October 2021

Human Mobility Monitoring using WiFi: Analysis, Modeling, and Applications

Amee Trivedi
University of Massachusetts Amherst

HUMAN MOBILITY MONITORING USING WiFi : ANALYSIS, MODELING, AND APPLICATIONS

A Dissertation Presented

by

AMEE TRIVEDI

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2021

CICS

HUMAN MOBILITY MONITORING USING WiFi : ANALYSIS, MODELING, AND APPLICATIONS

A Dissertation Presented

by

AMEE TRIVEDI

Approved as to style and content by:

Dr. Prashant Shenoy, Chair

Dr. Deepak Ganesan, Member

Dr. Jeremy Gummeson, Member

Dr. Tauhidur Rahman, Member

James Allan, Chair of the Faculty
CICS

ABSTRACT

HUMAN MOBILITY MONITORING USING WiFi : ANALYSIS, MODELING, AND APPLICATIONS

SEPTEMBER 2021

AMEE TRIVEDI

B.E., K.J.SOMAIYA, MUMBAI UNIVERSITY

M.Sc., UNIVERSITY OF MASSACHUSETTS, AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Dr. Prashant Shenoy

Understanding and modeling humans and device mobility has fundamental importance in mobile computing, with implications ranging from network design and location-aware technologies to urban infrastructure planning. Today's users carry a plethora of devices such as smartphones, laptops, tablets, and smartwatches, with each device offering a different set of services resulting in different usage and mobility leading to the research question of understanding and modeling multiple user device trajectories. Additionally, prior research on mobility focuses on outdoor mobility when it is known that users spend 80% of their time indoors resulting in wide gaps in knowledge in the area of indoor mobility of users and devices. Here, I try to fill the gaps in mobility modeling in the areas of understanding and modeling indoor-outdoor human mobility as well as multi-device mobility.

In this thesis, I propose the characterization and modeling of human and device mobility. Further, I design and deploy mobility-aware applications for contact tracing of infectious diseases and energy-aware Heating, Ventilation, and Air Conditioning (HVAC) scheduling. I try and answer a sequence of four primary inter-related questions : (1) how is indoor and outdoor user mobility different, (2) are multiple device trajectories belonging to a single user correlated, (3) how to model indoor mobility of users and (4) how to design effective mobility aware applications that are easily deployable and align with long term goals of sustainability as well relay positive societal impact. The insights gained from each question serves as a base to build up on the next question in the series.

I present answers to these questions across three main parts of my thesis. The first part comprises of characterization and analysis of human and device mobility. In this part I design and develop tool to extract device trajectories from WiFi system logs "syslog" and map devices to users. These extracted trajectories and device to user mapping are used to characterize and empirically analyze the mobility of users at varying spatial granularity (indoor, outdoor) and extract device mobility correlations between multiple devices of users and forms the first part of my thesis. In the second part, based on the insights gained from the multi-granular and multi-device mobility characterization stated above, I argue that mobility is inherently hierarchical in nature and propose novel indoor human mobility modeling approach. Third, I leverage the passively observed mobility to design mobility-aware applications that either look back or look ahead in time. WiFiTrace is a look back or backtracking application that is a network-centric contact tracing tool to aid healthcare workers in manual contact tracing of infectious diseases and iSchedule is a look ahead machine learning based mobility-aware energy-saving application that predicts Heating, Ventilation, and Air Conditioning (HVAC) schedule for higher energy savings while increasing user comfort.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

How, why, where, and when people move - has intrigued us for a long time. Understanding the mobility of users and their devices has become very important in the era of mobile Internet with implications from the design of mobile services to urban infrastructure. With the ever so increasing availability of digital traces of mobile devices, understanding user and device mobility through passive sensing has become feasible. This thesis explores the challenges and opportunities in leveraging WiFi system logs to infer actionable insights for modeling mobility and designing effective mobility-aware applications by applying several techniques from Data Mining, Machine Learning, Deep Learning, and Natural Language Processing.

1.1 Motivation

A wide range of applications, from system design and urban planning to personalized assistants and simulations for spread of diseases, depend on understanding, modeling, and predicting human mobility. Uncovering the anatomy of human mobility and understanding its statistical properties have been a favorite among researchers across a variety of fields ranging from computer science to epidemiology.

In the past, a lot of effort to understand human mobility was through surveys, paths traversed by banknotes, and letters leading to a very high-level coarse understanding of human mobility. However, with the advent of the internet and advancement in mobile technologies modern day efforts to understand human mobility use datasets curated from digital traces Global Positioning System (GPS), Call Data

Record (CDR) of mobile devices (smartphones, tablets), user social media check-ins, vehicle logs, etc. While most of these modern day datasets are collected passively they represent a very macro level or outdoor view of human mobility.

Studies have shown that humans spend over 80% of their lives indoors and inside buildings [68]. Consequently, understanding *indoor* mobility is equally important for purposes of modeling and system design, but this area has seen much less work than *outdoor* mobility of users [122, 114, 52, 32]. Recent research has recognized that indoor mobility of users inside buildings, where many users spend a significant portion of the day, is very different from outdoor mobility exhibited when walking in a city or traveling in vehicles [124, 125]. While a few recent efforts have specifically focused on indoor mobility [69, 125] it has been from the network perspective and many research questions remain unanswered.

Additionally, in today's world users carry a plethora of mobile devices such as smartphones, laptops, tablets, etc. Understanding the mobility of these devices and the correlations among their mobility patterns serve as a base for collaborative multi-device task management and this area has not been explored.

To push the frontier of human mobility research, in this thesis proposal, I mainly explore the following questions :

- What are the differences in mobility patterns of users and devices at various spatial granularities (indoor/outdoor)?
- How correlated are the mobility patterns exhibited by different devices belonging to the same user?
- What are the implications of these findings on mobility modeling research? How to design mobility models that reflect the insights gained from the empirical analysis?

- How can the above analysis be used in designing effective and accurate data-driven mobility-aware applications that are easy to deploy, align with long term goals of sustainability, and relay a positive societal impact.

I believe that this confluence of factors — i) ubiquity of mobile devices and ii) availability of digital traces of the mobile devices — permits the use of sophisticated data analytic techniques to infer actionable insights. As more mobile devices become increasingly prevalent and ubiquitous, and smartphone mobility becomes a proxy for user mobility, I argue that modeling device and user mobility patterns with passive sensing using data-driven modeling will be crucial in designing mobility-aware applications.

1.2 Thesis Contributions

In this thesis, I propose tools, empirical analysis, mobility models, and mobility-aware systems based on passive mobility derived from WiFi traces. The primary components and contributions of this thesis are as follows:

- *Trajectory Extraction and Empirical Analysis :* A tool to extract device trajectories from WiFi logs and empirical analysis of extracted spatio-temporal trajectories to understand the characteristics of human and device mobility at multiple spatial granularity.

- *Mobility Modeling :* An indoor human mobility model for accurately predicting the future mobility or entire day trajectory of users across buildings at multiple spatial scale encapsulating the correlations across the outdoor and indoor mobility features.

- *Mobility-aware Applications:* Based on the passively sensed trajectories we can design applications that broadly fall into 2 categories- (i) applications where we backtrack the past observed mobility after an event happens in time to infer

attributes, causes, or preventive actions associated to the event, (ii) applications based on predicting future mobility or aggregated mobility (such as occupancy prediction) over a horizon. In this thesis, I present one application per category:

- *Network-centric Contact Tracing :* WiFiTrace, a network-centric approach for contact tracing of infectious diseases using passive WiFi sensing. We backtrack observed user trajectories for infectious disease containment.
- *Mobility-aware HVAC Scheduling :* iSchedule, a machine learning-driven technique to automatically learn custom occupancy-based Heating, Ventilation, and Air Conditioning (HVAC) schedules for buildings across a large campus. An application based on predicting future building occupancies by observing past mobility and occupancy of each building.

Each component is described in more detail below.

1.2.1 Trajectory Extraction, Analysis, and Modeling

With the large amount of digital traces left by user devices or user check-ins, there has been a great interest in understanding human mobility. There is extensive work on understanding mobility at urban scales [122, 114, 52, 32], activity identification [73, 116, 51], and point of interest areas [117]. This body of work has focused on characterizing and modeling *outdoor* mobility at large spatial scales, such as cities and campuses [87, 63], as well as different temporal scales, by using a variety of data sources such as cellular, WiFi, social media check-ins, and vehicular data [48, 107, 49, 61].

However, users spend around 80% of their time indoors [68]. Consequently, understanding *indoor* mobility is equally important for purposes of modeling and system design, but this area has seen much less work than *outdoor* mobility of users. In this work, I present a detailed empirical analysis of indoor building mobility in a university campus and present key insights on differences in mobility at various spatial scales. I

also explore the mobility correlations between trajectories of multiple devices belonging to the same user. Overall, this study reveals significant differences between indoor mobility and prior results on outdoor mobility, indicating that system designers need to carefully consider our findings when designing systems for indoor use.

1.2.2 Mobility Modeling

Modeling human mobility has a wide range of applications from urban planning to simulations of disease spread. It is well known that humans spend 80% of their time indoors [12] but modeling indoor human mobility is challenging due to three main reasons: (i) the absence of easily acquirable, reliable, low-cost indoor mobility dataset, (ii) high prediction space in modeling the frequent indoor mobility, and (iii) multi-scalar periodicity and correlations in mobility. To deal with all these challenges, I propose WiFiMod, a transformer based data-driven approach that models indoor human mobility at multiple spatial scales using WiFi system logs. WiFiMod takes as input enterprise WiFi system logs to extract human mobility trajectories from smartphone digital traces. Next, for each extracted trajectory it identifies the mobility features at multiple spatial scales, *macro and micro*, to design a multi-modal embedding transformer that predicts user mobility for several hours to an entire day across multiple spatial granularities. Multi-modal embedding captures the mobility periodicity and correlations across various scales while transformers capture long term mobility dependencies boosting model prediction performance. Further, the multi-scale approach significantly reduces the prediction space by first predicting *macro mobility*, then modeling indoor scale mobility, *micro mobility*, conditioned on the estimated macro mobility distribution, thereby using the topological constraint of the macro-scale.

1.2.3 Mobility-aware Applications

1.2.3.1 Network-Centric Contact Tracing

Contact tracing is a well-established and effective approach for the containment of the spread of infectious diseases. While manual methods for contact tracing are well established in the medical community [33], such methods are laborious and do not scale well. As a result, there has been a push for digital contact tracing solutions. While bluetooth-based contact tracing method using phones have become popular recently, these approaches suffer from the need for a critical mass of adoption in order to be effective. I present an alternate contact tracing solution, WiFiTrace, a network-centric approach for contact tracing that relies on passive WiFi sensing with no client-side involvement. WiFiTrace exploits WiFi network logs gathered by enterprise networks for performance and security monitoring and utilizes them for reconstructing device trajectories for contact tracing. WiFiTrace enhances the efficacy of traditional methods, rather than to supplant them with a new technology. WiFiTrace is based on an efficient graph algorithm to scale the approach to large networks with tens of thousands of users.

1.2.3.2 Mobility-aware Energy Efficient Application

Heating, ventilation, and air conditioning (HVAC) systems account for over 50% of the energy consumed by commercial buildings. Though "smart" HVAC technologies, such as learning thermostats, are widely available for residential use, commercial buildings rely on legacy systems that are difficult to upgrade and require facility managers to manually set HVAC schedules. Static HVAC schedules set by Building Facility Managers are neither energy-efficient nor provide high user comfort.

To address this issue, I propose a novel Machine Learning-driven technique to automatically learn custom occupancy-based HVAC schedules for buildings based on the past occupancy observed by the building across various spatio-temporal granular-

ities. The system is highly adaptive and can be customized to generate schedules per building per day of the week. The building occupancy is derived passively across a large campus that leverages the existing omnipresent wireless networking infrastructure in a modern campus. I present an algorithm that learns from these patterns to derive a custom HVAC schedule. This approach is adaptive and dynamically adjusts its schedules as occupancy patterns change, much like a learning thermostat.

1.3 Outline

I structure the remainder of this thesis as follows. Chapter 2 dives through the foundations of human mobility, prior work, WiFi dataset details used throughout the thesis, and ethical considerations of data collection. Chapter 3 presents a tool to extract device trajectories from WiFi traces and characterization of device and user mobility. In chapter 4 I present a different take on user and multi-device mobility modeling. I argue that mobility modeling should be hierarchical in nature and the mobility of multiple devices belonging to a single user should be modeled as a collection. In chapter 5 and 6, I discuss 2 mobility-aware applications from 2 different perspectives. In chapter 5 I propose WiFiTrace, a network-centric approach for contact tracing that relies on passive WiFi sensing with no client-side involvement and look at a novel graph based algorithm. Whereas in chapter 6 I present iSchedule, a machine learning-driven technique to automatically learn custom occupancy-based HVAC schedules for buildings across a large campus. Finally, in Chapter 7 I wrap up my thesis with conclusions.

CHAPTER 2

RELATED WORK

This chapter presents background on mobility, the dataset used throughout the thesis, and a discussion on ethical considerations of user data collection and usage.

2.1 Human Mobility

Human mobile behavior is assumed to be *nomadic* in nature, where nomadicity involves traveling to a location, staying at that location for a period of time, followed by travel to a new location, and so on [67]. The process of moving between two successive locations is referred to as a *transition*, while the stationary behavior at a location is denoted as a *stationary* period. The path of a mobile user over time is referred to as their *trajectory*.

2.1.1 Understanding Mobility

Early work in mobile computing focused on characterizing such nomadic behavior as *random walks*, where the choice of the next location was random [95, 23, 15, 95]. However subsequent research showed that human activities follow daily and weekly routines, and there are significant spatial and temporal correlations as well as recurring patterns in the locations visited by mobile users [46, 14, 60, 123, 42]. A variety of data sources have been used to understand mobility, ranging from GPS, cellular, WiFi logs of mobile devices, social media check-in data [20, 13, 100, 123, 48, 25], as well as transportation data such as taxi logs [42, 100].

Challenges

While much prior work focuses on outdoor mobility, there have been a few campus-scale mobility characterization studies, at university campuses such as Dartmouth [69, 50], Singapore Management University (SMU) [55, 56] and Tsinghua [125] and in corporate campuses [18]. However, none of these studies have examined mobility of multi-device users or that at different spatial scales. These studies do not analyze mobility at micro- and macro-scales (nor are we aware of efforts to characterize outdoor urban-scale mobility at multiple spatial scales). Further, prior studies have not focused on users with multiple mobile devices—prior work from the early 2000s was conducted in the pre-smartphone era and implicitly assumed a single device per user environment, with laptops being the primary user device[69, 18]. More recent studies [55, 56, 125] did not focus on this specific research question, and have instead focused on other issues such as crowd activities [125], group behavior [55, 56] or networking aspects [18].

Our work in mobility characterization presented in Chapter 3 differs in a few keys ways, most notably our focus is on outdoor and indoor mobility characterization and analysis of multi-device mobility. Our study reveals significant differences between indoor mobility and prior results on outdoor mobility and brings insights indicating that system designers need to carefully consider our findings when designing systems for indoor use.

2.1.2 Mobility Modeling

There has been significant work on using Markov Models or Hidden Markov Models (HMMs) to capture the sequential nature of human mobility. However, capturing long-term dependence in the data or recurring patterns is challenging when using Markov Models; doing so requires the use of higher-order Markov models, which quickly grow in complexity and computational overheads.

Most of the mobility modeling work focusses on outdoor mobility modeling at urban-scales [52, 75, 71, 63] , next location prediction [31, 76, 74, 41, 45, 79], and point of interest areas [117] using a variety of data sources such as cellular, WiFi, social media check-ins, and vehicular data [42, 48, 107, 49, 61, 17]. All these outdoor models cater to a discrete mobility models where mobility is infrequent compared to fine grain indoor mobility hence these outdoor models cannot be directly applied to indoor environments.

More recent work in this area has focused on urban mobility modeling using cellular, transportation or social media data using data driven methods, specifically deep learning. Recurrent Neural Networks (RNNs) have emerged as a popular approach for urban mobility modeling [75, 37, 98, 121, 59, 121] taking inspiration from Natural Language Processing (NLP) to learn long term dependencies. SpatioTemporal-Recurring Neural Network (ST-RNN) [9] models spatial and temporal contexts of mobility but is too complicated with the need to tune a lot of parameters and cannot be easily deployed for indoor mobility which is very frequent. In DeepMove [37] the authors propose using RNN to model sparse trajectories. It does not cater to either indoor mobility or capturing the mutli-scale hierarchical mobility correlations.

Other efforts in indoor mobility modeling comprise of [57] but this approach is for modeling mobility based on groups and social friendship ties. Additionally they use fingerprinting to acquire the dataset and it requires human effort and cost in acquisition. WiFiMod uses passively sensed WiFi syslogs and doesn't need any new infrastructure or human feedback for data collection. There has been work on using WiFi probes for sensing where the probe requests from mobile devices that steadily scan the APs close by for access are used for monitoring the crowds or activity flow for monitoring users [113, 109]. These prior works do not focus on individual mobility and instead look at crowd and activity behavior of aggregated users. In our work we focus on individual user mobility trajectories using WiFi syslogs and not WiFi

probes. Additionally, our model supports all applications that need individual as well as aggregated human mobility unlike only aggregated behavior as analyzed by prior work.

Note: All our work is coded in python and heavily uses python pandas [89].

2.2 Dataset and Methodology

2.2.1 Campus WiFi Dataset

Our campus-scale indoor mobility study is based on WiFi data from a large university campus (University of Massachusetts, Amherst). The campus comprises 156 buildings spread over 1460 acres and is a residential campus where most undergraduate students live on campus. Wireless connectivity is available in all campus buildings and also in many outdoor spaces. The campus WiFi network consists of 5104 HP Aruba access points (APs) that are managed by seven wireless controllers. The controllers receive syslog messages of all events seen by the APs; these logs contain many types of events, of which six events types are relevant to our study: (i) association, (ii) disassociation, (iii) re-association, (iv) user authentication, (v) deauthentication, and (vi) drift events. Since the campus WiFi network uses enterprise RADIUS authentication, all user devices must first authenticate themselves before they connect to the network. Doing so generates authentication and deauthentication log messages, which allows the network to associate each device with a particular user. Once authenticated, the device can then associate with a nearby access point, which generates an association message in the event logs. If the device moves out of range or wakes up from sleep, it may generate disassociation, reassociation, or drift messages.

Finally, for Cisco networks, we log WiFi data directly from the network using the Cisco Connected Devices (CMX) Location API v3 [99]. All of these preprocessor scripts convert raw logs into the following standard record format:

```
Timestamp, AP Name or Id, Device MAC Id, event type, (optional) User Name
```

Item Description	Value
Duration	Fall 2018 (Sep-Dec)
Num. events in log	9.6 billion
Num. of Buildings	156
Num. of APs	5104
Num. of devices	70,040
Num. of Student users	24,791
Num. of Faculty-staff users	5293

Table 2.1: Dataset Description

By default, we assume anonymized (or hashed) device MACs and usernames. We also assume a separate secure file containing a mapping of real names to hashes. While the association, disassociation, reassociation, and drift messages from the syslog give us spatio-temporal details about the various devices on the network, the authorization events provide us with details about MAC ID and username, aiding us to create a device-user mapping. The device-user mapping helps us count each user once by considering only the highly mobile user device among multiple user-owned devices. The identified highly mobile device is most commonly a smartphone because it gets carried around by users everywhere.

WiFi Logs: Each event in the log consists of a timestamp, the event type, the MAC address of the device, and the Access Point ID. In addition, authentication and deauthentication events also include the user name and user type (which can be one of student, faculty-staff, or guest). For privacy reasons, all device MAC addresses and user names are anonymized using a SHA-1 hash function. Since the location of all access points are known (in terms of the building and floor where they are deployed), each of six event types represents a *"presence"* event, since it indicates the presence of that device in the vicinity of the access point (and its corresponding location). The sequence of presence events generated by a device over the course of the day then reveals all the AP (and building-specific) locations visited by that device and the time spent at each location. Further, since each device must first authenticate to

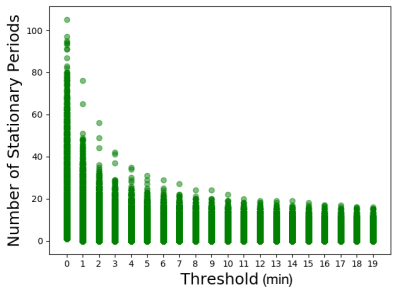


Figure 2.1: Impact of different thresholds for determining stationary periods.

the network using its owner's user ID, the owner of each device is known, which in turn reveals the collection of devices owned by each user.

This data collection has been ongoing since 2013 for various longitudinal studies. Unless specified otherwise, for computational tractability, our analysis focuses on a single university semester, namely Fall 2018, which spans from September to December 2018 (see Table 4.1). The event log for this one semester is over 260GB in size and contains 9.6 billion events spanning 70,040 devices, 24,791 student users and 5293 faculty-staff users.

2.2.2 Trajectory Extraction Over Noisy Data

We now describe our methodology for extracting the mobility trajectory exhibited by each device using noisy syslog traces and validation of the generated trajectories. From the overall trace, we first extract a trace of events generated by that device using its hashed MAC address enabling us to reconstruct the trajectory of the device over each day. Specifically, the timestamped presence messages reveal the sequence of locations visited by the device during each day and the time spent at each location. For students who stay on campus in residence halls, we can reconstruct a trajectory for the entire day, while for faculty, staff and student who live off-campus, we can reconstruct device trajectories for the portion of each day spent on campus. Each device trace reveals a trajectory comprising a sequence of stationary and transition periods. A *stationary period* indicates that the device is stationary at a particular

location (by virtue of being associated with the same AP for a period of time). A *transition period* indicates that the device is "on the move," which is seen as a sequence of association events with different APs, each of a short time duration.

For the purpose of this study, we use a 10 minute threshold to distinguish between a stationary and transition state for a device—if a device is associated with an AP for a duration greater than 10 minutes, we label the current location as a stationary period, otherwise the location is assumed to be part of a transition period in the overall device trajectory. The 10 minute threshold was chosen after a careful analysis of the data. Figure 2.1 depicts the number of stationary periods (i.e., locations) visited by a device obtained for different thresholds. A smaller threshold implies that even short stays at an access point will get labeled as stationary periods. The figure depicts that the curve flattens at 10 minutes and stays flat beyond this threshold value; such a 10 minute threshold, also employed by others [63], aligns with human notions of visiting a location versus transiting through one.

Handling Noise: The trajectories extracted from raw traces will be inherently noisy. For example, mobile devices may connect to access points that not the most proximal, or "ping-pong" between nearby access points even though the user is stationary. Similarly, when the user is walking to a new location, devices may connect to distant APs in weakly connected regions or exhibit similar ping-pong switching effects. Since we are using the AP location to determine the device location, all of these effects introduce noise or spurious location changes into our extracted trajectories. To address these issues, the raw traces are subjected to a multiple filtering and smoothing steps during trajectory extraction to remove noise and obtain clean trajectories.

Validating Trajectories: We conducted a small-scale study to validate that device trajectories derived from the WiFi dataset corresponds to the ground-truth device trajectory. To do so, we had volunteers mimic user behavior by walking with a phone to various campus buildings, spend some time inside each building, and then walk

to next building and so on. The ground truth trajectory (i.e., locations and times) recorded by the user were compared to that extracted from the WiFi log of the phone. Our validation study revealed more than 99% accuracy between the extracted locations and the ground truth for indoor locations and deviations of no more than 20-40 meters in outdoor locations when walking outside, providing confidence that our dataset enables us to study campus-scale mobility behavior.

