacks away. 4 is an open region of the layout, i.e. the corridor. We segregate the retail layout in (C) into geometric elements based on the relative position of beacon and receiver.

Then, we parameterize our link function with a variable, γ that denotes the geometric-element separation. We represent the new link function as $g_\gamma(d, R, p_0)$. Now, at training time, we estimate parameters for the functions—free space g_{F-S}, one stack g_{1-S}, two stack g_{2-S} and corridor (C) model g_C. We use a Bayesian training procedure similar to the free-space scenario. We segment our training data into the different scenarios, and use the segment-specific data to learn the parameters in each g_γ. For example, the data with one stack separation is used to train g_{1-S}. During testing, B-PRP uses the maximum likelihood model to identify the underlying location as well as stack separation.

At first blush, it might seem very complex to identify γ for each of the B beacons. We exploit the knowledge of the store geometry and beacon arrangements within the store to significantly reduce the number of unknowns. Assume that beacons a and b are in the same aisle adjacent to each other. Then, *regardless of where the individual is, beacons a and b must have the same model type γ with respect to the receiver.* Similarly, if beacons a and b are in neighboring aisles, and the model type γ is $F - S$ for a, then γ must be $1 - S$ for b. Thus, given a location x_t, y_t, knowledge of store geometry and beacon arrangements help fix the model type for *all* beacons, given the model type for *any one* beacon.

3.5.3 Leveraging human mobility

Finally, to further improve localization accuracy, we leverage the fact that human location across time is not independent. Humans motion is continuous in space. We update our Bayesian formulation to incorporate this constraint.Assume that we wish to track individuals at temporal resolution δ, corresponding to T time intervals. Let the unknown location variables be $l_t = (x_t, y_t), t \in \{0, \dots, T-1\}$. We observe that the locations are not independent across time. Since we know little about the initial location of the customer, we set the prior distribution of l_0 to drawn uniformly at random from the entire testing area. Let the unknown speed of the individual when reaching location l_t be s_t. Then:

$$
\begin{aligned}
s_t &\sim U(0, S_{max}), && \text{speed,} && (3.8)\\
x_0 &\sim U(0, W), && \text{initial } x, && (3.9)\\
y_0 &\sim U(0, L), && \text{initial } y, && (3.10)\\
x_t \mid x_{t-1} &\sim \mathcal{N}(0, s_t * \delta), && x_t \text{ constrained by } s_t \times \delta, && (3.11)\\
y_t \mid y_{t-1} &\sim \mathcal{N}(0, s_t * \delta), && y_t \text{ constrained by } s_t \times \delta. && (3.12)
\end{aligned}
$$

Where, S_{max}, which is a constant in our model denoting maximum movement speed of a human (similar to [95]); $W \times L$ is the tracked area; and the conditional distributions $x_t \mid x_{t-1}$ and $y_t \mid y_{t-1}$ are Normally distributed with zero means and variances equal to $s_t \times \delta$. Incorporating the speed parameter in our formulation enforces smoothing in our inferred location estimates.

Recall, our observed data contains the number of packets received $c_{i,t}$ from each beacon i at each time t, where $N = R \times \delta$. We know the location of each beacon (x_i, y_i), their sending rate R and power p_0. We also know g_{F-S} from training. Thus, during the test phase, we estimate using our Bayesian framework (c.f. Figure 3.2(b)), the posterior distribution $P([l, s] \mid D)$ of the unknown location and speed parameters $[l, s] = \{\{l_t\}, \{s_t\}\}$ given this data D.

3.6 SYSTEM DEPLOYMENT AND OPTIMIZATION

To summarize, the B-PRP system operates in following steps:

- **Deployment:** We deploy BLE beacons at known locations in an environment like a retail store. The location of the beacons as well as the floor plan is uploaded to a B-PRP server. The server can reside on the cloud or be an edge device local to each environment.
- **Training:** A user walks to fixed locations in the store with a smartphone app or another BLE receiver and measures the PRP values. The PRP values are uploaded to a server. The server uses these labelled PRP values, the beacon locations, and the floor plan to train the B-PRP model.

- **Localization:** Finally, when new users walk in, they measure PRP for beacons already deployed in the store. The app on the smartphone uploads the PRP values to the server. The server uses the trained model to infer location of the users and sends it back to the user. Note that, this system is centered on the user. If the user chooses not to share the PRP values with the server, no location estimation and contact tracing can be performed. Furthermore, the design also conserves power on the smartphone because the user never has to transmit any BLE packets.

Finally, we transmit beacons using BLE advertising mode. This prevents the need for making any explicit connection between the user device and the beacon. The user device can ignore the advertising beacons to avoid localization.

Reducing Deployment Overhead: Deploying the localization infrastructure has two major overheads—setting up the beacons at exact locations, and training. Knowing the location for beacons deployed by a large store is labor intensive. Similarly, training involves standing at multiple known locations inside the layout and collecting data for certain period of time. We ask two questions—*1. Instead of costly human labor, can we infer most beacon locations from training data? 2. Can we leverage data from unlabeled locations of store workers to train our model?*

As it turns out, we can affirmatively answer both these questions in our formulation. We can leverage unlabelled data (without location information) that is collected by store workers as they move around the store to help train the model as well as to infer most beacon locations. We use data collected by store workers D to solve both problems. D contains number of packets received from all B beacons at all N_R training locations. Let us assume that we know the locations of a small number $b \ll B$ *primary* beacons, with the remaining $B - b$ beacon locations unknown; Ideally we will like b to be as close to 0 as possible. Also assume that only a small number $r \ll N_R$ locations are known, with the remaining $N_R - r$ locations unknown. Our goal is to infer $B - b$ beacon and $N_R - r$ training locations from D along with the packet reception model parameters $[w]$.

To enable this, we view the model through a generative process. We initialize the $(B - b)$ beacon and $(N_R - r)$ unknown reception locations from a uniform prior over the testing area which is of dimension $W \times L$. We want to jointly estimate the distribution of the unknown beacon locations $\{l_j\}, j \in \{1, \ldots, B - b\}$, $\{l_k\}, k \in \{1, \ldots, N_R - r\}$ and packet reception model parameters $[w]$, given data D. In other words, we want to estimate the posterior distribution $P([l_j, l_k, w] \mid D)$. This can be easily achieved, given the Bayesian nature of our model. We use standard **Markov Chain Monte Carlo (MCMC)** based Bayesian inference techniques to compute the posterior distribution over the unlabelled data points and beacons. We use **No-U-Turn sampling (NUTS)** [97] included with PyMC3 [96] to perform MCMC sampling. Therefore, B-PRP can leverage unlabeled data as well as unlabelled beacon locations to improve its estimates and reduce the deployment overhead.

3.7 EXPERIMENTAL SET UP

We evaluate B-PRP in two testbeds—an academic library and a retail store. Both spaces have shelves segregating the floor space into rectangular regions, i.e. aisles and corridors. The two environments differ in three main aspects—difference in layout, i.e. arrangement of rectangular areas and the presence of walls around the space, difference in material of shelves, and human interference. The retail store had more dynamic customer traffic movement during the experiments.

Library: We show the layout of the library space, $14m$ by $8m$, in Figure 3.5(a). It has three wooden shelves (each $11m$ long & $0.5m$ wide). The aisles between two stacks are $0.7m$ wide. We placed two rows of 12 beacons on each stack. We manually measured each inter-beacon distance. The distance between two adjacent beacons on the same row is $0.91m$. The distance between two devices kept opposite each other on the same shelf, but facing two different aisles is $0.43m$. We carried out our experiments during regular library hours.

Retail Store: Figure 3.5(b) shows a retail store with dimensions: $10m$ by $10m$. The environment has four steel stacks ($1.27m$ wide each; three are $7.5m$ long, one is $6m$ long). The aisles between two stacks are $1.8m$ wide. We place two rows of beacons on each stack. The inter-beacon distance on the same row is $1m$. Retail store is a challenging environment due to the presence of steel structures as well as worker and customer movement during the experiments.

3.7.1 Devices

We use following devices for our experiments—Bluvision iBeeks [98] , BluFi [99], TI packet sniffer, a laptop and Android smartphones(Nexus5x, NuuA4L). iBeeks or iBeacons are battery operated BLE beacons. They support a wide range of broadcasting power from $-40dBm$ to $+5dBm$. $-40dBm$ translates to $3m$ line of sight range, while $+5dBm$ gives us a range of $150m$. For our experiments, the beacons send 10 packets per second at -15 dBm power. We deploy 60 iBeacons in the library and 38 beacons in the retail store. As discussed before, we do not need to know all of these beacon location a priori.

We use three receiver devices for BLE: Texas Instrument Packet Sniffer (CC2540 dongle), Nexus 5X smartphone, NuuA4L smartphone. iBeacons broadcast BLE packets in three

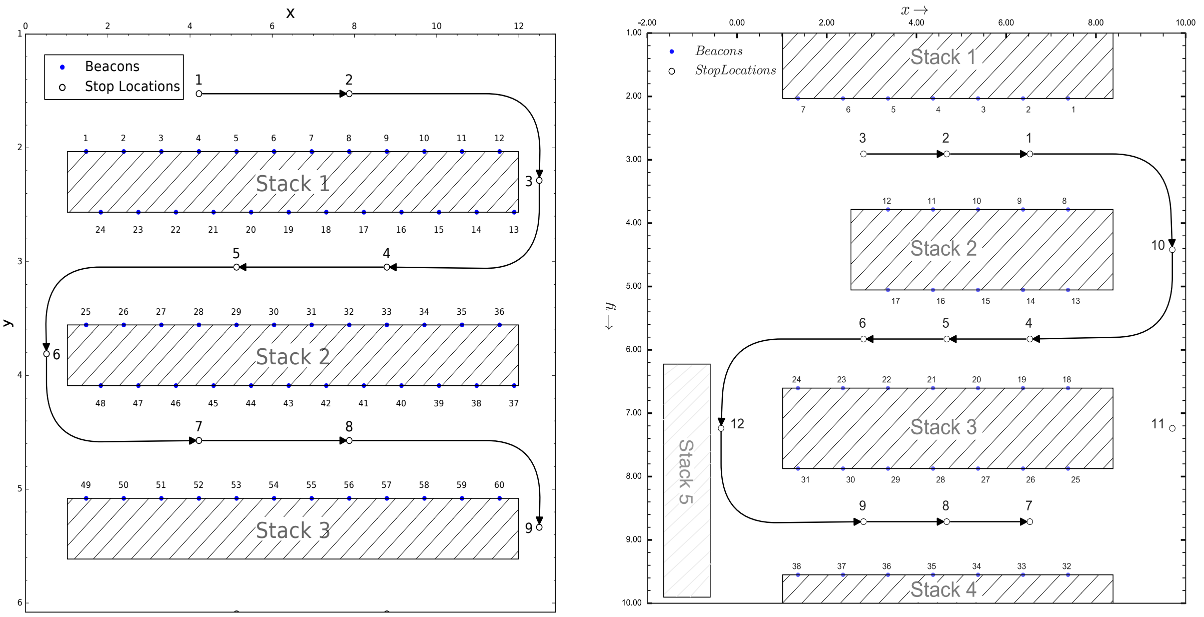
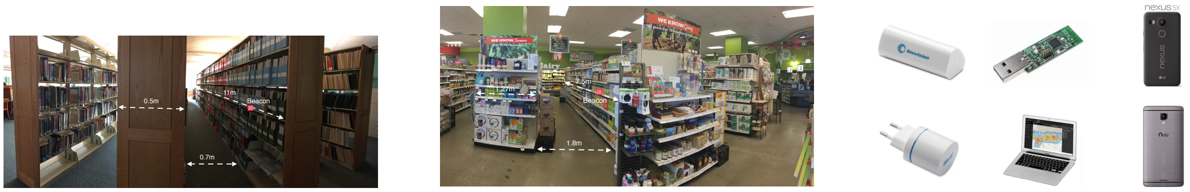


Figure 3.4: Library and Retail Store layout. We show one specific movement sequence used in our experiments.



(a) Library (b) Retail Store (c) Devices

Figure 3.5: **Experimental Testbed:** We conduct our experiments in a library (a) and a retail store (b) using devices shown in (c)—Beacon, Blufi, Sniffer, Laptop, Nexus5X and NuuA4L android smartphone.

channels— 37, 38 and 39. The sniffer can filter out packets from specific channels. We connect the sniffer to a Windows laptop and use it for packet reception from beacons. For the Android phones, we built an android app using Altbeacon [100] library to scan BLE channels.

3.7.2 Baselines

We compare B-PRP against state-of-the-art in RSSI-based positioning:

- **Horus** [17] is an RSSI fingerprinting technique that was originally tested with WiFi. We extend it to BLE. For fairness, we use Horus with the same number of training locations as other baselines—12 for library and 9 for retail store. The inter-state distance is 3.5m for library and 1.85m for retail store.

- **Bayesian RSSI** [39] uses a generative model based on RSSI to determine location. We set the priors and parameter values following recommendations in [39].
- **Bayesian RSSI Fingerprinting** (or Bayesian FP) [47] is a Bayesian Fusion technique applied to a fingerprinting based method for BLE devices. It stores fingerprints like Horus, but employs fusion technique to combine current RSSI and prior location information.
- **MCL** [87] is a range-free localization technique and uses proximity rather than ranging information to localize nodes. It observes whether a packet was received from a device and infers whether the reception location is inside or outside a threshold distance from the beacon.

To ensure fair comparison, we use the same training data across all techniques. Furthermore, for RSSI based techniques, we use mean RSSI values over all packets used by the PRP technique. That is, **if PRP uses k packets at a location, we use the mean RSSI value over the *same* k packets.** This removes inter-packet RSSI variance at the same location, *improving* RSSI localization. RSSI results are significantly worse without averaging.

There is more recent work in CSI-based positioning [4, 5, 81], but CSI data is not available on most commercial smartphones. Hence, we do not cover these baselines. For reference, the state-of-the-art CSI-based method achieves a median localization error of 86 cm[101]. However, this work requires CSI data on phones and multi-antenna beacons, both of which are not mainstream yet, and hence, cannot be deployed at scale for applications like contact tracing.

3.7.3 Data collection

We collected data for both layouts in two phases—training and localization. We collected data at stationary spots to train B-PRP and competing baselines. We marked some fixed places for each layout and stood there for 1 minute to receive data from the beacons. We used 12 such spots for the library layout and 9 locations for the retail store layout.

We collected data in both test-beds to compare the accuracy of localization and contact tracing techniques. To track and test on data from a moving person, we asked users to naturally move inside the layout with the laptop and sniffer in hand. We used fixed movement paths and marked spots along the path. Each path or trace is a simulated movement carried out in real time between such marked spots. We stop at each marked place for 10 seconds, and we move at a normal walking speed of $0.5m/sec$ between the spots. We can now calculate the ground truth location at any time within the movement trace. Please note that **we**

evaluate our location estimates throughout the movement trajectory. They are not restricted to the marked fixed spots.

3.8 MICRO BENCHMARK

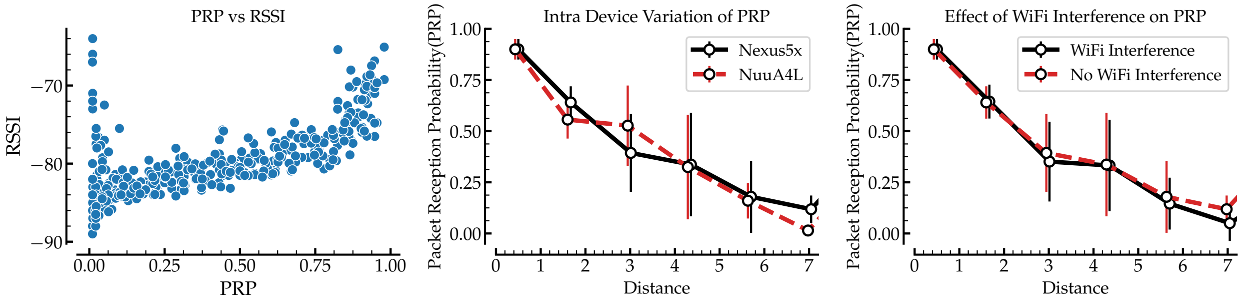


Figure 3.6: **Microbenchmarks:** (left) PRP vs RSSI are not directly related, but follow an expected trend. (middle) PRP variation is similar across two different android devices—Nexus5X and NuuA4L. (right) PRP variation is robust to ambient WiFi interference.

We present microbenchmarks to better understand PRP:

Relationship to RSSI: First we ask how PRP varies with RSSI and if packet reception is directly dependent on RSSI. We plot this relationship in Figure 3.6. As seen in the figure, there is an expected trend between the two parameters, but there is also significant variance for each value of PRP. This implies that the relationship between packet reception and RSSI is not determined by a hard threshold, but is instead more probabilistic. The probability of packet reception goes down with RSSI but several other factors including random noise come into play.

Translation across devices: Does the relationship of PRP with distance depend on a device? To answer, we collect PRP values at the same location with two android smartphones: Nexus5X and NuuA4L. As shown in Figure 3.6, we see very close trends in PRP vs distance with minor variations[5].

Robustness to interference: Does interference from other in-band transmissions like WiFi hurt PRP? To understand this, we conduct the following experiment. We setup a WiFi router on 2.4GHz WiFi band and use two laptops to saturate the link using the iperf utility[102]. We measure the PRP-distance relationship with WiFi interference turned on and off. We see negligible variation in the relationship between PRP and distance (Figure 3.6(right)). This is because the three advertising channels of BLE fall between or outside the main frequencies used for IEEE 802.11, allowing for better coexistence with WiFi.

[5] The experiments were conducted on different days for each smartphone

Does interference from many co-located beacons hurt PRP? BLE beacons send out short advertising messages in passive mode containing a payload of at most 31 bytes. As pointed out in [84], the small size of the advertising messages helps in avoiding any significant collisions of upto 200 or more co-located devices. Similarly, the co-location of many receivers or scanning devices will not impact PRP. In our set-up, the receiver receives the advertising message in a passive scanning mode, and does not respond in any way. As a result, many scanning devices do not lead to any interference.

3.9 RESULTS

We compare the localization performance of baselines against B-PRP in Section 3.9.1. For these results, we assume that all beacon and reception locations are known (for all methods). In Section 3.9.2, we evaluate the robustness to the number and placement of beacons. In Section 3.9.3 and Section 3.9.4, we list the results for B-PRP when we reduce the beacon set-up costs and the number of labelled training locations. In summary:

- Median error for B-PRP is 1.03m and 1.45m in library and retail store. The corresponding errors for the best baseline, Bayesian RSSI are 1.3m and 2.05m.
- B-PRP is more robust than RSSI to decreasing number of beacons. With 5 beacons, B-PRP performance is 65% better in the library and 50% better in the retail store.
- B-PRP performs better than Bayesian RSSI when we use only Non Line-of-Sight(NLOS) or far away beacons. With beacons placed greater than 6m distance, B-PRP gives error of 1.53m and 2.07m in LOS and NLOS. RSSI errors are 3.85m and 5.15m.
- B-PRP can reduce set-up cost by learning most beacon locations. Given data from 12 training locations, B-PRP needs to know exact location of only 6 beacons and it can infer the remaining 54 beacon locations while giving an accuracy of 1.05m.
- B-PRP can reduce retraining efforts by leveraging data from unknown locations. Having data from 12 known locations vs (6 known + 6 unknown) locations gives the same accuracy level. We can improve accuracy $\sim$ 40% by adding data from unlabeled spots.

3.9.1 Localization accuracy evaluation

We compare the accuracy of B-PRP against baselines. We use Euclidean distance to measure the error between actual and estimated locations for each time window. We show cumulative distribution over errors in Figure 3.7 and median error in Table 3.1.

Environment	**B-PRP**	B-PRP + RSSI	Bayesian RSSI [39]	Horus [17]	Bayesian FP [47]	MCL [87]
Library	1.03m	0.91m (↓ 11.6%)	1.3m (↑ 26.2%)	1.83m (↑ 77.6%)	1.93m (↑ 87.4%)	2.26m (↑ 119%)
Retail Store	1.45m	1.46m (↑ 0.6%)	2.05m(↑ 41.4%)	1.85m(↑ 27.6%)	1.95m(↑ 34.5%)	2.93m (↑ 102%)

Table 3.1: **Median error (in m) of B-PRP and baselines:** B-PRP perfoms best in both environments followed by Bayesian-RSSI in the library and Horus in retail store. Fusion of B-PRP and RSSI performs slightly better in the ideal library environment with many beacons at close distance. Horus and Bayesian FP underperform as they require more training states

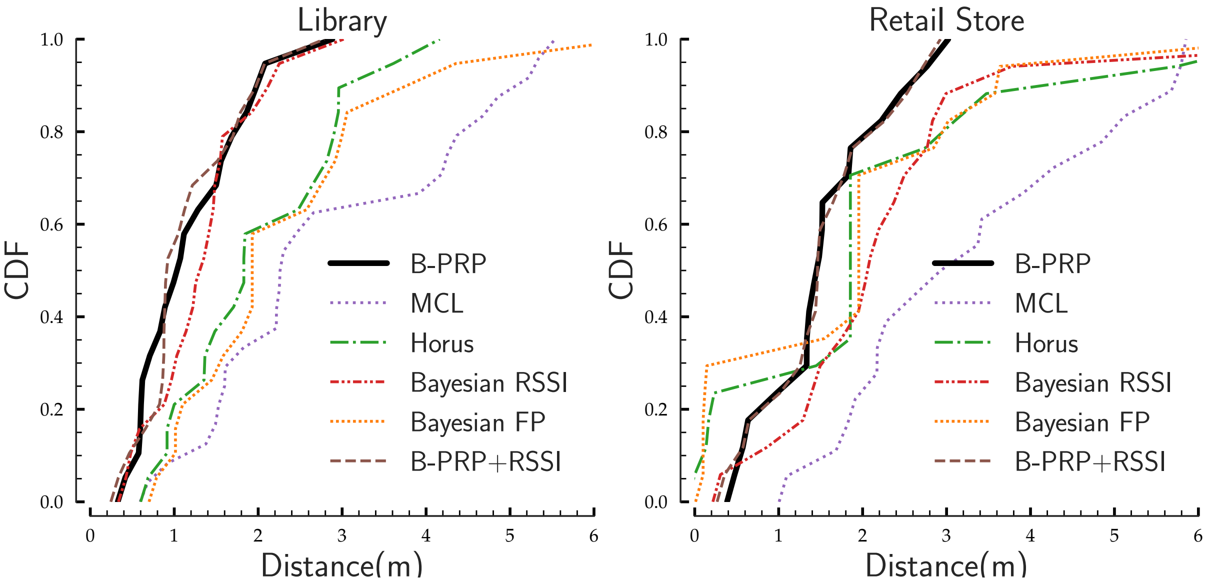


Figure 3.7: **CDF error distribution** for Bayesian PRP and baselines in library and retail store. For RSSI techniques, we have averaged RSSI value across the same number of packets that was used by PRP technique.

First, observe that B-PRP achieves a median error of 1.03m and 1.45m in the library and retail store. The next best method, Bayesian RSSI, achieves errors of 1.3m and 2.05m. The errors for other methods are higher, so for the rest of the document, we will present detailed results for Bayesian RSSI. The errors for all methods are higher for the retail store which has more human traffic than the library. B-PRP can significantly outperform all other baselines due to two reasons: (a) B-PRP can extract information even from lost packets, and (b) It incorporates a new multipath-model that can work in the presence of obstacles. The stack model helps to increase the median accuracy of B-PRP from 1.41m to 1.03m in the library and from 1.6m to 1.45m in the retail store.

Non-Line of Sight: How does Bayesian RSSI and B-PRP compare in line-of-sight (LoS) vs non-line-of-sight (NLoS) scenarios? In Figure 3.8, we can see that the median error for Bayesian RSSI is 2.34m with NLoS beacons in the library, which is far more than 1.3m error that it achieves using all beacons. B-PRP achieves a better median error of 1.63m with NLoS

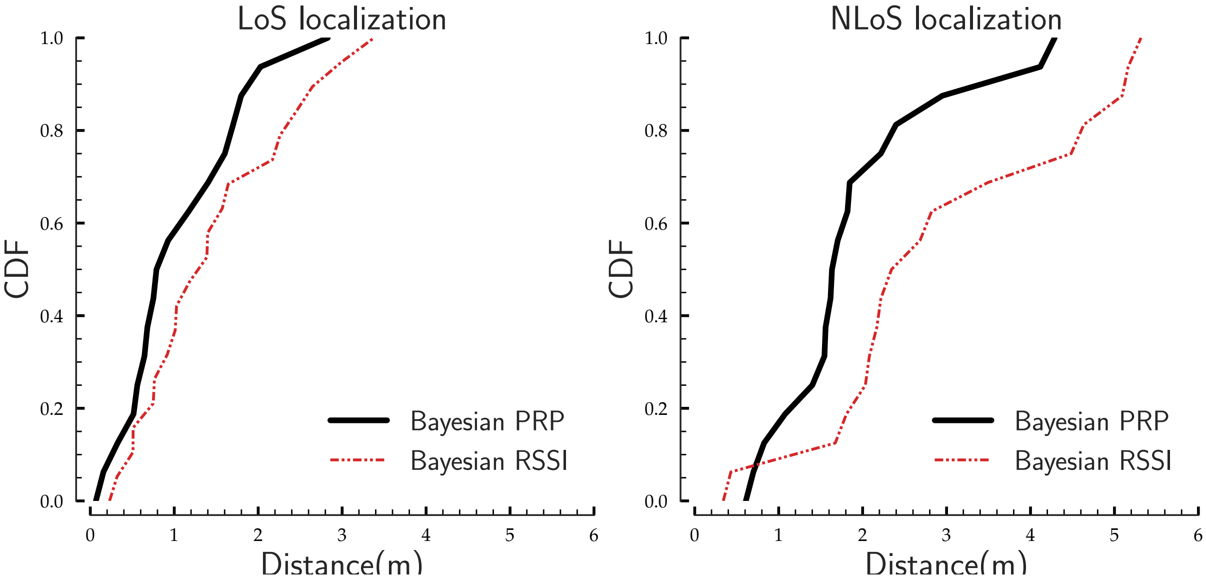


Figure 3.8: **CDF of localization errors** for B-PRP and RSSI for different scenarios. In library, B-PRP performs much better when asked to localize with *only* non-line-of-sight beacons.

devices. This shows an additional advantage of B-PRP over traditional approaches using RSSI. It can perform significantly better even when no beacon is in line-of-sight. Note that this error difference is significant because an additional error of $0.8m$ can significantly worsen false-positive and false-negative rates in end user applications.

B-PRP+RSSI: One might wonder if B-PRP can be augmented with RSSI to achieve even better performance. We augment B-PRP with RSSI to test this hypothesis. As shown in Figure 3.7, the method works approximately similar to B-PRP. As we demonstrate in the next subsection, this is because at smaller distances, RSSI experiences little packet loss and helps our model make better inference. However, at large distances, RSSI experiences larger sampling bias and consequentially, just acts as noise, thereby hurting the model.

3.9.2 Beacon number and placement

We evaluate the robustness of localization performance of B-PRP against the best performing baseline—Bayesian-RSSI to two factors—the number of beacons and the placement of beacons.

Beacon Number: In our test beds, we initially set up the beacons at a distance of $1m$. That resulted in 60 beacons in total in the library and 38 beacons in the retail store. A large retail store will need to place hundreds of beacons to maintain this inter-beacon distance, which in turn increases the localization infrastructure and configuration cost. So, it's natural

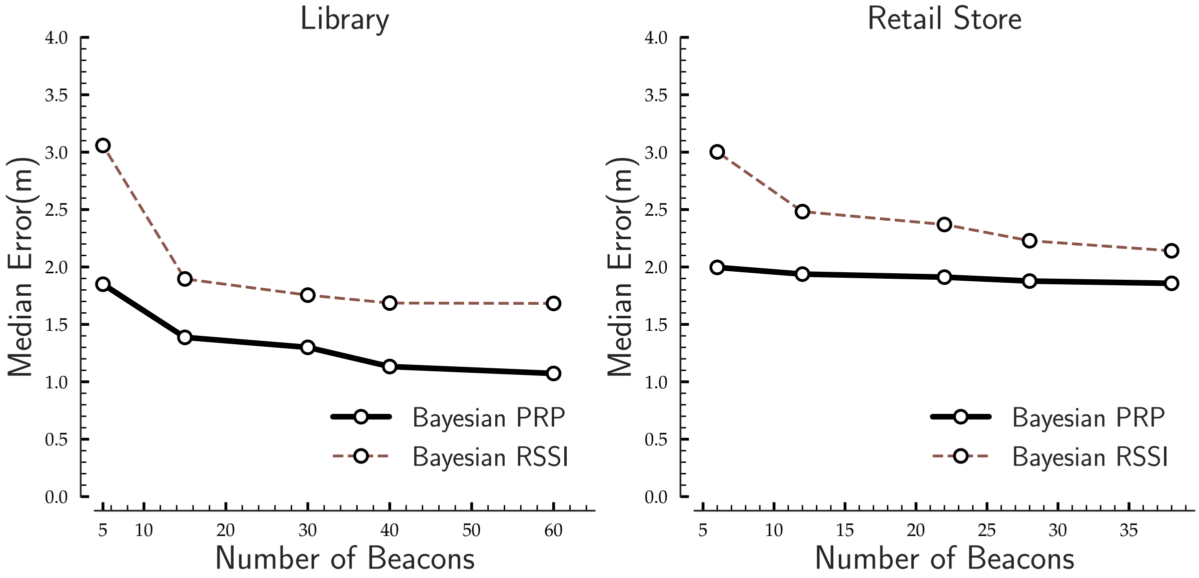


Figure 3.9: **Variation in median error for B-PRP with beacon number**. The error is within $2m$ for all cases. With 5 beacons, B-PRP performance is better than Bayesian RSSI: 65% (library) and 50% (retail store).

Line-Of-Sight Condition	distance $< 2m$			$2m <$ distance $< 6m$			distance $> 6m$		
	B-PRP	RSSI	B-PRP + RSSI	B-PRP	RSSI	B-PRP +RSSI	B-PRP	RSSI	B-PRP +RSSI
LOS	0.89m	0.5m	0.5m	0.63m	3.57m	0.62m	1.53m	3.85m	1.56m
Non-LOS	0.85m	1.05m	0.57m	5m	5.95m	5.62m	2.07m	5.15m	2.7m

Table 3.2: **Robustness To Beacon Placement:** Recall, our median error using all beacons is $1.03m$. If we use only beacons that are closer than $2m$ to the receiver, both PRP and RSSI errors are low. Fusion of PRP and RSSI gives even lower errors of $0.5m$. In this range, RSSI values have less variance and more distance information. With beacons further than $2m$, RSSI variances increase which cause fusion results to be worse. PRP gives high error if beacons are between $2m$ and $6m$ in a Non Line-of-Sight scenario. This is expected because the variance in the PRP estimates are high in this region. PRP again gives low error if beacons are further than $6m$.

to ask :*how do we impact localization accuracy when we decrease the number of beacons?*. We evaluated the accuracy with fewer number of beacons (lower bound is set to three beacons – the minimum required to localize).

In Figure 3.9, we see B-PRP performance degrades slowly than Bayesian-RSSI to decreasing beacon density. The median error of localization for B-PRP is always within $2m$. For Bayesian-RSSI, with lower beacons, the error is as high as $3m$. With 5 beacons, B-PRP performance is 65% better than Bayesian-RSSI in the library and 50% better in the retail store. Also, note that just with 5 beacons, B-PRP performs better or equal to Bayesian-RSSI with upto 60 beacons. This, yet again, demonstrates that the errors in RSSI-based positioning cannot be solved by just additional deployments, but are fundamental (sampling bias and multipath).

Beacon Placement: *How does the placement of a beacon with respect to the receiver impact the localization accuracy by B-PRP and Bayesian RSSI?* If we use only beacons that are closer than $2m$ to the reception location, both PRP and RSSI errors are good (c.f. Table 3.2). In fact, RSSI performs slightly better in Line-of-Sight scenario due to the less variance in RSSI values and more distance information at very close range. Fusion of B-PRP and RSSI also yields lower errors. When beacon distances become greater than $2m$, RSSI errors dramatically increase due to variance in RSSI values caused by multi-path and sampling bias. In comparison, PRP errors are much lower in order of $1.53m$ and $2.07m$ when beacons are more than $6m$ away from the receiver. Errors in RSSI also cause fusion results to be worse. Error for all approaches is high when we use only beacons, all of which are in a Non Line-of-Sight(NLOS) scenario and are at a distance between $2m$ and $6m$ from the receiver. This experiment highlights the importance of PRP. As RSSI estimates suffer from higher sampling bias with increasing distance, the underlying location information gets corrupted. This is why at larger distances, both Bayesian RSSI and B-PRP+RSSI do worse.

3.9.3 Minimizing beacon set-up cost

N_R	$b = 60$	$b = 6$	$b = 3$	$b = 1$
12	1.03	1.05	1.24	1.38
8	1.05	1.22	1.82	2.15
4	1.05	2.88	3.74	3.48

Table 3.3: B-PRP's median localization error (in m) with varying number of known beacon locations b, and number of training locations N_R . Error increases as we decrease b (each row) and decrease N_R (each column). For $N_R = 12$, performance is almost same as with $b = 60$ and $b = 6$. Decreasing N_R impacts accuracy more than does b.

So far, we have used location information of all B beacons. **Now, we will use the location information for only $b \ll B$ primary beacons.** For the rest of the section, we use B as total number of beacons and b as the number of beacons with known location information. We use data to estimate $B - b$ unknown beacon locations. We then use these estimated values to track a receiver.

We vary the number of primary beacons $b = \{1, 3, 6, 60\}$. $b = 60$ corresponds to when we know all beacon locations. B-PRP with $b = 60$ serves as our baseline. We also vary the value of N_R i.e. the total number of training locations. Since we make the beacon locations an unknown parameter in our framework, we also want to get a sense of the number of training locations required to efficiently estimate all these unknowns and retain accuracy level. Ideally,

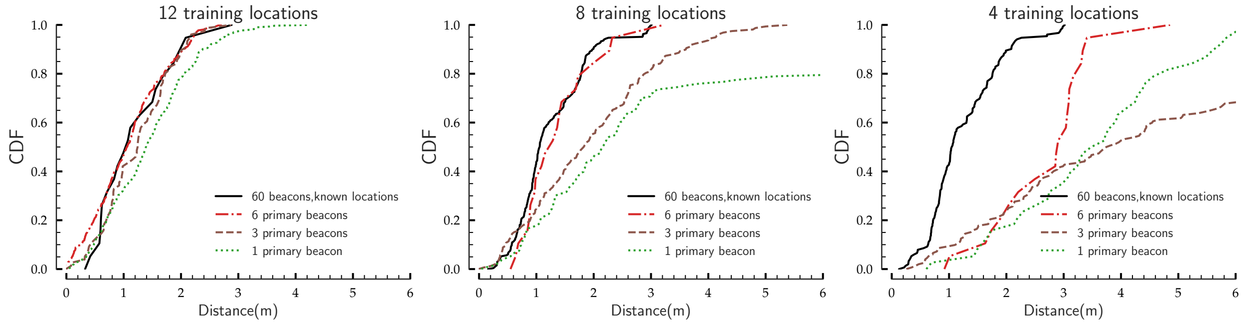


Figure 3.10: **Reducing beacon set-up cost:** CDF for comparing localization errors of B-PRP as we vary the number of known beacon locations(or primary beacons). As we decrease the number of known beacon locations by order of magnitude from 60 to 6, we hardly see any increase in error when we have 12 training locations. If we reduce training locations to 4, we need more primary beacons to retain same accuracy.

we would not like to use a lot of training locations to compensate for the unknown beacon location, as this adds to the set-up cost.

We show the results in Figure 3.10 and Table 3.3. We highlight three observations. *First*, when ($N_R = \{12, 8\}$) there is negligible difference in the CDF of the tracking errors between the cases of $b = 60$ and $b = 6$. *Second*, for any value of N_R, the errors increase when we decrease b, with the effects most pronounced for $N_R = 4$. *Finally*, the figures suggest that the effect of unknown beacon locations is *less significant* than the effect of the number of training locations. B-PRP can give the same level of performance with as low as $b = 3$ primary beacons when the number of training locations N_R is high. If we reduce N_R to 8, we need at least $b = 6$ known beacons.

These results highlight that B-PRP can be a low-overhead method for public spaces with little deployment overhead. A retail store operator may just place these beacons across each aisle, and move around with a smartphone to some known locations. B-PRP can infer the beacon location on its own (for most beacons) and still achieve competitive performance.

3.9.4 Reducing training efforts

Till now, we have used the location information of all training spots N_R while training. Now, **let's use the information for only** $r < N_R$ **training spots** and estimate the remaining $N_R - r$ locations using our framework.

We change the value of known training locations $r = \{12, 8, 6, 4, 2, 0\}$, with $N_R = 12$. Figure 3.11 shows the results. In the leftmost sub-figure, we see that as r decreases, error increases; but notice that we can cut the known locations in half, from $r = 12$ to $r = 6$,

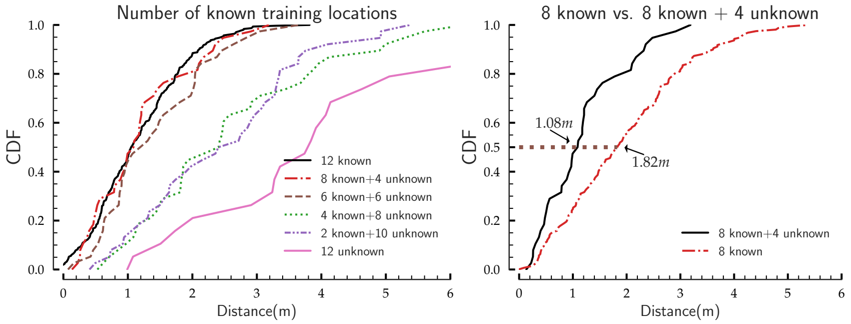


Figure 3.11: **Reducing retraining efforts:** CDF for comparing errors of B-PRP when we train using data from some known and mostly unknown locations. If we have data from 12 known locations vs (6 known + 6 unknown) locations, we get the same accuracy level. In the right two subfigures, we show that we improved accuracy $\sim 40\%$ by adding data from unknown spots rather than only using data from known spots.

without appreciable increase in error. This means that we can collect data from 12 spots but need to annotate only half of those and B-PRP can still maintain the same accuracy level. One might wonder, *do we really gain any performance improvement by adding data from unknown locations?* Figure 3.11 (two right sub-figures) validate that conjecture. Suppose, our training dataset contains data from 12 training locations in total. Now, 8 of those are labeled with location information while 4 are unlabeled. If we train PRP parameters using only 8 labeled data locations, our median error from the trained model is $1.82m$. In contrast, if we use the entire dataset and treat the location of the 4 unlabeled data points as random variables in our framework, we improve the median error to $1.08m$. Similarly, if we have 4 labeled and 8 unlabeled locations, by using all the locations our errors improve from $3.8m$ to $2.4m$. Thus, data from un-labled locations are valuable for training PRP parameters. This further eases the deployment cost by allowing operators to collect fewer labelled data points.

3.10 DISCUSSION AND LIMITATIONS

Now we discuss few application scenarios and limitations of B-PRP technique.

Applicability to general indoor environments: We design B-PRP with a focus on public indoor environments like retail stores that have stacked layouts. This layout is applicable to multiple spaces like libraries, warehouses, pharmacies, etc. and covers an important application area. While the current multipath-resilience model of B-PRP does not directly apply to other environments like homes, we believe PRP itself is applicable to such environments and provides the unique advantage of robustness at large distances. Furthermore, in such environments, obstacles like walls can be modelled using the approach followed in B-PRP.

Access to Layouts: We design the layout requirement for B-PRP to be low-effort. The layout and stacks can be extracted from the store floorplan, either manually or through an app. This makes the deployment effort low. Furthermore, B-PRP can apply to store layouts with more stacks. We may encounter geometric elements like *three stacks away* $(3 - S)$, *four stacks away* $(4 - S)$ etc. We do not necessarily need a separate PRP function for each of these elements. Since PRP becomes very low after certain number of stacks, we can club these spaces into one geometric element and learn a single model.

Computational complexity: Bayesian MCMC techniques may take more time to infer location. We ran our computations in python on a MacBook Pro laptop with $2.5GHz$ Intel Core i7 processor and $16GB$ RAM. With 60 beacons, it took us ~ 3 seconds to find the next location, within our time resolution ($\delta = 10$s) for localization. We can further speed-up by using native code and parallelizing the inference.

Scalability to the number of packets: One limitation of B-PRP is that it needs more than one packet to localize. We can reduce the number of packets used for localization by changing the advertising frequency. We observe in our experiments that as we lower the sending rate from $10Hz$ to $1Hz$, while keeping the localization rate to once per 10 seconds, the median error increases by just $0.2m$.

3.11 CONCLUSION

This work establishes the feasibility of using Bluetooth Low-Energy (BLE) to provide a robust, scalable indoor localization solution using commodity hardware. Demonstrating the feasibility of BLE based distance estimation technique is particularly important during the current pandemic, where BLE has emerged as key technology for contact-tracing. BLE-based distance estimation today relies on either RSSI or just presence, both of which have publicly documented failure modes. We analyze the fundamental underpinnings of these failure modes and demonstrate robust localization through the Bayesian formulation of a new metric—Packet Reception Probability–that *exploits the absence of received packets.* We show significant improvements over the state of the art RSSI methods in two typical public spaces—a retail store and a library. We show that fusing B-PRP with RSSI is beneficial at short distances ($\leq$2 m). Beyond $\geq$2 m, fusion is worse than B-PRP, as RSSI based estimates beyond $\geq$2 m are effectively de-correlated with distance. Our solution does not require any hardware, firmware, or driver-level changes to off-the-shelf devices, and involves minimal deployment and re-training costs. In the next chapters, we extend our framework for peer-to-peer distance estimations in outdoor settings where we cannot install any beacons, and use only smartphones to estimate distance.

CHAPTER 4: CONTACT TRACING WITH MINIMAL INFRASTRUCTURE

4.1 INTRODUCTION

Infectious diseases spread due to close contact between people. As per CDC guidelines, COVID-19 can spread from an infected person to another if they are closer than 6ft for over 15 minutes. An infected person can potentially spread the virus to many other people before the appearance of any symptoms [103]. A potentially exposed person needs to receive quick alerts about avoiding contact with other people. Minimizing contact between infected persons and non-infected people is an essential requirement to curb the spread of the disease [60, 61]. Current contact tracing approaches mainly rely on manual interviews.

We can enable automated contact tracing through apps deployed on our smartphones. Bluetooth Low-Energy (BLE) is emerging as the key technology in contact-tracing apps around the world. These apps continuously broadcast BLE packets, which are received by surrounding devices. The apps on the receiver side use certain properties of the received packets to infer distance from the sender. The Aarogya Setu contact-tracing app [7] in India and PACT program [104] use BLE packet signal strength to infer contact. The open-source, privacy-preserving contact-tracing framework, BlueTrace [8] (deployed in Singapore), uses BLE packets to detect the presence of another device. (i.e., a smartphone that can hear another must be in proximity of the other.)

BLE apps that trivially rely on pairwise measurements like received signal strength to measure contact between two people have a well-known shortcoming. Latent factors like device relative positioning on the human body, the orientation of the people carrying the devices, and environmental multi-path effect can impact the measured power besides distance. We can receive the same signal strength value for two people standing 3ft apart facing away from each other and two people facing each other but standing more than 6ft apart. BLE apps that use presence suffer higher errors. Presence is a poor proxy for distance since devices can hear Bluetooth beacons beyond 6 ft social distancing radius and hear them across aisles and walls. In this chapter, depending on the context of the environment, we provide two solutions to solve challenges with peer-to-peer measurements.

4.1.1 Indoor, minimal infrastructure

First, we ask the question—*Can we provide better estimates of contact tracing distances between two people using minimalistic infrastructure in indoor environments?* For example, two people are present in a cafe. We are interested in inferring whether or not they are in contact. Direct BLE measurements between the phones of these two people are prone to errors due to factors like—are the two people facing each other or away from each other, are the phones in their pockets or on their tables. Can we provide better estimates of contact if the cafe had few beacons installed in the environment? We assume that the beacons are installed at known locations, and we know the distance between these beacons. The phones will be receiving packets from the beacons. We want to infer the distance between the two phones based on the properties of received packets from the beacons and using a known distance value between the beacons.

To solve the challenge with direct peer-to-peer measurements, we make the following contributions:

Correlated errors in localization-based solution: An easy way to obtain the distance between two people is through localization: we first estimate locations of two persons independently and then calculate the Euclidean distance between the two locations. We identify that this approach is sub-optimal—we are estimating location (a tuple) while we are interested only in the distance (a scalar). Also, if we have localization errors for a particular individual, these errors will impact *all* the distance estimations between this individual and other nearby persons. In other words, we end up with correlated errors for distances between different pairs of individuals.