2.3 Data Ethics and Privacy Considerations

2.3.1 Privacy and Ethical Considerations

Collection of user data raises important privacy concerns, especially for everyday users whose whereabouts would essentially be documented in logs. Mobility aware applications bear the same concern, especially since users are passively tracked when connected to the WiFi network. To reduce the privacy risk, several safeguards have been put into practice and a few suggested as below.

1. **No direct access to the user or device data:** In all my work, all WiFi network data such as MAC ID and username that can identify a user is hashed (e.g., using the SHA-2 hash) to maintain anonymity. The hashing key is known only to one person in the IT department. Hashing of MAC ID and usernames is performed before they are shared with the researchers.

2. **Leveraging existing operational security standards:** WiFi network data is used by many IT departments for network maintenance and security surveillance. For example, our campus uses the same WiFi data used throughout all projects in this thesis to track down compromised devices that may be responsible for internal DDOS attacks and identify student hackers who, most notably, might be attempting to change course grades. Additionally, in many regions, audit and compliance laws also necessitate gathering network logs for subsequent analysis and audits. These routine evaluations have operational security

standards in place to protect user privacy. Using this WiFi data for contact tracing requires compliance with the same high operational security standards.

3. **Obtain user consent:** Data protection acts in many countries require organizations to acquire user consent before starting any data collection operations. So, before any applications described in this thesis can be used in production, users must provide informed consent to contact tracing upon connection to WiFi in an enterprise network. Similarly, case investigators must also be authorized to retrieve de-anonymized information when necessary, and all data sharing must follow the approved guidelines. For example, a contact tracing team could decide to directly contact an individual at high risk of contracting Covid-19 or publish at-risk locations like a public alert to appeal to potentially infected individuals to contact health authorities. In the latter case, the proximity data report is used for further contact tracing when at-risk individuals contact health officials. We currently use this latter approach at our USA campus.

2.3.2 IRB Approval

Data collection for experimentally validating the efficacy of our approach across all projects has been approved by our Institutional Review Board (IRB). It is conducted under a Data Usage Agreement (DUA) with the campus network IT group that restricts and safeguards all WiFi data. To avoid private data leakage, all the MAC ids and usernames in the syslogs are anonymized using a strong hashing algorithm. The hashing is performed before syslog data is stored on disk under the campus IT manager's guidance, who is the only person aware of the hash key of the algorithm. Any data analysis that results in the users' de-anonymization is strictly prohibited by our IRB agreement and the signed DUA.

CHAPTER 3

EMPIRICAL ANALYSIS AND CHARACTERIZATION

Understanding the mobility of users and their devices has become ever more important in the era of the mobile Internet—mobile behavior has broad implications on the design of mobile services, wireless networks, edge computing, and urban infrastructure. Over the past decade, there has been extensive work on understanding human mobility at urban scales [122, 114, 52, 32] and on modeling such mobility [75, 112, 42, 71, 31, 63, 90, 74] by using a variety of sources such as cellular, WiFi, social media check-ins, and vehicular data [48, 107, 49, 61]. This body of work has focused on characterizing and modeling *outdoor* mobility at large spatial scales, such as cities and campuses [87, 63], as well as different temporal scales, by using a variety of data sources such as cellular, WiFi, social media check-ins, and vehicular data [48, 107, 49, 61]. This body of work has largely assumed mobile devices to be independent, an assumption that no longer holds in an era of mobile Internet users who own a multitude of devices that exhibit correlated mobility patterns. Further, prior work has analyzed or modeled mobility patterns at a single spatial scale–often that of the underlying dataset—and has not considered the impact of mobility at different spatial scales on system design. In this chapter, we challenge the above stated assumptions with insights gained from empirical analysis and discuss the implications of the insights on system design.

3.1 Motivation

Studies have shown that humans spend over 80% of their lives indoors and inside buildings [68]. Consequently, understanding *indoor* mobility is equally important for purposes of modeling and system design, but this area has seen much less work than *outdoor* mobility of users [122, 114, 52, 32]. Recent research has recognized that indoor mobility of users inside buildings, where many users spend a significant portion of the day, is very different from outdoor mobility exhibited when walking in a city or traveling in vehicles [124, 125]. While a few recent efforts have specifically focused on indoor mobility [69, 125], many research questions remain unanswered.

In this chapter, I focus on empirically characterizing the indoor mobility of modern mobile Internet users with a view to address three questions: (1) How do indoor mobility patterns differ from those observed in outdoor settings? (2) Since modern users own multiple mobile devices, how correlated or different are the indoor mobility patterns exhibited by different devices belonging to the same user? (3) What are the implications of these findings on mobility modeling research?

I argue that to fully understand indoor mobility, mobility patterns should be analyzed at multiple spatial scales. For example, how users and devices move from one building to another over the course of a day, and what mobility do they exhibit inside a building? That is, both *macro-scale* inter-building mobility as well as *micro-scale* intra-building mobility should be analyzed to understand the richness of indoor mobility patterns. We also argue that since today's mobile users own a multitude of mobile devices and use various devices differently, it is important to characterize how these usage patterns translate to similarities and differences in their exhibited mobile trajectories. That is, it is important to analyze the collection of each user's devices rather than doing so independently.

I address these research questions by conducting a large-scale characterization study using the campus WiFi dataset.

3.2 Background

In this section, we present background on indoor and outdoor mobility.

Outdoor versus Indoor Mobility: Much of the above work has focused on *outdoor mobility* to understand how humans move from one location to another at the spatial scale of a city or community (i.e., at urban-scales) [52, 87, 117, 98, 120]. As noted earlier, humans spend over 80% of their lifetime indoors [68], and indoor mobility is known to be different from outdoor mobility patterns [124, 125]. Specifically, indoor mobility is concerned with how users and their devices exhibit nomadic behavior within and across buildings—that is, what locations (buildings, rooms) users visit, how long they stay at each location, and the transition path between locations; since indoor movements are based on walking, we are not concerned with the velocity of transitions—unlike outdoor mobility in vehicles, for instance.

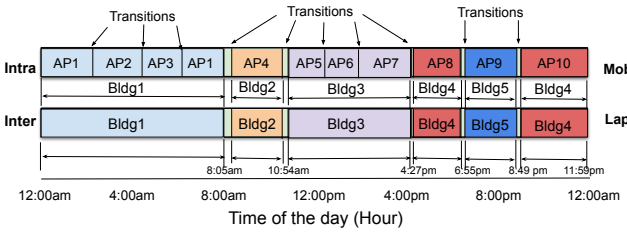


Figure 3.1: Intra and Inter-building Mobility

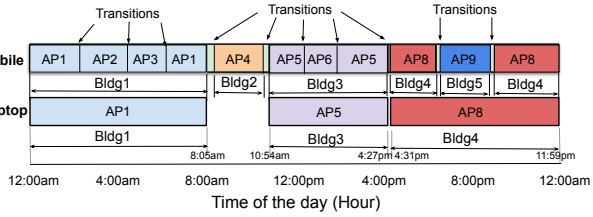


Figure 3.2: Different devices of a user exhibit dissimilar trajectories.

Building-level Indoor Mobility: Our work characterizes indoor building mobility at two different spatial spaces: inter- and intra-building scales. Inter-building mobility is concerned with macro-scale mobility from one building to another; in this case, the *entire* building is assumed to be single spatial location and we characterize nomadic behavior in terms of time spent in a building, transition time to the next building, and so on. Intra-building mobility is concerned with micro-scale mobility *inside* a building–i.e., how the time spent in a single building can be further broken down into mobility across rooms or locations within that building; in this case, rooms, specific areas within the building or even access points are assumed to specific locations and

mobility is viewed at the finer spatial scale across these locations. We argue that a multi-spatial scale approach captures both macro and micro mobility patterns, and in doing so yields a better understanding of indoor mobility behavior. This is illustrated in Figure 3.1 which depicts the inter-building and intra-building trajectories of an actual user; at inter-building scale, the trajectory reveals the sequence of buildings visited over a day and the times spent in each, while at intra-building scale, the trajectory reveals what locations were visited by the user when visiting each of those buildings.

User versus Device Mobility for Modern Users: Modern Internet users carry multiple devices. A common use case is to own a smartphone and a laptop, but many users will own more than two devices that include tablets, e-readers, and wearable smartwatches, among others. In such multi-device environments, it is important to distinguish between *the mobility of a user from that of their devices.* Prior studies that have implicitly assumed device mobility to be a proxy for user mobility, since the former is used to characterize the latter. In our case, each user owns multiple mobile devices, and perhaps more importantly, uses these devices differently, causing them to exhibit different mobility patterns. For example, a user may carry their phone everywhere they go while at work, but they may not take their laptop to activities such as lunch that do not require the laptop. This will cause the mobile behavior of the laptop to deviate from that of the phone even though both are owned by the same user. Thus, we distinguish between user mobility and device mobility and assume that each mobile device exhibits a distinct mobility pattern, which approximates to varying degrees the true mobility of their owner. Further, the user's device that exhibits the greatest mobility—often the one that the user takes with them "everywhere"—yields the best approximation of the user's mobility. Figure 3.2 illustrates these differences by depicting mobile and laptop trajectory for an actual user. As shown, both devices visit the same location whenever the user brings both

devices to that location; the figure shows that the user often leaves the laptop at a location but takes the phone with them when visiting other locations inside a building or even other buildings, causing trajectory deviations. The trajectories converge again when the user returns to the previous location.

Mobility in Campus Environments: Campus settings are well-suited for studying indoor mobility for several reasons. Users in university or corporate campuses spend a significant portion of the work day working or studying in such settings. Such users tend to be tech-savvy and own a multitude of mobile devices, and campus environments tend to have ubiquitous WiFi network coverage, which yields data for mobility studies. Finally, since a campus comprises multiple buildings, it enables both intra-building and inter-building mobility of users to be analyzed over the course of the day.

Study	Data Source	Spatial Scale	Multi-device users	Use case
Tsinghua campus [125]	WiFi, SNMP, Apps	Indoor flat	No	Student behavior
SMU campus [55, 56]	WiFi, Apps	Indoor flat	No	Group/user behavior
Dartmouth campus [69]	Network WLAN	Indoor flat	No	Network optimization
Corporate campus [18]	Network WLAN	Indoor flat	No	WLAN characterization
Our study	WiFi syslog	Inter- & Intra-building (Multiple Floors)	Yes	Modeling

Table 3.1: Comparison with Prior Indoor Mobility Studies

Relation to prior work on indoor mobility: While much prior work focuses on outdoor mobility, there have been a few campus-scale mobility characterization studies, at university campuses such as Dartmouth [69], SMU [55, 56] and Tsinghua [125] and in corporate campuses [18]. However, none of these studies have examined indoor mobility at micro- and macro-scales (nor are we aware of efforts to characterize outdoor urban-scale mobility at multiple spatial scales). Further, prior studies have not focused on users with multiple mobile devices—prior work from the early 2000s were conducted in the pre-smartphone era and implicitly assumed a single device per

user environment, with laptops being the primary user device[69, 18], while more recent studies [55, 56, 125] did not focus on this specific research question.

Table 3.1 summarizes the differences between our work and prior indoor mobility efforts. While these efforts have analyzed various aspects of building-level mobility, their primary emphasis has been on other mobility issues. For example, [125] emphasizes crowd activities in a campus and the impact of spatial context on physical activities at a university scale. The SMU study [55, 56] focuses on behavioral aspects of users or groups of users, such as understanding the difference in mobility of single users versus groups of users as well as non-conformance predictor in indoor user mobility. The Dartmouth Study [69, 50], which is the closest to ours, analyzed WiFi based mobility trace of the campus devices but did not consider multiple devices or multiple spatial scales. Finally, [18] studied mobility patterns in corporate settings, extracted common WLAN usage characteristics and introduced prevalence and persistence metrics to model user mobility.

3.3 Device Classification

Our WiFi logs allow us to associate a device to a user and determine all devices belonging to a user, but they do not include any information to determine the *type* of each device—anonymized MAC addresses alone do not reveal device type.[1] Consequently, we develop a simple classification technique that uses the network behavior exhibited by each device to infer its device type.

First, we observe that differences in OS power management results in different network behavior during idle periods for different device types. Devices such as phones, and many tablets, tend to be powered on at all times—even when the user is not actively using the device. When these devices become idle, their network

[1]For privacy reasons, even partial MAC address prefixes, which reveal vendor and device type, are unavailable to us.

interfaces enter a low-power listen state but stay powered up (e.g., to receive push notifications, chat messages, or video calls). Hence the device continues to maintain a network presence and is periodically visible to the WiFi network. Consequently, when a user walks from one location to another, the access points along the path periodically see the presence of the device (through scans, association or disassociation messages). Of course, if the user actively uses their device when walking, the device maintains a continuous, rather than periodic, network presence along the path. Either way, the trajectory of such always-on devices is visible to the network during a location transition.

In contrast, mobile devices such as laptops tend to hibernate when not in active use (e.g., when the laptop lid is closed). The hibernate power state results in network interfaces being powered down, and the device no longer maintains a network presence while hibernating. Consequently, when a user walks with a laptop from one location to another, the device is not visible to the network during the walk and only becomes visible when a user begins using the device at the new location.

This difference in network behavior during transition periods and the resulting network visibility of the device (or lack thereof) enables us to distinguish between, and classify, *always-on* and *hibernating* devices. The most common always-on device in our current environment is a phone[2] while the most common hibernating device is a laptop.

Validation: Since the above classification method is a heuristic, we conducted a small-scale study to validate its accuracy. We had a volunteer mimic actual user behavior by visiting various buildings and walking within and across buildings over a period of 3 weeks and 8 hours each day; the user used two iOS devices (phone and tablet), one android phone, and 3 laptops with MacOS, Windows and Linux.

[2]Smart watches, which are another type of always-on device, were not present in our dataset since they do not support RADIUS authentication for enterprise WiFi networks.

Device Characteristics	% Devices
Hibernating	26.91%
Always-on	73.08%

Table 3.2: Distribution of always-on and hibernating devices.

Device Type	% Users
Primary	89.54%
Secondary	94.52%

Table 3.3: Ownership of primary and secondary devices.

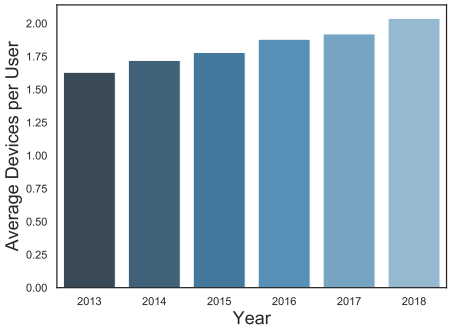


Figure 3.3: Device ownership has grown steadily since 2013.

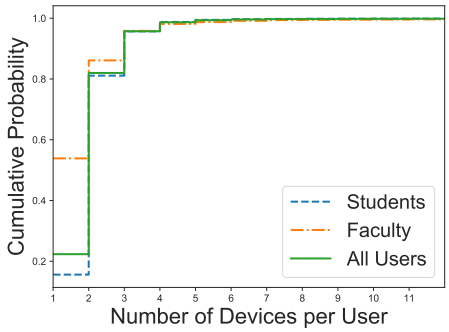


Figure 3.4: CDF of the number of devices owned per user.

In each case, we used the daily trace from the device to classify its type using the above heuristic. We found that all devices get classified with perfect accuracy as an always-on or hibernating device when using a day long trace. The only way a laptop can get mis-classified as an always-on device is to walk with the laptop's lid open so that it shows network presence during location transitions. However, it is highly unlikely that users will *always* use a laptop in this manner over multiple days, even if they occasionally walk to a location with an open laptop. Further, our analysis uses day-long traces over multiple randomly-chosen days from the 4 month dataset to classify each device, which significantly reduces the changes of mis-classification. Consequently, our heuristic is able to classify devices as always-on or hibernating with high accuracy.

Result: We classified all devices in our WiFi log using the above method, and Table 3.2 depicts our results. As shown, 73.08% of all devices maintain a network presence during location transitions and are classified as always-on devices. The

remaining 26.91% devices do not exhibit any presence during location transitions and are classified as hibernating devices.

For the purpose of our study, we further classify all devices belonging to each user as primary and secondary. A user's *primary* device is defined as the always-on device that exhibits the greatest mobility (greatest number of stationary periods per day) across all always-on devices owned by that user. All other devices belonging to that user, whether always-on or hibernating, are defined as *secondary* devices. By virtue of being the user's most mobile device, the primary device also provides the best approximation of the user's actual mobile behavior. With a high likelihood, a user's primary device is likely to be a mobile phone. Further, with high likelihood, a hibernating secondary device belonging to the user is likely to a laptop. After labeling all devices as always-on and hibernating and then labeling each user's device as primary and secondary, we see in Table 3.3 that 89.54% of users own a primary device. This also implies that 10.45% of our users either do not own a smartphone or do not connect their phone to the campus WiFi network. The table shows that 94.52% of our users own at least one secondary device. Since multi-device ownership is common on our campus, there is a substantial overlap between these two user groups, as discussed next.

Key Takeaway

- *Network visibility and network behavior of a mobile device are strong indicators of the device type.*

3.4 Multi-device Users

In this section, we analyze the mobility of different devices owned by a user and the mobility across device types.

3.4.1 Multi-device Ownership

To analyze device ownership, we first consider our entire longitudinal dataset spanning 2013 to 2018. Figure 3.3 shows the mean number of devices per user over this five year period. As shown, device ownership has grown steadily in recent years—the mean number of mobile devices per user grew from 1.63 in 2013 to 2.04 in 2018. Next, we focus on the Fall 2018 semester used for this study and analyze multi-device ownership across broad classes of users. Figure 3.4 plots the probability distribution of device ownership across students, faculty-staff, and the combination of the two. The CDF shows that the majority of the users—84.4% of the students and 46.1% of faculty-staff—own two or more devices. The average student owns 2.1 devices, while the average faculty-staff user owns 1.7 devices, indicating that students own more devices, on average than other user types. This is not surprising since younger users tend to be more tech-savvy, and furthermore, a large majority of students ($> 60\%$) stay on-campus and connect all of their devices to the networks (while faculty and staff who live off-campus may not bring all of their devices to work). Interestingly, the figure also shows that 18% of users own three or more devices, with 1.33% users owning five or more devices.

Key Takeaways

- Device ownership has increased steadily over time, and the typical user now owns 2.04 devices.
- 18% of the users own three devices or more.

3.4.2 Characterizing Device Mobility

To characterize the mobility of different types of devices owned by a user, we considered the primary device (i.e., the phone) and the hibernating secondary device (i.e., the laptop) for each user, which is also the common case for device ownership on our campus.

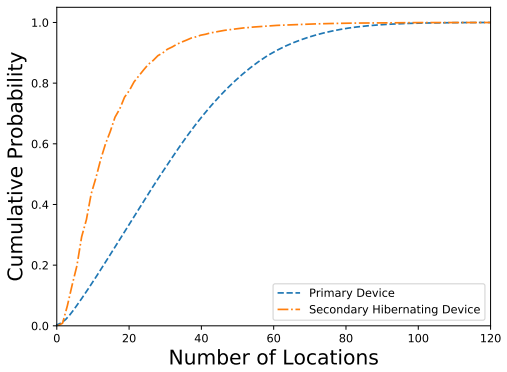


Figure 3.5: CDF of number of locations visited by device type.

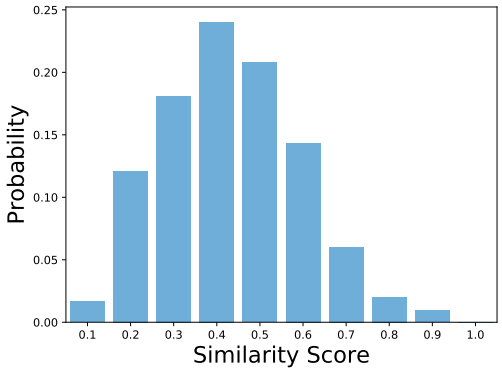


Figure 3.6: PDF of similarity scores of primary and secondary device trajectories.

Figure 3.5 depicts the CDF of intra-building locations visited per day for the two device types. The CDF reveals that phones visit 35.9 intra-building locations, on average, per day, while laptops visit 10.2 locations, which yields a 3.5X more mobility for phones than laptops. More generally, since devices as laptops are less portable than phones, users will carry them to fewer locations, causing them to exhibit lower mobility—the "less portable" the device, the lower its mobility. In the future, as wearable devices such as smartwatches become common, we expect them to be even more mobile than today's phones.

Recall also from Figure 3.2 that while the phone trajectory will often deviate from the laptop and the two will converge again when the user returns to the laptop's location. Thus, despite having lower mobility, the laptop's trajectory is correlated to the phone's trajectory since both depend on the user's mobility behavior. To understand the degree of similarity between the two, we computed the pairwise similarities in the stationary location trajectories for each user's phone and laptop using Longest Common Subsequence (LCSS) score. We choose LCSS as a measure of similarity since it is robust to noise and can handle synchronous or random shifts of the location sequence [108]. Thus, small variations in trajectories do not have a large impact on the similarity measure. We compute the similarity score as a ratio of the length of LCSS

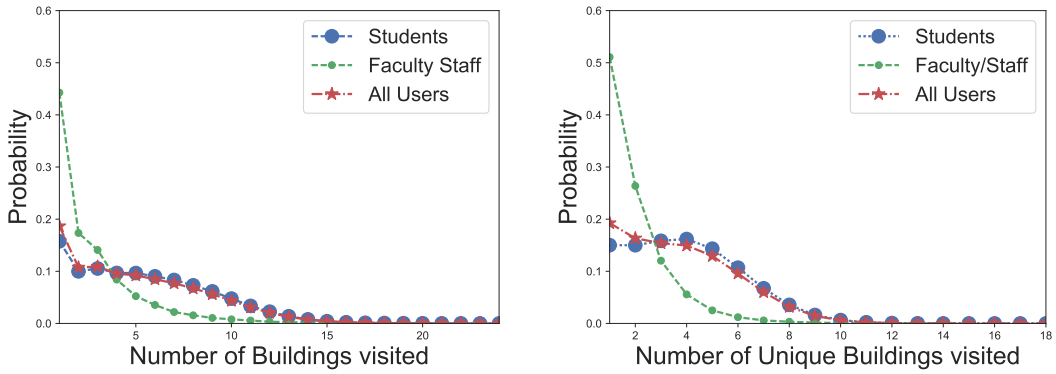
to the length of the union of the primary and secondary trajectories; the higher the score the higher is the device trajectory co-relation and vice-versa. Figure 3.6 depicts the PDF of the similarity scores obtained for all users. The similarity scores range from 0.06-0.86, with a mean of 0.37. The plot shows that 31.9% device pairs have a weak similarity score of less than 0.3, 59.1% device pairs have a moderate similarity score between 0.3 and 0.6 and 9% device pairs have a high similarity score of 0.6 or more. Thus, more than two-thirds of the users use their phones and laptops such that the two device trajectories show moderate to strong correlations, and this behavior is true despite phones having 3.5X higher mobility than laptops. The maximum similarity score is 0.86 which indicates that *even the most correlated pair of devices nevertheless see some dissimilarities in their trajectories.* Our analysis shows that a user's mobile devices should not be viewed as independent due to their moderate to strongly correlated mobility patterns. Further, these correlations vary significantly across users, which should also be considered in system design and modeling.

Key Takeaways

- More portable devices such as phones exhibit 3.5X more mobility in terms of location visits than laptops.
- Primary and secondary device trajectories for over two-thirds of the users show moderate to strong correlations.

3.5 Macro and Micro-scale Mobility

In this section, we analyze mobility at macro (inter-building) and micro (intra-building) spatial scales. Unless specified otherwise, all results in this section are based on the users' primary device.



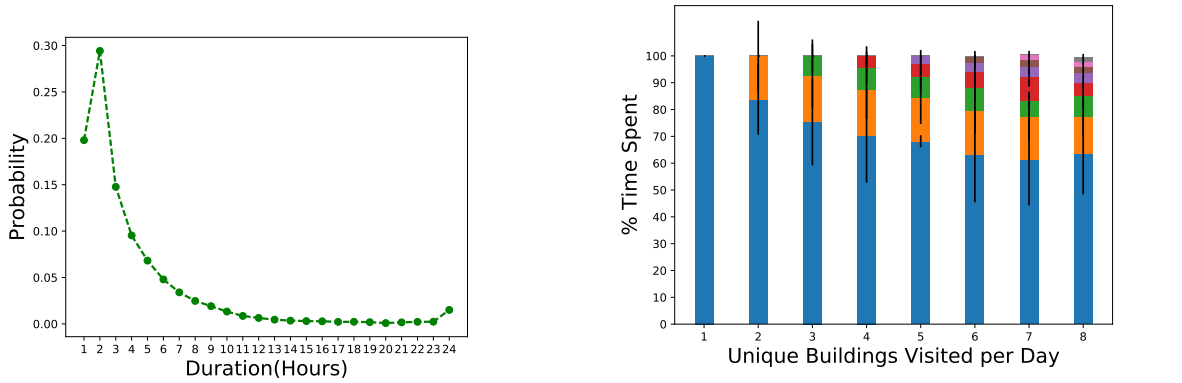
(a) Buildings visited (b) Unique building visited

Figure 3.7: Distribution of the number of buildings visited per day by campus users.

3.5.1 Macro-scale Inter-building Mobility

To analyze mobility at the macro scale of entire buildings, consider the trajectory of each user's primary device, which is a sequence of AP locations visited by that user. Since building- and floor-specific locations of each AP are known, we can assign a building label to each visited AP, and then aggregate a consecutive sequence of APs with the same building label as a *single* location, representing a visit to that building with a corresponding aggregated visit duration. The transformed inter-building trajectories then yield a sequence of buildings visited by each user, time spent in each building, and transitions across buildings. At this macro scale, user trajectories are only concerned with visits to buildings and transitions between buildings, but not what happens inside a building.

Stationary period analysis: Figure 3.7(a) plots the distribution of the number of buildings visited by a user over a day. The distribution reveals that the average user visits 4.1 buildings per day; highly mobile users, depicted by the 90-th percentile of the distribution, visit 9.8 buildings per day. The distribution also shows that students are more mobile than faculty and staff, with students visiting 4.4 buildings per day, on average, and faculty and staff visiting 1.2 buildings, on average. Figure 3.7(b)



(a) PDF of time spent in a building (b) Distribution of time spent across buildings

Figure 3.8: Distribution of time spent by users in campus buildings

plots the distribution of the *unique* number of buildings visited by users each day (where multiple visits to a building count as a single unique location). The figure shows that a user visits 2.7 unique buildings, on average, each day, which implies users often return to their primary office building after a visit to another building or visit the same building (e.g., dining hall) multiple times per day.

Figure 3.8(a) shows the PDF of the time spent in each building by a user. The PDF shows that a campus user spends 109 minutes, on average, visiting a campus building. Further analysis of this distribution reveals that about 30% of building visits last less than 1.5 hours; 29% of all building visits are long visits, lasting an average of 5.8 hours, indicating that the tail of the distribution has a substantial mass. Figure 3.8(b) plots the total time spent in each unique building visited versus the number of buildings visited by users each day. The figure shows that both less mobile as well as highly mobile users spend between 60 to 80% of their day in a single building, with the remainder of the day spent visiting other buildings for shorter periods. This result shows that most users spend a majority of their day in a single "home" building (e.g., office or residence hall)

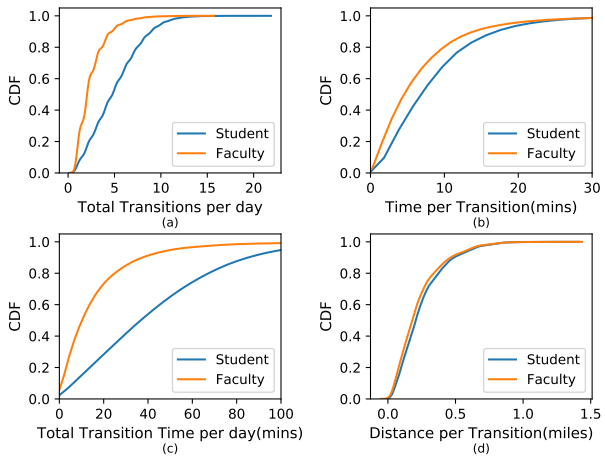
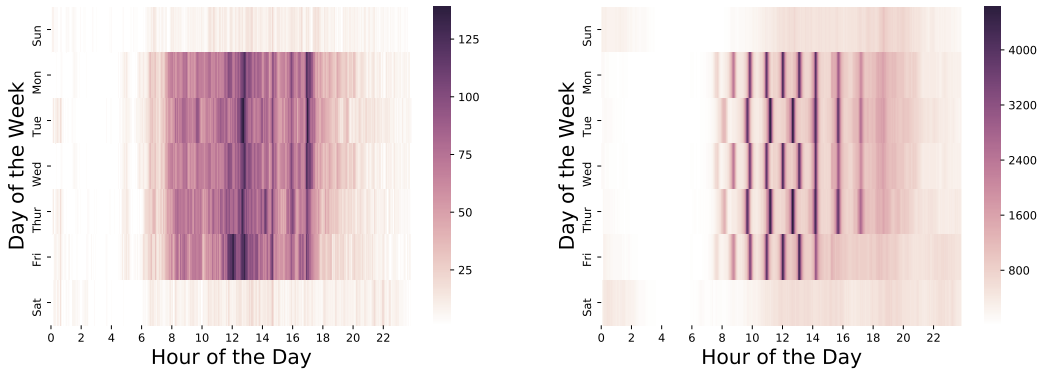


Figure 3.9: Inter-building mobility analysis: (a) CDF of transitions per day (b) CDF of duration of each transition (c) CDF of total transition time per day (d) CDF of mean distance traveled per day

Transition analysis: Next, we analyze temporal aspects of inter-building mobility by focusing on inter-building transitions. Figure 3.9 (a) depicts the CDF of the number of transitions per day made by campus users. Since each visit to a building must be preceded by a transition, the mean number of transitions is the same as (or, for off-campus users, one more than) the number of buildings visited, with a mean of 4.1 transitions per user per day. Figure 3.9(b) depicts the CDF of the duration of each inter-building transition. The figure shows that average transition time, which is usually a walk between two buildings, lasts 8.4 minutes and 6.5 minutes for students and faculty-staff users, respectively. Figure 3.9(c) shows the CDF of the total time spent in walking between buildings over the entire day. The figure shows that the average student spends 42.2 minutes per day walking between buildings, while the average faculty-staff user spends 16.1 minutes. Finally, Figure 3.9(d) depicts the CDF of the distance traveled when walking from buildings to another, and shows the average walk between campus building is 0.22miles.

Interestingly, we also find that about 15% of all inter-building transitions on campus are loops, with the same origin and destination. Such transitions last 15 minutes,

on average, and involve a walk of 0.6 miles. We believe that such transitions occur when users go for a walk during break, or walk to another building to for an errand and return to their previous building.



(a) Faculty-staff (b) Students

Figure 3.10: Heatmap of inter-building transitions. Student mobility is aligned with class and meal times, while faculty and staff mobility is more dispersed with a cluster around the lunch hour.

Figure 3.10(a) and (b) depicts the heatmap of *when* users move between buildings. Faculty and staff users make inter-building visits at all times of the day during working hours, as shown in Figure 3.10(a), with a significant density of transitions during the noon lunch hour. Transition times during evenings, nights, and weekends are more diffused for these users. In contrast, Figure 3.10(b) shows that student transitions during weekdays are highly aligned with lecture start and end times between 8 to 6pm, and are more dispersed during other hours. The weekend transitions do not show such patterns since there are no classes on weekends.

Key Takeaways

- Users visit 4.1 buildings per day and spends nearly 2 hours, on average, in a building.
- Highly-mobile users visit 2.4X more buildings than the typical user.
- A third of the building visits are short, lasting 90 min or less, while a third of the visits last 5.8 hours, indicating the tail of the distribution has a significant mass.
- Even in a large campus, users tend to show a primary affinity to a single building, spending over 60% of their day in that building.
- A typical transition (walk) between buildings lasts 8.1 minutes; the timings of such transitions is strongly correlated with class and meal times during daytime hours.

3.5.2 Micro-scale Intra-building Mobility

Next, we examine intra-building mobility by characterizing micro-scale behavior of what users do while inside a building. We do so by analyzing trajectories of AP locations visited by the user's primary device inside each building.

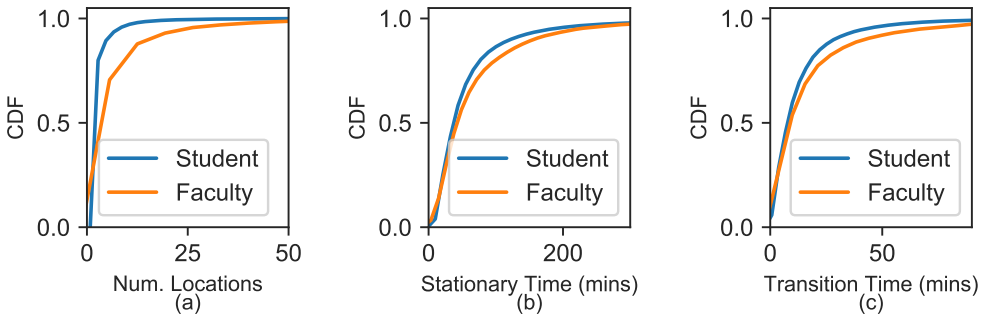


Figure 3.11: Intra-building mobility analysis: (a) CDF of number of locations visited, (b) CDF of time spent at each location, and (c) CDF of intra-building transition times.

Figure 3.11(a) shows the CDFs of locations visited (i.e., stationary periods) by a user inside a building. Interestingly, our results show that students visit 8.6 locations, on average, inside a building, while faculty and staff users visit 12.1 locations. In other words, at intra-building scale, faculty and staff exhibit higher mobility (by 1.4X) than students, which can be attributed to them spending more time inside each building due to the *lower* inter-building mobility. Since each stationary period is preceded by a transition, Figure 3.11(a) also represents the mean number of transitions inside a building (not counting the final transition when the user departs from the building). Figure 3.11(b) shows the CDF of time spent at each location inside a building. The figure shows that students and faculty spend 37 and 40 minutes, respectively, on average, when visiting a location inside a building, indicating the mobility is similar across user types. Figure 3.11(c) analyzes the duration of each intra-building transition. Such transitions result from users walking inside a building to see a colleague, go to a class or meeting, or to take a restroom break. The CDF shows that the average transition within a building takes 1.5 minutes for faculty-staff and 1.48 minutes for students, which is again similar across user types.

Importantly, over the course of a day, the typical user makes 35.9 intra-building transitions across all visited buildings. Thus, we see 8X more mobility at intra-building (micro) scale than at inter-building (macro) scale, implying that mobility decreases at higher spatial scales (37.8 intra-building vs 4.1 inter-buildings locations visited). Faculty-staff users make 14.8 intra-building transitions, while students make 37.9 transitions[3]; in doing so, they spend 22.3 and 56 minutes walking inside buildings, respectively. Highly mobile users representing the 90-th percentile of the distribution make 59.8 intra-building transitions across all visited buildings, and spend a total of 90.4 minutes walking inside buildings.

[3]Despite making fewer intra-building transition per visit, the higher number of buildings visited per day still yields an overall higher number of transitions for students.

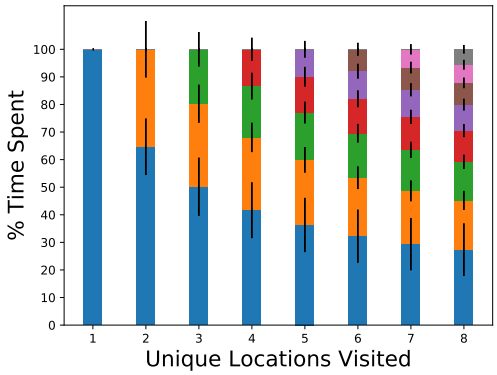


Figure 3.12: Distribution of times spent across unique locations inside a building

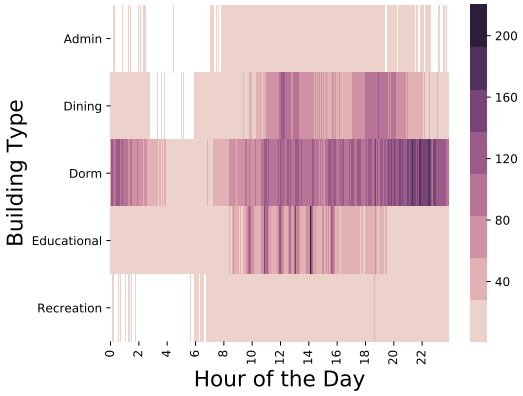


Figure 3.13: Heatmap showing transitions by building type

Figure 3.12 plots the distribution of time spent at unique locations visited inside a building versus the number of visited locations. Unlike the inter-building scale, where users spent 60 to 80% of the time inside a single building, at inter-building scale, we find that users spend only 30 to 60% of the time at their most visited location; that number rises to 60% or greater when we consider the top three most frequented location for each user.

Finally, Figure 3.13 shows a heat map of when intra-building transitions occur over the day. We find that mobility patterns inside a building are highly dependent on the type of the building—an academic building sees very different intra-building mobility patterns than a residence hall. Figure 3.13 shows four different buildings from our overall analysis: a dining hall, a residence hall, an academic building, and an administration building. The figure shows residence hall users making intra-building transitions at all times of the day, while the academic buildings see transitions correlated with lecture start and end times as well as arrival and departure times. The dining halls see a high concentration of transitions at meal times such as breakfast, lunch, and dinner, while the administration building sees transitions during AM arrivals and PM departures and uniform mobility in-between. More broadly, we find

that the type of indoor space governs the type of intra-building mobility patterns that will be seen in that space.

Key Takeaways

- The typical campus user visits 8.97 locations inside each building during a building visit. Over a day, a typical user visits 35.9 intra-building locations across all visited buildings.
- Users exhibit nearly 8x more mobility at intra-building scale than inter-building scale.
- While the amount of mobility decreases (by 8x) with increasing spatial scale, the time spent at each visited location increases (by a factor of 2) with increasing scale.
- Unlike inter-building mobility, users do not exhibit any affinity to a single intra-building location; over 60% of the time during a building visit is spent at the three or fewer indoor locations.
- The type of indoor space governs the intra-building mobility patterns that are seen in that space.

3.6 Implications of Our Results

We now discuss the broader implications of our results.

Campus-scale Mobility: Overall, we find that campus-scale mobility in building depends on *five* key factors:

Spatial Scale: Our results show that as the spatial scale becomes coarser, the amount of mobility in terms of locations visited decreases with a corresponding increase in time spent at each visited location. We found that intra-building mobility was 8x more frequent with shorter stays at each location than inter-building mobility.

Since mobility is more frequent at micro scales than macro scales, a judicious choice of the correct spatial scale is necessary when addressing system design problems.

Device type: Our results indicate that less portable devices have lower mobility, since phones were found to be 3.5x more mobile than laptops. This key finding implies that all mobile devices should not be treated as equal and optimizing systems based on device type (or size-based groupings) may yield a better overall design.

Multi-device ownership: Given the prevalence of multi-device ownership, treating devices as being independent of others is no longer a reasonable approach. Our results showed that device trajectories of various devices owned by a user exhibit moderate to strong correlations but also that the degree of correlations varies considerably from user to user. Thus jointly modeling group of devices owned by a user or exploiting mobility pattern of one device to predict those of others for that user may yield better results.

User behavior: Some users will naturally be more mobile than others, and this mobile behavior manifests differently at different scales. At a given scale, highly-mobile users visit several times more locations than the average user. Across spatial scales, users who visit more buildings per day are *less mobile*, on a per-visit basis, at the intra-building scale, since their higher inter-building mobility results in shorter stays and few intra-building location visits per building. These findings manifested themselves in our study as students being more mobile, as a group, at inter-building scales, and faculty-staff, as a group, visiting more locations per building visit at intra-building scale.

Building type: Our results show that the intra-building mobility patterns are heavily dependent on the type of the building; the functions served by a building determine how frequently and when indoor mobility will be seen. The same user will exhibit different mobile behavior in different types of buildings, which implies that

mobile behavior is not just a user characteristic but also depends on the building type.

Outdoor versus Indoor Mobility: Our study reveals important differences between outdoor and indoor mobility and also some similarities. First, similar to the findings in [125], we find that mobility in buildings is far more frequent than urban-scale outdoor mobility in terms of the number of locations visited. Of course, transition times and distance traveled will naturally be smaller inside buildings than in outdoor spaces. Thus, results from outdoor mobility should not be directly employed when designing systems that will be primarily deployed and used inside buildings or on campuses. Interestingly, outdoor mobility can be viewed as a natural progression in the hierarchy from inter-building mobility, and when viewed from this standpoint, it naturally follows that mobility (in term of number of locations visited) will be lower at outdoor scales than finer indoor building scales—in line with our hierarchical spatial scale findings. A hierarchical study that combines both outdoor and indoor mobility in an integrated fashion is left to future work.

3.7 Summary and Status

In this chapter, I presented empirical characterization of human and device mobility based on mobility derived passively from WiFi logs. First, the study reveals that indoor mobility decreases with increasing spatial scales—we find 8x more mobility at intra-building (micro) scale than inter-building (macro) scale. The opposite is true for time spent at each visited location—inter-building visits have 2x higher stay durations than intra-building ones. Second, we find that the type and size of mobile device has a significant impact on its mobility—the larger the device, the lower the observed mobility. For example, phones exhibit 3.7x more mobility in terms of locations visited per day than laptops by virtue of being smaller and more portable. Third, different devices owned by a user exhibit moderate to strong correlations in

their daily trajectories, and we also find that the degree of correlation can vary in significant ways based on the user. We also find that the building type—type of indoor space—has a significant impact on the observed mobility patterns seen inside that space. Finally, We find indoor mobility to be far more frequent, in terms of locations visited per day, than outdoor mobility results from prior work, confirming our hypothesis that the two types of mobility are very different.

CHAPTER 4

WiFiMod: INDOOR MOBILITY MODELING

Modeling human mobility has a wide range of applications from urban planning to simulations of disease spread. While outdoor mobility has been studied well, indoor mobility has not received much attention in spite of the fact that humans spend 80% of their time indoors. In this chapter, we propose WiFiMod, a transformer-based approach to model indoor human mobility at multiple spatial scales using Passive Sensing.

4.1 Motivation

Understanding human mobility is fundamental to location based services, urban transportation, and smart cities among many other applications and paramount to improve sustainability. Lately, with the advances in networking technologies and ubiquity of mobile phones, a large amount of mobility data is generated and collected in the form of GPS logs, cellular data, social media check-ins, and vehicular data giving rise to data-driven human mobility modeling [48, 107, 49, 61]. This prior work seeks to capture human mobility at urban scales [52] using transportation, social media, and phone data. While taxi or public transit data [42, 107] allow urban-scale mobility of users to be captured from a vehicular or transportation standpoint, social media check-in data [61] enables users' mobility to be tracked at various points of interest [49]. GPS and cellular data from phones have also been used to capture urban mobility patterns, with GPS capturing fine outdoor mobility and cellular data capturing coarse mobility [51]. However, all these modeling efforts focus on outdoor

or *macro scale* human mobility across various Point of Interest (POI), locations, or city regions.

Studies have shown that humans spend over 80% of their lives indoors [12] resulting in indoor or *micro mobility*. Recent research has recognized that indoor mobility of users inside buildings, where many users spend a significant portion of the day, is very different from outdoor mobility exhibited when walking in a city or traveling in vehicles [124, 125]. As we model mobility at a finer spatial scale, mobility becomes more frequent and the prediction space expands. We argue that the motivation for indoor mobility, as well as the region of movement, is time-dependent and micro mobility shows high correlations to the macro mobility features of context, location type, and location name. Moreover, indoor mobility displays a complex sequential periodicity correlated to the macro, outdoor or coarse grained, features of mobility. Due to the above stated reasons, we cannot directly use outdoor mobility models that capture mobility at large grid or POI levels at a single spatial scale for indoor mobility modeling.

In this work, we present WiFiMod , a transformer-based multi-scale indoor mobility model that uses existing WiFi infrastructure to passively sense human mobility. In pursuit of this model, we have three specific goals. First, we argue that human mobility is inherently hierarchical, where macro-mobility, user type, and time of the day determine the micro mobility. Second, we capture the multi-modal features of macro as well as micro mobility patterns by creating a joint embedding and learn the correlations to generate sequences of context (Work or Home), building type (describes the space usage), building name (unique building identifier), and indoor location (room number, floor, or zone). The multi-modal embedding captures how individuals move between indoor spaces across and within buildings and takes into account how different space types exhibit distinct mobility patterns over time due to differences in space utilization. Third, we provide a ready-to-deploy system that uses

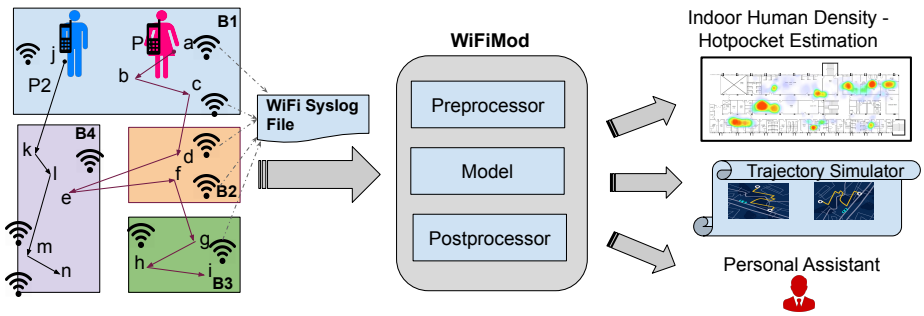


Figure 4.1: WiFiMoD : An indoor mobility modeling approach using WiFi sensing.

existing ubiquitous WiFi infrastructure present at all enterprise networks and uses system log (*syslog*) messages to extract indoor human mobility.

Our main contributions in this work can be summarized as follows:

- We design an end-to-end data-driven approach to model indoor human mobility using passive WiFi sensing. WiFi log based passive sensing approaches use already existing WiFi infrastructure in an enterprise or campus network providing a reliable indoor mobility dataset.

- We propose the use of multi-modal embedding to capture the macro and micro mobility features along with their correlations to improve the model prediction accuracy.

- We demonstrate the efficacy of our model by evaluating it against a real world dataset of 2500 users in a large campus setting and show that our model shows superior performance by at least 10% points over other indoor mobility models.

- We present three case studies that demonstrate the use of our model in predicting indoor hot pockets or high human density zones, generating user mobility trajectories, and designing personal assistants.

4.2 Background

In this section, we present the background for our work on data-driven indoor mobility modeling.

Mobility as Nomadic Behavior: Some mobility models, such as the classic random waypoint model, emphasize modeling the physical movement of users such as velocity, acceleration, and direction of movement [24, 29]. In contrast, several other models, including our work, view user mobility as inherently nomadic. Nomadic user mobility can be seen as a sequence of location visits, where users visit a location to spend some time at that location known as a *dwell* period then *transition* to another location, followed by a dwell period at the new location and so on [93]. Figure 4.1 shows two users P1 and P2 visiting multiple buildings B1 through B4 and spending time at various locations. Each dwell period at buildings B1 through B4 for P1 and P2 is followed by a transition. In this case, the emphasis is on *which* locations are visited at various times of the day, across multiple buildings, building types, and context, revealing the semantic meaning of the nomadic behavior. Since humans are creatures of habits and tend to follow a routine [97], we need to capture the correlations such as repeating visits to a location, repeating sequences resulting from daily or weekly routines, long-term dependencies, and affinity to certain locations, to name a few. While transitions from one location to another also need to be modeled, the emphasis is on capturing nomadic behavior, rather than factors such as the speed of mobility, the direction of movement, mode of transport, etc. Since, our primary focus is on modeling indoor mobility, modeling nomadic behavior is more appropriate since users are often stationary inside the building - in their office, in meetings, etc.