Triangle inequality-based solution: We directly estimate the distance between two individuals *without localization* by exploiting the well-known triangle inequality constraints in Euclidean geometry. We extend our Bayesian framework to estimate distances between pairs of individuals independently. We form triangles with the known beacons, and we impose triangle inequalities on these distances to rule out many distance configurations in the real world. The improvements are significant: we improve our distance estimates by $\sim 10\%$ by moving from a location-based distance estimation to a triangle inequality-based distance estimation.

We conducted experiments in public spaces: a library and a retail store with known distances from Bluetooth beacons and made the following findings.

Distance estimation for contact tracing: Our contact tracing distance estimation framework achieves median error of 0.89 m (library) and 1.07 m (retail store) with PRP values. The corresponding errors with RSSI are 1.36 m (library, 52.8% more error) and 1.34 m (retail store, 25.2% more error). Using the covid risk metric proposed in [41], we see that PRP does 1000X better than RSSI in the library.

Error with decreasing beacon number: With only five beacons, PRP error for contact tracing remains around 1.5 m for both library and retail store.

4.1.2 Outdoor, no centralized infrastructure

Second, we extend our work to outdoor environments where we cannot install extra infrastructure. Instead, we propose using collaboration to solve distances robustly. Without installing added infrastructure, we cannot leverage known distances in our triangle inequality framework. However, we can leverage triangles formed between three or more people in the vicinity. We constrain the unknown distances using triangle inequality. With more people in the vicinity, we can form more triangles and constrain the unknown distances further.

In the second part of our work, we make the following contributions:

Collaborative triangle inequality framework: We designed a Bayesian framework that includes the likelihood function based on pairwise measurements between people and triangle inequality-based likelihood values.

Android and iOS contact tracing application: We developed Android and iOS applications to exchange Bluetooth packets for contact tracing.

BLE packet structure: We designed and implemented BLE packet structure and protocol to enable collaborative contact tracing between people present in the vicinity.

Our empirical experiments revealed that PRP does well even when there are interfering $\in [1, 2]$ human bodies between two phones. In a sample experiment, for one interfering human body, PRP is 0.6 at $3ft$ while it is 0.2 at $12ft$. With no interference, PRP remains constant at 0.7 for both $3ft$ and $12ft$, while mean RSSI is $-70db$ at $3ft$ and $-80db$ at $12ft$. The experiment suggests that PRP and RSSI should be used in conjunction to give better estimates of the distance across all types of interference. Our experiments also reveal a new challenge of intra-device variation in PRP and RSSI values due to different kinds of antenna designs in different phones. To deploy PRP/RSSI-based contact tracing applications at scale in different phones, we need future research into Model Adaptive Machine Learning to learn RSSI and PRP models that can be adapted/generalized to different devices.

4.2 RELATED WORK

Several contact tracing applications exist worldwide. [60, 61] shows the applications have been useful in curbing the spread of the disease. [62] contains a full list. These apps fall into four broad categories based on the technology they use to find the proximity of two persons.

1. **GPS/Cellular**: The apps like ViruSafe [63], Rakning c-19 [64] use GPS or triangulation from nearby cell towers to find the location of a phone on a person. If two phones have been in the exact location simultaneously, the apps infer the two people to be in proximity. Applications relying on GPS or cellular data are less accurate than Bluetooth-based technologies since they can infer far away people to be in proximity. They are more privacy-invasive than Bluetooth since they can find the absolute location of people.

2. **Bluetooth**: The apps like Virusradar [65], Covidradar [66] use Bluetooth in phones to swap encrypted tokens with any other nearby phones. If a phone receives the Bluetooth token from another phone, then the two persons owning the phones are inferred to be close. Apps relying on Bluetooth perform better than GPS on privacy since tokens can be anonymized. However, Bluetooth packets can reach as far as $30ft$ and detect faraway people in proximity.

3. **Apple/Google Exposure**: Apps like Stopp-corona [67] in Austria, Covid-alert [68] in Canada, GuideSafe [69] in Alabama USA, CovidDefense [70] in Louisana, etc. rely on the joint API developed by Apple and Google that allows iOS and Android phones to communicate with each other over Bluetooth.

4. **DP-3T**: Apps like SwissCovid [71], Coronalert [72] use decentralized privacy-preserving proximity tracing protocol over Bluetooth. It's an open-source protocol in which an individual phone's contact logs are only stored locally, so no central authority can know who has been exposed.

Our work is different in three main aspects—we measure distance based on packet reception probability of BLE signals, a new metric that we described in Chapter 3. We use minimalistic infrastructure installed in the environment to improve our contact tracing distance estimates. We constrain distance estimates using geometric constraints—these constraints sharpen with an increase in the number of individuals. Next, we will discuss the infrastructure-assisted contact tracing system, where we will see how we can improve all Bluetooth-based contact tracing applications by installing few Bluetooth devices in an environment that can communicate with Bluetooth phones on a person.

4.3 INFRASTRUCTURE ASSISTED CONTACT TRACING

Now, we will discuss the infrastructure-assisted contact tracing system where we install a few beacons in the environment to estimate contact between people. We will discuss the problem statement in Section 4.3.1. We will detail the challenge of trying to get distance through finding locations in Section 4.3.2 and then propose a Euclidean constraint-based framework where we can directly infer distance.

4.3.1 Problem statement

In this work, we focus on finding distance between individuals in indoor public spaces. We will use BLE signals to infer distance. BLE offers a unique advantage for contact tracing. Due to its low power budget, it can be turned on frequently and hence, enable more frequent updates as compared to high power protocols like Wi-Fi. Recall that BLE's maximum transmit power (10 dBm) is ten times lower than that of Wi-Fi (20 dBm). This factor, in addition to its ubiquitous presence on off-the-shelf smartphones, has made BLE the natural choice for such applications.

We deploy few BLE beacons at fixed locations in the indoor environment to aid distance estimation. We show that some minimal infrastructure support will aid in more accurate indoor contact tracing. The beacons are configured to emit Bluetooth packets, power p_t, and a fixed sending rate, R. The smartphone of an individual navigating the public space listens to these packets and logs them in the following form:

$$L = \{(b_1, t_1), (b_2, t_2), \ldots, (b_N, t_N)\} \tag{4.1}$$

where b_i refers to the BLE beacon `id` heard at time t_i. The phones then upload the log to a central server. We use packet log L_1 and L_2 uploaded by two individuals to find the distance between them.

4.3.2 Solution using Euclidean constraints

We could trivially extend the Bayesian framework discussed in Chapter 3 to estimate the distance between two individuals using a two-step approach: first, estimate their locations independently, and second, calculate the Euclidean distance between the two locations. Then, we could use this distance to ascertain whether two individuals were in contact for contact tracing. However, this approach is sub-optimal. It requires us to determine four unknowns — (x, y) for the two devices, while we are only concerned about the final distance estimate between the two devices. Also, if we make errors in location estimation for an individual, that impacts all the distance estimations of this individual with other neighboring persons. In other words, we get correlated errors for independent distances between different pairs of individuals. **Can we do better?**

At a high level, we can improve the distance estimation process using two insights. *First*, we don't need to model individual locations if we just care about distance. Therefore, we explicitly incorporate the distance between two devices as part of our Bayesian model. This helps us reduce the number of unknowns in our framework and also helps to model the distance between each pair of individuals as an independent unknown. *Second*, we leverage the triangle inequality. The triangle inequality states that given a triangle, the sum of two edges has to be greater than or equal to the third edge. This helps us rule out many triangular distance configurations. We present a detailed formulation of these insights below.

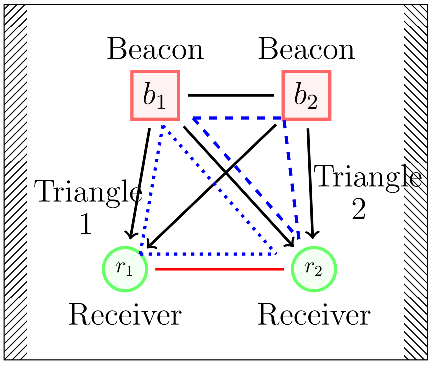


Figure 4.1: Modeling contact tracing distance optimizing joint likelihood of observed PRP values and triangle inequalities

Finally, one might wonder: why do we need all this complexity? Why don't we just use the direct transmission between two devices to estimate distance—device A transmits to device B, device B measures PRP, and we convert that to distance? This approach would create a posterior distribution for distance but with a large variance because interference by other nearby persons increases the uncertainty in our posterior distance distribution. To shrink this variance, we need other distance measurements, either to known beacons or many other peers.

Indoors, we adopt an infrastructure-assisted approach described above to help compute distances between pairs of individuals. Given that public indoor spaces like retail stores or restaurants (or even businesses) are more likely to be crowded, the infrastructure-assisted approach is reasonable in these settings.

Objectives and Data: We have N_R people. We are interested in finding the $\binom{N_R}{2}$ contact tracing distances $\{d_{i,j}^{R-R}\}, i, j \in \{1, \ldots, N_R\}, i \neq j$. For each person $r \in N_R$, we have PRP data to each beacon $\{d_{r,b}^{R-B}\}, b \in \{1, \ldots, B\}$ allowing us to estimate distance to the beacon. Additionally, we assume that we know the inter-beacon distances $\{d_{i,j}^{B-B}\}, i, j \in \{1, \ldots, B\}, i \neq j$. In this problem, we have three types of distances—receiver-to-receiver

$(R-R)$ which is our final objective, receiver-to-beacon $(R-B)$ which are latent variables on which we have PRP data, and beacon-to-beacon $(B-B)$ which are known constants in our framework.

We solve the problem in two phases—first, estimating the latent variables, receiver-to-beacon distances d^{R-B} by maximizing a joint likelihood function of the observed prp data and triangle inequalities containing all possible triangles with combinations of two beacons and one receiver. Second, given the estimated values of the latent variables in the first step, we estimate the contact tracing distances d^{R-R} by maximizing the likelihood of inequalities containing all possible triangles with the combination of two receivers and one beacon.

Toy Example: We explain our solution using a toy example (pictorially represented in Figure 4.1) which contains two beacons b_1 , b_2 and two receivers r_1 , r_2. We are interested in finding the distance d_{r_1,r_2}. The latent variables are $(d_{b_1,r_1}, d_{b_1,r_2}, d_{b_2,r_1}, d_{b_2,r_2})$, on which we have prp data. d_{b_1,b_2} is known.

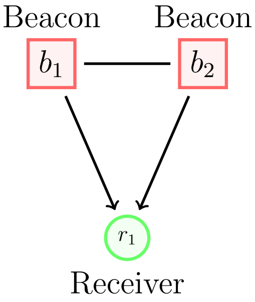


Figure 4.2: Triangle inequality

In the first phase, we construct a joint likelihood function that contains two components—observed PRP values and triangle inequalities. The first component is similar to PRP based location estimation where we maximize the likelihood $Pr(prp_{x,y}|d_{x,y}), x \in \{b_1, b_2\}, y \in \{r_1, r_2\}$. For the second component, recall the geometric property that for any triangle, the sum of two sides of the triangle must be greater than the third side. Taking the receiver r_1 as an example, we have the triangle (r_1, b_1, b_2) which gives us three triangle inequalities that can be converted to likelihood values as

$$P_T = Pr(d_{r_1,b_1}+d_{r_1,b_2}-d_{b_1,b_2} > 0)\times Pr(d_{r_1,b_1}+d_{b_1,b_2}-d_{r_1,b_2} > 0)\times Pr(d_{r_1,b_2}+d_{b_1,b_2}-d_{r_1,b_1} > 0) \tag{4.2}$$

For estimating the latent variables involving the receiver r_1, we can write down the joint log likelihood function as:

$$L_{PRP} = \max_{(d_{r_1,b_1}, d_{r_1,b_2})} [\log Pr(prp_{r_1,b_1}|d_{r_1,b_1}) + \log Pr(prp_{r_1,b_2}|d_{r_1,b_2}) + \log P_T] \tag{4.3}$$

Intuitively, if three distances on a geometric plane form a triangle, they should satisfy Euclidean constraints by default. However, if we solve the distances using PRP values only, we are solving each distance independently, and the samples drawn for one distance do not affect the other distance. We need to explicitly enforce the euclidean constraints between three distances that form a triangle.

Similarly, we can write down the joint likelihood function involving r_2. Note that in the joint log-likelihood for N_R receivers and B beacons, we will have $N_R \times B$ terms for the observed PRP values, and $3 \times N_R \times \binom{B}{2}$ terms for the triangle inequalities. We apply MCMC sampling techniques to solve these joint likelihood functions.

In the second phase, we maximize the likelihood of triangle inequalities involving triangles with two receivers and one beacon. For the distance d_{r_1,r_2}, we have two triangles $T_1 = (r_1, r_2, b_1)$ and $T_2 = (r_1, r_2, b_2)$. Note that here we maximize the likelihoods given the latent variables that we inferred in the previous step i.e. $(d_{b_1,r_1}, d_{b_1,r_2}, d_{b_2,r_1}, d_{b_2,r_2})$. Now, let $A = d_{r_1,r_2}, B = d_{r_1,b_1}, C = d_{r_2,b_1}$ and $\Delta_{A+B-C} = Pr(d_{r_1,r_2} + d_{r_1,b_1} - d_{r_2,b_1} > 0)$

We can construct the euclidean constraint likelihood function as below:

$$L_{T_1} = \log \Delta_{A+B-C} + \log \Delta_{A+C-B} + \log \Delta_{B+C-A} \tag{4.4}$$

Similarly, we can write the euclidean likelihood for triangle T_2. We can write the combined likelihood as

$$L = \max_{(d_{r_1,r_2})} [L_{T_1} + L_{T_2} | d_{b_1,r_1}, d_{b_1,r_2}, d_{b_2,r_1}, d_{b_2,r_2}] \tag{4.5}$$

Note that in the joint log-likelihood for N_R receivers and B beacons, we will have $3 \times B$ terms in the likelihood function for each receiver-to-receiver distances d^{R-R}. We measure the absolute error between estimated and ground-truth d^{R-R}. In Section 4.4, we compare the median errors of PRP with baselines that use the same likelihood estimation above, but with different observed values—RSSI and PRP+RSSI. For RSSI based technique, we write down Equation (4.3) as:

$$L_{RSSI} = \max_{(d_{r_1,b_1}, d_{r_1,b_2})} [\log Pr(rssi_{r_1,b_1} | d_{r_1,b_1}) + \log Pr(rssi_{r_1,b_2} | d_{r_1,b_2}) + \log P_T] \tag{4.6}$$

The triangle inequality component remains the same, while we use RSSI as observed value in the first component. Similarly for PRP+RSSI, the formulation becomes:

$$\max_{(d_{r_1,b_1}, d_{r_1,b_2})} [L_{PRP} + L_{RSSI} - \log P_T] \tag{4.7}$$

In all three cases, the second phase of the estimation remains the same. The generalization to N_R receivers and B beacons from Equation (4.6) and Equation (4.7) is straightforward.

4.4 EVALUATING CONTACT TRACING DISTANCE ESTIMATES

We conducted empirical experiments in two testbeds—an academic library and a retail store. The library has a floor area of $14m$ by $8m$ while the retail store dimensions are $10m$ by $10m$. We installed iBeeks or iBeacon in our environment to act as infrastructure devices. We used Texas Instrument Packet Sniffer (CC2540 dongle) as the receiving device. We collected data at multiple reception spots throughout the layout. Each reception spot serves as a proxy for one person. Our goal is to find the contact tracing distance between each pair of people. In other words, we want to find the distance between each pair of reception spots.

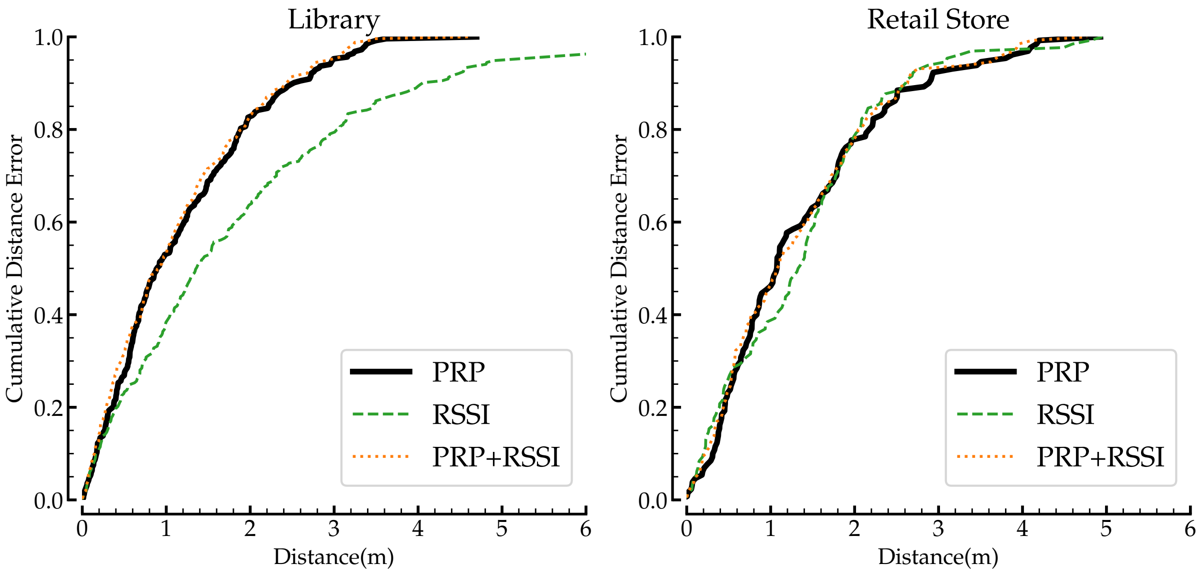


Figure 4.3: **CDF error distribution of contact tracing distance** for PRP, RSSI and PRP+RSSI in library and retail store. PRP and PRP+RSSI gives the best median error in both environments.

We compare the accuracy of PRP against RSSI and PRP+RSSI in contact tracing distance estimation. We measure the absolute error between actual and estimated distances for each pair of person or receivers. We show cumulative distribution over errors in Figure 4.3.

First, observe that PRP achieves a median distance error of $0.89m$ and $1.07m$ in the library and retail store. RSSI achieves distance errors of $1.36m$ and $1.34m$. PRP+RSSI gives median distance errors of $0.91m$ and $1.08m$.

Now, notice that for a contact tracing application, errors in all distance estimations $d_{i,j}$ between a pair of individuals i, j are not equally important. When the actual distance is

$d_{i,j} < 2m$ (the social distancing range), the errors become more important. When two individuals are far away from $d_{i,j} \gg 2m$, errors become less important. We used a standard risk function described in [41] to map the distance values $d_{i,j}$ between a pair of individuals i, j to a risk metric $R_{i,j}$:

$$R_{i,j} = \frac{1.0}{1 + \exp\{\alpha(d_{i,j} - 2m)\}} \tag{4.8}$$

,where, $\alpha = 9.17$.

Notice that the risk is a sigmoid function that takes high values $R_{i,j} \approx 1$ when $d_{i,j} < 2m$, and low values $R_{i,j} \approx 0$ when $d_{i,j} \geq 2m$. We calculate the absolute error between the actual risk (based on ground truth distance) and predicted risk (based on the estimated distance). PRP gives a median risk error of 5×10^{-6} and 0.16×10^{-6} in the library and retail store, while RSSI achieves 4×10^{-3} and 0.22×10^{-6} respectively. Note that the median risk error with RSSI in the library is 1000 times larger compared to PRP. PRP+RSSI has risk error of 1.3×10^{-6} and 0.06×10^{-6}. Also, though the risk errors look small, at population scale and across multiple interactions, the risk is very high.

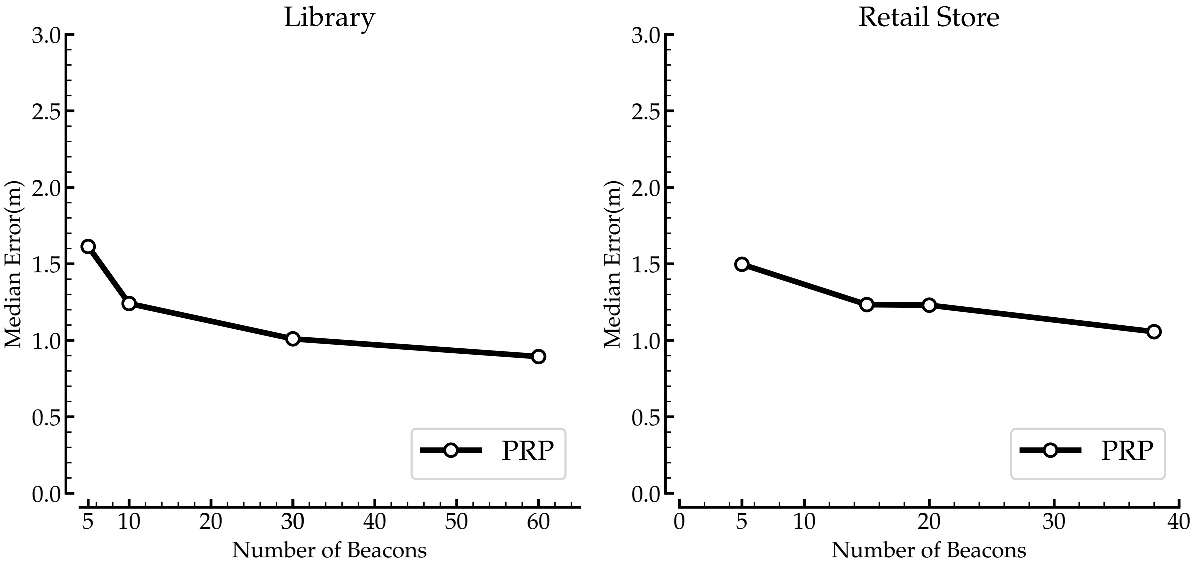


Figure 4.4: **Variation in median error for contact tracing performance with decreasing beacon number**. The error is around $1.5m$ for all cases.

4.5 INDOORS: IMPACT OF BEACON NUMBER

Now, we look at the contact tracing performance when we decrease the number of beacons. In our test beds, we initially had 60 beacons in the library, and 38 beacons in the retail store.

Now we evaluate the accuracy with fewer number of beacons.

In Figure 4.4, contact tracing performances degrades slowly with decreasing beacon number, but always stays around $1.5m$.

4.6 OUTDOORS: INFRASTRUCTURE FREE CONTACT TRACING

Till now, we discussed a contact tracing solution where we proposed the installation of a few BLE beacons in an indoor environment to obtain a robust estimate of distances between pairs of people. However, the infrastructure-assisted solution will be infeasible for outdoor environments where we cannot install such infrastructure. Now we discuss an application that infers contact tracing distances between people without any added infrastructure. Here, we wish to utilize the measurements made between other people in the vicinity collaboratively to improve each contact tracing distance estimate.

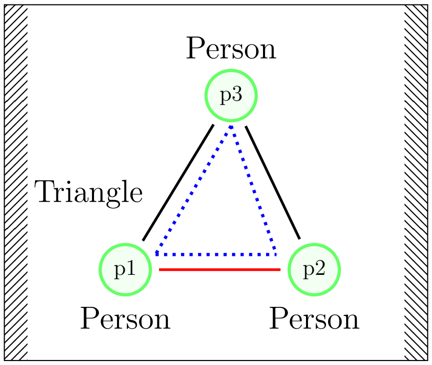
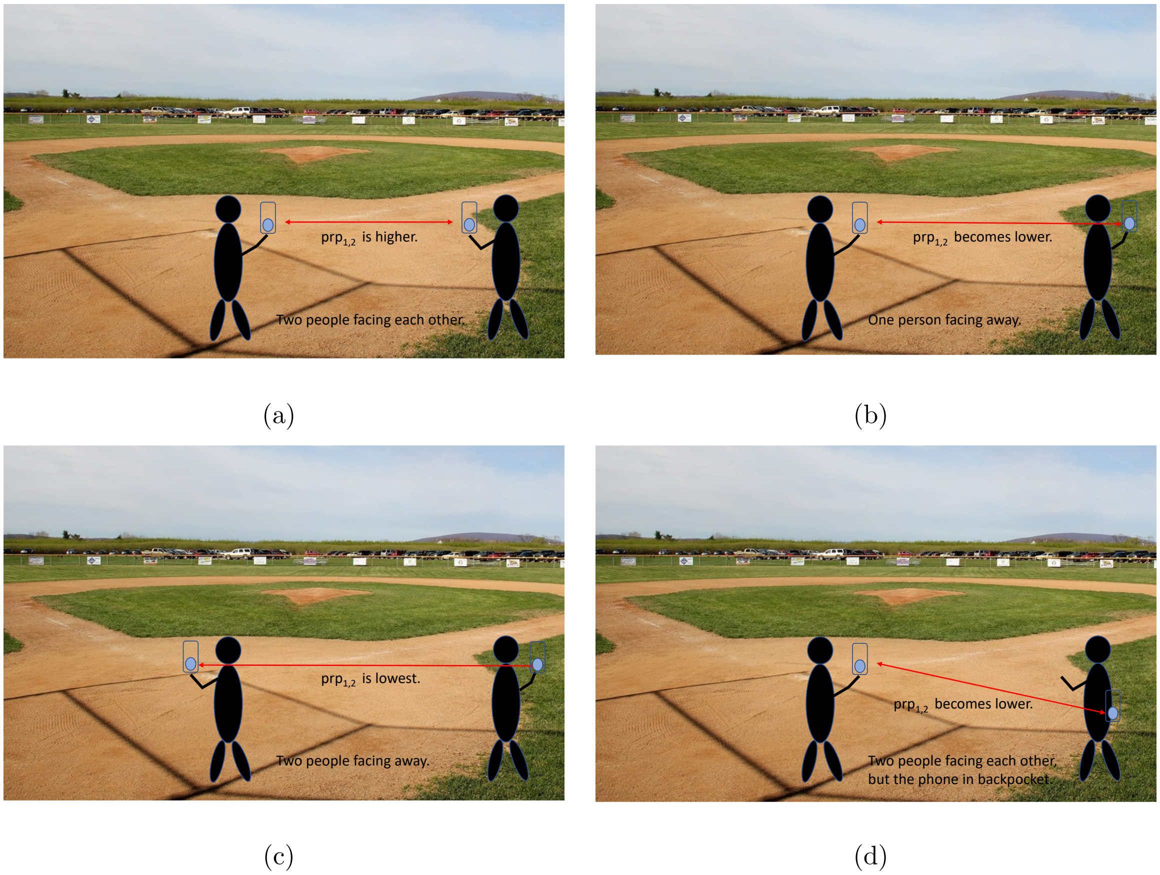


Figure 4.5: Infrastructure free contact tracing: optimizing joint likelihood of observed PRP values and triangle inequality between peers.

4.6.1 Challenge

One might wonder: why can't we just use the direct transmission between two devices to estimate distance—device A transmits to device B, device B measures PRP, and we convert that to distance? Besides distance, other latent factors impact PRP—orientation of the people and relative location of the device on a person. Standing at the same distance, if two people are facing each other Figure 4.6(a), we will measure higher PRP compared to if they are facing away from each other Figures 4.6(b) and 4.6(c) This is because human bodies act as interference to signals and cause signal strength attenuation. Similarly, we will get more PRP when two people have their phones in the palm of their hands vs. they have kept the phones in their pockets Figure 4.6(d).


$prp_{1,2}$ is higher.
Two people facing each other.
$prp_{1,2}$ becomes lower.
One person facing away.
$prp_{1,2}$ is lowest.
Two people facing away.
$prp_{1,2}$ becomes lower.
Two people facing each other,
but the phone in backpocket.

(a)

(b)

(c)

(d)

Figure 4.6: (a) 2 people are facing each other and there is a clear line-of-sight between the 2 phones, PRP value is high. (b) When one of the person faces away, there is now a human body blocking the line-of-sight and prp decreases. (c) If both people face away, there are 2 human bodies blocking the line-of-sight and prp decreases further. (d) When two people face each other, but one of them is keeping the phone in their backpocket, then also we have a human body blocking the line-of-sight.

4.6.2 Solution

For each contact tracing edge, we will need to solve a latent variable in addition to distance—the attenuation/interference factor. If two people are facing each other and have the phones in their hands, the attenuation factor is low. If they have the phones in their back pockets, the attenuation value is high. We will use collaboration between people in the crowd and impose triangle inequality constraints similar to Section 4.3 to jointly solve distance and attenuation factors on each edge.

We explain our solution using a toy example (pictorially represented in Figure 4.5) which contains three people p_1 , p_2 and p_3. We are interested in finding the distance $d_{p_1,p_2}, d_{p_2,p_3}, d_{p_1,p_3}$.

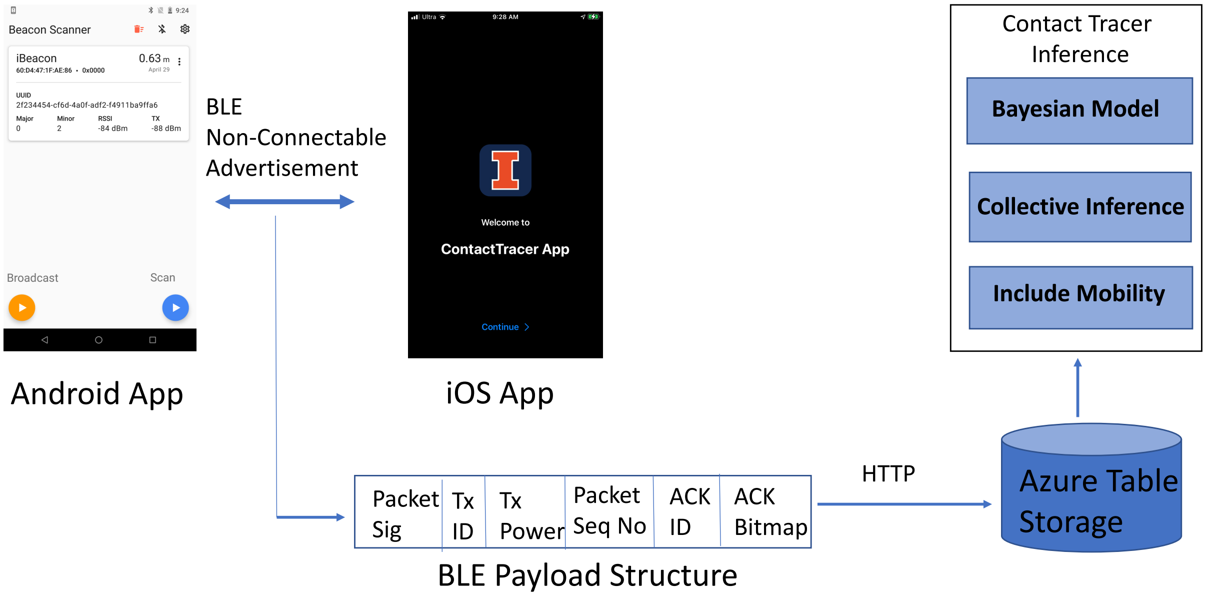


Figure 4.7: Contact Tracer Architecture, which includes a fully functioning android and iOS app, payload structure of the BLE packets that are sent out by our apps, cloud infrastructure in Azure to receive and store the BLE packet logs through HTTP requests, and the contact tracer inference engine. The inference engine includes the Bayesian distance estimation model, the collective triangle inequality model and a mobility based model.

We construct a joint likelihood function that contains two components—observed prp values and triangle inequalities. We can write down the equations as

$$L_T = \log P(d_{p_1,p_2} + d_{p_2,p_3} - d_{p_1,p_3} > 0) + \log P(d_{p_1,p_2} + d_{p_1,p_3} - d_{p_2,p_3} > 0) \tag{4.9}$$

$$+ \log P(d_{p_1,p_3} + d_{p_2,p_3} - d_{p_1,p_2} > 0), \tag{4.10}$$

$$\max_{(d_{p_1,p_2}, d_{p_2,p_3}, d_{p_1,p_3})} [\log P(prp_{p_1,p_2} | d_{p_1,p_2}) + \log P(prp_{p_2,p_3} | d_{p_2,p_3}) \tag{4.11}$$

$$+ \log P(prp_{p_1,p_3} | d_{p_1,p_3}) + L_T] \tag{4.12}$$

As we have more people in the vicinity, we can form more triangles and constrain the distances further. If we have N people in the vicinity, we can form $\binom{N}{3}$ triangles, giving us $3 \times \binom{N}{3}$ triangle inequalities, and $\binom{N}{2}$ prp value to solve the distances and latent factors.

4.7 SYSTEM DEPLOYMENT

We developed an end-to-end system for the contact tracing application from designing the BLE payload structure, packet transmission protocol, packet transmission applications for android and iOS, and finally, creating the Azure storage tables which receive the packet information through HTTP requests and store them.

4.7.1 Payload structure

We carefully designed the BLE packet structure (Figure 4.7) to contain the necessary information required by our contact tracing algorithms. It has the following components—1-byte packet signature, 2 bytes TxID or transmitter id, 1-byte TxPower or transmitter power, 2 bytes for the packet sequence number, 2 bytes of ACK ID or acknowledgment id, and 4 bytes of ACK bitmap or acknowledgment bitmap.

One byte of packet signature is needed to communicate that this is a packet sent by our contact tracing application and hence contains the required payload structure. Using this signature, we can easily filter out non-contact tracing packets sent by other Bluetooth applications. 2 bytes TxID is a random identifier for the transmitter of the packet. One byte TxPower is the transmitter power used for the packet. For iOS and Android applications, this value varies from a fixed low value (ULTRA-LOW for android) to a specific high value (HIGH for android). The transmitter power will help calibrate the signal measurements made on the receiver, like the received signal strength or RSSI.

We store 2 bytes of a sequence number in each packet which helps to get a detailed log of received and dropped packets on the receiver end. On the transmitter side, we maintain a counter which gets incremented before sending out each packet. The sequence number on the packet contains the current value of the counter. On the receiver end, between the first and last sequence value received from a specific transmitter, we can easily derive that a particular packet was dropped if the corresponding sequence number is not present in the log. Maintaining sequence numbers can help us segment the packet trace into subparts and extract a time series of PRP values for fine-grained distance estimation and changes in the distance over time.

In each packet, a transmitter attaches the packet log that it received from another sender. The sender ID, for which the packet log is being attached or acknowledged, is present in 2 bytes of ACK ID, while the log is present in 4 bytes of ACK bitmap. The packet log will help in collaborative distance calculation between people in the vicinity. Recall that in Section 4.6, we mentioned that for accurate distance calculation between two peers, we would also use the packet exchange information between other peers in the vicinity. Since each node calculates the distance in a decentralized fashion, the ACK fields in the packet ensure that the necessary collaborative information is available during distance estimation.

ACK bitmap contains a bitmap of received packets, where bit = 1 indicates that the corresponding packet sequence number was received from the source, while bit = 0 indicates the packet was not received. We used the most significant bit in the bitmap for the most recent packet. Next, we will describe the packet transmission and reception protocol.

4.7.2 Protocol

Now, we will describe the protocol for sending and receiving packets in our contact tracer application. Specifically, we will detail how each node creates a log of the packets it receives from a sender and then creates a bitmap of that log to attach to the transmitted packets.

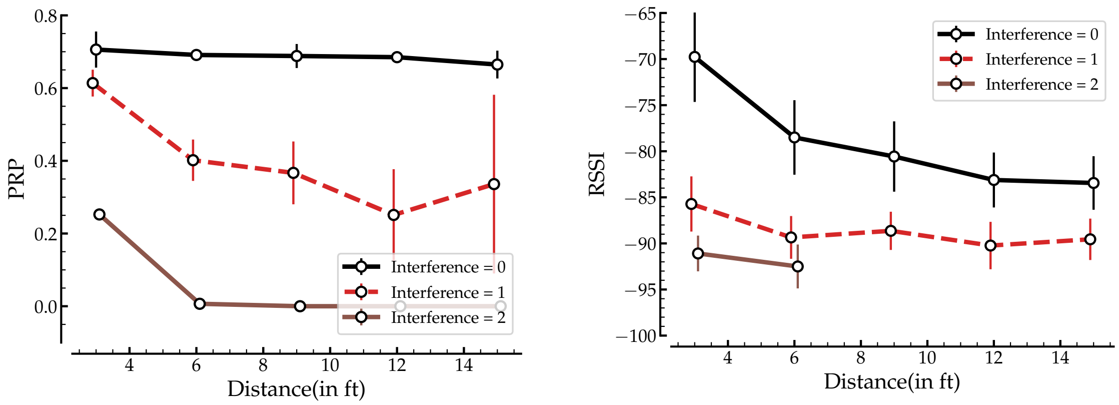
On the transmitter end, let us assume that our frequency is T_f. We set T_f to be $100ms$ in our app. At the beginning of transmission, the packet sequence number is 0. After every T_f seconds when we send out a new packet, we increment the sequence number. In each new packet, the transmitter acknowledges another nearby transmitter by attaching the current packet information that it has received from the other one. If there are many transmitters in the vicinity, we choose the one which has the oldest unacknowledged packet. We pick up the bitmap, which contains all the current unacknowledged packets from that transmitter. We mark all the packets in the bitmap to have been acknowledged.

On the receiver end, we maintain a log of the packets from each transmitter in the vicinity. For every received packet, we first check if it is a packet from our application based on the packet signature. If yes, we extract the Tx ID and sequence number of the packet. We add the sequence number in memory to a log for this Tx ID and also set the status for the current (Tx ID, sequence number) packet to be unacknowledged.

4.7.3 Android and iOS applications

We developed an app on Android using Java/Kotlin and an app on iOS using Swift that can transmit and receive Bluetooth packets. We encoded our BLE packet structure inside the apps. We used non-connectable BLE advertisements to exchange these packets.

We initially designed an experimental version for both Android and iOS apps where we initiated and stopped BLE packet transmission and reception through the press of a button. In Android, we set the advertising frequency to be $250ms$, while we experimented with advertising powers from ultra-low to high. In iOS, we used the system's default advertisement power of $12dBm$ and advertisement frequency of $100ms$. iOS does not provide the levers to control these parameters. However, in iOS, we can control the number of packets we send out by restarting the advertisements at frequent intervals. For example, after sending out one packet, we can stop the current advertisement and then use a timer to restart after a



(a)

(b)

Figure 4.8: Variation of PRP and RSSI with distance. (a) PRP does well with interfering human bodies. (b) RSSI does well with no interference.

specific time. In the production version of these apps, we can use the restarting technique to switch to a low frequency and save power when no other transmitter is in the vicinity. Low-frequency mode can be the default state of the app. Instead of using a button to start high-frequency advertisement bursts, we will use the reception of BLE packets as a trigger to switch to the high-frequency mode.

An issue with iOS BLE applications is background transmission and reception. iOS decreases advertising frequency by default in background mode. Similarly, in background scanning mode, iOS coalesces multiple packets from the transmitter into one record. We record a reduced number of received packets in iOS background mode. The reduced number can make PRP inference a challenge. We have to attribute whether fewer received packets are due to more considerable distance or background mode. One trivial solution to this problem is to use RSSI inference instead of PRP, which does not rely on the number of packets. Also, packets sent in the background mode have a specific format that can distinguish them from the foreground packets, and we can adjust the expected advertisement frequency accordingly.

4.7.4 Azure storage

Our apps use HTTP transfer protocol to POST the packet information to the azure table API. Azure table parses and stores data in POST.

4.8 EMPIRICAL BENCHMARK AND CHALLENGES

We experimented with our contact tracing application on iOS and android in an outdoor environment in our university quad. One person carrying a phone in their hand stood at

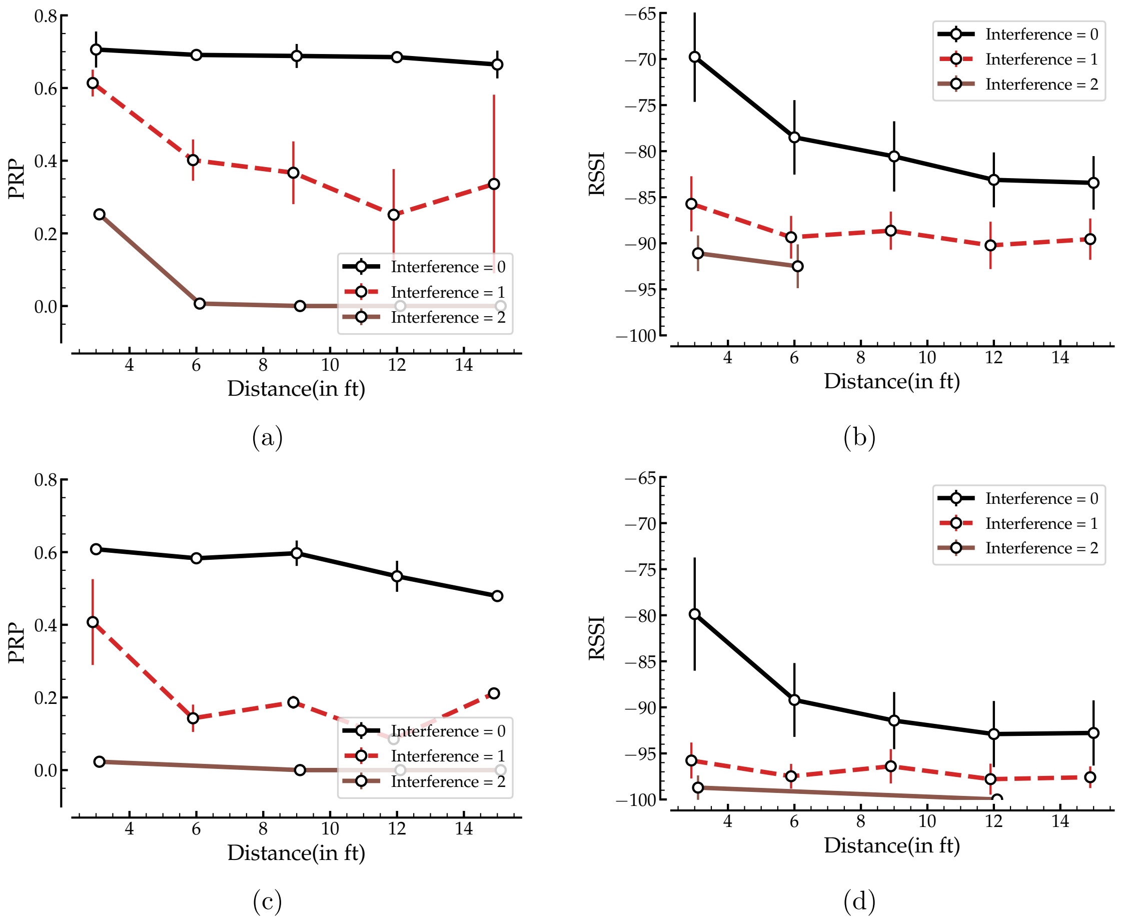


Figure 4.9: Intra-device variation in PRP and RSSI due to different antenna designs. PRP is lower on the Motorola phone (c) compared to the Nexus phone (a). Similarly, RSSI values received are also lower on the Motorola phone (d) compared to Nexus phone (b).

a fixed location, while the other person moved with the phone in their hand to various locations to cover different distance scenarios. We covered distance values from $3ft$ to $15ft$. Each person stood in different orientations: facing each other or facing away from each other for each distance value. When both persons are facing each other, there are zero interfering human bodies between the two phones. When one person faces away, there is one interfering human body, while when both face away, we have two interfering human bodies. This experiment setup helps us in analyzing PRP and RSSI values at different distances and interference values.

We show the results of variance in PRP and RSSI for BLE with distance and interfering human bodies in outdoor environments in Figure 4.8. We can see that PRP does well with interference, giving good variations with distance. For interference $\in [1, 2]$, PRP can be easily used to identify between the cases of distance $= 3ft$ and distance $= 12ft$ as seen in

Figure 4.8(a). However, PRP remains almost constant across distances for the no interference case. On the other hand, RSSI does well with no interference giving good variations with distance Figure 4.8(b), while it becomes almost constant with high variance in values for interference $\in [1, 2]$. The finding reinforces our earlier conclusion that we should augment PRP with RSSI to get better localization accuracies.

Our empirical experiments revealed a new challenge with deploying ranging-based contact tracing applications at a large scale. We observed inter-device variations for both PRP and RSSI, as can be seen in Figure 4.9. We did experiments on two android phones–a Motorola E6 and a Nexus 5X. We observed that PRP is lower on the Motorola phone Figure 4.9(c) compared to the Nexus phone Figure 4.9(a). At distance $= 3ft$ and no interference, while PRP is 0.8 for the Nexus phone, it is only 0.6 for the Motorola phone. Similarly, RSSI values received are also lower on the Motorola phone Figure 4.9(d) compared to the Nexus phone Figure 4.9(b). Distance $= 3ft$ and no interference, mean RSSI is $-70db$ on Nexus, while it is $-80db$ on Motorola.

Based on the empirical experiments, we realize that different antenna designs in different phones lead to variation in PRP and RSSI values. To implement contact tracing applications at scale, we need to learn models that generalize well to different devices. However, learning such generalizable models will require new research.