Modeling Trajectories: Mobility models come in many different flavors depending on what aspects of mobility the model is attempting to capture. A common type of mobility modeling to capture nomadic behavior is next location prediction [31, 76, 74, 41, 45, 79] where the model attempts to predict the next location that will

be visited by the user. Next location prediction can be used in mobile systems for location-aware services, caching, etc. In contrast, our modeling approach focuses on modeling and predicting the entire trajectory of the user (and devices) over the next few hours to an entire day. Modeling and predicting trajectory over many hours or entire day can be viewed as a generalized and more complex problem than next location prediction, since, doing so involves predicting a long sequence of future location and not just the next one. A trajectory is essentially a temporally ordered sequence of locations visited, duration of stay at each location, with transitions between two successive locations where the transit is the path used to move from the previous location to the next one. Figure 4.1 shows the trajectory of users P1 and P2 as a sequence of locations each visited for a specific time duration at a certain time of the day. Modeling the entire trajectory provides a holistic view of how users and devices move throughout a day.

Modeling Different Spatial Scales: A key design consideration in indoor mobility modeling is the spatial scale for capturing the nomadic movement of users and devices. Generally, models are designed to capture mobility or nomadicity at a single spatial scale and this spatial scale is often the same as that in the underlying dataset used to derive the models. For example, cellular data sets have been used to model mobility at the spatial scale of cell towers. In this work, we argue that indoor mobility models should be capable of modeling nomadic movement at different spatial scales and the choice of which spatial scale to choose should depend on what higher-level problems need to be solved using the model. While some prior work has focused on context-aware modeling they do not take into consideration the multiple spatial scales of mobility [73, 31].

In the case of indoor mobility within and across buildings, at least two spatial scales are desirable from a modeling perspective. For models that are derived using WiFi traces, the finest spatial scale for nomadic movement is that of an Access Point

(AP), which roughly translates to mobility at the scale of a room or a group of rooms in the span of a single AP. This spatial scale reveals *micro-scale* nomadic movement inside each building. It is also useful to consider coarser spatial scales such as considerably larger spatial regions (e.g. an entire floor) as a single location and consider nomadic movement across such coarser spatial regions. Another useful spatial scale is to consider an entire building as a single coarse-grained location to model *macro-scale* nomadic movement. In this case, a trajectory comprises visit to buildings, time spend inside a building, visit time of buildings, and transitions between buildings; at this scale, we are only concerned with which building (e.g. in a university campus) users visit and not how they move inside that building.

Different spatial scale models lend themselves to solving different types of problems. For example, a macro-scale model is useful for designing location-aware recommendations when a user visits a building, while a micro-scale mobility model is useful for indoor resource scheduling and hot pocket identification. As noted earlier, we employ a hierarchical approach for modeling mobility at multiple scales. Doing so not only enables our models to predict both macro- as well as micro-scale mobility patterns, it is also more efficient—it reduces the prediction space by first predicting mobility patterns at the macro scale and then modeling micro scale patterns conditioned on the estimated macro scale patterns.

WiFi Log Based Passive Sensing: Today, WiFi is ubiquitous at university campus, enterprise, and urban locations. When users move across the campus with their mobile devices, the devices get associated and disassociated with access points (AP) along the user's mobility route. These device associations and disassociations get logged as events into the system log, syslog, of each AP. We use the AP syslog file to passively observe the user devices as they move across the network and derive user mobility by using the smartphone as an alias for user mobility since users carry their mobile phones with them everywhere. The key benefits of using WiFi syslog

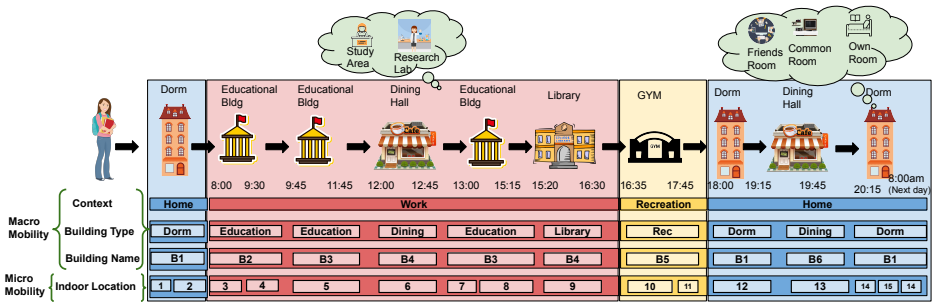
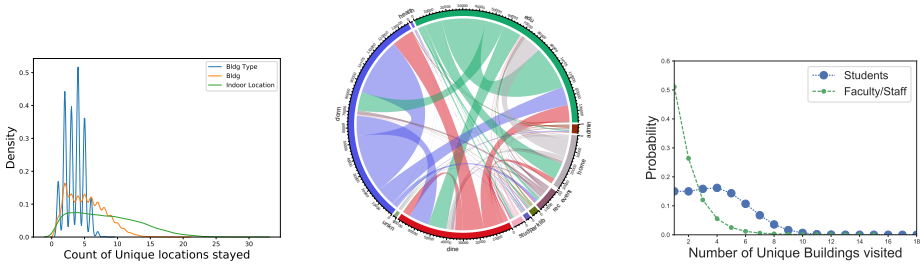


Figure 4.2: Mobility Hierarchical View

for passive sensing are (i) we do not need any new installation or deployment of any devices as, in most places, the WiFi syslogs are collected by the Information Technology (IT) department to analyze network performance or network attacks; (ii) no data collection on the user device needs to be done and no user intervention is needed to collect the data, and (iii) WiFi is present indoors and thus WiFi logs provide a viable method to learn indoor mobility.

Multi-Scale Mobility While it has been shown that user mobility displays recurring patterns at a scale, we argue that human mobility is inherently hierarchical, where hierarchy is represented by spatial granularity scale as it becomes fine grained micro mobility from a coarse grained macro mobility representing context, building type, and building name. As shown in Figure 4.2, a user who visits several locations to accomplish their daily tasks seems extremely mobile at the scale of indoor location, visiting 14 locations throughout the day. As we change the spatial granularity to a coarser grain, we find that the mobility becomes infrequent at the building scale, where the user visits 10 buildings. Finally, at the context level—which defines the overall span of activities the user performs in the part of the day—the user shows mobility across only 4 contexts. Thus, showing that human mobility becomes more frequent as the spatial scale becomes fine grained. Also, each indoor location space shows high affinity to the context and building type displaying dependencies and correlations between macro and micro scale mobility features.



(a) (b) (c)

Figure 4.3: Features influencing indoor mobility prediction (a) Spatial Scale (b) Building Type (c) User Type

Features Impacting Indoor Mobility Prediction We conducted empirical analysis on a large campus WiFi syslog dataset described in §4.4 and found that four main factors impact micro mobility:

- **Spatial Scale:** Figure 4.3(a) is a density plot of count of dwell locations of users across an entire day at each spatial scale. Dwell location is defined as a location where users spend at least 10 minutes. *Context* describes the situational factors such as work or home. *Building type* indicates the building usage activity: for example, a food court is used for dining, while a building with classrooms is used for education. *Building name* is the location name visited, and the *indoor location* is the location inside the building visited as shown in Figure 4.2. We see that the average number of visits are 4, 5, and 11 at building type, building name, and indoor location level respectively. Giving us the insight that *as the spatial scale becomes more fine-grained, from context to indoor mobility, the user mobility becomes more frequent.*

- **Building Type:** Figure 4.3(b) is the chord diagram showing user movements within and across different building types. We see that an educational building, as well as dorms, see more dwell locations within the buildings where other building types such as admin, and dining see relatively less within building

dwell locations. The main reason is that students move from one classroom to another within and across educational buildings during work hours resulting in a high number of dwell locations in education buildings. This indicates that the *space type that governs the primary activity inside the building plays is an important feature in indoor human mobility.*

- **User Type:** our campus dataset has two types of users, students, and faculty, as identified by the role field in the authentication events of syslog messages. Figure 4.3(c) shows the distribution of the unique number of buildings visited by users (students and faculty/staff), here multiple visits to a building count as a single unique location. We see that on an average a faculty/staff visits 1.2 unique buildings per day while students visit an average of 3 unique buildings per day. Thus, illustrating that user type influences the observed user mobility.

- **Past Behavior:** We find that the future mobility of a user is highly dependent on past behavior. Users who display high conformance behavior in the past continue to do so in the future. This observation is inline with the findings in prior work [57].

4.3 Problem and Approach

4.3.1 Problem Statement

We focus on the problem of modeling indoor mobility trajectories of users over the timescale of several hours to a day. We assume that historical indoor mobility data for each user is available for purposes of modeling. A trajectory of a user over a duration such as a day is defined to be a sequence of tuples (c,s,b,l), where each tuple comprises of context (c), space type (s), building location name (b) and indoor location name (l). Our model seeks to predict the trajectory of each user while learning the correlation between the c,s,b, and l at multiple spatial granularities.

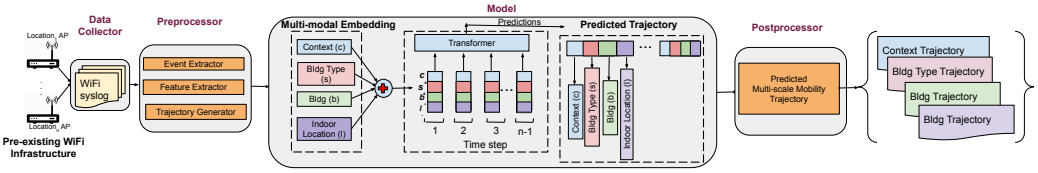


Figure 4.4: System Architecture Diagram

Further, we model trajectories inside a single building as well as those that span a collection of nearby buildings.

4.3.2 System Overview

Figure 4.4 shows the architectural overview of WiFiMod . WiFiMod is a pipeline of 3 main modules: data collector, data preprocessor, and model. The main objective of the data collector is to collect the WiFi syslog files across all the APs in the enterprise network. Most IT departments already have the network logging turned on; if disabled then the IT admins would need to turn "on" the network logging to enable data collection. The output of this module is an aggregated syslog file from all APs across the campus. The aggregated syslog file is fed to the data preprocessor, which extracts the events and fields needed to generate user trajectories from the raw syslog files. This module is vendor-specific, depending on the vendor of the deployed AP. Currently, WiFiMod supports HP-Aruba syslog files. Once the user trajectories are extracted they are fed into the model, which extracts the macro and micro mobility features, creates a multi-modal feature embedding, and feeds it to the Transformer model. The output of the model are predictions, which can be used to generate reports or are aggregated to predict space usage and occupancy for various applications.

4.3.3 WiFi-Based Modeling Approach

A large, campus-like infrastructure comprises of various building types such as dormitory, educational, dining, student union, research labs, health center, recreational

center and administrative. Campus users move across multiple buildings everyday to accomplish their tasks and use resources scattered across campus. A campus or enterprise WiFi network provides seamless WiFi coverage inside buildings and between buildings through Access Points (AP) installed across the geographical area of the institution. As users move within this geographical area, their devices connect and hop APs. Each AP maintains an internal log that consists of a list of all events observed by the AP.

When a user connects their device, it associates with a nearby AP. Each AP has a fixed location identified by the room, floor and building of installation. As a user moves across multiple locations on the campus, the device gets associated and disassociated with multiple APs on the user's path. The association and disassociation events, along with timestamp, Device MAC, AP ID, and event type get logged in the internal syslog file maintained by each AP. Extracting all the association, disassociation or drift events from syslog files of all APs on the campus and indexing them by timestamp gives us a sequence of APs visited and duration of visit by each user device. Since all AP locations are known in terms of building, level and room of installation, it further helps us derive user device trajectory information at multiple spatial scales.

The enterprise WiFi network on campus is operated with RADIUS authentication that mandates all users to authenticate before connecting to the network. Since today's users carry a plethora of mobile devices we extract these authorization messages from syslog files to create a user-to-device map and use this to identify the mobile devices (typically the smartphone) of each user and use its trajectory as an alias of the user trajectory.

Now, to train a data-driven model, we collect syslogs and extract trajectories for each user for a few weeks and create a historic trajectories dataset for training the model. From the extracted trajectories, we derive macro and micro mobility features

based on the building type and heuristic rule for context defined above. This serves as the input to the model, which is a global model trained on all user trajectories. We use the multi-level spatial features of each trajectory to create a multi-modal embedding and train the Transformer. The predictions of this model are then used as is for individual mobility or can be aggregated.

4.3.4 System Architecture

4.3.4.1 Preprocessor:

The syslogs collected from the APs are a deluge of data mainly used for system diagnosis or analysis of attacks on the enterprise network. A typical syslog is a collection of diverse timestamped events, where each event has a pre-specified format. The goal of the preprocessor is to extract the relevant events from the syslog file and convert the events into a trajectory. The preprocessor is a sequence of 3 main steps: event extraction, data dependency resolution, and trajectory generator.

In the first step, we extract association, disassociation, reassociation, authorization, deauthorization, and drift event messages, hereby referred to as presence messages, from the syslog file. The event format is as shown below:

```
<Timestamp> <hh:mm:ss> <controller_name> <event_id>
<message_body : MAC_ID , AP_ID, other text>
```

The timestamp field gives us the time of event; *event_id* gives us the event type; *message_body* consists of device MAC_ID, which identifies each device uniquely, and AP_ID, which gives us the AP details namely building name, level and room number. Authorization and deauthorization messages additionally have username and role fields that help create a mapping between users and their devices, used for selecting the most mobile device from the collection of devices owned by each user, along with the role of the user on campus identified as student or faculty/staff.

The event logging in syslog has lots of inconsistencies such as dropped events, time sequence events overlap, multiple similar events, incorrect order of events, multiple disparate event types logged for the same device at the same timestamp, to name a few. Such inconsistencies need to be resolved before the mobility trajectory of the device is computed. The main objective of this step is to resolve these inconsistencies, estimate the missing entries, clean the data, and generate a timestamped sequence of rows of association and disassociation of devices with AP.

After that, we gather all events per user device and create a timestamp indexed sequence to identify the APs visited, along with the time of visit, to generate a mobility trajectory. Then, for each generated indoor mobility trajectory, we add the corresponding context, building type, and building name to each visited indoor location. We generate the context based on a simple heuristic that campus working hours are between 8:30am and 4:30pm, so all user activities between these times are marked as "work" context and the rest are marked as "home" context. We find that students who stay on-campus display both these contexts whereas for off-campus users, we generally see only the work context except for students in research labs who work outside the work context hours and students who stay on campus to use recreation and student union facilities later or early during the day. Each building on our campus has a specific usage assigned to it (e.g. educational building have classrooms, dining has food courts, recreational building has swimming pools, squash courts, gymnasium). We use the designated space activity as the location space type. Thus, for each indoor location visited in the extracted WiFi trajectory we compute the corresponding context, space type, and building name resulting in a sequence of (c,s,b,l) tuples as the multiple spatial granularity trajectory.

4.3.4.2 Transformers for Sequential Prediction:

The Transformer neural network architecture [106], originally introduced for the task of machine translation, follows an encoder-decoder structure. The encoder maps a sequence of inputs $\mathbf{x}$ consisting of the inputs x_i at each position i to a sequence of continuous representations $\mathbf{z}$. These representations are provided as the input to a decoder that autoregressively generates an output sequence of labels $\mathbf{y}$, with the prediction at each output timestep conditioned on the entire input sequence $\mathbf{z}$. The length of the output sequence is not tied to the length of the input sequence. In the Transformer architecture, the encoder and the decoder share the same neural network architecture structure, except that in the decoder, the representation at position i is prevented from observing representations at subsequent positions. We describe this architecture in more detail below.

First, a sequence of input tokens is first mapped to corresponding d_{model}-dimensional input embeddings via an embedding lookup table. These embeddings are then fed to the encoder of the Transformer, which is comprised of L layers of the same form. Each layer j passes its inputs through two sub-layers, multi-head self-attention and a feed-forward layer, with residual connections (addition followed by normalization) between each:

$$
\begin{aligned}
h_1 &= MultiHeadAttention(\mathbf{x^{(j-1)}}) \\
h_2 &= LayerNorm(\mathbf{x^{(j-1)}} + \mathbf{h_1}) \\
h_3 &= FeedForward(h_2) \\
\mathbf{x^{(j)}} &= LayerNorm(h_2 + h_3)
\end{aligned}
$$

For the representation x_i at a given position in the sequence, self-attention computes scores between x_i and every other representation in $\mathbf{x}$, and uses those scores to

compute a weighted average (attention) over the representations at all positions. In multi-head self-attention, this operation is performed k times, so that k different attention functions can be learnt, to model different dependencies between elements in the sequence. For more low-level details, see [106]. For each attention head, three matrices Q, K, V are created by multiplying the input (a sequence of embeddings) with weight matrices W^Q, W^K, W^V (of dimension $d_{model} \times d_{model}$, $d_{model} \times d_k$, and $d_{model} \times d_v$, respectively).[1] Using the terminology from [106], Q represents "queries", K represents "keys", and V represents "values". The multi-head attention mechanism allows for the model to jointly attend to information from different representation subspaces along different positions. Layer normalization is applied after residual connections to improve optimization.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^o$$

where,

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Attention is given by the formula:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt[2]{d_k}}\right)V$$

It is also worth noting that, unlike recurrent neural networks (RNNs) or convolutional neural networks (CNNs), Transformers are not inherently sensitive to sequential order, and it is therefore common to inject information about the position of elements in the input sequence via *positional encodings*. These positional encodings are added to the input representations.

[1] In practice, it is common to set $d_k = d_v = d_{model}$, as we do in this work.

4.3.4.3 Multi-Modal Transformer Model:

To learn the sequential as well as long-term dependencies from the input trajectories we use a Transformer-based autoregressive (sometimes called "causal") language model. We use an off-the shelf Transformer implementation based on GPT-2 [94] and train it from scratch on our dataset.[2] We treat the task of predicting the next set of locations visited by the user as a task of language modeling, where language modeling task is defined as the task of predicting next character or word in a document.

Our Transformer model takes as input $m = 4$ trajectories corresponding to the *context, space type, building*, and *indoor location* spatial modalities generated by the preprocessor. We map the events in each raw trajectory to an index in a shared vocabulary of size V, to obtain m integer-valued sequences $T_1, ..., T_m$, each with length n. Then, we map the entries in each sequence to learned, d-dimensional *event embeddings* $E_1, ..., E_m$, where $E_i = \langle e_1, ..., e_n \rangle$ and each $e_i \in \mathbb{R}^d$. Since the vocabulary is shared, we use a separate set of event embeddings for each modality to avoid collisions when event ids from different modalities happen to overlap. Since Transformer models inherently lack an inductive bias that would allow them to be sensitive to different sequential orderings, we also learn d-dimensional *position embeddings* P for each of the n positions. We obtain a single joint embedding by summing together $Je = E_1 + ... + E_m + P$. The derived joint embedding Je is passed through m stacked Transformer encoder layers, each of which has m attention heads.

We train this model using a self-supervised autoregressive training objective: given the events at timesteps $1, ..., i-1$ in each modality, the model is trained to predict the events that occur at timestep i. In other words, the model estimates $p(c_i, s_i, b_i, l_i | c_{1:i-1}, s_{1:i-1}, b_{1:i-1}, l_{1:i-1})$. To make predictions, we pass the outputs obtained from the Transformer encoder through an additional linear layer of dimension

[2]We use the GPT-2 [94] implementation available in the HuggingFace Transformers [110] library, version 4.4.2.

$d \times V$, which is shared across all modalities. We hypothesize that using a shared output layer encourages the embeddings for different modalities to maintain a coherent geometry relative to one another; however, we leave in-depth analysis of different architectural choices for future work.

During training, we convert the logits obtained from the output layer to (log) probabilities via the softmax function, then compute the batchwise-mean cross entropy loss for each modality. We sum these together to obtain a final combined loss. We minimize this objective over 15 epochs using the Adam optimizer [64] with a learning rate of 0.01 and a mini-batch size of 100. As mentioned above, we use $m = 4$ Transformer layers and $m = 4$ attention heads per layer. We set the embedding dimension d to 64.

Postprocessor: The predictions of the transformer are fed to the postprocessor to generate trajectories of desired length for each of multiple spatial granularities by segregating the predicted location values of mobility at each spatial scale and then arranging them as a sequence indexed by the time of the day.

4.4 Experimental Evaluation

4.4.1 Dataset and Parameter Setting

Dataset For the evaluation of our model we use campus-scale device trajectory dataset extracted from WiFi logs of a large university campus as stated in section §2.2.1. Table 4.1 provides dataset details. The event log for the 2 months of Fall'19 is over 150GB in size and contains 6.4 billion events.

Parameter Setting: To evaluate the robustness of our proposed model we use a train-dev-test split of 80-10-10 where we use the first 80% data of each user as training data, next 10% as dev and rest 10% as testing data. For the selection of model hyper-parameters, we use a grid search over the parameter space and select

Item Description	Value
Number of Users	2500
Number of Building Types	13
Number of Buildings	156
Number of APs	5104
Avg. Buildings visited	4
Avg. Indoor locations stayed	8.97/building
Time Span	Fall'19: Sep-Nov '19

Table 4.1: Dataset Description

the optimal parameter settings using the dev dataset. Parameter optimization is performed using mini-batch Adam optimizer and with a batch size of 100.

4.4.2 Baseline Comparison

To evaluate the effectiveness of our model we compare our proposed model with the following:

N-gram: An n-gram model is one of the most important tools in speech, language and text processing. An n-gram model is used to estimate the conditional probability of visiting a location given the sequence of previously visited locations. We include evaluations against first and second order Markov chains as the baseline. A bi-gram model uses past location to estimate the probabilities (using MLE), whereas tri-gram approach conditions on past 2 locations.

HMM: In a Hidden Markov Model (HMM) we regard all visited locations as state and build a transition matrix based on the sequence of locations visited. We train one HMM for all users and each hidden state generates locations over a Gaussian distribution.

LSTM Long Short Term Memory (LSTM) has shown superior performance for sequential data and encoding long term dependencies, so we use LSTM as one of our baselines.

Model Name	15 mins	30 mins	60 mins
bi-gram	5.32%	6.24%	7.94%
tri-gram	9.51%	12 .3%	19.24%
four-gram	14.45%	17.5%	24.8%
HMM	15.16%	20.21%	25.36%
LSTM	57.48%	62.3%	71.96%
Simple Transformer	64.28%	69.93%	75.81%
Our Model	**68.39%**	**79.4%**	**83.2%**

Table 4.2: WiFiMod indoor mobility prediction comparison with Baseline Models.

Simple Transformer is an adaption of our model, which does not perform multi-modal embedding. For indoor modeling we train the simple transformer with the historic indoor trajectories. It is a basic autoregressive language model implemented with Transformers.

Results: Table 4.2 shows the comparison results between our proposed model and the baseline models. For the evaluation, we predict the entire indoor trajectory generated by each model for each user at a temporal granularity of 15 mins, 30 mins, and 60 mins and check the predictions against the ground-truth locations to compute the model accuracy. We evaluate WiFiMod against other baselines and find that Transformer-based WiFiMod outperforms both the LSTM model and HMM. Transformers have a higher-order transition modelling capacity than a HMM. In addition, the multi-head self-attention mechanism allows it to capture long-term dependencies more effectively than an LSTM.

In general, the deep neural network (DNN) based models show superior performance to n-gram models and HMMs, demonstrating that long-term historic information is important for mobility modeling and prediction. The DNN approach captures long-term regularities— e.g. if the start location of a trajectory is a dormitory, the likelihood of the trajectory ending in the same dormitory is high—whereas this information is not captured by n-gram or HMM models.

Additionally, we observe that, due to variations in human behavior, there are errors in prediction too. For example, students frequently change the dining halls visited based on the menu at each dining hall or based on the dining location visited by their friends. Also, non-regular mobility such as visits to university health center or to an administrative office are hard to predict and the model does not capture such high variations from users' routine mobility. We also find that our model captures the recurring mobility patterns at the inter-building level with a very high accuracy of 90% but, due to variations introduced by human behavior such as visiting a different dining hall or carrying out an unexpected errand at an administrative building, etc results in the induction of errors. Another interesting observation is that varying the temporal scale of trajectory has an impact on the prediction accuracy.

Impact of Temporal Granularity In this experiment, we vary the temporal granularity of training and prediction. Trajectory temporal granularity refers to the sampling rate of trajectories. We represent the user trajectory as a sequence of locations, where the location is sampled every n minutes. We train the model on trajectories with a temporal granularity of 15 mins, 30 mins, and 60 mins, here on referred to as T15, T30, and T60 respectively. As shown in Table 4.2, we find that across all models with different sampling frequency T60 prediction is highest followed by T30 and then T15. We find that as the model's temporal granularity becomes coarser, the indoor mobility accuracy increases because indoor human mobility is more frequent at fine granularity. When we learn mobility at a coarser temporal scale of 60 mins, frequent short mobility observed at 15 min temporal scale such as a break to visit the vending machine or stop by a colleague's office for a chat gets masked. Additionally, such short micro events have high variability and cannot be predicted accurately at a fine temporal granularity resulting in reduced accuracy at a fine temporal granularity as seen in T15 trajectories, location sampled every 15 mins.

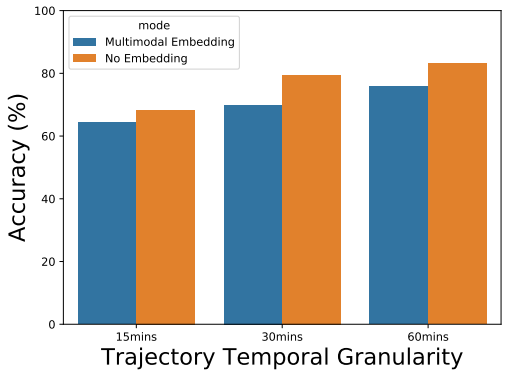


Figure 4.5: Multimodal embedding effectiveness. Comparison of indoor location prediction of a transformer for multimodal and non-multimodal embedding input

4.4.3 Effectiveness of Multi-Modal Embedding

In figure 4.5 we see that the indoor mobility prediction accuracy of our model is higher than a single Transformer implementation that has a flat input structure of only indoor locations. To compare the two models on prediction accuracy, we predict the next top-1 location with both the models for the same test dataset on indoor location granularity. The multi-modal embedding model shows an accuracy of 83.2% while a simple transformer with no embedding has an accuracy of 75.81% for T60 trajectories. The multi-modal embedding approach outperforms the non-embedding approach even for T15 and T30 trajectories demonstrating that modeling mobility from a hierarchical perspective where the model learns the correlations across multiple spatial scale mobility using multi-modal embedding results in higher prediction accuracy. The intuition behind higher accuracy is that the multi-modal approach significantly reduces the prediction space by learning the correlations between macro and micro scale mobility, conditioning the prediction on the estimated macro and micro scale mobility distribution, thereby using the topological constraint of the multiple spatial scales. Additionally, the model also captures the correlation and periodicity in mobility across varying spatial scales.

Spatial Scale	WiFiMod	Non-Hierarchical
Building Type	89.58%	89.23%
Building Name	87.39%	81.12%
Indoor Location	83.2%	75.81%

Table 4.3: Comparison of accuracy of hierarchical and non-hierarchical model across multiple spatial scale.

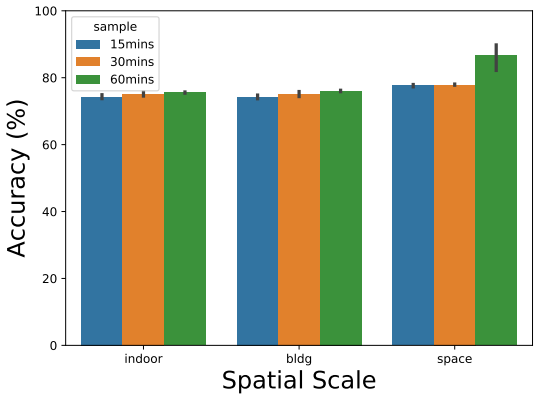


Figure 4.6: Space Type Prediction

4.4.4 Importance of Space Type Prediction

While analyzing the indoor location predictions made by the model, we find that most of the errors are in predicting food court location and space inside food courts, indoor library locations of use, indoor location inside the recreation center, etc. These locations have a high variance when predicting the indoor location. However, we find that the model displays a high accuracy in predicting the context, and location types and low accuracy on building name, in the case of multiple food courts, or indoor location of use. Figure 4.6 analyzes the model accuracy by space type. We see that the model has high prediction accuracy for building type followed by building name and lowest for indoor location across all 3 sampling frequencies T15, T30, and T60. This is mainly because, while routine activities such as visiting the classrooms, office space, research labs have fixed building type, building name, and indoor location while visits to high variance locations such as library or dining hall has a fixed building type

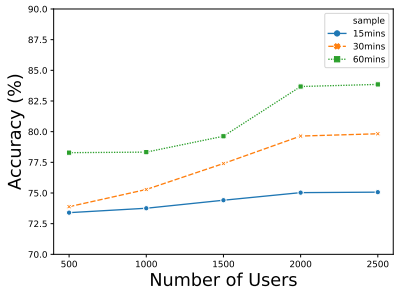
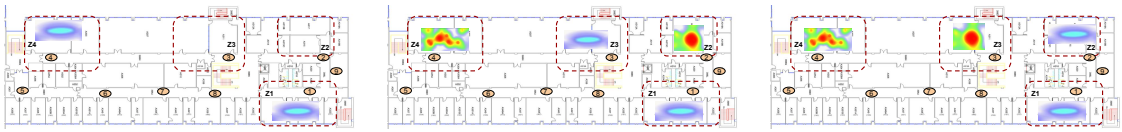


Figure 4.7: Impact of number of users on model accuracy

but variance in indoor locations (since the person might not sit at the same location always) and building name (since the person might not visit the same building under the building type, such as dining hall). Additionally, we find that most errors are found in indoor location prediction, fine spatial granularity for high sampling rate of 1 sample every 15 mins in T15 because this trajectory captures the most unscheduled high variance micro mobility at a fine spatial scale.

4.4.5 Impact of the number of trajectories

We vary the training dataset by using a subset of trajectories, with subset sizes of 500, 1000, 1500, 2000, and 2500 user trajectories. We train the models on the subset of user trajectories for the first 7 weeks of the semester and predict the user trajectories for the next 2 weeks. We find that the transformer based model displays higher accuracy for larger training set size indicating that the model has better generalizability and higher performance for more and new data. The model accuracy for T15 increases the most from 73.4% to 75.06%, whereas model accuracy for T60 increases from 78.28% to 83.2%. Across all trajectories, with different temporal binning we see that the model accuracy increases as we increase the number of user trajectories in the training dataset.



(a) 8:30am (b) 12:00pm(Noon) (c) 3:30pm

Figure 4.8: Heatmap of predicted indoor occupancy of educational bldg with classrooms, research labs, faculty office, kitchenette

4.5 Case Studies

In this section we discuss three case studies of our proposed system WiFiMod.

4.5.1 Case study 1: Indoor COVID19/ILI Hotspot Prediction

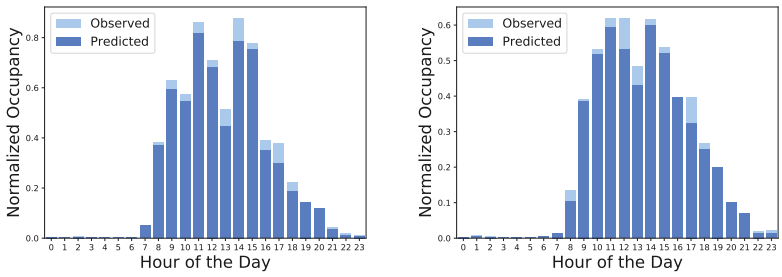
With the current COVID-19 pandemic, building occupancy scheduling and resource allocation for de-densification is a key component in designing re-opening policies. Here, we present a case study of using WiFiMod to predict indoor mobility of a building across the coarse of an entire day(s) to identify indoor spaces with high space utilization that can become a hot pocket zone and needs dedensification so that the number of users inside the building is always below the 50% or 25% usage constraint for space usage..

Here, we use the trained model to predict the user trajectories of all users on campus for the entire day. We then aggregate all these predicted trajectories across the temporal attribute to compute the occupancy at each indoor location at various times of the day. In our case since we are using WiFi AP syslogs, the indoor spatial granularity is zone level where each AP captures occupancy per zone that might encapsulate a single room or across few rooms, based on the range of AP deployed. Fig 4.8 shows the floor map of an educational building with 9 deployed APs. The floor has a combination of faculty office, break room, research labs, and classrooms. We focus on APs 1-4 which have a range across zones Z1-Z4 respectively as indicated in figure 4.8. Zone Z1 encompasses few faculty offices and a research lab, Z2 spans

across a kitchenette and a research lab, Z3 across a conference room and a student work space, and Z4 across a classroom and a research lab. Fig 4.8 (a)-(d) shows the computed indoor user occupancy based on the model predictions, at 3 different times of the day. We find that at 8:30am fig 4.8(a) the space occupancy is very low with the start of the day across all zones, with some occupancy in Z1 and Z4. Fig 4.8(b) shows predicted space occupancy at noon and we observe high human density across Z4(classroom zone) with an in-person class (in 2019), Z2(kitchenette area) with the break room where students gather to eat lunch, and the rest zones show low to moderate occupancy. Fig 4.8(c) shows predicted space usage at 3pm and we see that zones Z3, and Z4 have high occupancy due to predicted recurring seminar, and classroom usage respectively while Z2 kitchennet and Z1 lab space have relatively low occupancy. Fig 4.8(d) shows predicted space usage at 5pm and we see some occupancy in zones Z1 and Z2 that comprise of research labs with students still working in late evenings. The computed occupancy across the 3 times of the day shows an accuracy of 96% as compared with observed ground truth indoor occupancy computed from WiFi logs.

In the heatmaps 4.8(a)-(c) the red areas indicate high human density or hotpocket zones on the floor map. We can generate such heatmaps for all indoor spaces across the times of the day to identify hotpockets and design policies or space usage schedules to de-densify them to lower the risk of disease spread and safe opening of indoor spaces.

Additionally, indoor location occupancy computed by aggregating indoor mobility can also be used to generate customized Heating, Ventilation, and Air Conditioning (HVAC) schedules per building. Such customized HVAC schedules can help reduce the energy consumption while increasing the user comfort by scheduling HVAC to turn-on with predicted indoor occupancy while turning it off with low to no indoor occupancy.



(a) Location1(r^2 0.989) (b) Location2(r^2 0.984)

Figure 4.9: Normalized hourly occupancy across 2 locations computed from actual and simulated trajectories.

4.5.2 Case study 2: Human Mobility Simulation

Mobility datasets are fundamental to evaluation of a system or applications such as simulation of disease spread. However, such datasets are hard to obtain due to privacy concerns. Majority of the mobility trajectory generators use deterministic models that have predefined mobility distributions or assume human mobility follows levy walk, random distribution or stochastic process, failing to capture realistic mobility. This leads to a gap in analyzing and fine tuning systems. Hence, we propose scenario simulation by generating synthetic mobility trace using our pre-trained hierarchical model. To demonstrate the efficacy of our model for synthetic trace generation we generate the trajectory of users and their devices using a pre-trained model on the campus mobility dataset. We compare the hourly occupancy at 2 different locations computed from the synthetic trace and observed trajectories. The simulation is performed at 20% of the total population, and the observed transition and occupancy are scaled down accordingly. Figure 4.9(a) and (b) compares the hourly occupancy of 2 different locations, loc1 and loc2, and the model demonstrates a high accuracy with the coefficient of determination, r^2 value as 0.989 and 0.984 for the loc1 and loc2 respectively.

For applications such as user profiling and behaviour analysis, which need to capture variations in human behavior we can introduce realism in capturing the variations in human behavior by changing the inference mechanism in the decoder from selecting the next location that gives least negative log-likelihood to sampling the next location from the top-5 possible next predictions. To validate if our synthetic traces are close to the real dataset, we do a domain search of the generated trace to actual observed trace and find trajectory similarity score of 82% on weekdays and 63% on weekends for indoor mobility.

4.5.3 Case study 3: Single User Personal Assistant

The last few years have seen the introduction of personal assistants that share the goal of presenting the user the right information at the right time. However, knowing when to present the information without any query from the user is an important criterion and a critical limitation in many of today's models. Since the information presented is mainly associated with current location, time of day, space type, and user type. We propose that a user mobility model derived by using WiFiMod serves as a foundation for a highly accurate personal assistant that can be used for informing users with a variety of tasks/events/updates. We use a globally trained model and fine tune it for each user by locally training it with the historic trajectories of each user to create a personalized model and use it to make macro and micro scale predictions. We find that our model shows high indoor mobility prediction accuracy in the top-1(accuracy of most likely location) prediction score is 89% for user type faculty/staff and 85% for user type student for weekdays. Such a model can be augmented with the user calendar or campus event calendar to notify the user with upcoming events of interest or prior scheduled classes or meetings.

4.6 Conclusions

Modeling indoor mobility and using the correct spatial granularity of mobility can substantially benefit a large range of applications. In this paper, we proposed WiFiMod, a data-driven approach to model indoor human mobility using passively sensed WiFi logs. We demonstrated that as the spatial and temporal scale of modeling mobility becomes more fine the model accuracy reduces because we encounter more frequent, unscheduled user mobility that gets masked in a coarse grain scale. In WiFiMod we jointly model mobility context, space type, outdoor location, and indoor location using a transformer to learn the correlations of mobility at various spatial granularities. We extensively evaluated our approach using available ground truth WiFi data at a large university campus comprising 156 buildings across 13 building types and found that our model outperforms the current state-of-the-art baselines significantly.

CHAPTER 5

WiFiTrace : NETWORK-CENTRIC CONTACT TRACING

In the earlier chapters, I presented empirical analysis and human mobility models. Aside from analyzing the mobility anatomy and modeling it, we can also design mobility-aware applications. In the current and following chapter, I will present 2 mobility-aware applications that are based on individual as well as aggregated user mobility. Here, I present WiFiTrace, which is a mobility-aware application that back-traces past individual user trajectories to identify or extract information after an event is observed in time .

COVID19 infecting 80,200,000 people worldwide and causing more than 1,750,010 deaths globally [111] has highlighted the pressing need for development of scalable tools for containment of infectious diseases. Spread of human communicable diseases such as measles, coronavirus, SARS can be significantly reduced by disease containment and mitigation. One of the most effective tool for containment is contact tracing, which is a method of tracing and testing all the users who have come in contact with an infected person and iterating the process for any contact that tests positive. Current method of contact tracing is mainly manual where the healthcare workers interview the infected person to identify the contacts in close proximity. This process of manual contact tracing is slow and due to gaps in human recollection results in incomplete and inconsistent contact identification.

This wide scale impact of COVID19 along with the need for quicker and better contact tracing solution has resulted in the emergence of research in technology backed-up solutions referred to *"Digital Contact Tracing"*. Many countries have de-

ployed apps such as AarogyaSetu in India, TraceTogether in Singapore, SwissCovid in Switzerland that have been downloaded by millions of users yet there has been a low adoption [58] and due to a number of reasons namely privacy, security, high battery drainage, user dependency on app installation. For any app or tool to be effective at least 80% [5] of the user population should adopt it. Such high dependency on the user population adoption has resulted in ineffectiveness and incomplete digital contact tracing using the bluetooth and GPS based apps. To over come this problem, we propose WiFiTrace, a network centric digital contact tracing solution that uses already existing enterprise network syslogs.

5.1 Background and Motivation

In this section, I provide background on contact tracing and present motivation for the network-centric approach.

5.1.1 Background

In this section, we provide background on contact tracing and present motivation for our network-centric approach.

Contact Tracing Procedure:

Contact tracing is a well-established method used by health professionals to track down the source of an infection and take pro-active measures to contain its spread [33]. The traditional method is based on questionnaires, whereupon diagnosis, the user is asked to list places visited, and other people they have contacted. This information is used to contact the set of potentially infected individuals, and the process is continued recursively until all possible infections are eliminated [33]. The contact tracing goals are two-fold: identify the potential infection source for the diagnosed individual and determine others who may have gotten infected due to proximity or contact.

Unfortunately, many illnesses have a 2 to 14 day incubation period between infection time and the onset of the illness. Thus, infected users might need to recall the places they had been to and the people they interacted with two weeks ago. This reliance on a person's possible fickle memory to trace the previous location and interaction footprints is a laborious manual process and erroneous.

Client-centric Digital Contact Tracing:

With the ubiquity of smartphones, digital contact tracing leveraging Bluetooth, and GPS [19, 2] has become popular. In this approach, a unique (often anonymized) identifier is transmitted, using Bluetooth, from every phone. Each phone also listens for such identifiers from other phones to determine users/phones in its proximity. This information is further used for contact tracing. Apple and Google have implemented this basic approach as part of their contact tracing API [2], which has been utilized by many standalone apps [4, 3, 1, 19, 6].

These solutions are client-centric and require high user adoption for successful contact tracing coverage. In particular, each user must download the standalone app, allow the necessary permissions to sense, and allow the app to run continuously in the background. The need for high user adoption proved to be a key hurdle for Singapore's TraceTogether app [6], which achieved only 1.1 million downloads despite needing a critical mass of 4 million active users (around two-thirds of the population) to be effective [5]. These challenges have led health experts to argue that while technology-based contact tracing solutions are useful, they should be seen as complementary to traditional yet still effective methods of contact tracing [38].

5.1.2 Motivation of WiFiTrace

Aid Health Experts:

WiFiTrace was designed to support health experts performing contact tracing in two key ways. First, WiFiTrace aids in acquiring detailed location histories of indi-

viduals for whom contact tracing is necessary. This significantly reduces the burden and errors caused by high dependence on human recollection to determine potentially infected contacts. Second, WiFiTrace processes the overwhelming amount of location information to prioritize co-locators at high risk of infection. This significantly reduces the number of contacts that need to be followed up by the health experts. In addition, unlike standalone contact tracing apps that require end-users to self-monitor and self-report their proximity to infected users, WiFiTrace is designed to integrate with a health expert's contact tracing workflow.

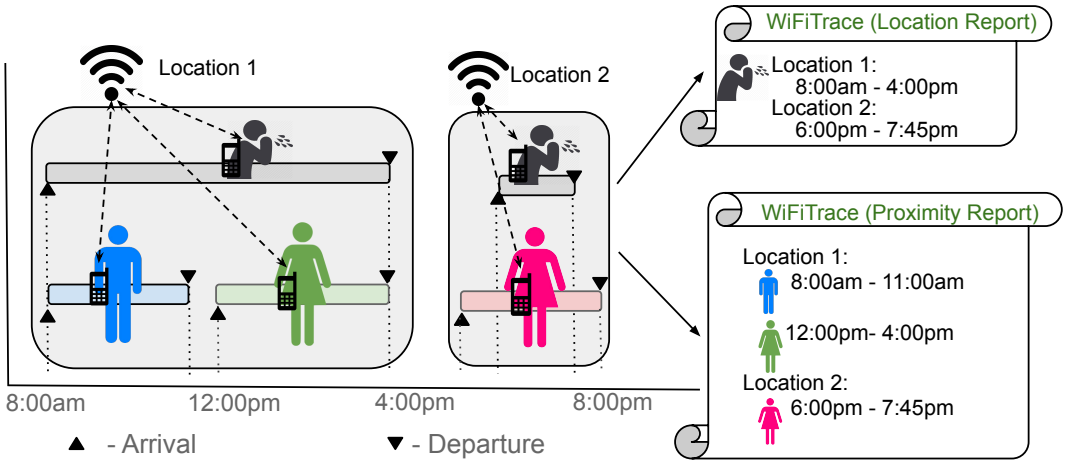


Figure 5.1: A network-centric approach to contact tracing using WiFi sensing

Network-centric Digital Contact Tracing:

For WiFiTrace to be useful to health experts, it must overcome the key adoption challenge faced by client-centric approaches. To achieve a high adoption rate, WiFiTrace uses a network-centric sensing approach using passive WiFi sensing, as illustrated in Figure 5.1.

These days, WiFi access is almost ubiquitous in key environments such as offices, university campuses, and shopping areas. WiFi sensing has thus emerged as a popular approach for addressing a range of analytic tasks [54, 126]. WiFi sensing can be client-based (i.e., done on the mobile device) or network-based (i.e., done from the network's

perspective). An example of client-side WiFi sensing is performing triangulation via RSSI or time-of-flight measurements to multiple WiFi access points and localize a device's position [82].

On the other hand, network-centric WiFi sensing uses the network's view of where individual WiFi devices are located to perform analytics. This approach has been used for monitoring the mobility of WiFi devices by analyzing the sequence of the access points that see the same device over a period of time [54].

WiFi sensing (both client and network-centric) has been used to characterize and model user mobility [69, 63] with more recent work leveraging WiFi sensing to track health [78], stress [118], and perform retail analytics [119].

We build on this prior body of work and leverage a network-centric WiFi sensing approach to assist with contact tracing. Our key insight is that the *mobility of a user's phone is visible to the network* through the sequence of WiFi access point associations performed by the device as the user moves. This allows the network to determine the locations visited by the users' device and other co-located devices present at those locations by being associated with those APs. Thus, the approach passively senses devices as they move through the WiFi network. The key advantages of this approach over a client-centric approach are:

1. **No mass user adoption:** Our approach does not require users to opt-in or download an app as the wireless network will automatically "see" all connected devices (associated) to it at all times. This approach makes WiFiTrace much easier to deploy at scale.

2. **No active data collection on user devices:** Data is collected directly by the WiFi network without requiring any direct interaction with a user's device (unlike a client-centric approach). Specifically, our method relies on collecting "syslog" network events, SNMP reports, or RTLS events that are already

commonly used by many enterprise networks for performance and security monitoring.

3. **Single sensing modality:** A client-centric method that uses Bluetooth for proximity sensing must use a second sensing modality such as GPS for sensing device location. In contrast, WiFi sensing is single modality sensing determining both location (based on AP locations) and proximity (based on AP associations). It is important to note that GPS does not work well indoors while WiFi does.

Design Challenges:

Our network-centric approach is not without several challenges. Firstly, WiFi sensing only provides coarse-grain proximity measurements (e.g., users co-located within the range of an access point). This is unlike fine-grain Bluetooth sensing, which promises accuracy values of up to a few feet of the user. Thus, using coarse-grained proximity sensing could increase the number of false positives. To overcome this, WiFiTrace capitalizes on the co-location duration as an indicator of the risk of infection to investigate individuals at higher risk [1] as demonstrated in the evaluation Section 5.5.