4.9 CONCLUSION AND FUTURE WORK

This work aims to achieve reliable contact tracing by solving the challenges present in peer-to-peer Bluetooth measurements. The challenges involve—latent factors like phone orientation and device location on the person can impact pairwise measurements besides distance. Also, we have well-known multipath effects that can add high variance to distance measurements made from a single pairwise observation. We propose two solutions to overcome this problem—using minimal infrastructure in indoor space and using collaboration in outdoor spaces. Both infrastructure and collaboration can help us leverage more pairwise measurements made between other nodes in the vicinity. Getting multiple samples of measurements can help us in two ways—reduce the variance in estimation caused by multipath. Also, we can constrain these measurements together through triangle inequalities.

First, our indoor experiments have shown that leveraging known distances between few beacons in a triangle inequality based likelihood framework can help us achieve a median error of 0.89 m in a library and 1.22 m in a retail store environment. Even with just five beacons in a $100sqm$ area, we achieve a median error of 1.5 m. We have improve our distance estimates by $\sim 10\%$ by moving from a location-based distance estimation to a triangle inequality-based distance estimation.

Second, in the absence of infrastructure, we can use many unknown distances in collaboration and constrain them through triangle inequalities to better solve distances. We designed the contact tracer application that will run on smartphones to transmit and receive packets. We conducted empirical benchmarking experiments to see how PRP and RSSI vary with distance and human interference. PRP does better for high interference giving good variation with distance. RSSI captures better distance variation at low interference when PRP almost remains constant. The analysis exemplifies the complementary nature of PRP to RSSI and its potential to further indoor location art by using it in combination with RSSI.

However, our empirical experiments also revealed the challenge of intra-device variation in PRP and RSSI values, making it challenging to deploy contact tracing applications that measure PRP or RSSI at scale across many devices. Future research needs to learn RSSI and PRP models that can adapt/generalize to different devices. Specifically, we will need to investigate Model Adaptive Machine Learning techniques to learn device generalizable models.

CHAPTER 5: CROWDSOURCED ESTIMATE OF SPACE ANONYMITY

5.1 INTRODUCTION

Indoor positioning has led to a wide array of intelligent location-based services in public spaces. The omnipresence of sensor-rich mobile phones has led to applications like in-building guidance and navigation in malls, inventory management in warehouses, easy wayfinding, and discount notifications in public places such as airports, railway stations. More recently, location services have helped automate digital contact tracing in these spaces where the COVID-19 virus is known to spread faster.

Location services have raised a fundamental problem— more granular level *Location surveillance or tracking*. While people want services, they do not want to be surveilled. Location applications can collect data at different gradations or granularities. For example, an airport may be collecting location information in the range of $\sim 10m$ to do contact tracing at the level of gate number. A mall can be localizing in grids of $\sim 5m$ to find which shops are more crowded at different times of the day. However, the airport location system can also localize in $\sim 1m$ grids to find if a person browsed a vending machine. Similarly, a mall can localize in $< 1m$ range to automatically find our acquaintances based on similar patterns in the location signature. More granular level tracking can reveal much sensitive information about a person without taking their explicit consent. The possibility of such information leakage raises doubts in the minds of people about adapting location services.

People using an indoor location system do not know the granularity of the location information that the system collects about them. Lack of transparency on location granularity leads to privacy invasion or widespread distrust of all location technologies. However, the performance of a system highly depends on the nature of infrastructure installment in the environment. For example, an airport may have better localization with a dense device deployment than a railway station with only a few devices. The airport may be targeting more detailed information about a traveler, like browsing items in a particular vending machine, while the railway station may be collecting only coarse information for digital contact tracing like which platform or region a traveler was in at a particular time. While the airport system can meet the definition of privacy invasion for many people, most people will also like to use the railway system to keep themselves safe. The above examples show that the solutions at both ends of the spectrum—implementing and adapting location technologies without thinking about privacy concerns or discarding location technologies altogether comes with their own set of limitations. Instead, if people know the level of location tracking in a space, they can decide whether to use it based on their personal privacy-benefits trade-off. We are interested in auditing the level of location tracking in a space, not in developing a new privacy-preserving protocol.

One may argue that an indoor location system can reveal the level of tracking that it does in a particular space to increase goodwill and trust. However, to solidify the claims of tracking level made by a system, we need to verify this information externally or using a third party. Here we ask the question—*Can we detect the level of location tracking by consuming only the raw data that the system collects from us?*

This work introduces the insight of location tracking or surveillance as an error radius that varies across space. This insight is a novel contribution to the best of our knowledge since previous works have only thought about tracking accuracy and tried to measure a single number for the entire space. The reality is that tracking granularity is spatial, and we want to find the tracking error or uncertainty specific to a certain point in an environment. To clarify, we are not interested in tracking a person at a certain point. Instead, we want to know—given a location estimator, what is the error or uncertainty in tracking a person at that point.

We observe that surveillance levels may vary over different regions of the same space and claim that finding a single-valued metric of surveillance for a public space is insufficient. We propose finding a surveillance map or distribution over the space. Let us take a railway station as an example. Say the station wants to localize people more accurately near its ticketing counters which are more visited. At the same time, it is okay to have a coarser granularity localization in a specific platform that is less used. From empirical studies in the wireless localization domain, we know that more devices (such as WiFi APs and Bluetooth beacons) help better location finding as we can add more data while finding our unknowns. Accordingly, the station has installed many devices near its ticketing counters and only a few devices on the platform. Finding a single number for the level of surveillance, in this case, will be erroneous. That will either underestimate the surveillance level for regions near ticketing counters or over-estimate for the platform region. We need to capture variance in surveillance levels across a space.

In this work, we develop a framework for the crowd to audit the surveillance characteristics of public space. We make some assumptions around technology, data capture, and information symmetry. We focus on location technologies like RSSI that can deploy on a commercial device like a smartphone that most people carry today. An application on the person's phone communicates with the other infrastructure installed in the space through a wireless medium (such as WiFi and Bluetooth) and collects data. We assume that the infrastructure is the transmitter, as can be seen in many localization works like [16, 48, 76] and the person's phone is the receiver that captures the data. We further assume that both the infrastructure and our framework use this same raw data collected on a person's smartphone. The infrastructure combines this data with other auxiliary information (such as the location of infrastructure devices, position and orientation of antennas on devices, a wireless transmission model, or a fingerprinting map) to find the person's location. The auxiliary information is not available to the crowd framework. The current approach cannot guarantee surveillance findings if there is an information asymmetry in the raw wireless data, e.g., our system collects RSSI data, and the infrastructure uses CSI data for localization. Under these assumptions, we propose that each person collects raw wireless data while interacting with a public space in their specific way and then contributes that trace to a central repository. We will use the crowdsourced traces to infer the surveillance in space.

We designed active learning strategies to find the location surveillance map with uniform accuracy all over the space. Since we are interested in finding a map over the space rather than a single number, we need to make sure that we estimate all map portions with certain desired accuracy levels. We need to collect a sufficient amount of data in all regions of space to ensure those accuracy levels. Continuing our example from the previous paragraph, we can see that the more frequented region near the ticketing counter of the railway station will have more crowdsourced data. At the same time, we will get less data for the platform. The natural movement of people inside a public space leads to the creation of such data hotspots in few regions, while we have sparse data in others. Here we ask the question—*Can we find a spot or region inside a public space where data collection will lead to maximal improvement of our estimate of the surveillance map?* We can then provide cues to people requesting them to collect data near that region. Our active learning strategies find regions of maximum uncertainty to collect data based on the current estimates of surveillance maps and infrastructure properties.

Our simulation studies based on actual-world data show that the active learning strategies can more uniformly determine the location surveillance map over the entire space. The best active learning strategy based on received data uncertainty can detect location surveillance with an error of $0.4m$ and all beacon locations with less than $1m$ error for RSSI-based location estimator.

Overall, our work suggests a new research direction— auditing the location surveillance capabilities of a location finding infrastructure. We believe this line of work is essential for establishing trust among people to use location-based services in public spaces. It is also the first step towards future work of achieving declarative privacy where people can specify their comfortable level of surveillance. Specifically, we make the following contributions.

- **CrowdEstimator**: We build a crowdsourcing system that harnesses the power of the crowd to audit the nature of the infrastructure and estimate the surveillance or level of tracking in space.
- **Location surveillance is not a single number, rather spatial**. All previous works have only thought about tracking accuracy and tried to measure a single number for the entire space. However we want to estimate a spatial map.
- **Active Learning to learn spatial map with uniform accuracy**. We have designed active learning strategies to pick up regions inside a public space where data collection will lead to a maximal improvement of the estimate of the overall surveillance map, not just specific regions.

5.2 RELATED WORK

This section will talk briefly about indoor location state-of-art systems and then focus on a few privacy-preserving algorithms in indoor localization space. We will then outline two important previous works that have audited location tracking infrastructure towards achieving different goals. Finally, we will discuss some active learning tools that we have utilized in our work.

- **Indoor location tracking:** Majority of indoor location systems use WiFi access points [1, 17] for localization. Recently, there has been much interest in Bluetooth-based indoor localization [19, 20, 21] as well to achieve digital contact tracing [62]. Most indoor location systems use some properties of signals exchanged with access points to infer the location of a target. Most widely used properties are RSSI or received signal strength [1, 16, 31], and CSI or channel state information [85, 86]. The granularity and accuracy of the location information depend on the type of signal properties used. While we can get decimeter-level accuracy with CSI property, we need to make hardware changes on commercially available smartphones to access this information. In this work, we have focussed on RSSI and PRP systems that can be accessible to many people today through smartphones.
- **Privacy Preserving Indoor location tracking:** There are many proposed solutions for Privacy-Preserving Indoor Localization (PPIL) [73] using Homomorphic Encryption (HE) such as the Paillier cryptosystem [74] or k-Nearest Neighbors (k-NN) algorithm [75]. [73] analyzed the privacy issues of WiFi fingerprint-based localization system and then used the Paillier cryptosystem to protect both the client's location privacy and the service

provider's fingerprint data privacy. [75] enables a user to localize privately through an Indoor Positioning System by making k camouflaged localization requests. They design a Temporal Vector Map (TVM) algorithm, which guarantees that the IPS system cannot find a user's location with a probability higher than a user-defined preference. These works are specific to WiFi fingerprinting systems where the localization space is segmented into N discrete points. In our work, we are looking at a continuous localization space where instead of k-anonymity, a user wants to achieve α-anonymity, i.e., do not localize the user to a circle of radius smaller than α. Besides, we also assume that the user does not have access to any system-side auxiliary information such as fingerprinting maps and access point locations. The current granularity of location inference is unknown on the user side. To achieve anonymity, we first need to know the level of location inference or surveillance that depends on the infrastructure installed in that space. This work focuses on inferring the infrastructure properties from the data collected on the user side, which is the first step towards achieving α-anonymity.

- **Auditing of location tracking systems:** The previous bullet highlighted that we want to estimate the granularity of location estimation in a place. To estimate location granularity, we need to audit the location infrastructure that currently exists in that place. Specifically for RSSI-based location systems, we need to find information about the number of access points and their locations. For CSI-based systems, we will need to find extra information like the number and orientation of antennas. We looked at previous systems like EZ [16], and LocBLE [76] which inferred properties of the infrastructure. EZ aims to achieve indoor localization without any pain or less manual overhead. To that end, they estimated the location of WiFi access points from RSSI data and few GPS labels rather than requiring manual labeling of those locations. LocBLE wanted to locate nearby BLE beacons. They combined BLE RSSI data with inertial sensor data to find the beacon locations. While the goal of these works was to get an accurate estimate of the infrastructure properties, our goal is to accurately estimate surveillance level, which is a function of the infrastructure properties.

- **Active Learning:** People have mostly studied active learning in the context of classification problems [77] where labeling a data point comes with the cost of getting a human annotator to label that point. In most cases, we need multiple human annotators on a single data point to avoid personal bias in data labeling. Active learning helps to pick the best data point so that we can learn more with less cost. In our problem setting, since we will rely on people to collect RSSI or PRP data (equivalent to labels) to estimate infrastructure properties, it makes sense to use active learning to collect data to learn more with the same amount of data. Classification setting assumes having a finite unlabeled set from which they need to choose the data point. In contrast, in our problem setting, we are faced with a continuous space. In the current work, we discretize the space into a fine grid and then treat the set of points in the grid as our unlabeled set. There are different kinds of active learning techniques that have been proposed—uncertainty sampling [78], expected error reduction [79], information density framework [80]. In this work, we want to select locations (equivalent to features in the classification scenario) to collect PRP or RSSI data (equivalent to labels) to reduce the uncertainty in some objective distributions. The objective distributions can be PRP or RSSI data distributions for all locations, all infrastructure device locations, or directly the surveillance level distribution.

5.3 MOTIVATION

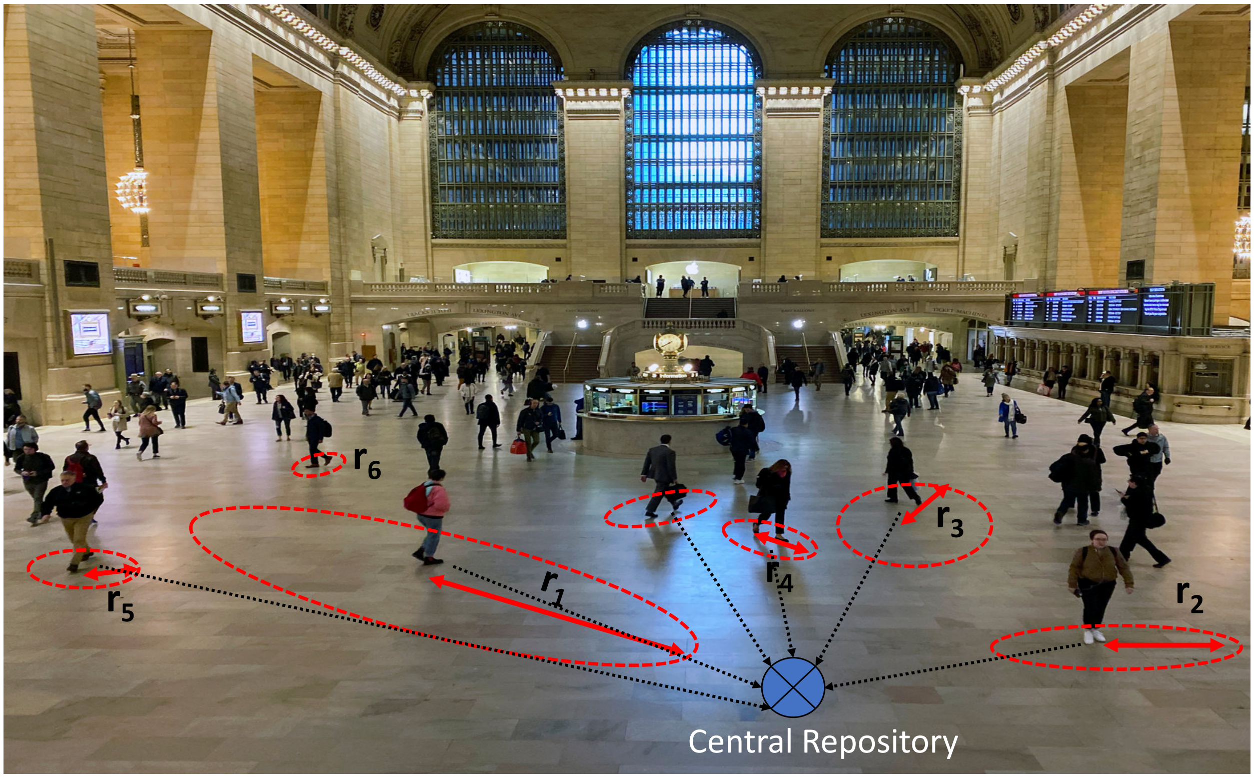


Figure 5.1: Figuring out Location Surveillance has 2 components—how much (the radius) and where (the distribution). We use crowdsourcing to estimate the location surveillance map of public spaces like airports, malls, railway stations etc.

We have seen much work in the last two decades in the indoor localization domain [1, 4, 5, 6, 16, 17, 18] that tracks people in indoor spaces. The objective of such works was to increase the accuracy with which they track people. They measure tracking accuracy through a median error in location estimation that they observe over a certain space. For example, RSSI estimators [1, 16, 17] have achieved median errors of $2m$ while CSI estimators [4, 5, 6] have achieved median errors in decimeters.

This work aims not to decrease tracking error or increase accuracy but rather to measure tracking error in an indoor space given that a localization infrastructure exists in that environment (on which we do not have any control) and the absence of any well-calibrated training data. We assume that the infrastructure already exists in an indoor space, and we, as a third party, are interested in measuring the tracking possible with this infrastructure. We also assume that we are not privy to any calibration information like the number of devices (such as WiFi APs and Bluetooth beacons) present in the infrastructure, locations of such devices, or properties of wireless propagation in that environment.

Reporting a single number like the median error is not enough to measure tracking error for a space. We need to estimate a map or a spatial distribution. Tracking error in a certain location depends on multiple factors such as the number of infrastructure devices present in the vicinity, distance to such devices, the number, position, and material of reflectors present nearby. We need to determine a number for tracking errors for each location in space. In other words, we estimate localization error $\hat{l_e}(x, y)$ as a function of x, y such that

$$|\hat{l_e}(x, y) - l_e(x, y)| < \delta \quad \forall x, y \tag{5.1}$$

where $l_e(x, y)$ is the ground truth error. We can have different definitions for $l_e(x, y)$ such as median localization error if we collect large number of data points at (x, y), or theoretical bounds on location estimation variance at (x, y) as we will discuss in Section 5.4.2. We want to estimate tracking error map with uniform accuracy δ over the space.

5.4 SYSTEM DESIGN

In this work, we propose an end-to-end system called **CrowdEstimator** that crowdsources data from many users and estimates location surveillance distribution over the layout of public space as shown in Figure 5.2. Our system currently makes a few design choices or assumptions about the localization infrastructure of public space(Section 5.4.1) and a formal definition of surveillance (Section 5.4.2). It incorporates a Bayesian framework (Section 5.4.3) that combines wireless and inertial sensor data-streams to obtain location surveillance estimates. Finally, it includes active learning algorithms (Section 5.4.4) that can find out better data collection locations for surveillance estimation.

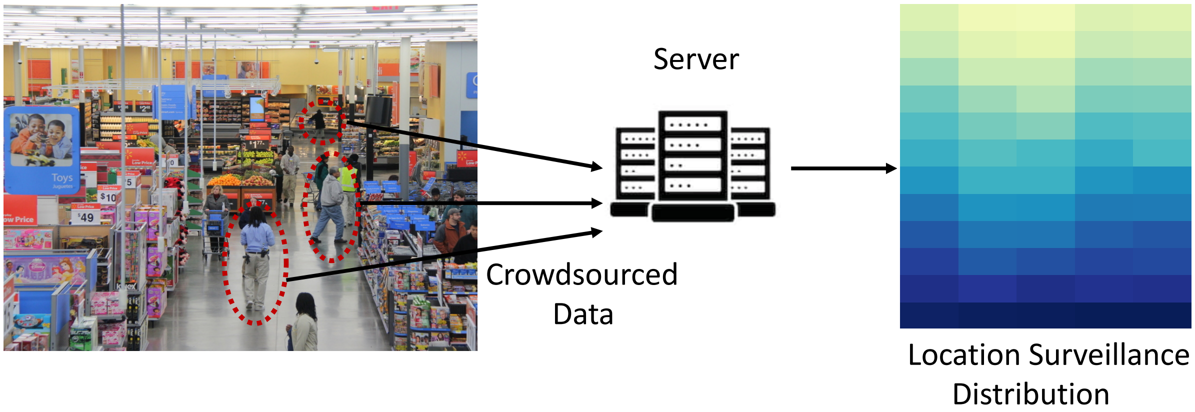


Figure 5.2: Our system CrowdEstimator collects data from many users, updates and builds a distribution of location surveillance over the space.

5.4.1 A few design choices

Location surveillance in a space depends on the localization system's infrastructure and wireless signal properties. The infrastructure can use different types of devices like WiFi access points or Bluetooth beacons. Location is inferred by extracting signal properties from the communication between a device on the person and infrastructure devices. An infrastructure can use different signal properties for localization like signal strength or RSSI, angle-of-arrival or AoA, time-of-flight, or ToF. AoA and ToF techniques are more broadly called Channel State information or CSI-based techniques. Most state-of-art localization techniques [4, 6, 17] use either RSSI or CSI based properties. In this work, we make a few assumptions about the localization system.

Assumption about technology: In this work, we focus on location surveillance with signal properties available on ubiquitously available technology like smartphones. Localization on a large scale can be made available today only through signal properties like RSSI since they are available on smartphones. Though CSI is a more accurate feature for localization, it is not available on most smartphones today. However, we can extend our insights and algorithms to CSI-based location systems. For CSI, we will need to infer additional properties like antennae orientation from the crowdsourced data besides device locations.

Assumption about data capture: We assume that smartphones scan wireless packets sent out by a localization infrastructure installed in a public space. The phones upload scanned packets to our central server. The same algorithms can be trivially applied to the reverse scenario when phones broadcast packets and infrastructure devices receive them.

However, instead of crowdsourcing data from each person, now the infrastructure needs to send the scanned data stream to our server for location surveillance estimation. We can imagine an audit scenario where infrastructure submits its collected data to a third party (our server in this case) who can independently verify its claims on the level of surveillance they are conducting.

Assumption about information symmetry: We assume that our system uses the same raw data that the infrastructure is collecting for location estimation. Both the infrastructure and our system obtain this data from the smartphones of each user. It does not access any other auxiliary information (such as device location or antennas orientation) present with the infrastructure. The current approach cannot guarantee surveillance estimation if there is an information asymmetry, e.g., our system collects RSSI data while infrastructure uses CSI data for localization.

Our system uses the scanned wireless packets for location surveillance level estimation, while the infrastructure uses the wireless packets along with other auxiliary information for location estimation. The auxiliary information can be the exact location of its devices, parameters for relationships between signal properties and distance, or fingerprinting maps across its space. Infrastructure may have used extensive training mechanisms to generate such auxiliary information for accurate localization. Our system does not use these auxiliary pieces of information.

5.4.2 Defining location surveillance

Location surveillance in a space depends on the infrastructure (number of beacons/WiFi routers, their locations) and wireless signal property (RSSI or CSI) used for localization. However, other dynamic factors like reflectors (walls, ceiling, or glass), multi-path effects, signal-to-noise ratio (SNR) also dynamically impact the received signal properties, affecting the surveillance performance.