Secondly, WiFi-based contact tracing is limited to areas with WiFi coverage that tend to be popular indoor spaces and bounded outdoor spaces. In contrast, Bluetooth sensing does not need a network to listen to other devices and work "anywhere". We acknowledge this limitation of our network-centric approach but note that this approach is still highly effective in environments that do provide WiFi. In particular, these effective environments include most university campuses and corporate offices – i.e., places where individuals spend a significant portion of their day. Table 5.1

[1]Traditional contact tracing can then process this candidate set to eliminate those who are not at risk

	Bluetooth	Passive WiFi
Architecture	Client-based	Network-based
Location sensing	GPS	AP-level or WiFi locationing
Proximity sensing	Bluetooth	AP-level co-locators
Distance sensing	Fine-grain	Coarse-grain
Proximity duration	Fine-grain	Fine-grain
Data collection	On-device	In-network
Target environment	Indoors or outdoors	Indoors, limited outdoors
Key technical hurdle	Mass adoption needed	Does not work outdoors
Privacy Issues	Yes (see [26])	Yes (See Sec 5.1.3)

Table 5.1: Comparison of Bluetooth as a client-centric vs. WiFi as a network-centric approach to contact tracing.

summarizes the comparison between a Bluetooth based client-centric approach and a WiFi-based network-centric approach.

5.1.3 Privacy and Ethical Considerations

Client-centric contact tracing apps raise important privacy concerns, especially for everyday users whose whereabouts would essentially be documented [26]. Network-centric WiFi-based sensing bears the same concern, especially since users are automatically and passively tracked when connected to the WiFi network. However, unlike most Bluetooth apps that are focused on assisting the population at large, our tool aims to assist case investigators, as our primary end-users, in performing the formal contact tracing procedure. To reduce this risk, several privacy safeguards can be put into practice.

1. **No direct access to user or device data:** Unlike client-centric contact tracing apps, users do not need to provide their information or change device permissions to provide apps access to device data. In our work, all WiFi network data such as MAC ID and username that can identify a user was hashed (e.g., using the SHA-2 hash) to maintain anonymity.

2. **Leveraging existing operational security standards:** WiFi network data is used by many IT departments for network maintenance and security surveillance. For example, our campus uses the same WiFi data used by WiFiTrace to track down compromised devices that may be responsible for internal DDOS attacks and identify student hackers who, most notably, might be attempting to change course grades. Additionally, in many regions, audit and compliance laws also necessitate gathering network logs for subsequent analysis and audits. These routine evaluations have operational security standards in place to protect user privacy. Using this WiFi data for contact tracing requires compliance with the same high operational security standards.

3. **Emergency disclosure request:** A larger outbreak of a disease such as Covid-19 will require de- anonymization of location histories of high-risk individuals. This information will strictly be shared with case investigators performing the contact tracing. A typical procedure for WiFiTrace will be to query only hashed identities and MAC addresses. Note that the hashed data is stored separately from the tool and only accessible to a small, trusted group. When an individual is identified at-risk, only an authorized case investigator can obtain a de-anonymized copy of the information.

4. **Obtain user consent:** Data protection acts enacted in many countries require organizations to acquire user consent before starting any data collection operations. Similarly, before WiFiTrace can be used in production, users must provide informed consent to contact tracing upon connection to WiFi in an enterprise network. Similarly, case investigators must also be authorized to retrieve de-anonymized information when necessary, and all data sharing must follow the approved guidelines. For example, a contact tracing team could decide to directly contact an individual at high risk of contracting Covid-19 or

publish at-risk locations like a public alert to appeal to potentially infected individuals to contact health authorities. In the latter case, the proximity data report is used for further contact tracing when at-risk individuals contact health officials. We currently use this latter approach at our USA campus.

Data Ethics & IRB Approval

Data collection for experimentally validating the efficacy of our approach has been approved by our Institutional Review Board (IRB). It is conducted under a Data Usage Agreement (DUA) with the campus network IT group that restricts and safeguards all WiFi data. To avoid private data leakage, all the MAC ids and usernames in the syslogs are anonymized using a strong hashing algorithm. The hashing is performed before syslog data is stored on disk under the campus IT manager's guidance, who is the only person aware of the hash key of the algorithm. Any data analysis that results in the users' de-anonymization is strictly prohibited by our IRB agreement and the signed DUA.

5.2 Network Centric Contact Tracing Approach

This section presents an overview of our approach, followed by the details of our graph-based contact tracing algorithm.

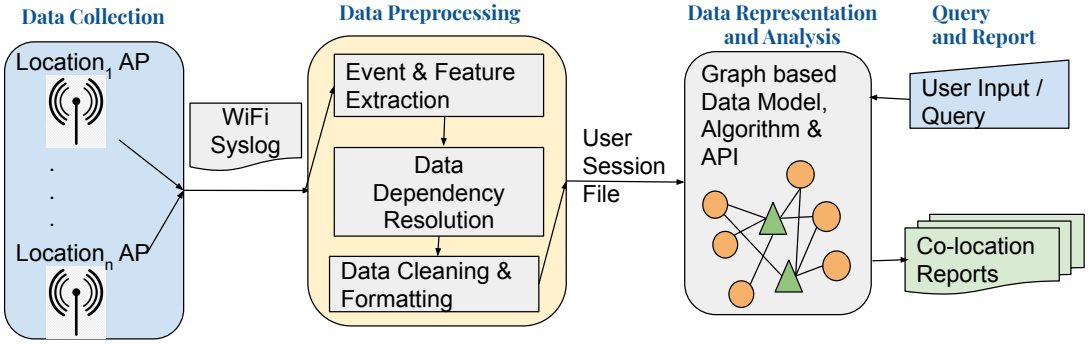


Figure 5.2: An overview of the tiered architecture used by our approach.

5.2.1 System Overview

Figure 5.2 shows that WiFiTrace uses a three-tier pipelined architecture. The data collection tier uses network logging capabilities that are already present in enterprise WiFi systems to collect the *WiFi logs of device associations to access points* within the network. Many enterprise IT administrators already collect this data for network monitoring, in which case this already-collected data can also be fed into WiFiTrace . Otherwise, the IT admins need to turn on WiFi logging to start gathering this data.

The next tier ingests this raw data and converts it into a standard, intermediate format – i.e., it performs pre-processing of the data. Since the raw log files will have vendor-specific formats, this tier implements vendor-specific pre-processing modules specific to each WiFi manufacturer and its logging format. This tier processes log files in batches every so often and generates data in intermediate form.

The final tier ingests the intermediate-form data produced by the vendor-specific pre-processor and creates a graph structure that captures the trajectories of all user devices. This tier also exposes a query interface for contact tracing. For each query, it uses the computed trajectories over the query duration to produce (i) a location report listing locations visited by the infected user and (2) a proximity report listing other users in the proximity and their respective durations. As discussed below, this tier uses efficient time-evolving graphs and algorithms to efficiently intersect the trajectories of a large number of devices (typically tens of thousands of users that may be present on a university campus) to produce its report.

5.2.2 Basic Contact Tracing Approach

Consider a WiFi network with N wireless access points that serves M users with D devices. We assume that the N access points are distributed across buildings and other key spaces in an academic or corporate campus and that the location of each access point (e.g., building, floor, room) is known. Large enterprises such as a

residential university will install thousands of access points to serve tens of thousands of users. For example, our work uses data from two large universities, one based in the Northeastern USA with 5500 access points and one based in Singapore with 13,000 access points. To manage such a large network, enterprise WiFi networks use controller nodes to administer and manage the APs and the network traffic and provide detailed logging and reporting capabilities.

As a user moves from one location to another, their mobile device (typically a phone) associates with a nearby access point. Since the locations of APs (building, floor, room) are known, the sequence of AP associations over the course of a day reveals the user's trajectory and visited locations. We extract these association events from the WiFi controller logs to reconstruct this trajectory. Typically this information is of the form: `t`imestamp, AP MAC address, Device MAC, optional user-id, event-type, where event-type can be one of association, disassociation, reassociation (when a device wakes up from sleep), un/authorization (the device is not authenticated). Typically when a device switches to a new AP due to user mobility, this is visible to the network in the form of a disassociation event with the previous AP and an association event with a new AP.

Given this log information, the process of contact tracing a particular user involves two steps:

1. determine all APs visited by the user in the specified time period and
2. determine all users who were associated with each of those APs concurrently with the infected user.

To do this, we analyze the WiFi logs to construct the time-ordered sequence of sessions that represent the time that the devices spend at each AP. A session is the time period between an association event and a subsequent disassociation event for

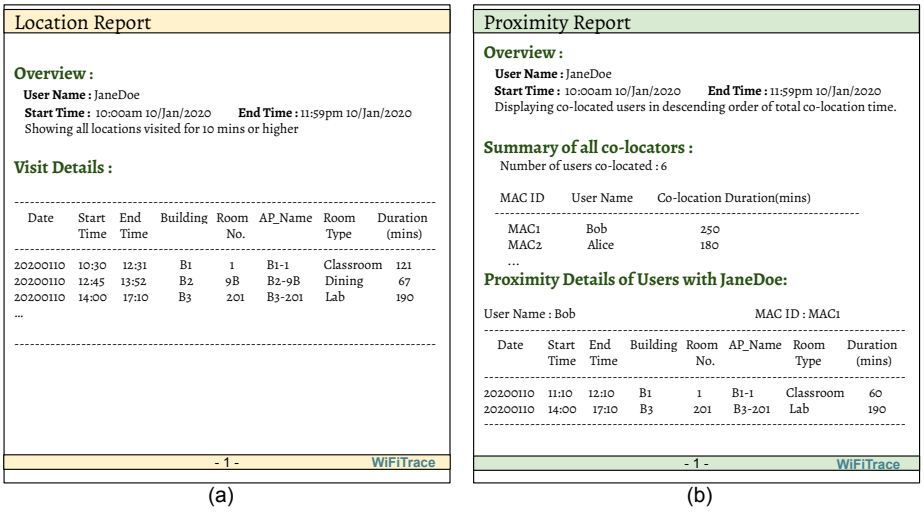


Figure 5.3: An example contact tracing report produced by WiFiTrace : (a) Patient Report (b) Proximity Report

that AP and device. Since AP locations are known, this sequence of sessions also represents the user's location.

Next, we analyze the log for each AP session to determine all other users who have overlapping sessions for the same AP. These are users (i.e., their devices) who were present in the infected user's proximity. Of course, the WiFi log does not reveal the distance between the two users or whether physical contact occurred. Nevertheless, it enables us to determine users at risk by computing the *duration* for which the two users were in proximity to one another. In some cases, the location where they were co-located may reveal the degree of risk (e.g., an hour-long meeting in a small conference room or a lecture classroom). To allow health professionals to assess the risk, we generate a location report, showing locations visited by the user, how long, and a proximity report of co-located users at each location and co-location duration. Figure 5.3 depicts a sample report resulting from the process.

One practical problem is that an enterprise network with thousands of APs and tens of thousands of devices will generate large log files. For example, one of our campuses' log file contains more than 9 billion events over a 4 month semester pe-

riod. Thus, scanning the log to compute the location and proximity can be slow and inefficient. To address this, we present an efficient graph-based algorithm based on time-evolving graphs in the next section.

5.2.3 Efficient Contact Tracing Using Graphs

To efficiently process contact tracing queries, we model the data as a bipartite graph between devices and APs. Each device in the WiFi log is modeled as a node in the graph; each AP is similarly modeled as a node. An edge between a device node and an AP node indicates that it was associated with that AP. The time interval annotates each edge (t_1, t_2) that denotes the start and end times of the association session between that device and the AP. Note that data is continuously logged to the log files, which causes new edges to be added to the graph as new associations are observed and new nodes to be added as new devices are observed in the logs. Thus, our bipartite graph is time-evolving.

For computational efficiency, each device and AP node in the graph is limited to a time duration, say an hour or a day. This is done to limit the number of edges on each node, which can keep growing as the device associates with new APs or APs see a new association session. As a result of associating a time duration with each node, *each* device or AP is represented by *multiple* nodes in the graph, one for each time duration where there is activity. In this case, we can view the node ID as the mac address concatenated by the time duration. For example, MAC_1[10:00,10:59], MAC_1[11:00,11:59], represent two nodes for the same device, each capturing AP association edges seen within that period. In the case of AP nodes, this would capture all device association to that AP within those time periods (see Figure 5.4). The duration for partitioning each node's activity in the graph is a configurable parameter. This duration can be chosen independently for a device node and an AP node if needed.

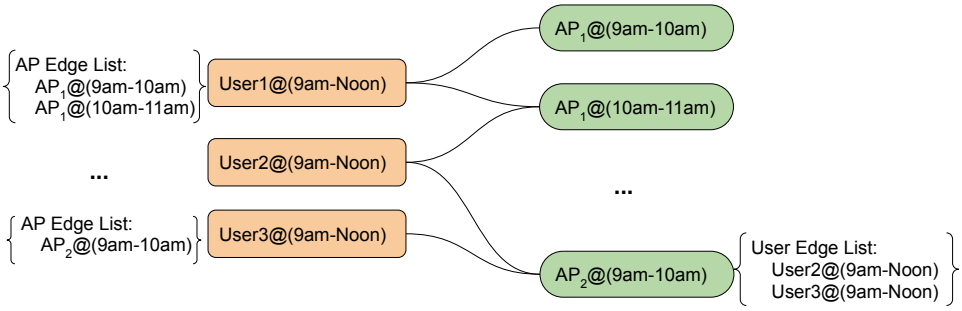


Figure 5.4: An example bipartite graph shows device to AP association and time-based partitioning of node activity.

Given such a bipartite graph, a contact tracing request is specified by providing a device MAC address and a duration (T^{start}, T^{end}) over which a contact trace report should be generated. The query also takes a threshold τ that specifies only AP sessions of duration longer than τ should be considered.

The graph algorithm first identifies all device nodes corresponding to this user that lies within the (T^{start}, T^{end}) interval and identifies all edges from these nodes. These edges represent all AP locations visited by the user, and the session duration represents the time spent at each location. Only edges with the following constraints are considered: (1) the session must lie within the query time interval, i.e., $[t_1, t_2] \in [start, T^{end}]$ and (2) the session duration must be at least τ, i.e., $(t_2 - t_1) \geq \tau$. Edges that do not satisfy either of the above criteria are ignored. The remaining edges are used to enumerate the AP locations visited by the device and the time duration spent at each location.

To compute the proximity report, the algorithm traverses each edge and examines the corresponding AP node. For each AP node, the list of incident edges corresponds to all devices that had active sessions with that AP. The session duration $[t_3, t_4]$ on each edge is compared to the infected user's session $[t_1, t_2]$, and the edge is included only of the two-session overlap. This process yields a list of all other users who had an overlapping session with the infected user. The algorithm can also take an optimal

parameter w that indicates the minimum overlap in session between the two for the user to be included in the proximity representation, i.e., $w \geq [t_1, t_2] \cap [t_3, t_4]$. The parameter w specifies the minimum duration of the co-location necessary for a user to be included in the proximity report. Algorithm 1 lists the pseudo-code for our graph algorithm. Thus, a time-evolving bipartite graph allows for efficient processing of contact tracing queries over a large dataset.

Algorithm 1 Contact Tracing

procedure CONTACTTRACING($Graph$, $UserID$, τ, w, T^{start}, T^{end})
devNodesList ← find all device nodes for UserID in interval [T^{start}, T^{end}]
for node in devNodesList **do**
Filter out all edges where session(t_1, t_2) $\notin$ [T^{start}, T^{end}] and (t_2 - t_1) $< \tau$
for each remaining edge **do**
Add edge.APLocation and edge Session(t_1, t_2) to list of locations
Add APnode corresponding edge to APnodesList
for `node in APnodesList` **do**
Filter edges where user session(t_1, t_2) doesn't overlap with colocator session (t_3, t_4)
and $w \geq [t_1, t_2] \cap [t_3, t_4]$
for each remaining edge **do**
Add user and device information corresponding to edge list of co-locators.

5.3 System Implementation

This section presents the implementation of WiFiTrace , and it is available as open-source code to researchers and organizations who wish to deploy it (source code is available at http://wifitrace.github.io).

5.3.1 Three-tier Architecture

From the three-tier architecture (see Figure 5.2), our system implementation occurs in the second and third-tier; the first tier is based on the logging capabilities already supported by enterprise-grade WiFi networks, and we currently support WiFi Access Points from Cisco and HP/Aruba, two large enterprise WiFi equipment vendors.

Service	Description
get_id	Return Node Id (MAC id for user node/Location id for location node)
add_neighbor	Add a location neighbor and init edge weight
get_connections	Return all visited locations
get_weight(location)	Return edge weight for a location
get_sessions(user)	List sessions for the user at current location node
get_name	Return non-time indexed location name
get_users	Return list of all users
get_location	Return list of all locations
add_user_node	Add a new user node
add_loc_node	Add a new location node
add_edge	Add a new edge in graph
get_user_node	Return user node from graph user node dictionary

Table 5.2: Graph APIs implemented by our graph-based data representation.

Data Preprocessing

We implemented preprocessing code for both Cisco and Aruba networks that takes raw monitoring data and converts it to a standard intermediate data format for our second tier. For HP/Aruba network, WiFiTrace supports the processing of both syslogs (generated by Aruba WiFi controllers) and RTLS logs generated by Aruba APs. Both types of logs provide association and disassociation information. In the case of Aruba RTLS, we log WiFi data directly from the controller nodes using either real-time location services (RTLS) APIs [83]. For Aruba syslogs, we periodically copy the raw syslogs generated by the controller and preprocess this raw data. Finally, for Cisco networks, we log WiFi data directly from the network using the Cisco Connected Devices (CMX) Location API v3 [99]. All of these preprocessor scripts convert raw logs into the following standard record format:

```
Timestamp, AP Name or Id, Device MAC Id, event type, (optional) User Name
```

By default, we assume anonymized (or hashed) device MACs and usernames. We also assume a separate secure file containing a mapping of real names to hashes. While

the association, disassociation, reassociation, and drift messages from the syslog give us spatio-temporal details about the various devices on the network, the authorization events provide us with details about MAC ID and username, aiding us to create a device-user mapping. The device-user mapping helps us count each user once by considering only the highly mobile user device among multiple user-owned devices. The identified highly mobile device is most commonly a smartphone because it gets carried around by users everywhere.

Graph Representation and Analysis

Our third tier supports contact trace querying. A query is of the form (hashed) username or device MAC, start duration, end duration, threshold τ, and co-locator threshold w. Internally the data generated by the pre-processing code is represented as a bipartite graph, as discussed in Section 5.2.3. Our system supports various queries on this graph through a graph API depicted in Table 5.2. This graph API is used to implement the graph algorithm described in Section 5.2.3.

The algorithm yields a *location report*, which shows all locations (APs) visited by the user for longer than τ and a *proximity report* that shows all users who were connected to those APs for a duration greater than w. Figure 5.3 shows a sample location and proximity report generated by our system.

In addition to human-readable query reports, our system can optionally output query results in JSON format, convenient for visualization or subsequent processing. Our system also supports additional report types beyond location and proximity reports. For example, it can produce reports of additional users who visit a location *after* the infected user has departed from that location. This is useful when a location has high-contact surfaces that may transmit a contagious disease even after an infected user departs. Such a report can be produced by specifying a window parameter that

specifies the time other users are identified as at-risk at each location after they depart.

5.4 Deployment and Validation

In this section, we describe our deployment and validation results for contact tracing.

5.4.1 Deployment

We deployed WiFiTrace on a university campus in the northeastern USA and one in Singapore. Both campuses have large WiFi networks, one with 5500 HP/Aruba APs and a mixed Cisco/Aruba network of 13,000 APs. While we had originally developed this tool to fight the outbreak of meningitis on our campus, university health officials from both campuses viewed our solution as a scalable contact tracing method for Covid-19.

Even though WiFiTrace has been operational for several months now, as of May 2020, neither campus has experienced a situation that requires using WiFiTrace . This was due to strict lockdown measures with both residential universities switching to online learning. Students were instructed to vacate their dorms, and work-from-home policies were enforced. Except for a small number of students unable to return to their home countries (due to global lockdown), both campuses have been largely empty until recently (Sep 2020 onwards) when the Singapore campus started letting more students on-site. One of our campuses saw a single case of an employee at a high-risk of Covid-19. However, manual contact tracing had determined the case as a low threat to others. As such, university health professionals did not need to perform further contact tracing.

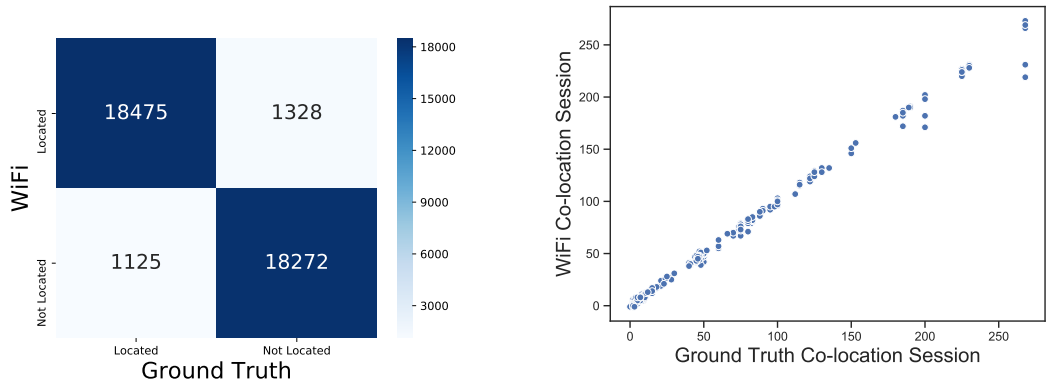
5.4.2 Validation

We conducted a small-scale user study to gather ground truth data to validate four questions related to the use of passive WiFi sensing for contact tracing:

- V1: How accurate are WiFi access point associations at revealing the true user location?
- V2: How accurate are WiFi session durations at revealing the true duration a user spent at a location?
- V3: Do co-located WiFi device sessions accurately show co-located users at those locations?
- V4: How do AP density, phone activity, user movement type, and phone make/model/OS impact device registration events seen by the WiFi network? In particular, do these factors result in a temporal lag in the extracted trajectories?

Dataset

We collected data from a group of eighteen volunteers over a period of 37 days with each volunteer walking to various locations, and spending variable lengths of time at each location, around our campus while carrying their mobile devices. Each user manually logged the entry and exit times at each location, and the paths they took from one location to the next. For each user, we computed a location report containing all visited locations (assuming a threshold $\tau = 0$) and compared the locations as seen by the WiFi network to the ground truth locations recorded by each user. Some of the user's devices' trajectories were correlated, which meant the users were co-located whenever (their devices were) connected to the same AP concurrently. Overall, about 19,000 unique locations were visited by our volunteers.



(a) (b)

Figure 5.5: (a) Confusion Matrix (b) Scatter Plot displaying ground truth and WiFi based session duration

Results

To answer V1, we compared the locations reported by WiFi with the actual ground truth location recorded manually by each volunteer. Figure 5.5(a) shows the confusion matrix for this comparison. Specifically, an overlap in WiFi and ground truth ***located*** labels indicate the inferred location from a WiFi AP of a user matches their logged location – true positive. An agreement in WiFi and ground truth ***non-located*** labels imply a user was, in fact, not physically situated where the WiFi AP had not inferred them to be – true negative. Similarly, two mismatch possibilities can occur. The first is the WiFi had missed inferring the user location to where a user was actually localized (WiFi ***non-located*** and ground truth ***located***) – false negative. The second is that WiFi had inferred the user to be situated in a spot the user was not in (WiFi ***located*** and ground truth ***non-located***) – false positive. Both errors could arise due to temporal delay in AP hopping or skipping, where the user's device had maintained a previous AP connection, despite the user already transitioning to a different spot.

Overall, WiFiTrace yields a precision of 0.93, a recall of 0.94, and a high F1-score of 0.93. In other words, the inferred location matches the ground truth location with high accuracy. A deeper analysis of the errors shows that they mainly occur when a user transitions between different locations quickly – on the order of tens of seconds.

Even in these cases, the true location of these quickly moving users is usually just off by one AP. Figure 5.5(b) shows a scatter plot of the session durations reported by WiFiTrace compared to the ground truth and demonstrates a good match between WiFiTrace and the ground truth. The small errors in the figure occur when devices enter or exit the area as these devices will need additional time to switch networks.

To answer V2, Figure 5.6 plots the inferred location's accuracy for varying session lengths observed across four different mobile devices made by Apple, Samsung, Motorola, and LG. Our tool yields 100% accuracy whenever a session length exceeds 3 minutes, which is sufficient accuracy for a contact tracing application.

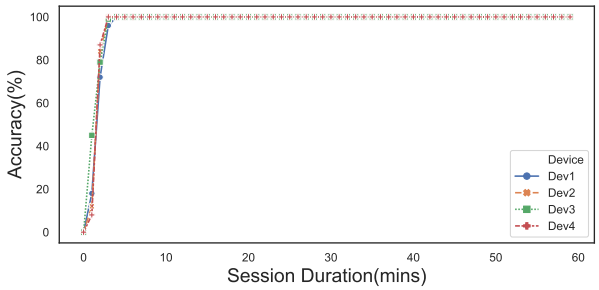


Figure 5.6: Accuracy of inferring user locations for varying WiFi session duration.

Next, to answer V3, we validated the accuracy of co-locations by using WiFiTrace to generate the proximity report for each device and comparing it to the ground truth trajectories reported for each device. WiFiTrace can capture co-located devices (and users) with high accuracy for sessions exceeding 3 minutes, as shown in Figure 5.6.

However, as noted earlier, if a user moves fast (i.e., there are short transitions), the locations can be off by one AP cell. This implies that the network can see two fast-moving devices near one another connected to adjacent APs, rather than the same one. Fortunately, this does not hamper the efficacy of contact tracing as two users need to be near one another for a period of time (e.g., 15 minutes or more) to be considered at-risk. WiFiTrace accurately capture these longer co-located times.

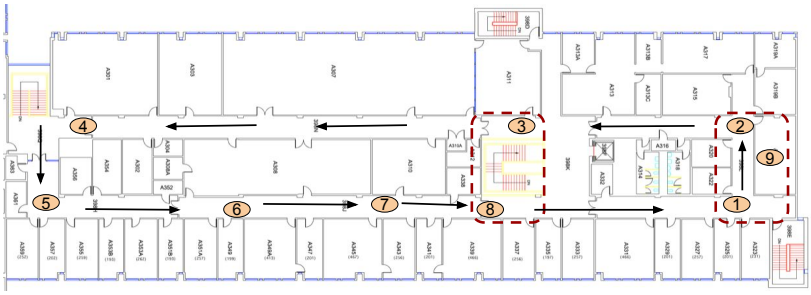


Figure 5.7: Floor Map with AP locations

Finally, to answer V4, we measured the time lag in WiFi logs extracted against ground truth manual trajectory logs. This experiment used four different smartphones (Apple, Samsung, Motorola, and LG) that were streaming a YouTube video continuously (i.e., they were always active on the WiFi network) and traveling through the same paths and floors. Figure 5.7 shows the single floor route covered by 9 APs. The path chosen moved from AP1 to AP2, AP3, etc., and eventually circled back to AP1. AP9 was never exclusively visited, as shown in the ground truth trajectory in figure 5.8. The floor was intentionally selected for having both a sparse and dense AP dispersion. We repeated this experiment using 5 different transition styles: walking casually, hustling through several locations, and being in stationary spots between 1, 5, and 20 minutes.

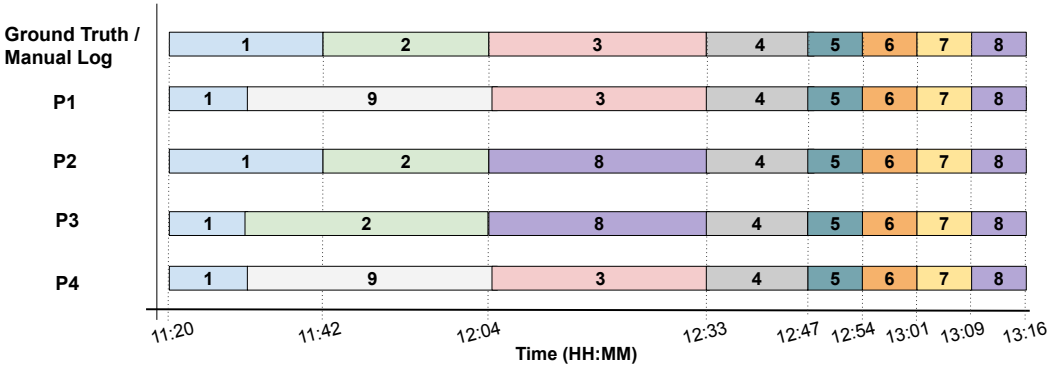


Figure 5.8: Temporal lag between computed WiFi Trajectories of active Android Phones of different OS versions as compared to Ground Truth User Trajectory.

Figure 5.8 shows the trajectory connections across APs for four devices P1 to P4, that moved from AP1 through A9 and were stationary at each AP for 5 to 20 minutes. For example, as a user with device P1 moved from AP1 through AP8 along the same path as indicated by the ground truth trajectory. However, device P1 was first connected to AP1 for a short duration and then connected to AP9 even though the user moved to AP 2. P1 then connected correctly to AP 3, 4, 5, etc., as the user moved across the path. In contrast, P2 was connected to AP1 and 2 but connected to AP8 instead of AP3, even though it followed the same path as P1. We believe this was because AP3 and AP8 were on opposite sides of a large open area, and thus the signal from AP8 was still strong even near AP3. P3 and P4 encountered similar AP connectivity with AP2/9 and AP3/8, respectively.

To rectify the errors due to devices connecting to nearby APs in open areas or areas where there are many APs, we can create zones as indicated by the boxes in figure 5.7. For example, APs 1, 2, and 9 form a zone Z1 while APs 3 and 8 form another zone Z2. This grouping of APs into logical zones helps eliminate the errors shown above caused by devices connecting to nearby APs instead of the closest AP.

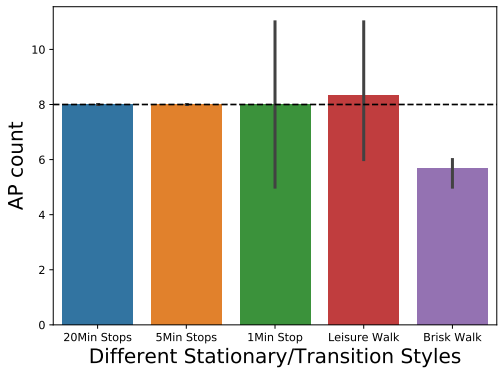


Figure 5.9: Number of AP Hops encountered for various trajectory styles with varying stop durations for different models, make and OS versions of Android Mobile Phones

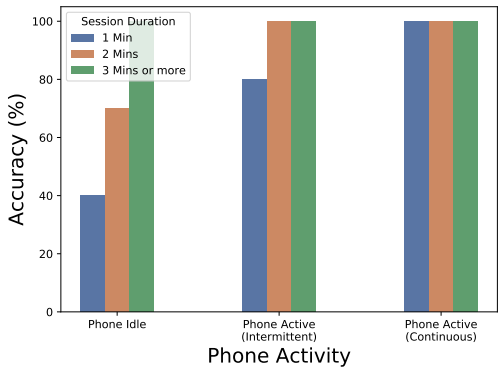


Figure 5.10: WiFi location Sensitivity for various phone activity scenarios across various session durations as computed across various phone models, makes, and OS (iOS and Android)

Figure 5.9 plots the number of AP hops due to different transition styles. Overall, each device experienced 8 AP hops when moving from AP1 back through AP8 using different transition styles. We observed more variance in AP hops when a user was moving continuously – i.e., without making stops. For example, a user walking leisurely through the planned path recorded between 6-11 AP hops, while a brisk walk recorded 5 AP hops on average. This result shows that when no/short stationary stops are made, there is a higher tendency for AP scans to increase.

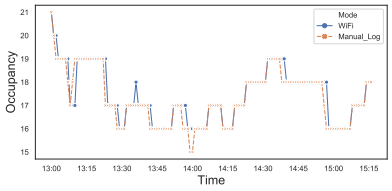


Figure 5.11: Comparison of the number of users co-located at an access point with the ground truth.

Item	US Univ.	Singapore Univ.
Users	≈ 38000	≈ 50000
AP	≈ 5500	≈ 13000
Buildings	230	240
Time Span	Jan-May 2020	Feb-May 2020

Table 5.3: Dataset Characteristics across the two Universities (Univ.)

We measured the accuracy of the spatio-temporal values from the WiFi logs when phones were engaging in different activities such as idling, intermittent use (e.g., occasionally responding to messages), and continuous use (e.g., video streaming). Figure 5.10 shows that the accuracy when the phone was idling was the worst, while continuous phone usage had the best accuracy. Specifically, we find that the temporal accuracy values for phone idling are approximately 40% for sessions less than a minute, 75% for sessions between 1-2 minutes, and 100% for session durations more than 3 minutes. On the other hand, continuous phone use guarantees device associations with AP changes indicating that the user's presence will be recorded. User sessions less than a minute for intermittent use maintains an accuracy of at least 80%. In the most practical use-case, we expect that users transitioning around campus would be engaged in intermittent phone use as they periodically reply to messages or emails. In the worst case, WiFiTrace still maintains high accuracy even when the phone is

idling for at least 3 minutes long stationary periods at a location. Thus, WiFiTrace is still useful as a contact tracing application for use cases, such as Covid-19, where individuals need to spend at least a 15-minute overlap time over a 24 hour window period with an infected person [40] for the infection to spread.

Finally, we count the number of users entering and leaving library rooms and compare it to the number of devices (users) reported by our tool. As shown in Figure 5.11, the automatic count generated by WiFiTrace closely mimics the ground truth. The small mismatches occur due to short WiFi sessions (implying a user is present only for a brief period or when their devices did not switch to a new AP from a previous one). The user counts remain accurate for all sessions that exceed a few minutes as their devices will eventually switch to the closest AP.

Overall, our user scale study shows that WiFiTrace can answer questions V1 to V4 with sufficient accuracy to support its use as a contact tracing application.

5.5 Experimental Evaluation

In this section, we describe case studies that evaluate our contact tracing tool and our graph algorithms. Further, we discuss the limitations of utilizing WiFi sensing.

5.5.1 Dataset and Methodology

To validate WiFiTrace , we used the production WiFi logs from our university WiFi network. Table 5.3 summarizes the characteristics of the WiFi logs. Our US university uses an Aruba network of 5,500 APs deployed across 230 buildings. It has approximately 38,000 users comprising 30,000 students and 8,000 facultystaff. The dataset spans between Jan 2020 to May 2020, including a full campus-wide Covid-19 lockdown during Spring break (mid-March).

The APs are spread across the US campus with a mean of 23 APs per building. We found that 90% of the buildings have up to 45 APs, and few tall buildings (dorms

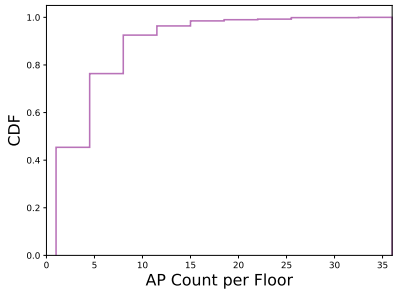


Figure 5.12: Distribution of APs per floor

and library) with a high number of floors have more than 100 APs installed inside. As shown in Figure 5.12 an average of 6 APs is installed per floor.

We also collected data from a Singapore university that has a mixed Aruba and Cisco network comprising 13,000 APs deployed across 240 buildings. It has approximately 50,000 users comprising 40,000 students and 10,000 faculty/staff. The Singapore dataset spans between Feb 2020 to May 2020 and includes a Covid-19 quarantine plan, which was progressively introduced by the Singaporean government, ending with a full lockdown similar to the US University.

5.5.2 Case Study

5.5.2.1 Efficacy of Contact Tracing

Since a real disease outbreak is yet to occur on either campus, we emulate how our tool works under emulated diseases. We pick three diseases, each with a different incubation period, which would require contact tracing for a different number of days.

- Seasonal Influenza: Incubation period $\approx$ 1-3 days; contact tracing 2 days
- Covid-19: Incubation period $\approx$ 4 days, contact tracing 4 days
- Measles: Incubation period $\approx$ 8-12 days, contact tracing 10 days

We randomly choose a user from our dataset and assume they are infected with one of the above diseases and use our tool to compute the number of locations visited

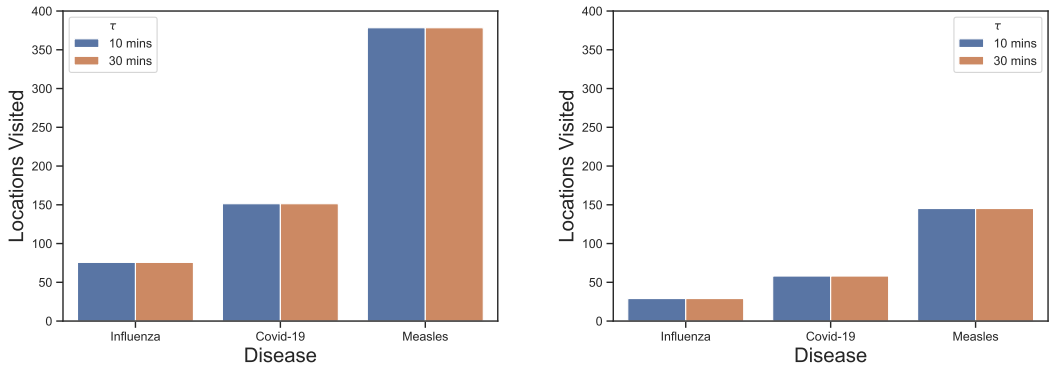


Figure 5.13: Cumulative location count for various diseases for τ (10minutes, and 30minutes) for (a) Student User (b) Non-Teaching Faculty Staff

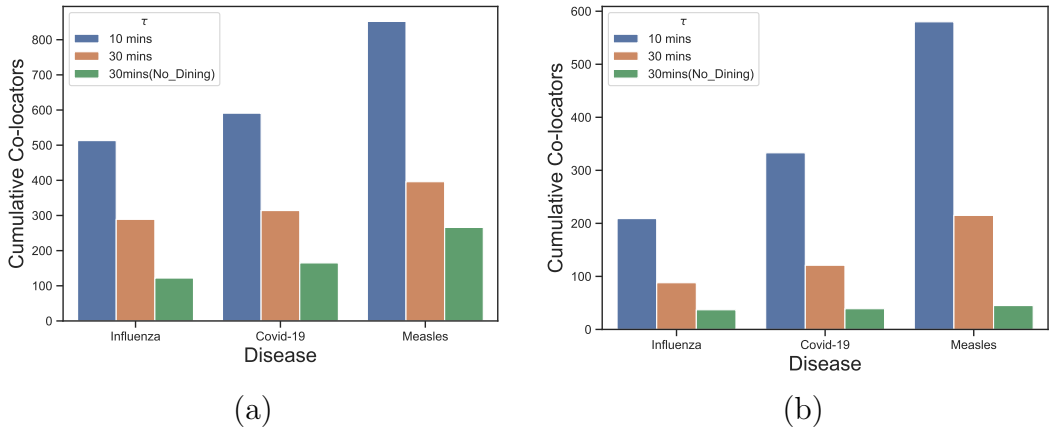


Figure 5.14: Cumulative Co-locator count for various diseases for τ (10minutes, 30minutes and 30minutes excluding dining) for (a) Student User (b) Non-Teaching Faculty Staff

by the user over that period and the number of co-located users. We perform contact tracing assuming $\tau = \omega = 10$ minutes and $\tau = \omega = 30$ minutes, which implies location visited for at least 10 (or 30) minutes and co-location of at least 10 (or 30) minutes. For each disease, we repeat each contact tracing experiment for 50 randomly selected students, and then 50 randomly chosen faculty or staff users.

Figure 5.13 shows how the number of locations visited by an infected user grows as the duration of contact tracing increases between 2 - 10 days (for Influenza and Measles, respectively). We find that the number of location visits is insensitive to τ

beyond $\tau > 10$ minutes (we discuss this in detail in the next section). A student visits ≈ 37 locations, while a faculty/staff user is less likely to transition around campus and visits ≈ 15 locations per day.

Figure 5.14 depicts the proximity results from our contact tracing experiment. With $\tau = 10$ min, the system yields many co-located users, specifically 500 co-located users for Flu over a 2-day period and over 800 users for Measles over a 10-day period for a student. With $\tau = 30$ min, the number of co-located users decreases slightly to 300 users for flu and 400 users for Measles. In contrast, the co-location count is lower for staff users, when $\tau = 10$ minutes, but projects to affect about 200 users (during a Flu) and 500 users (during a Measles outbreak). $\tau = 30$ minutes will substantially lower the numbers to 100 and 200, respectively. These results yield the following insights:

1. We note that the number of co-locators does not increase linearly with an increase in contact trace duration. The growth is *sublinear*, indicating that users have a limited social circle (of users), and these interactions last over several days.

2. It is infeasible to contact trace several hundred users for each infected user manually. Instead, WiFiTrace can address this problem by setting the parameters τ and ω carefully, balancing the duration of locations and co-locations based on the disease at hand. For example, $\tau = 10$ min could result in a high rate of chance co-location. Choosing $\tau = 30$ minutes and ω be 15 or 30 minutes may yield better results. The tool subsequently outputs a manageable number of cases for manual contact tracing investigation.

Further, our results show that common areas such as cafeterias substantially increase the co-location counts. Accordingly, it is easier to filter out these AP sessions to determine users with higher risk. As shown in Figure 5.14 (a) and (b), the num-

ber of co-locators drops considerably once cafeteria visits are excluded. Given these representations, WiFiTrace will produce a *proximity report* (see Figure 5.3), summarizing the total time spent with the co-locator and the respective locations in sorted order. In a practical use-case, case investigators can consider the top N users (e.g., N = 15) with the most proximity minutes or consider specific locations suspected of at high-risk. Such strategies are already used by professional contact tracers to hone in on the most probable at-risk co-locators while eliminating users who may be false positives.

5.5.2.2 Efficacy of Iterative Contact Tracing

While the above experiment involved a single level of contact tracing, contact tracing is, in most cases, an iterative process, with each co-locator subjected to an investigation. Given that a user may come in contact with more than a hundred users in a single day (e.g., if a user attends a lecture and then visits the cafeteria), tracing for only two locations can be limiting.

We had explained how the co-locators list needs to be pruned at each step to identify the users at most risk. We had also suggested using a carefully chosen τ and ω to filter out low-risk locations. These strategies may be susceptible to missing some "true positive" cases. An alternative is to "test and trace", which combines testing with contact tracing - a strategy used by many countries for Covid-19. In this case, each co-located user is administered a test to verify their infection medically. Accordingly, only verified infected cases are subjected to iterative contact tracing while excluding others.

Contact Tracing	Round 1	Round 2	Round 3	Round 4
Selective	275	438	764	1752

Table 5.4: Count of co-located Users by "test and trace" strategy

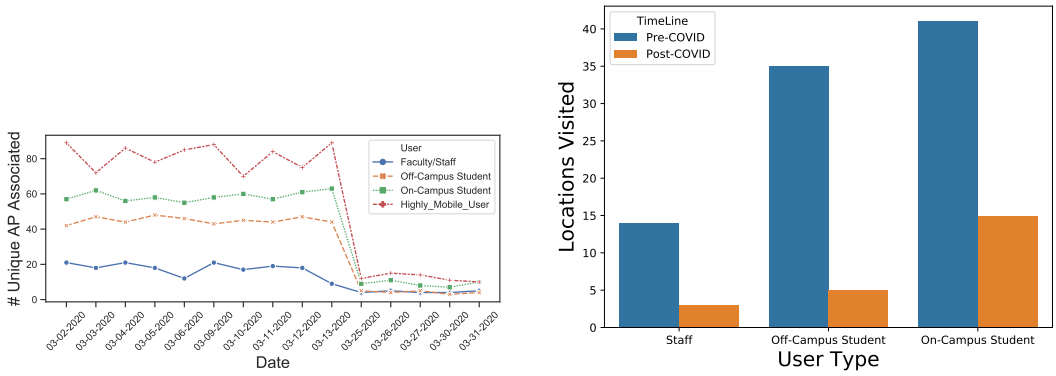
With "test and trace", the number of users subject to contact tracing increases based on the transmission rate, R. For example, if R=2, then only 2 out of the several tens of users identified by our tool will be subjected to additional tracing in each step (we assume that all users are tested to find R users who are infected). Table 5.4 depicts the number of users identified by "test and trace"; we observe the growth is much lower than a naive iterative strategy.

With testing constrained by availability, cost, and time, the proximity report can once again be utilized to *"prioritize"* and *"filter"*. We prioritize by sorting the co-locators based on the amount of overlapped time duration with an infected patient and picking the top co-locators. Then, we filter to identify the users co-located with the patient for a duration of at least ω minutes. Responding to Covid-19 CDC guidelines [40], ω value is 15 minutes for the infection to spread. This procedure is performed iteratively until all high-risk co-locators are traced. Another capability is to identify at-risk co-locators by considering the number of at-risk users exposed to and exposure duration. Case investigators can carry forward these shortlisted users for manual investigation.

5.5.2.3 Contact Tracing during Quarantine Periods

We had presented experiments during the pre-Covid semester, observing routine mobility among students and staff. Here, we examine how WiFiTrace's contact tracing results changes with strict lockdown policies.

Figure 5.15(a) shows the number of locations visited by different campus user types per day. While users visited between 20-80 locations for $\tau = 10$ minutes during the normal period, we observe a sharp drop in AP location visits for all user types due to lockdown policies (after March 25^{th}). The change in mobility will significantly alter WiFiTrace 's contact tracing results for infected during the lockdown period. Specifically, Figure 5.15(b) shows the number of locations visited by one user ruled



(a) (b)

Figure 5.15: (a) Number of Locations visited pre-covid and post-covid by 4 different user types (b) WiFi based location count for $\tau = 10$ for different user types pre-COVID19 and post-COVID19

for Covid-19 contact tracing (duration of 4 days). As shown, the number of locations visited varies from 5 to 20 for $\tau = 10$, and it drops to 1-6 locations visits for $\tau = 30$ minutes (or greater).

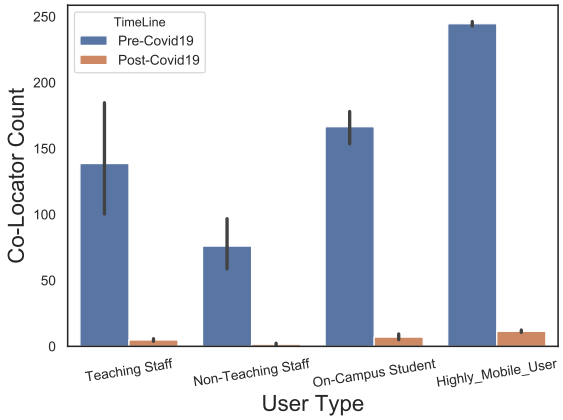


Table 5.5: Number of Co-locators pre-covid and post-covid for each of the 4 different user types $\tau = 30$minutes and $\omega = 30$minutes

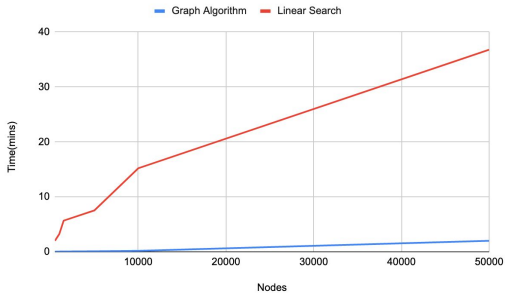


Table 5.6: Efficiency of our graph algorithm

Figure 5.5 depicts the number of co-locators for $\tau = 30$ minutes for several users during pre-Covid and lockdown periods. The safety policies introduced, primarily social distancing and lockdown, have lowered the co-locator count to be less than ten

for all user types, an order of magnitude reduction. In such cases, *comprehensive* contract tracing of all co-locators is feasible through manual means.