It is hard to quantify environmental factors like SNR for every region within a space. SNR has a complicated relationship with many attributes like nearby reflectors, area, number of reflectors, the material of reflectors, distance to reflectors. Crowdsourcing data to accurately measure each of these attributes for all regions in space will involve a considerable overload. One can try to collect data at all regions in space and empirically measure the SNR in each region. However, SNR may vary over time due to changes in the environment.

In this work, we make a simplifying assumption: we assume that SNR is constant across the space (same SNR impacts surveillance in all regions), and we define location surveillance as a Cramer-Rao-based theoretical lower bound or CRB[105]. We believe CRB is a good starting point for location auditing work since it captures the impact of infrastructure devices and incorporates a constant SNR into its framework. Note that our system can be trivially extended to other definitions of location surveillance as well.

We can write the Cramer-Rao Bound (CRB) on the covariance matrix of any unbiased estimator $\hat{\theta}$ as $cov(\hat{\theta}) \geq F_{\theta}^{-1}$ where the Fisher information matrix (FIM) F_{θ} is defined as

$$F_{\theta} = -E\nabla_{\theta}(\nabla_{\theta} f(X;\theta))^{T} \tag{5.2}$$

Since we care about location estimators deployed on smartphones, and RSSI is the state-of-art localization feature on smartphones, we define location surveillance as CRB of RSSI-based estimator. A standard log-normal RSSI model can be defined as $P(dBm) = 10\log_{10} P$ where P is the measured RSSI in milliWatts. $P(dBm)$ is Gaussian.

$$P(dBm) \sim \mathcal{N}(\bar{P}(dBm), \sigma_{dB}^{2}) \tag{5.3}$$

$$\bar{P}(dBm) = P_0(dBm) - 10\eta_p \log_{10}(\frac{d}{d_0}) \tag{5.4}$$

where $\bar{P}(dBm)$ is the mean power in decibel milliwatts, σ_{dB} is the variance of shadowing, $P_0(dBm)$ is the received power in decibel milliwatts at a reference distance d_0. The path loss exponent η_p is a function of the environment and captures the SNR (signal-to-noise ratio) effects in the environment. For a standard log normal RSSI model, [105] derived the CRB as

$$CRB = \frac{1}{b}\frac{\sum_{i=2}^{m+1} d_{1,i}^{-2}}{\sum_{i=2}^{m}\sum_{j=i+1}^{m+1}(\frac{d_{1\perp i,j}d_{i,j}}{d_{1,i}^{2}d_{1,j}^{2}})^{2}} \tag{5.5}$$

where we assume that the infrastructure has m devices installed in a public space. The locations of these devices are given as $(x_i, y_i), i \in [2, \ldots, m+1]$. (x_1, y_1) is the target location to find the CRB or surveillance level. $d_{1,i}$ is the distance from the target location to the device i. $d_{i,j}$ is the distance between two devices i and j. $d_{1\perp i,j}$ is the shortest distance from target location to the line connecting devices i and j. b captures the SNR effects present in the space. Currently, the CRB formula assumes that SNR effects remain the same over the entire space. However, in future iterations, we can extend the CRB derivation to consider that SNR varies across space or try to capture the variation in SNR from empirical data.

$$b = \left(\frac{10\eta_p}{\sigma_{dB}\log 10}\right)^{2} \tag{5.6}$$

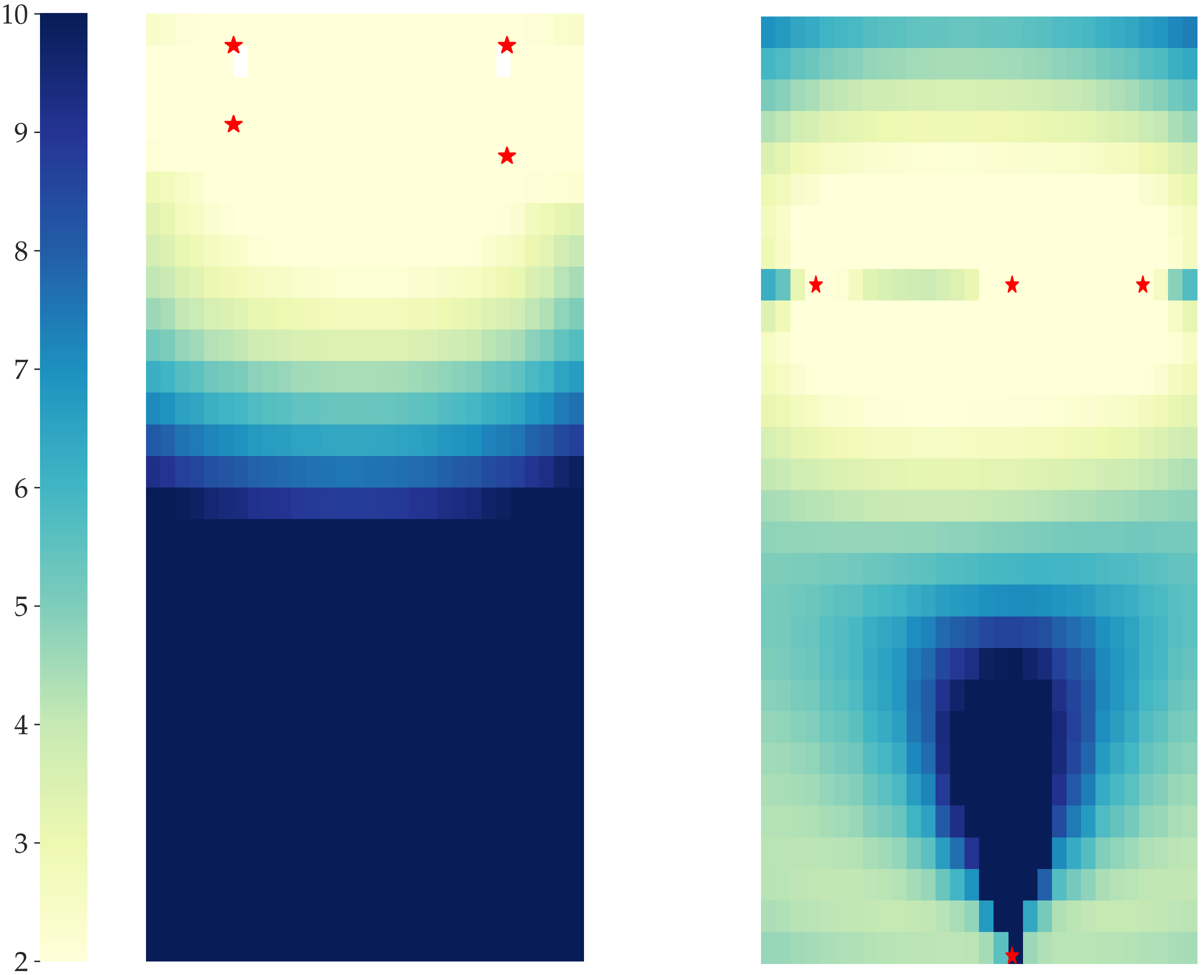


Figure 5.3: Cramer-Rao Bound (CRB) for RSSI location estimators. The red stars denote the location of the devices. CRB is low near devices, high far away from them. CRB is high when close to only one device. CRB is high along the line joining two devices because dilution of precision is high in those regions.

Let us look at the geometric interpretation of the CRB formula through two sample placements of four devices in an environment in Figure 5.3. The first figure shows that CRB is lower when we are close to devices and higher when far away from them. This observation is intuitive since RSSI values are more accurate when we are close to a device, and the values start accumulating noise as we move far away from devices. The second figure shows that CRB is also high if we are close to only one device while further away from all others. Accurate RSSI from a single anchor device can only identify the distance or radius around that device, but not both x and y locations. Specifically, CRB is higher close to the straight line joining two anchor devices because the dilution of precision is higher in these regions. Minor errors in RSSI values lead to more extensive regions of uncertainty.

CRB values help in obtaining a lower bound on the surveillance levels. A high value of the lower bound for public space will encourage more people towards using its location services since the space may be only doing coarse-grained localization for essential applications like digital contact tracing. However, since CRB depends only on anchor device locations and does not consider environmental factors like multi-path, we may obtain lower values of surveillance in a region than is realistically possible. As future work, we want to incorporate multi-path effects in our definition of location surveillance.

To summarize, we defined location surveillance as Cramer-Rao Bound or CRB. CRB is a function of the target location, and hence we obtain a CRB distribution or map over the space. CRB depends on the device locations. Next, we estimate the device locations from crowdsourced data.

5.4.3 Estimating location surveillance

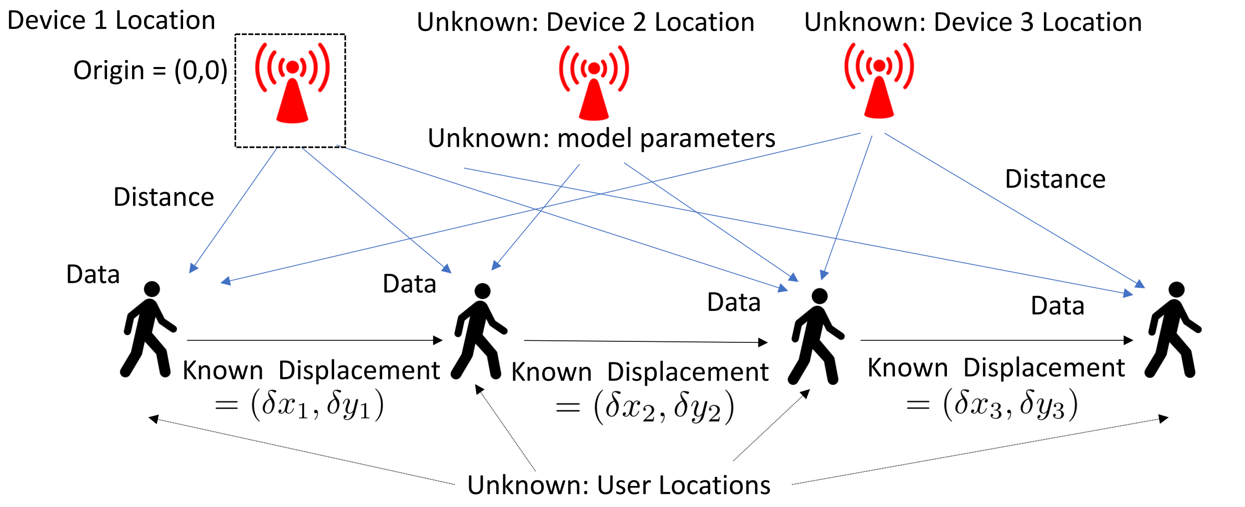

Device 1 Location
Origin = (0,0)
Unknown: Device 2 Location
Unknown: Device 3 Location
Unknown: model parameters
Distance
Distance
Data
Data
Data
Data
Known Displacement
$= (\delta x_1, \delta y_1)$
Known Displacement
$= (\delta x_2, \delta y_2)$
Known Displacement
$= (\delta x_3, \delta y_3)$
Unknown: User Locations

Figure 5.4: **Surveillance Estimation System**: We crowdsource raw wireless traces collected in a person's phone, and also the displacement data obtained through inertial sensors. We fix one of the device locations as the origin of our co-ordinate system across all traces. We estimate three unknowns—device locations, user locations and model parameters that connect wireless properties (like RSSI) with distance.

We pointed out in Section 5.4.2 that CRB-based location surveillance for RSSI localizers is a function of the location of the infrastructure devices. To estimate CRB for RSSI, we need to solve three types of unknowns—device locations, user locations, and data reception model parameters. We have only one data stream—wireless interaction data between users and infrastructure devices. We incorporate inertial sensor data with wireless data to solve the unknowns.

Trying to estimate three unknowns—device locations, user locations, and model parameters solely from the crowdsourced wireless data can lead to translation and scaling problems. The crowdsourced data is a function of the distance between the user and device locations, which are unknown to us. The function between wireless data and distance is also unknown due to a lack of training. As a result, the same data can fit different configurations of devices and user locations. For example, a possible device and user location configuration can be translated in space due to a lack of fixed origin to get many solutions. Also, in our solutions, we can scale up distances between devices and users by a particular factor while scaling down the function parameters by the same factor, again leading to many solutions.

We have provided two key insights to solve the problem of too many unknowns—first, we solve the scaling problem by making ground truth measurements of physical scale through inertial sensors. Inertial sensors like accelerometers and gyroscopes are available on almost all smartphones and can provide displacement data $(\delta x, \delta y)$ between two locations of the same person. The displacement data can constrain the user locations and thereby helps to get rid of the scaling problem. Also, we fix an arbitrarily chosen device location to be origin $(0, 0)$ across all data points, thereby eliminating the axes translation problems. For example, in Figure 5.4, we are interested in finding the location of the three devices. Other unknowns are the four locations of the person who has collected the data and the model parameters that capture the relation between data and distance. To solve the unknowns, we use the three displacement data between the consecutive locations of the person available from inertial sensors. We also fix the location of the first device to be $(0, 0)$, i.e., origin across all data.

Second, we observe that wireless transmission or packet transfer physics remains the same across an environment at a particular time. We can use multiple data points collected from multiple people to constrain and solve the same wireless transmission model parameters. For example, in Figure 5.4, the relationship between the data received by the person at 4 locations and distance is captured by the same set of model parameters. As the number of locations increases, the data points increase, but we still need to solve the same number of model parameters.

We use a Bayesian model Figure 5.5 to estimate the posterior distribution of our unknown parameters. Let us assume that the infrastructure has B devices in the environment, whose locations are $\{(x_1, y_1), \dots, (x_B, y_B)\}$. The locations are unknown to the crowd but known to the system infrastructure. Say we have N users, and each of these users measures the space $r_i, i \in 1, \dots, N$ where r_i is a sequence of the form

$$\{x_1, y_1, q_1\}, \{x_2, y_2, q_2\}, \dots, \{x_M, y_M, q_M\} \tag{5.7}$$

Individual i first visits (x_1, y_1) and takes the measurement q_1, then visits (x_2, y_2) and takes the measurement q_2 etc. Individual locations $\{(x_1, y_1), \ldots, (x_M, y_M)\}$ are also unknown to us. We know the displacement between consecutive user locations $\{(\delta x_1, \delta y_1), \ldots, (\delta x_{M-1}, \delta y_{M-1})\}$ from the inertial sensors, where

$$\delta x_1 = x_2 - x_1 \tag{5.8}$$

$$\delta y_1 = y_2 - y_1 \tag{5.9}$$

Each measurement q_j that the individual i makes looks of the format

$$q_j = \{q_{j,1}, \ldots, q_{j,B}\} \tag{5.10}$$

i.e. there is one measurement made for each infrastructure device.

For the individual i, we can relate the $j-th$ measurement made for the $k-th$ device $q_{j,k}$ to the unknowns in the following manner

$$q_{j,k} = f(d_{j,k}), \tag{5.11}$$

$$d_{j,k} = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2}, \tag{5.12}$$

$$x_j \sim \mathcal{N}(x_{j-1} + \delta x_{j-1}, \epsilon), \epsilon = \text{ inertial sensors error}, \tag{5.13}$$

$$y_j \sim \mathcal{N}(y_{j-1} + \delta y_{j-1}, \epsilon). \tag{5.14}$$

Each measurement $q_{j,k}$ is either a PRP or RSSI value. The function f that related q to distance is the standard log normal rssi model or the prp model described in Chapter 3. The same f and device locations (x_k, y_k) impacts all measurements made by an user. The more measurements we add, the more number of constraints we can put on the unknowns. Also we draw each user location (x_j, y_j) based on a normal distribution that centers around the previous location (x_{j-1}, y_{j-1}) added to the displacement data $(\delta x_{j-1}, \delta y_{j-1})$. We use the ϵ parameter to capture the noise in displacement data.

To summarize, in our model, we have as unknowns: B infrastructure device locations, $N \times M$ user locations (one for each user and each measurement), and parameters in model function f. Our knowns are $N \times M \times B$ measurement data points made by each user at different locations for each infrastructure device, and $N \times (M-1)$ displacement data points between each consecutive location of the same person. We use Bayesian MCMC inference engine encoded in PyMC3 [96] framework to do the inference.

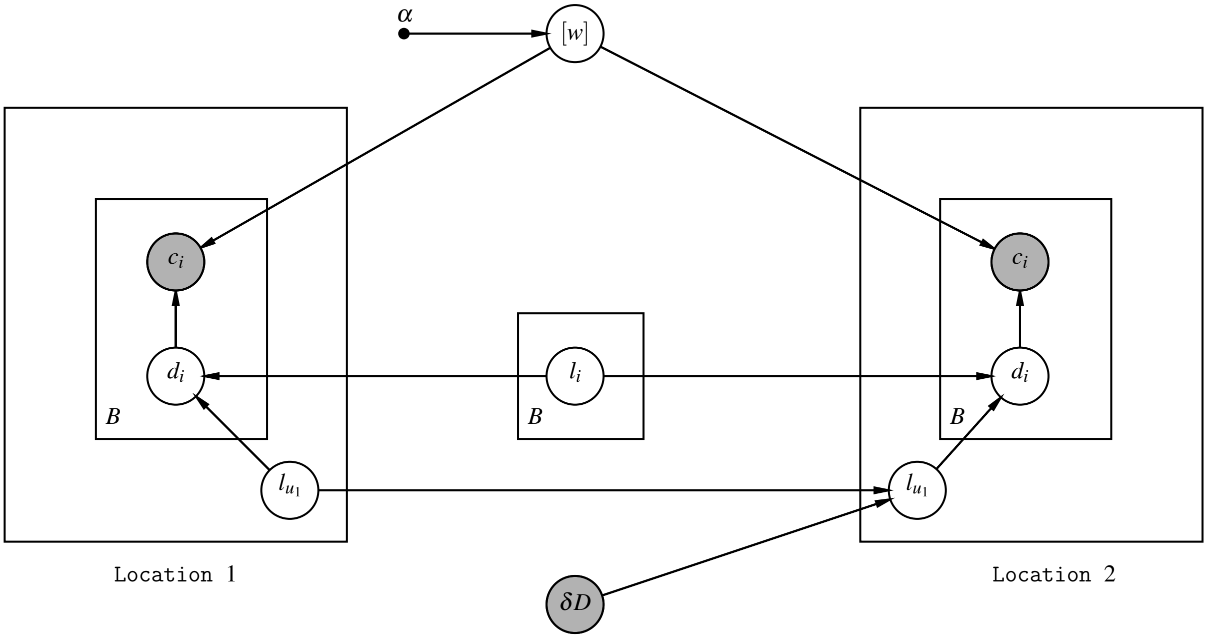


Figure 5.5: **Graphical model**: Shaded nodes are observed, while we need to estimate the unshaded ones. We use the data on number of received packets c_i measured from B beacons to estimate the PRP parameters $[w]$ and beacon locations $l_i, i \in \{1, \ldots, B\}$. We assume that we know the displacement data between two consecutive locations of the same person.

5.4.4 Active learning location surveillance

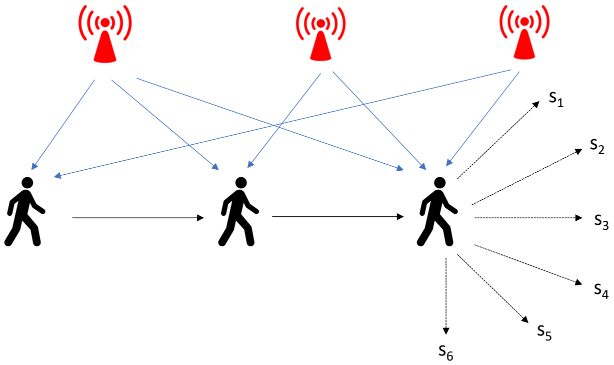


Figure 5.6: We will use active learning to ask a person to go to location s_i that maximizes score for certain Active Learning strategy functions.

We pointed out in Section 5.3 that estimating location surveillance in space is not about finding a single number, rather a distribution over space. Based on the nature of installation in a local region of space, we may have high or low surveillance. To estimate a location surveillance distribution, we need to solve the challenge of obtaining uniform accuracy in estimation all over the space. We incorporate active learning algorithms to get better estimations of surveillance all over the space.

Imagine that all N traces were not informed (i.e., independent) by each other and past traces. *Why might independent traces be a problem?* The convergence rate to the final distribution will be slow since the estimates will be non-uniform. Some hotspots within each public region get more visits and have better estimates, while others will have more erroneous estimates. Thus we need an active learning strategy where the previous N traces inform the trace of the $(N+1)$-th person.

Our key idea to solve the problem is to use active learning to guide people to the best locations for data collection. In addition to people moving around in the space in their way, we can provide visual cues to people requesting them to move to locations which will help us to better estimate location surveillance. If certain people choose to do so, then we can better improve our location surveillance estimates. For example, in Figure 5.6, the person has six possible candidate locations they can move to, and we want to rank them based on some active learning strategy. Instead of providing cues at each step, we could also use active learning to specify the entire sequence of movement $(x_1, y_1), \ldots, (x_k, y_k)$ for a person.

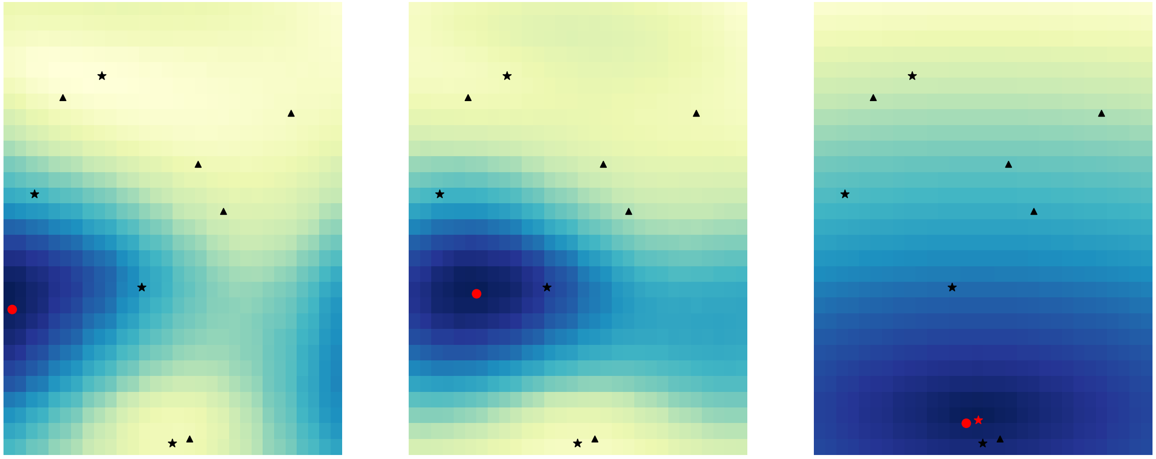
We will now elaborate on active learning algorithms that can estimate location surveillance with better accuracy over the entire space. We have designed three different strategies—choosing locations close to uncertain devices, choosing locations with high data uncertainty, and choosing locations that reduce data uncertainty over the entire space.

High Data Reception Uncertainty: In this strategy, we score locations based on the current uncertainty or entropy in the distribution of data values that we can receive at a location from infrastructure devices. We calculate the entropy of data distribution at a location k through the following observations:

1. In Chapter 3 for a device i, we modeled the PRP value through a binomial distribution and RSSI value through a log normal distribution. For both these distributions, the entropy is proportional to the log of the variance of the current distribution at location k i.e $\log\{\sigma_{data(k,i)}\}$, where σ is the variance of distribution $data(k, i)$.

2. The total entropy at a location will be the sum of the entropies for each device since each device observation is made independently of the others.

We estimate the current data distribution at location k for device i, i.e., $data(k, i)$ from the estimates of device location i and model f that we have obtained through the measurements



(a) (b) (c)

Figure 5.7: The stars represent the ground truth beacon locations. The triangles represent the initial locations chosen for cold-starting the active learning strategies. The red circle indicates the location picked by the strategy. (a) The blue region indicates where data reception uncertainty is high, and the red dot indicates the point selected by the strategy. (b) When we try to reduce uncertainty over the entire space by including representativeness to unsampled points, the red dot moves more towards the right or center of the space. (c) The red circle is next to the device (red star) with the highest uncertainty in its location estimate for the high device location uncertainty strategy.

.

made by the users so far. For each candidate location k, we generate a score as:

$$s_k = \frac{1}{B} \sum_{i=1}^{B} \log\{\sigma_{data(k,i)}\} \tag{5.15}$$

$$data(k,i) = \hat{f}(\hat{d}_{k,i}) \tag{5.16}$$

$$\hat{d}_{k,i} = \sqrt{(\hat{x}_i - x_k)^2 + (\hat{y}_i - y_k)^2} \tag{5.17}$$

where $\hat{f}$ is the current posterior for the model parameters, $(\hat{x}_i, \hat{y}_i)$ is the current posterior for i-th device location. $\sigma_{data(k,i)}$ is the current variance of the posterior distribution of possible data values at location k for device i. We choose candidate location k with the highest score s_k. For example, in Figure 5.7(a), the ground truth locations of the beacons or devices are marked by stars. The triangles indicate ground truth locations from where we have already collected data. Now we want to pick the next best possible location. Blue regions indicate where data reception uncertainty is high and red dot indicates our selected point.

Reduce Reception Uncertainty over entire space: In this strategy, we will choose a location that can maximally reduce entropy in reception data distribution over the entire layout. Similar to the expected error reduction technique discussed in [77], we can generate a score for each candidate location k as:

$$s_k = E_{data(s_k)}[\sum_{i=1}^{K} Var_{+s_k}(data_{s_i})] \tag{5.18}$$

$$s_k = \frac{1}{M}\sum_{i=1}^{M}[\sum_{i=1}^{K} Var_{+s_{k,m}}(data_{s_i})] \tag{5.19}$$

We want to know that if we add a data observation ($+s_k$) from a location k, how that will reduce the data reception uncertainty for all K points in our layout. However, we have not observed location k yet, and we want to do this computation before the actual measurement. We use the expected data distribution $data(s_k)$ at location k instead, where $data(s_k)$ is calculated based on the current distribution estimate of beacon locations and model parameters. Once we generate scores $s_k, k \in [1, \ldots, K]$ for each of K locations, we will choose the location with lowest s_k i.e. lowest expected variance after data addition. We will make a real measurement at that location, add those real measurements to our existing data, and re-train the model to obtain new estimates.

Since our data distributions are continuous, we can use Monte Carlo sampling to generate M samples and calculate the score for each location. For each sample $s_{k,m}$ in a location k, we add the sample ($+s_{k,m}$) to current data, retrain the model, and re-estimate the variance at each location. Then, we take the average across all M samples to finally estimate the overall variance if we were to add a data point from location k. The process is computationally expensive and would need many samples M to generate the score accurately. We propose a heuristic approximation to the computationally expensive uncertainty reduction technique. We score a location based on its current data uncertainty and its representation of unsampled points in space. For each candidate location k, we generate a score as:

$$\begin{aligned} s_k &= \left[\frac{1}{B}\sum_{i=1}^{B}\log\{\sigma_{data(k,i)}\}\right] \times \left[\frac{1}{K}\sum_{i=1,d_{i,S}>\alpha}^{K}\frac{1.0}{1.0+d_{k,i}}\right]^{\beta} \\ d_{k,i} &= \sqrt{(x_i-x_k)^2+(y_i-y_k)^2} \end{aligned} \tag{5.20}$$

In Equation (5.20), the left component measures the uncertainty in data values similar to the previous strategy. The right component measures the similarity of the location to other unsampled points in the layout, i.e., we focus on points where we have not collected data yet. We calculate the similarity as the inverse of the euclidean distance between two points. We consider points unsampled if they are a threshold distance α away from the already sampled

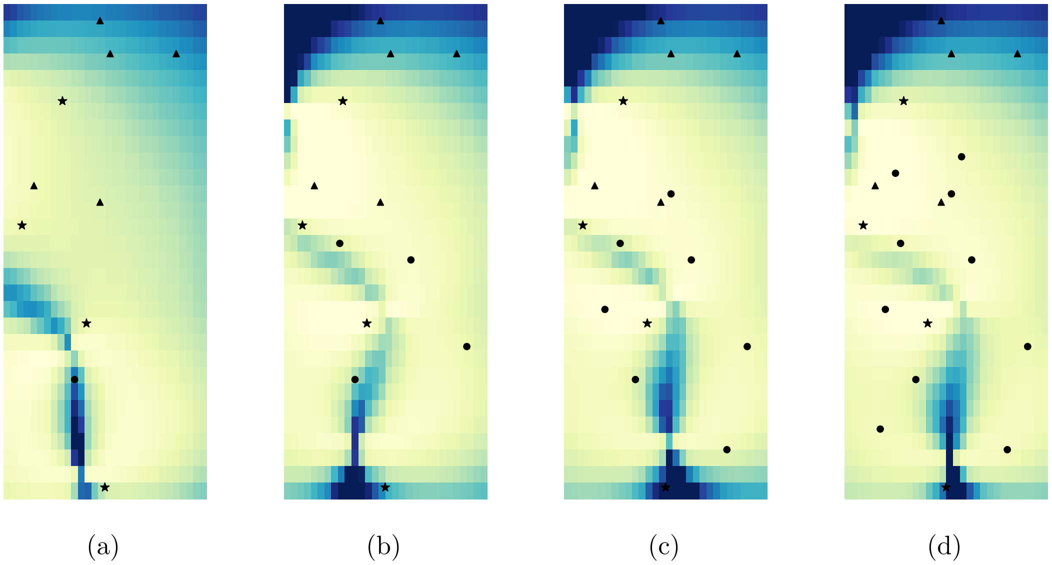


Figure 5.8: Heat map generated by Data Reception Uncertainty strategy over multiple iterations. The stars represent the ground truth beacon locations. The triangles represent the initial locations chosen for cold-starting the strategy. The black circles are the points sampled by the strategy. (a) denotes the initiation with only one point, while (d) denotes the final iteration. The points are well distributed over space. It gives reasonable estimates of all the CRB patterns present over the entire space.

.

regions S. β acts as a hyperparameter that controls the weight of the two components in the score. As we can see in Figure 5.7(b), when we add representativeness to unsampled regions, our selected point marked by a red dot moves towards the right or center of the space.

High Device Location Uncertainty: Here we try to reduce the device location uncertainty directly rather than the received data uncertainty. From empirical experiments, we have seen that collecting data close to a device helps to reduce its location uncertainty maximally. Based on the above observations, we designed a heuristic strategy to pick the location closest to the most uncertain beacon. We measure the beacon uncertainty by variance in its estimated location distribution from the current model. For example, in Figure 5.7(c), the red triangle indicates the estimated location of the beacon for which there is currently high uncertainty, and we collect data from the location closest to it.

Next we will describe our experimental set-up for testing out the active learning strategies.

5.5 EXPERIMENTAL SET-UP

In this work, we experimented with Bluetooth beacons as infrastructure devices. Our testbed is an academic library with three wooden shelves (each $11m$ long & $0.5m$ wide). We

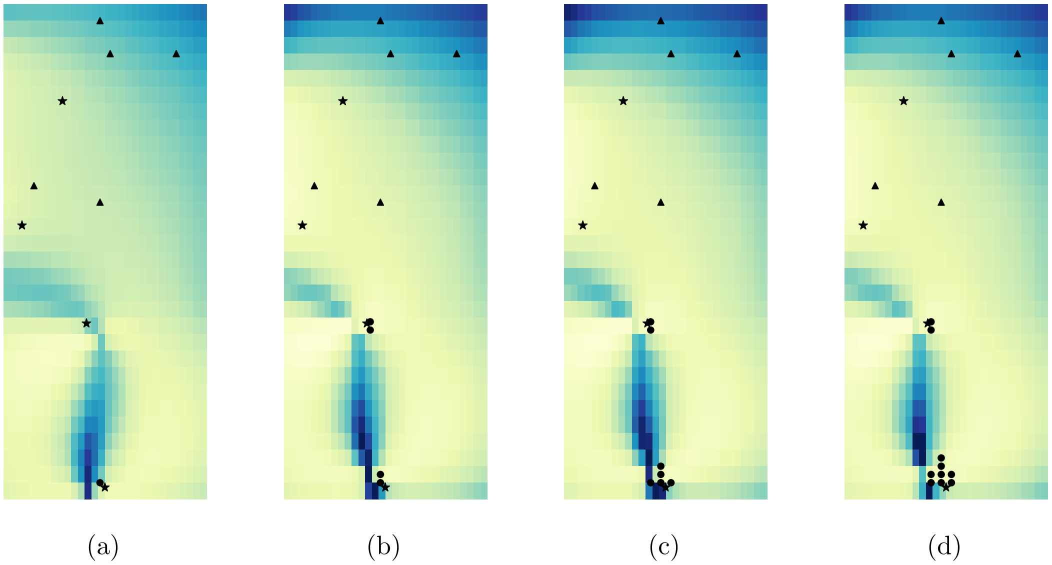


Figure 5.9: Heat map generated by Device Location Uncertainty strategy over multiple iterations. The stars represent the ground truth beacon locations. The triangles represent the initial locations chosen for cold-starting the strategy. The black circles are the points sampled by the strategy. (a) is the initial iteration, while (d) is the final iteration. The points are clustered in lower region of the space where it thinks the most uncertain beacon might be. It gives reasonable estimates of CRB patterns only in the lower region of the space.

placed 60 beacons in the space with an inter-beacon distance of $1m$. We collected data at 61 stationary locations through the layout.

To cold start our active learning strategies, we picked a few locations initially at random. We used data from the initial points to obtain an initial estimate of our models. Now out active learning strategies, based on initial model estimates, pick one additional location at each time step, adds data from that point to the existing data, and re-estimate the models. We repeated the experiment for different random initiations.

Active learning strategies got limited by the discretization of continuous space into 61 discrete points. Since we had actual world data from few discrete points in ample continuous space, the active learning strategies had to sample from those points. However, the most favorable location for data collection may be separate from those points. We needed a finer grid of data collection.

We could not collect data on a finer grid due to the onset of the pandemic, which made our testbed inaccessible. Instead, we used the collected data and the known device locations to train ground truth models, which could then simulate data for us over the entire continuous space. We used the same simulated data at a particular location across all strategies.

5.6 RESULTS

We compare the active learning strategies with a baseline strategy where we randomly sample points from the space. Note that these results are based on simulations. We used the empirical data collected from a university library space to train a model and then used that model to simulate data over the library space. While calculating location surveillance errors, we use the ground truth CRB values based on the ground truth locations of the Bluetooth beacons. We have shown the ground truth CRB map in Figure 5.10(a) and the final CRB-based location surveillance map generated by different strategies for one sample run in Figure 5.10. We show the aggregate errors over different random iterations in Figure 5.11.

First, lets take a detailed look at multiple iterations of Data Reception Uncertainty strategy(Figure 5.8) and Device Location Uncertainty strategy(Figure 5.9). As we can see in Figure 5.9(a) and Figure 5.8(a), both strategies started with similar estimates of the CRB map. However, as iterations progressed, Data Reception Uncertainty sampled points all over the space, while Device Location Uncertainty sampled points only in the lower region of the space. Hence, we observed that Device Location Uncertainty gives a good CRB map only in the lower region Figure 5.9(d). In contrast, Data Reception Uncertainty gives an overall better CRB map Figure 5.8(d).

Now lets look at the final CRB map generated by the Data Reception Uncertainty (Figure 5.10(b)) and Device Location Uncertainty(Figure 5.10(c)) with the ground truth CRB map (Figure 5.10(a)) for one sample run. The red stars in the ground truth map are actual beacon locations. The black triangles indicate the initiation points chosen for this sample run. The initiation points are the same across both strategies. The black circles indicate the points that the two strategies added over multiple iterations. We can see that the map generated by Data Reception Uncertainty better estimates the ground truth. We get good estimates of the dark blue regions present in the ground truth map, while Device Location Uncertainty only estimates the blue region at the bottom. Similarly, the Data Reception Uncertainty strategy also better infer the white spaces around the dark blue regions.

In Figure 5.11(a), we plot the median absolute error in CRB estimation over iterations. Note that for each iteration, we aggregate the results across multiple runs for multiple initializations. Main observations—The random and device location-based strategy performs the worst, followed by high data reception uncertainty and reduce reception uncertainty strategies. For the best strategy, CRB errors fall below 0.4m in the final iteration. For the best strategy, the variance in error also decreases over iterations.

In Figure 5.11(b), we directly look at the median errors in estimated device locations.

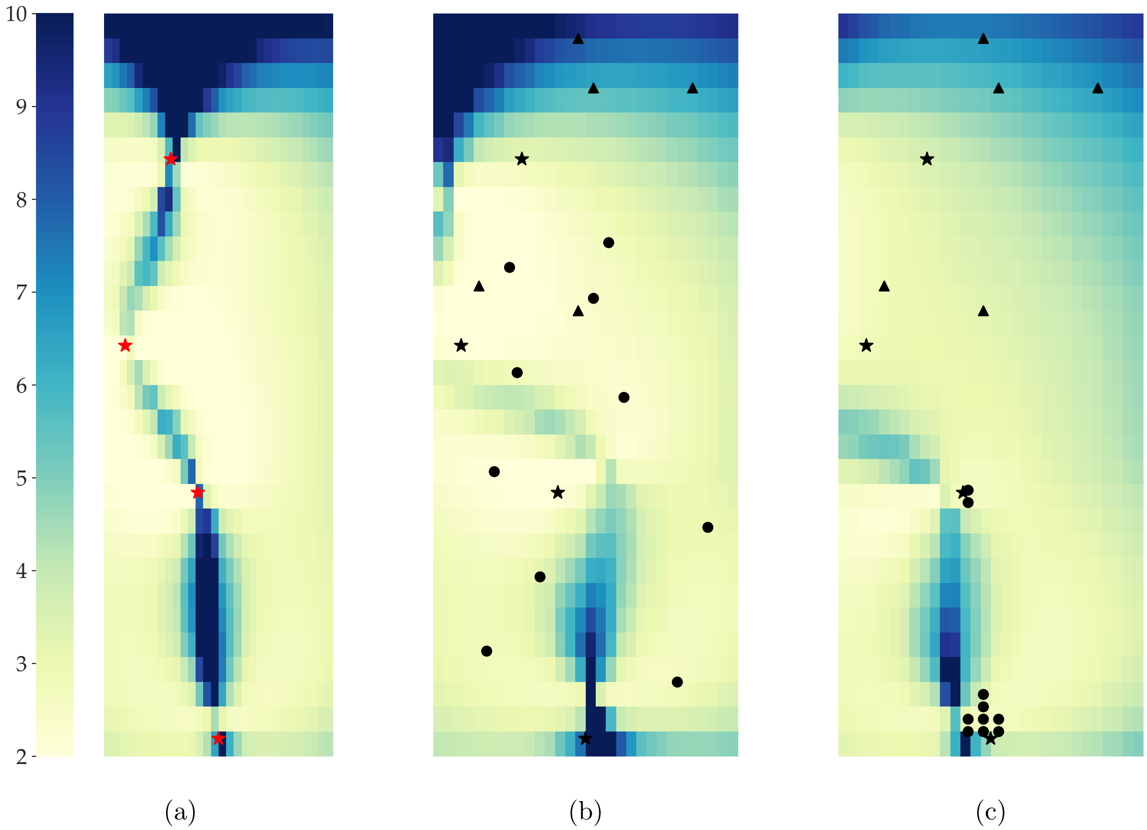


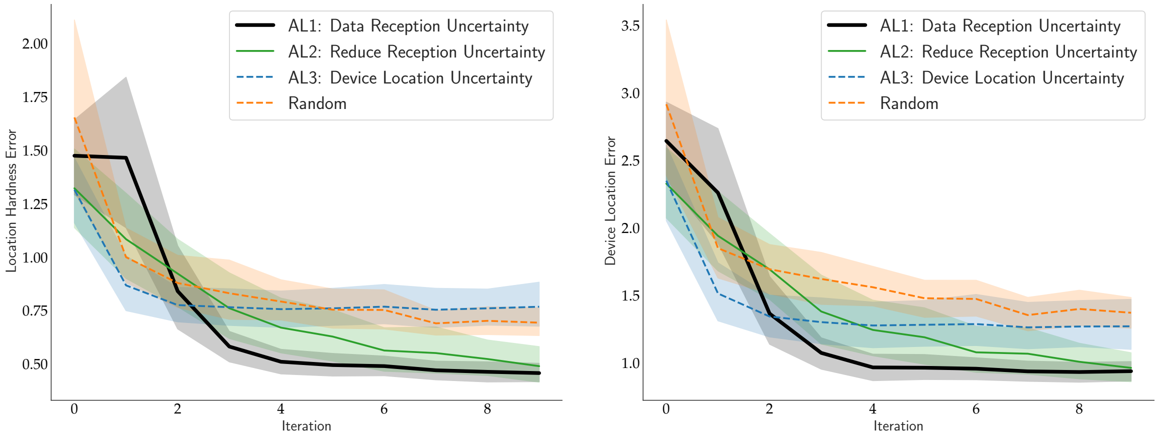
Figure 5.10: The stars represent the ground truth beacon locations. The triangles represent the initial locations chosen for cold-starting the active learning strategies. The black circles are the points sampled by the active learning strategy. (a) Ground Truth CRB-based Location Surveillance Map. We use ground truth device locations(red stars) to calculate this map. (b) Location Surveillance map estimated by Reception Data Uncertainty strategy. The sampled points(black circles) distribute well over the space and estimate a closer map to the ground truth. (c) Location Surveillance map estimated by Device Location-based strategy. Sampled points cluster near the uncertain device location and estimate a map that is better only in that region of the space

.

Similar to CRB results, the baseline performed the worst, and Data reception uncertainty performed the best. We can see that beacon error reduces over time from 3m to less than 1m with data reception.

5.7 CONCLUSION

This work introduces the idea of estimating location surveillance in a public space to increase people's trust towards location-finding technologies. Estimating the current level of



(a) (b)

Figure 5.11: (a) Median absolute error in **surveillance** over iterations. The baseline random and Device location strategy performs the worst, while reception data uncertainty strategies perform best. (b) Median error in estimated **device locations**. Reception data uncertainty strategies again perform better.

.

surveillance is an essential first step towards achieving declarative privacy where people can dictate their comfortable level of location surveillance. To estimate location surveillance, this work focuses on auditing the localization infrastructure installed in a particular environment and not on developing a new privacy-preserving protocol.

We propose two key insights in this work—location surveillance is a distribution (rather than a single number), and we will use the crowd to estimate that distribution. Our crowdsourcing framework uses a Bayesian model to aggregate data points collected by several people and estimates a single surveillance map for the space. Our active learning strategies can select good data collection points in the space to better estimate the surveillance map. We did studies based on actual-world data collected from an academic library. We saw that the best active learning strategy could detect location surveillance with an error of $0.4m$ for RSSI based location estimator.

CHAPTER 6: CONCLUSION AND FUTURE WORK

This thesis takes a practical approach towards indoor localization—what are the limitations in the existing localization solutions that impede their deployment towards scenarios like reliable contact tracing solutions? Despite 20 years of advance in indoor localization that led to the development of advanced features like RSSI [1, 16, 17] and CSI [4, 5, 6], why we could not use these features in contact tracing apps like Aarogya Setu and BlueTrace? We have identified three main limitations—deployability, accuracy, and privacy. We demonstrate these requirements in three different end-to-end problems—an infrastructure-based location estimation problem, a peer-to-peer distance estimation problem that uses minimal infrastructure, and a collaborative tracking estimation problem where people estimate properties of localization infrastructure.

6.1 RESEARCH CONTRIBUTIONS

We now present a summary of the work done in this thesis. We begin by summarizing our approach to location estimation.

6.1.1 Location estimation using packet reception probability

We built a new localization feature, Packet Reception Probability (PRP), which measures the probability that a receiver successfully receives packets from the transmitter. PRP is a statistical measure different from other traditional correlates of distance like RSSI or CSI, which are physical measurements. We have shown through empirical experiments that PRP encodes distance. Traditional measures like RSSI are highly erroneous on low power protocols like BLE. RSSI suffers from a sampling bias: they only use RSSI from decoded packets. Packet Reception Probability (PRP) uses negative information as a feature to solve the positive bias present in RSSI values and can achieve accurate localization using low-power BLE on easily deployable smartphones. RSSI also suffers from well-known multipath effects [3, 40] where signals travel along different paths and merge in different combinations at the receiver giving high variance in RSSI values for the exact location. PRP incorporates a new model of multipath for public indoor spaces. We have shown with actual-world experiments that PRP can achieve accuracies around $\sim 1m$. PRP is more ubiquitously deployable than CSI and more accurate than RSSI. Fusion of PRP and RSSI further improves the overall localization accuracy over PRP.

6.1.2 Distance estimation using minimal infrastructure

During the COVID 19 pandemic, a significant problem is finding peer-to-peer distances between people in indoor public spaces. These distance estimations can alert people who came within 6ft (social distancing threshold) of an infected person. BLE apps that rely on pairwise measurements like received signal strength to measure contact between two people suffer from the impact of latent factors like device relative positioning on the human body, the orientation of the people carrying the devices, and environmental multi-path. We built ContactTracer that can fight against latent factors in two ways—using known distances with minimal infrastructure in indoor environments and using a collaboration of unknown distances in outdoor environments. First, if we have a few infrastructure devices (e.g., Bluetooth beacons) installed at known locations in an environment, we can make more measurements between a person's phone and the BLE beacons. Also, since we install the beacons at known locations, we know the distances between these beacons. We have used these known distances in a triangle inequality framework to estimate the unknown distance between two persons. Second, in an outdoor environment where we cannot install infrastructure devices, we can collaborate between people to solve distances more accurately. We can impose triangle inequalities between each triplet of people. More people help us form more of these triplets, leading to more constraints and, hence, a better localization solution. We experimented with an infrastructure-based contact tracing solution in the library and retail store. We get median error of 0.89 m (library) and 1.07 m (retail store) with PRP values. The corresponding errors with RSSI are 1.36 m (library, 52.8% more error) and 1.34 m (retail store, 25.2% more error). We have also designed the architecture, protocol, packet structure, iOS and Android apps to deploy the collaborative contact tracing solution.

6.1.3 Location surveillance estimation using crowdsourcing

Location services raise a fundamental problem—*Location surveillance or tracking*. While people want services, they do not want to be surveilled. Furthermore, people using an indoor location system do not know the granularity of the location information that the system collects about them. The performance of a location system depends on the nature of the infrastructure. In this work, we build CrowdEstimator, a crowdsourcing system that harnesses the power of the crowd to audit the nature of the infrastructure and estimate the surveillance or level of tracking in space. We leverage the power of the crowd to collect many individual traces and consolidate them in a central repository. This work also introduces the insight of location tracking or surveillance as an error radius that varies across space.

This insight is a novel contribution to the best of our knowledge since previous works have only thought about tracking accuracy and tried to measure a single number for the entire space. The reality is that tracking granularity is spatial. We need to estimate all portions of the spatial surveillance map with a certain accuracy threshold. We have designed active learning strategies to pick up spots or regions inside a public space where data collection will lead to a maximal improvement of our overall estimate surveillance map. We did synthetic studies based on models trained by actual data collected from the library environment. Our studies showed that active learning strategies could more uniformly determine the location surveillance map over the entire space. With the best performing strategy, we were able to determine the tracking map with an accuracy of $0.4m$ and all the infrastructure device locations with an accuracy of less than $1m$.

6.2 FUTURE DIRECTIONS

Our work has taken the first step towards achieving two important goals besides accuracy that we identified for localization solutions—deployability on smartphones and privacy. Going forward, we will need to solve the following research directions to achieve those goals fully.

6.2.1 Device adaptive models for wireless transmission

Our empirical experiments with contact tracing applications observed inter-device variation in received wireless properties like RSSI or PRP. The different types of antenna designs in different smartphones lead to such variations. To measure location or distance accurately using smartphones, we need to use the correct model for wireless reception. If we try to use a model trained with smartphone A to localize using smartphone B, we will have high errors. Training an individual model for each smartphone is not scalable, given the wide range of phones available today. We need to invest in techniques like model adaptive machine learning [106] that can learn a general model using training data from some device and adapt well/quickly to other devices. Learning wireless models that can easily adapt to many devices will be crucial for the universal application of location-based services.

6.2.2 Capturing dynamic environmental effects in location surveillance

In our definition of location surveillance, we made a simplistic assumption that SNR (signal-to-noise ratio) is constant across space. The assumption helped us to define surveillance in terms of a theoretical lower bound. However, real-world spaces suffer from complicated interactions of environmental factors like nearby reflectors, area, number of reflectors, the material of reflectors, and distance to reflectors. Such effects are not captured in the theoretical lower bound formula. Hence, we often end up with a very loose lower bound for specific regions, which location estimators cannot achieve in reality. In future work, we need to investigate other location surveillance definitions that better encode these environmental factors.

6.2.3 Extending location surveillance estimation to CSI techniques

Our current work of location surveillance estimation focuses on signal properties like RSSI or PRP that are available on ubiquitous technology like smartphones. While this work helps us to concentrate on technologies that can deploy to people today, we have started seeing some work [35] that can enable CSI on selected smartphones in the future. Our current definition of location surveillance uses Cramer-Rao Bound (CRB) [105] as a theoretical lower bound for RSSI-based estimators. CRB for RSSI estimators depends only on the location of the infrastructure devices. We know that for CSI estimators, the number and the orientation of the antennas on the device are essential to estimate the angle of the received signal. Location surveillance bounds for CSI estimators need to consider antenna properties like the number, location, and orientation. Some works [107] have tried to find CRB bound for CSI-based estimators. In future work, we need to investigate definitions of location surveillance for CSI estimators and then come up with algorithms to estimate CSI surveillance.

6.2.4 Handling information asymmetry in location surveillance estimation

Our current work makes a strong assumption on information symmetry, i.e., our system uses the same raw data that the infrastructure is collecting for location estimation. Both the infrastructure and our system obtain this data from the smartphones of each user. We make a strong assumption about the goodwill of the infrastructure that they will reveal the same technology they are using for localization to a third-party crowd app. *What happens if the infrastructure is malicious?* For example, they reveal that they are using RSSI estimators, and our crowd app collects RSSI traces from smartphones. However, the infrastructure is also measuring CSI on their access points. Lower bounds obtained from RSSI CRB can give users a false sense of privacy, while CSI measurements help the infrastructure track people more accurately. The current approach cannot guarantee surveillance estimation in such cases of information asymmetry. We need to investigate if we can get proxies for CSI-based surveillance using the RSSI traces that we collect from smartphones in future work. Can we use RSSI or other data traces from smartphones to obtain a universal lower bound across all possible location estimators (RSSI, CSI, or PRP)?

6.2.5 Declarative privacy

Achieving a declarative definition of privacy is an essential requirement for the wide-scale adaption of localization solutions. We need to establish people or customers on the same level as these indoor location techniques to guide/provide input on their comfortable level of detection or surveillance. For example, a person in the airport should say,– do not locate which item the person is browsing in the vending machine. The system can locate the person's gate number at the airport for digital contact tracing. Detecting the level of location surveillance was an essential first step towards achieving privacy. Once we know the level of surveillance, we need to investigate appropriate hiding techniques that can enforce people's comfortable level of privacy.

REFERENCES

[1] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000*, vol. 2. Ieee, 2000, pp. 775–784.

[2] K. Heurtefeux and F. Valois, "Is rssi a good choice for localization in wireless sensor network?" in *AINA, 2012 IEEE 26th International Conference on.* IEEE, 2012, pp. 732–739.

[3] A. Zanella, "Best practice in rss measurements and ranging," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2662–2686, 2016.

[4] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 269–282.

[5] J. Xiong and K. Jamieson, "Arraytrack: a fine-grained indoor location system." Usenix, 2013.

[6] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single wifi access point." in *NSDI*, vol. 16, 2016, pp. 165–178.

[7] "Aarogya app," https://www.mygov.in/aarogya-setu-app/, 2020.

[8] "Bluetrace," https://bluetrace.io, 2020.

[9] "Pact: Private automated contact tracing," https://pact.mit.edu, 2020.

[10] "Inside amazon's surveillance-powered, no-checkout convenience store," https://tcrn.ch/2Bib823, 2018.

[11] "Shopkick. shopkick retail platform." http://www.shopkick.com/, 2009.

[12] "Techcrunch. wifarer brings indoor navigation to the royal bc museum." http://techcrunch.com/2012/08/01/wifarer-brings-indoor-navigation-to-the-royal-bc-museum/, 2012.

[13] "Zonedefense. zonedefense location-based mobile security platform." http://airpatrolcorp.com/products/zonedefense/.

[14] "Iot in indian railways." https://www.fabliantechnologies.com/eddystone-beacon-installation-at-indian-railway-stations-by-google/.

[15] "Iot in hong kong railways." https://railuk.com/rail-news/mtr-trials-bluetooth-beacon-navigation-at-one-of-its-stations/.

[16] K. Chintalapudi, A. Iyer, and V. Padmanabhan, “Indoor localization without the pain,” in *MobiCom*. ACM, 2010.

[17] M. Youssef and A. Agrawala, “The horus wlan location determination system,” in *MobiSys*. ACM, 2005.

[18] B. Grosswindhager, M. Rath, J. Kulmer, M. S. Bakr, C. A. Boano, K. Witrisal, and K. Römer, “Salma: Uwb-based single-anchor localization system using multipath assistance,” in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’18. New York, NY, USA: ACM, 2018. [Online]. Available: http://doi.acm.org/10.1145/3274783.3274844 pp. 132–144.

[19] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, “Does btle measure up against wifi? a comparison of indoor location performance,” in *European Wireless*. VDE, 2014.

[20] X. Wu, R. Shen, L. Fu, X. Tian, P. Liu, and X. Wang, “ibill: Using ibeacon and inertial sensors for accurate indoor localization in large open areas,” *IEEE Access*, vol. 5, pp. 14 589–14 599, 2017.

[21] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer, “A study of bluetooth low energy performance for human proximity detection in the workplace,” in *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*. IEEE, 2017, pp. 90–99.

[22] B. Großwindhager, M. Rath, J. Kulmer, M. S. Bakr, C. A. Boano, K. Witrisal, and K. Römer, “Salma: Uwb-based single-anchor localization system using multipath assistance,” in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2018, pp. 132–144.

[23] J. Wang and D. Katabi, “Dude, where’s my card?: Rfid positioning that works with multipath and non-line of sight,” in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 51–62.

[24] L. Yang, J. Cao, W. Zhu, and S. Tang, “Accurate and efficient object tracking based on passive rfid,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 11, pp. 2188–2200, 2015.

[25] C. Jiang, Y. He, X. Zheng, and Y. Liu, “Orientation-aware rfid tracking with centimeter-level accuracy,” in *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE Press, 2018, pp. 290–301.

[26] C. Jiang, Y. He, S. Yang, J. Guo, and Y. Liu, “3d-omnitrack: 3d tracking with COTS RFID systems,” in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks, IPSN 2019, Montreal, QC, Canada, April 16-18, 2019*, 2019. [Online]. Available: https://doi.org/10.1145/3302506.3310386 pp. 25–36.

[27] S. Liu and T. He, "Smartlight: Light-weight 3d indoor localization using a single led lamp," in *SenSys*, 2017.

[28] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *MobiCom*. ACM, 2014, pp. 447–458.

[29] C. Zhang and X. Zhang, "Pulsar: Towards ubiquitous visible light localization," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017, pp. 208–221.

[30] S. Zhu and X. Zhang, "Enabling high-precision visible light localization in today's buildings," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 96–108.

[31] J. Yang and Y. Chen, "Indoor localization using improved rss-based lateration methods," in *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*. IEEE, 2009, pp. 1–6.

[32] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 483–494.

[33] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "Sail: Single access point-based indoor localization," in *MobiSys*. ACM, 2014, pp. 315–328.

[34] "Cdc website," https://www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing/contact-tracing-plan/appendix.html#contact, 2020.

[35] M. Schulz, J. Link, F. Gringoli, and M. Hollick, "Shadow wi-fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over wi-fi," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 256–268.

[36] M. Abueg, R. Hinch, N. Wu, L. Liu, W. Probert, A. Wu, P. Eastham, Y. Shafi, M. Rosencrantz, M. Dikovsky et al., "Modeling the effect of exposure notification and non-pharmaceutical interventions on covid-19 transmission in washington state," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–10, 2021.

[37] "Covid apps effectiveness." https://www.technologyreview.com/2020/06/05/1002775/covid-apps-effective-at-less-than-60-percent-download/.

[38] D. Grande, N. Mitra, X. L. Marti, R. Merchant, D. Asch, A. Dolan, M. Sharma, and C. Cannuscio, "Consumer views on using digital data for covid-19 control in the united states," *JAMA network open*, vol. 4, no. 5, pp. e2 110 918–e2 110 918, 2021.

[39] D. Madigan, E. Einahrawy, R. P. Martin, W.-H. Ju, P. Krishnan, and A. Krishnakumar, "Bayesian indoor positioning systems," in *INFOCOM 2005*, vol. 2. IEEE, 2005, pp. 1217–1227.

[40] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental limits of rss fingerprinting based indoor localization," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 2479–2487.

[41] N. Van Doremalen, T. Bushmaker, D. H. Morris, M. G. Holbrook, A. Gamble, B. N. Williamson, A. Tamin, J. L. Harcourt, N. J. Thornburg, S. I. Gerber et al., "Aerosol and surface stability of sars-cov-2 as compared with sars-cov-1," *New England Journal of Medicine*, vol. 382, no. 16, pp. 1564–1567, 2020.

[42] G. Kaptchuk, D. G. Goldstein, E. Hargittai, J. Hofman, and E. M. Redmiles, "How good is good enough for covid19 apps? the influence of benefits, accuracy, and privacy on willingness to adopt," *arXiv preprint arXiv:2005.04343*, 2020.

[43] J. Xiong and K. Jamieson, "Secureangle: improving wireless security using angle-of-arrival information," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 2010, p. 11.

[44] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding wifi localization," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 2013, pp. 249–262.

[45] S. Sen, D. Kim, S. Laroche, K.-H. Kim, and J. Lee, "Bringing cupid indoor positioning system to practice," in *WWW*. International World Wide Web Conferences Steering Committee, 2015, pp. 938–948.

[46] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. Corner, "Virtual compass: relative positioning to sense mobile social interactions," in *International Conference on Pervasive Computing*. Springer, 2010, pp. 1–21.

[47] L. Chen, L. Pei, H. Kuusniemi, Y. Chen, T. Kröger, and R. Chen, "Bayesian fusion for indoor positioning using bluetooth fingerprints," *Wireless personal communications*, vol. 70, no. 4, pp. 1735–1745, 2013.

[48] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *MobiCom*. ACM, 2012, pp. 293–304.

[49] J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie, "Growing an organic indoor location system," in *MobiSys*. ACM, 2010, pp. 271–284.

[50] S. Yang, P. Dessai, M. Verma, and M. Gerla, "Freeloc: Calibration-free crowdsourced indoor localization," in *INFOCOM*. IEEE, 2013, pp. 2481–2489.

[51] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.

[52] Y. Chen, M. Guo, J. Shen, and J. Cao, "A graph-based method for indoor subarea localization with zero-configuration," in *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences.* IEEE, 2016, pp. 236–244.

[53] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "Fm-based indoor localization," in *MobiSys.* ACM, 2012.

[54] S.-Y. Lau, T.-H. Lin, T.-Y. Huang, I.-H. Ng, and P. Huang, "A measurement study of zigbee-based indoor localization systems under rf interference," in *Proceedings of the 4th ACM international workshop on Experimental evaluation and characterization.* ACM, 2009, pp. 35–42.

[55] W. Huang, Y. Xiong, X.-Y. Li, H. Lin, X. Mao, P. Yang, and Y. Liu, "Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones," in *INFOCOM, 2014 Proceedings IEEE.* IEEE, 2014, pp. 370–378.

[56] J. Palacios, G. Bielsa, P. Casari, and J. Widmer, "Single- and multiple-access point indoor localization for millimeter-wave networks," *IEEE Trans. Wireless Communications*, vol. 18, no. 3, pp. 1927–1942, 2019. [Online]. Available: https://doi.org/10.1109/TWC.2019.2899313

[57] G. Bielsa, J. Palacios, A. Loch, D. Steinmetzer, P. Casari, and J. Widmer, "Indoor localization using commercial off-the-shelf 60 ghz access points," in *2018 IEEE Conference on Computer Communications, INFOCOM 2018, Honolulu, HI, USA, April 16-19, 2018*, 2018. [Online]. Available: https://doi.org/10.1109/INFOCOM.2018.8486232 pp. 2384–2392.

[58] R. Nandakumar, V. Iyer, and S. Gollakota, "3d localization for sub-centimeter sized devices," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, SenSys 2018, Shenzhen, China, November 4-7, 2018*, 2018. [Online]. Available: https://doi.org/10.1145/3274783.3274851 pp. 108–119.

[59] S. Liu and A. Striegel, "Accurate extraction of face-to-face proximity using smartphones and bluetooth," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN).* IEEE, 2011, pp. 1–5.

[60] C. Wymant, L. Ferretti, D. Tsallis, M. Charalambides, L. Abeler-Dörner, D. Bonsall, R. Hinch, M. Kendall, L. Milsom, M. Ayres et al., "The epidemiological impact of the nhs covid-19 app," *Nature*, pp. 1–8, 2021.

[61] D. Menges, H. E. Aschmann, A. Moser, C. L. Althaus, and V. Von Wyl, "A data-driven simulation of the exposure notification cascade for digital contact tracing of sars-cov-2 in zurich, switzerland," *JAMA network open*, vol. 4, no. 4, pp. e218 184–e218 184, 2021.

[62] "Covid apps around the world." https://www.technologyreview.com/2020/12/16/1014878/covid-tracing-tracker/.

[63] "Virusafe," https://virusafe.info.

[64] "Rakning c-19," https://www.covid.is/app/is.

[65] "Virusradar," https://virusradar.hu.

[66] "Covidradar," https://covidradar.mx.

[67] "Stopp-corona," https://www.stopp-corona.at.

[68] "Covid-alert," https://www.canada.ca/en/public-health/services/diseases/coronavirus-disease-covid-19/covid-alert.htmlt.

[69] "Guidesafe," https://www.guidesafe.org/exposure-notification-app/.

[70] "Coviddefense," https://www.coviddefensela.com.

[71] "Swisscovid," https://foph-coronavirus.ch/swisscovid-app/#download.

[72] "Coronalert," https://coronalert.be/en/.

[73] H. Li, L. Sun, H. Zhu, X. Lu, and X. Cheng, "Achieving privacy preservation in wifi fingerprint-based localization," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications.* IEEE, 2014, pp. 2337–2345.

[74] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques.* Springer, 1999, pp. 223–238.

[75] A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, and Y. Theodoridis, "Privacy-preserving indoor localization on smartphones," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3042–3055, 2015.

[76] D. Chen, K. G. Shin, Y. Jiang, and K.-H. Kim, "Locating and tracking ble beacons with smartphones," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, 2017, pp. 263–275.

[77] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[78] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR'94.* Springer, 1994, pp. 3–12.

[79] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction. int. conf. on machine learning," 2001.

[80] B. Settles and M. Craven, “An analysis of active learning strategies for sequence labeling tasks,” in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 1070–1079.

[81] D. Vasisht, A. Jain, C.-Y. Hsu, Z. Kabelac, and D. Katabi, “Duet: Estimating user position and identity in smart homes using intermittent and incomplete rf-data,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, p. 84, 2018.

[82] S. He and S.-H. G. Chan, “Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[83] X. Tian, M. Wang, W. Li, B. Jiang, D. Xu, X. Wang, and J. Xu, “Improve accuracy of fingerprinting localization with temporal correlation of the rss,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 113–126, 2018.

[84] A. F. Harris III, V. Khanna, G. Tuncay, R. Want, and R. Kravets, “Bluetooth low energy in dense iot environments,” *IEEE Communications Magazine*, vol. 54, no. 12, pp. 30–36, 2016.

[85] X. Tian, S. Zhu, S. Xiong, B. Jiang, Y. Yang, and X. Wang, “Performance analysis of wi-fi indoor localization with channel state information,” *IEEE Transactions on Mobile Computing*, 2018.

[86] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, “E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures,” in *Proceedings of the 20th annual international conference on Mobile computing and networking.* ACM, 2014, pp. 617–628.

[87] L. Hu and D. Evans, “Localization for mobile sensor networks,” in *Proceedings of the 10th annual international conference on Mobile computing and networking.* ACM, 2004, pp. 45–57.

[88] M. Rudafshani and S. Datta, “Localization in wireless sensor networks,” in *IPSN 2007. 6th International Symposium on.* IEEE, 2007, pp. 51–60.

[89] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *Proceedings of the 9th annual international conference on Mobile computing and networking.* ACM, 2003, pp. 81–95.

[90] “Proximity marketing in retail,” https://www.proximity.directory/reports/, 2017.

[91] M. Radhakrishnan, S. Eswaran, A. Misra, D. Chander, and K. Dasgupta, “Iris: Tapping wearable sensing to capture in-store retail insights on shoppers,” 2016.

[92] M. Radhakrishnan, S. Sen, V. Subbaraju, A. Misra, and R. Balan, “Iot+ small data: Transforming in-store shopping analytics and services,” 2016.

[93] T. Liu, L. Yang, X.-Y. Li, H. Huang, and Y. Liu, "Tagbooth: Deep shopping data acquisition powered by rfid tags," in *Computer Communications (INFOCOM), 2015 IEEE Conference on.* IEEE, 2015, pp. 1670–1678.

[94] L. Shangguan, Z. Zhou, X. Zheng, L. Yang, Y. Liu, and J. Han, "Shopminer: Mining customer shopping behavior in physical clothing stores with cots rfid devices," in *Proceedings of the 13th ACM conference on embedded networked sensor systems.* ACM, 2015, pp. 113–125.

[95] S. De, S. Chowdhary, A. Shirke, Y. L. Lo, R. Kravets, and H. Sundaram, "Finding by counting: a probabilistic packet count model for indoor localization in ble environments," in *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization.* ACM, 2017, pp. 67–74.

[96] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck, "Probabilistic programming in python using pymc3," *PeerJ Computer Science*, vol. 2, p. e55, 2016.

[97] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1593–1623, 2014.

[98] "ibeek," http://bluvision.com/ibeek-5/, 2015.

[99] "Blufi,bluetooth to wifi gateway," http://bluvision.com/blufi/, 2014.

[100] "altbeacon," https://github.com/AltBeacon/android-beacon-library-reference, 2014.

[101] R. Ayyalasomayajula, D. Vasisht, and D. Bharadia, "Bloc: Csi-based accurate localization for ble tags." in *ACM CoNEXT*, 2018.

[102] "iperf," https://iperf.fr, 2003.

[103] W. E. Wei, Z. Li, C. J. Chiew, S. E. Yong, M. P. Toh, and V. J. Lee, "Presymptomatic transmission of sars-cov-2—singapore, january 23–march 16, 2020," *Morbidity and Mortality Weekly Report*, vol. 69, no. 14, p. 411, 2020.

[104] G. F. Hatke, M. Montanari, S. Appadwedula, M. Wentz, J. Meklenburg, L. Ivers, J. Watson, and P. Fiore, "Using bluetooth low energy (ble) signal strength estimation to facilitate contact tracing for covid-19," *arXiv preprint arXiv:2006.15711*, 2020.

[105] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on signal processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[106] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning.* PMLR, 2017, pp. 1126–1135.

[107] Y. Shen and M. Z. Win, "Fundamental limits of wideband localization—part i: A general framework," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 4956–4980, 2010.