5.5.3 Efficacy of our graph algorithm

To evaluate the efficiency of our graph algorithm, we compare the execution time of the naive linear search approach and our graph-based algorithm across varying co-locators. Since different users display a different mobility pattern, the number of co-locators seen for each user will differ. Searching the co-locators using linear search requires a complete scan of the entire dataset sequentially, resulting in a high overhead across all runs irrespective of the number of observed co-locators of the device. Additionally, as the number of nodes increase, the search overhead increases. In contrast, our graph algorithm efficiently identifies relevant edges and nodes relevant to the specified query, thereby reducing the search space overhead. Also, adding the constraint of τ results in further pruning of edges, resulting in reduced search space, reducing our algorithm's time and space complexity. This behavior is depicted in Figure 5.6 that compares the execution overhead of the two approaches for our campus dataset. As shown, our graph-based implementation outperforms the naive sequential search by a significant margin.

We further evaluated the graph-based algorithm with the performance of a PostgreSQL database with a single column and multiple column indexes. For the PostgreSQL database with a single index, we found the query response time for a various number of users (ranging from 5000 to 30000) across time spans of a few days to a month was similar to our solution. However, the memory overhead of the database was $\approx$ 4.5x higher than WiFiTrace .

As we update the PostgreSQL database with daily logs to add device trajectories, the time needed to update the database indexes is in the order of hrs ($\approx$1 to 5 for data from 5000 to 30000 users). Since indexes are synchronized with the tables, updating

the indexes blocks further updates. We also found that indexing created an additional memory overhead of at least 250MB for a simple datastore of 20GB comprising of 5000 user trajectories for 4 weeks. This indexing overhead would only grow as the number of users and locations increases.

We found that using multi-column indexing of the MAC and location or the MAC and date column produced higher overheads in terms of the index update time and the index memory consumption. This suggests that multi-column indexing is costlier than the single-column index. Overall, our Graph-based algorithm provides query performance similar to an indexed database without any indexing update or memory overheads.

5.5.4 Limits and Limitations of Technology

WiFi-Sensing has well-known limitations, and this section analyzes the implications of these limitations on contact tracing.

Multi-device Users : Researchers have previously studied multi-device users' behavior and shown that it is prevalent among users to own two or more devices [104]. A key consequence is *device count seen by an AP does not equal user count.* While all WiFi logs record device association information, not all of them provide user ownership information. If such information is missing, RADIUS authentication logs should be used to map devices to owners to avoid double counting devices as separate users.

Figure 5.16 shows the number of unique devices seen by APs in different campus buildings and the corresponding user count (e.g., ARUBA syslogs provide both types of information). As shown, locations such as dorms and classrooms see between 1.5x to 2x difference in unique devices and unique users (since users may connect a phone and a laptop to the network). Only dining areas (cafeteria) see low over counting since users are likely to carry only their phone when eating. This result highlights

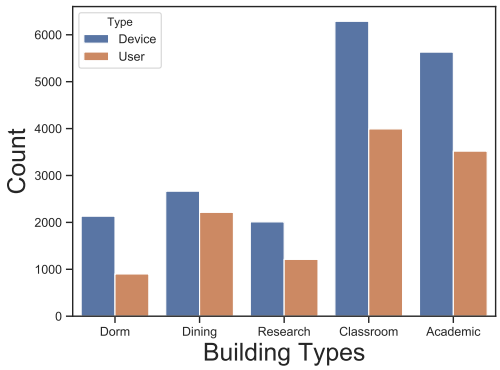
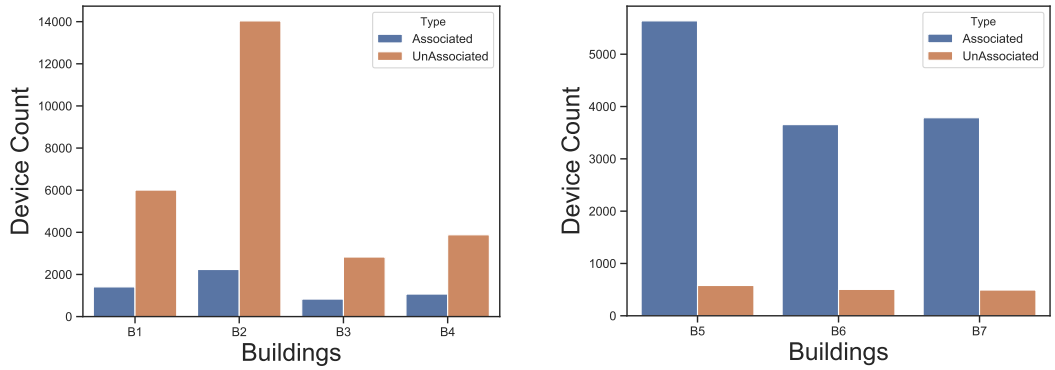


Figure 5.16: Unique Device and Unique User counts across buildings on a typical weekday. Need for user of device mapping information

the importance of considering device ownership to avoid over counting users by only considering connected devices.

Unassociated Devices : Not all users may connect their mobile devices to the WiFi network. Such devices are visible to the network when they perform SSID scans using a randomized MAC address. Unassociated devices can cause multiple challenges. Ignoring them altogether will result in under-counting users. Inversely, counting all devices can yield a large number of false positives.



(a) (b)

Figure 5.17: Associated devices v/s UnAssociated devices (a) Unfiltered and (b) Filtered

Figure 5.17(a) represents the number of unassociated devices seen in four buildings in our Singapore campus. Since the buildings are next to a public road or public bus stop, the number of unassociated devices per day is 5x greater than the number of associated users. Figure 5.17(b) shows that enforcing a session duration of 15 minutes filters out most of these *chance* associations, and the number of such devices (likely visitors) is around 12% of the total number of associated devices.

Impact of Session Duration: WiFiTrace uses two parameters τ and ω that are directly related to WiFi session durations. Judicious choice of these parameters can allow for a good tradeoff between eliminating false positives and eliminating true positives.

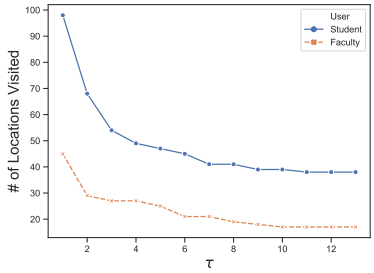
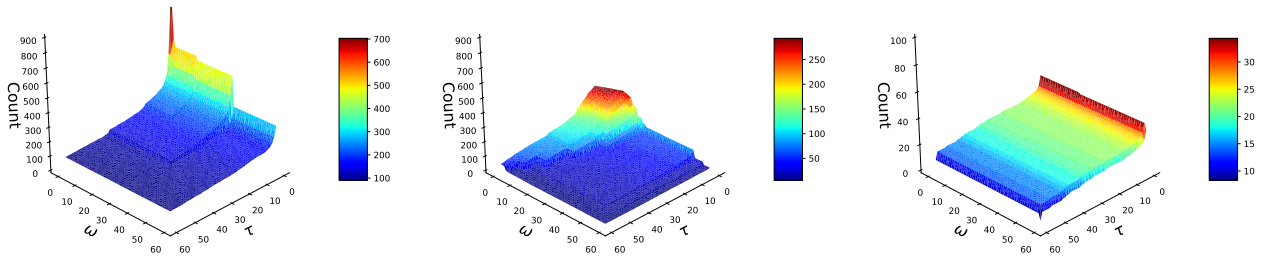


Figure 5.18: Count of Locations Visited by Student and FacultyStaff for varying values of τ

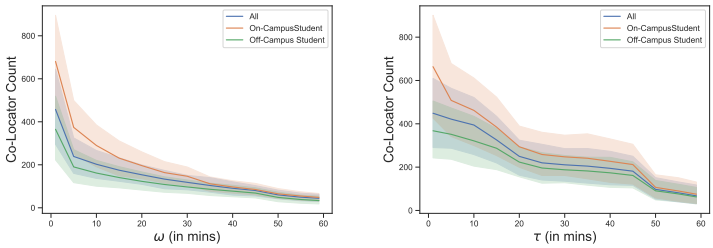
Figure 5.18 shows the number of AP locations visited by campus users for varying values of session length τ. The figure shows that the location visits stabilize around $\tau = 10$ minutes and then yield 20-40 location visits per day. Small values of τ include locations visited when in transit and should be ignored.

Figure 5.19 shows the impact of varying values of τ and ω, and the figure shows a decreasing gradient as both τ and ω are increased for all user types. Finally Figure 5.20 shows the number of colocated users for varying values of τ and ω. As shown, using values that are tens of minutes allows the tool to filter out overlapping sessions caused by users in transit. These results highlight the importance of carefully choosing τ and ω depending on the infectious nature of the disease and avoiding false positives.



(a) (b) (c)

Figure 5.19: Co-locator count for varying values of τ and ω (a) Student (b) Teaching Faculty Staff (c) Non-Teaching Faculty Staff



(a) (b)

Figure 5.20: (a) Co-locator count sliced with fixed value of ω (b) Sliced with fixed value of τ

5.6 Discussions

Here we discuss the implications and limitations of our findings.

5.6.1 Supporting Case Investigations

At its core, contact tracing remains a labor-intensive manual task performed by qualified caseworkers to determine infected patients and disease transmission chains in the community. WiFiTrace does not aim to replace caseworkers but instead aims to assist them by providing accurate information on possible infected individuals, locations, and contact events automatically – without required erroneous and time-consuming interviews. WiFiTrace was developed in collaboration with key healthcare professionals, including the director of a campus hospital and a nurse who has been running contact tracing operations for various outbreaks (such as meningitis) for

many years. The tool has been carefully designed with their input to fit into their workflow. Collectively, our efforts are centered on fulfilling two key requirements for a digital contact tracing tool, as defined by the Center for Disease Control (CDC) [39]:

1. **Aid the process of retrieving past location information from infected individuals:** In most cases, the use of GPS logs helps individuals accurately recall their daily activities' outdoor locations. However, GPS is unavailable indoors, and remembering all possible indoor locations visited is vital since the risk of spreading infections indoors is 20 times higher than outdoors [86]. WiFiTrace greatly assists with the recall process by providing accurate proximity locations of all users, at any location (either indoors or outdoors) where WiFi is available, without requiring additional environmental instrumentation or explicit user interaction.

2. **Improve investigators' efficiency in identifying and assessing the most relevant sources of threat:** The workload on health professionals increases dramatically during pandemic times, thus reducing their contact tracing burden is essential to prevent errors due to overwork. WiFiTrace achieves this by prioritizing highly threatening cases. It allows caseworkers to easily tweak key settings to change the thresholds used to determine the various threat levels. In particular, the thresholds specify how long a person needs to have spent in either a high-risk area or near an infected person before they are classified as "at-risk".

5.6.2 Limitations and Future Work

WiFi network events fundamentally drive our location proximity mechanisms. This constrains our solution to areas with WiFi coverage – most likely to be indoor spaces and a few key outdoor spaces. Enhancing WiFiTrace to use other data sources

(e.g., Bluetooth or GPS) will provide a richer source of location histories, but at the possible expense of user privacy and increased setup costs (to install an app to collect GPS or Bluetooth scans, etc.).

Another limitation is the inability to measure the physical distance between nearby users in proximity. WiFiTrace can discover individuals who are likely to be at risk because they were in a compromised area. But, it cannot determine if any specific set of individuals interacted with each other.

However, even this level of coarser interaction analysis is still useful as it provides an excellent first-cut of individuals who are at risk of infection. For example, in the case of Covid-19, researchers have found the virus's airborne transmission to remain viable in the air for three hours [105]. Hence, WiFiTrace can detect at-risk individuals who have been in a compromised area for too long. This information can then be used by case investigators to conduct follow-up and monitoring sessions to determine the exact severity of risk for each identified individual more accurately.

Beyond our experimental validation, more work needs to be done to identify how WiFiTrace can support routine clinical health investigations for Covid-19 and other diseases on other campuses and in other countries. We are currently expanding our efforts to deploy WiFiTrace to more campuses, and we welcome collaboration with any interested parties wanting to use WiFiTrace on their campus or environment.

5.7 Related Work

The prevalence of many infectious diseases in our society has increased the importance of contact tracing–the process of identifying people who may have come in contact with an infected person– for reducing its spread and disease containment [72, 96]. For performing contact tracing, the infected user needs to provide the places visited and persons in proximity or direct contact [33]. While the traditional method

relies on interviews, the Covid-19 pandemic has seen using a method such as GPS, Bluetooth [6], credit card records [30], and cellular localization.

Manual contact tracing as a mode for containment of diseases with a high transmission rate has proved to be too slow and cannot be scaled. Research [85, 38, 36] has shown that technology-aided contact tracing can aid reduce the disease transmission rate by quicker scalable tracing and help achieve quicker disease suppression.

5.7.1 BlueTooth Based Contact Tracing

Bluetooth and Bluetooth Low Energy (BLE) based contact tracing has emerged as a possible method for proximity detection [88]. A handful of systems based on Bluetooth or BLE have been rolled out, few of which have been supported by the government of various countries such as Singapore [6] and Australia [3]. The main limitation of these approaches is the need for mass adoption before it becomes effective [5] and its reliance on Bluetooth distance measurements, which may not always be accurate.

Authenticity and privacy attacks are other key issues in using Bluetooth for contact tracing. [28] has shown that authenticity attacks can be easily performed on Bluetooth based contact tracing apps. Such attacks can forge the location visited and create a fake history of a user introducing risk to the society, as shown in [28]. Bluetooth apps suffer from privacy issues as noted in [101, 35]. As a result, privacy issues for Bluetooth-based contact tracing has received significant attention [22, 26, 47]. Privacy-preserving methods include the use of homomorphic encryption for determining contacts [10] and private messaging to notify possible contacts [115], to name a few.

5.8 Conclusions

Technology-aided contact tracing is becoming an increasingly important tool for quick and accurate identification of co-locators. While the Bluetooth-based contact tracing method using phones has become popular recently, these approaches suffer from the need for a critical mass of adoption to be effective. In this chapter, we presented WiFiTrace a network-centric approach for contact tracing that relies on passive WiFi sensing with no client-side involvement. Our approach exploits WiFi network logs gathered by enterprise networks for performance and security monitoring and utilizes it for reconstructing device trajectories for contact tracing. Our approach is specifically designed to enhance the efficacy of traditional methods, rather than to supplant them with new technology. We presented an efficient graph algorithm to scale our approach to large networks with tens of thousands of users. We implemented a full prototype of our system and deployed it on two large university campuses. We validate our approach and demonstrate its efficacy using case studies and detailed experiments using real-world WiFi datasets. Finally, we discussed the limitations and privacy concerns of our work and have made our source code available to other researchers under an open-source license.

CHAPTER 6

iSchedule : MOBILITY-AWARE HVAC SCHEDULING

In the previous chapter, I presented a mobility aware application for contact tracing that back-traced individual user trajectories after an event (user tests positive for a contagious disease) was observed. In this chapter, I present another mobility aware application that proposes Heating, ventilation, and air conditioning (HVAC) schedule by predicting building occupancy as derived from past aggregated user mobility.

HVAC systems account for over 50% of the energy consumed by commercial buildings. While "smart" HVAC technologies, such as learning thermostats, are widely available for residential use, commercial buildings typically rely on legacy systems that are difficult to upgrade and require facility managers to manually set HVAC schedules, which are typically based on intuitions of building managers. These manual operations are often cumbersome, error-prone, and do not scale well on a campus environment with multiple types of buildings such as education, library, food courts, recreational, dormitories, etc resulting in user discomfort and energy loss. In this chapter, we present iSchedule, a novel Machine Learning-driven technique to automatically learn custom occupancy-based HVAC schedules for buildings across campus that leverages aggregated human mobility extracted from the existing omnipresent wireless networking infrastructure in a modern campus.

6.1 Motivation and Problem Statement

Each building's heating and cooling are controlled by a commercial HVAC system. Unlike a residential HVAC system, which is controlled by a thermostat, a commercial

HVAC system is typically controlled through a Building Management System (BMS). The building's facility manager interacts with the BMS to set a heating and cooling schedule and temperature setpoints: this schedule specifies when the HVAC equipment should be turned on over the course of a day and the temperature setpoints for the high and low occupancy periods. The BMS then automatically operates the building's HVAC equipment based on the specified schedule, a process we refer to as *schedule-based HVAC control*.

The schedule is typically determined based on the facility manager's intuitive understanding of the building's occupancy patterns. For instance, in a typical office environment, employees may arrive between 8 am and 9 am and leave for home between 4 pm and 5 pm, and the building may be lightly occupied during non-business hours or on weekends. In this example, the facility manager may program the BMS to heat or cool the building between 8am to 6pm and use a higher cooling or lower heating temperature during the other off-peak hours. Doing so, ensures users are comfortable when the building is highly occupied while saving energy when it is largely unoccupied.

Modern BMS systems enable a different schedule to be set in different parts of a building—e.g., a different schedule on different floors—if the per-floor occupancy patterns differ. However, to fully exploit this functionality, a facility manager needs to determine fine-grain schedules for different parts of a building, and dynamically fine-tune it as occupancy patterns change over time. Such manual operation is cumbersome and error-prone and does not scale across a large campus, where a facility manager may oversee tens-to-hundreds of buildings. As a result, existing schedule-based HVAC control tends to be driven by simple, manually-chosen static schedules, which miss many opportunities for reducing energy use by carefully exploiting temporal and spatial occupancy differences within and across campus buildings.

(a) Schedule-based HVAC Control (b) Occupancy-based HVAC Control

Figure 6.1: Schedule- versus Occupancy-based HVAC Control

To address these limitations, we argue that the process of deriving schedules for commercial HVAC systems should be *automated*. To do so, we need a system that monitors occupancy patterns in campus buildings, automatically learns an optimal schedule for each part of a building based on the observed occupancy, and dynamically modifies the schedule as occupancy patterns change. Such a system should be sufficiently robust to tailor its schedules to the different types of *spatial* occupancy patterns seen across campus buildings, e.g. classrooms, academic units, library, dining halls, on a university campus. It should also automatically tailor schedules for *temporal* variations seen across weekdays and weekends and across seasons. Figure 6.1(a) depicts the architecture of such a schedule-based HVAC control system—hard or soft sensors are first used to infer occupancy in each building. Occupancy data is then "mined" to extract repeating patterns, from which an "optimal" schedule is learned and fed to the BMS for schedule-based control of the commercial HVAC system.

An essential first step in such a learning-based system is to derive occupancy, which captures *how many* people are present in each part of a building and at *what times*. Hard sensors such as motion or door sensors can be used to track occupancy within each building [21, 8, 7, 84, 102]. However, such instrumentation is not ubiquitous in office buildings and can be expensive and laborious to install in existing buildings. Researchers have shown that occupancy can also be learned through "soft sensors" that are already deployed for other reasons. For example, occupancy can be learned through swipe card door access systems, calendar software, or through wireless network activity [44, 80, 103, 81]. Since WiFi infrastructure is now ubiquitous in offices and campus buildings, our work uses existing wireless networks rather than

requiring hard sensors to infer occupancy information. Doing so enables easy deployment of our system in today's campuses without requiring the expensive deployment of hard sensors.

Specifically, this work assumes that most occupants carry mobile smartphones and the presence of a phone in the vicinity of a wireless access point indicates a user (occupant) at that location. We further assume that the exact location of each access point within a building is known a priori. Consequently, simply tracking the number of mobile devices associated with each AP over time is a proxy ("soft sensor") for the number of occupants in that part of the building. As shown in Chapter 3 wireless network infrastructure provides a log of when a mobile device connects and disconnects to each access point, which is then used to count the number of active occupants over time. To avoid double counting users, only smartphone log entries are considered and other devices, such as laptops or stationary devices are filtered out[1].

Problem Statement: Given an office campus with multiple buildings, anonymized WiFi association and dissociation logs, and the mappings of the Access Points to specific locations in each building, our goal is to automatically learn an HVAC schedule that optimizes user comfort and energy usage on a fine-grain spatial basis and dynamically adjusts learned schedules when observed occupancy patterns change.

6.2 Campus-scale Analysis of Building Occupancy

A key hypothesis of our work is that occupancy patterns often vary spatially within a building (across floors and zones) and across building types, and temporally across days and seasons.

[1]Note that the system only needs to count the number of active devices at an AP and does not need to track individual users—identifiable information such as device MAC addresses can thus be anonymized.

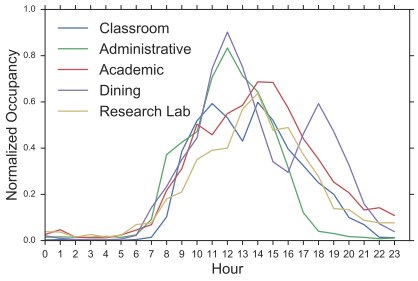
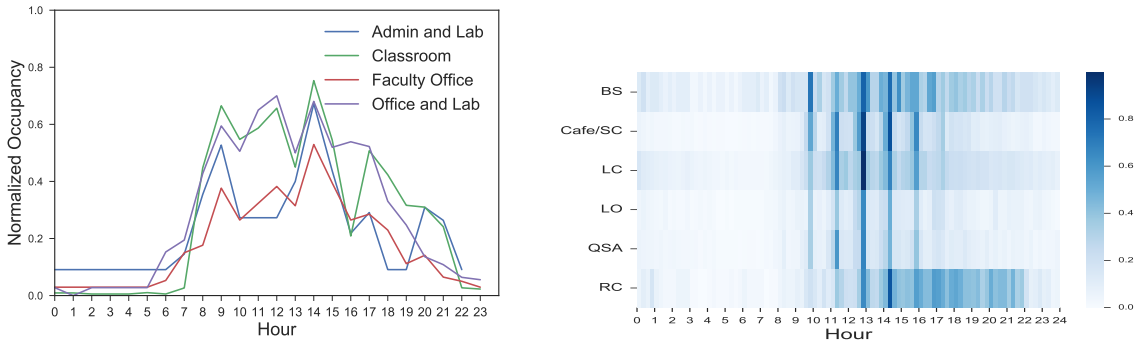


Figure 6.2: Normalized occupancy across campus buildings.

To validate this hypothesis, we analyzed occupancy patterns across all buildings on our university campus over several months. WiFi connectivity is ubiquitous on our campus—4674 access points are deployed in 112 campus buildings. We gathered anonymous WiFi association and disassociation logs for all 4674 APs over a 6 month period ranging from the beginning of Fall to the end of Spring; we included intervals when campus was closed (winter and spring breaks). The mapping of each AP to a building and a specific floor within that building is maintained by our campus IT office, which enabled us to analyze the number of active WiFi users in each building and each floor over the course of the day. We used the number of active smartphone users as a proxy for actual occupancy—a reasonable assumption due to the ubiquity of smartphones today. Due to the scale of our campus level analysis we use occupation computed by counting WiFi devices as our ground truth when evaluating the efficiency of various schedules; this technique has been demonstrated to be sufficiently accurate in prior work [16].

Impact of Building Type: Figure 6.2 shows the mean weekday occupancy for different hours of the day for several different buildings on our campus. Specifically, the figure shows occupancy for an academic department, a classroom building, an administrative building, a research lab and a dining hall. To ensure comparison across buildings, normalized occupancy is shown and we assume for our discussion here that a building is occupied whenever occupancy levels are more than 20% of the peak, while a building is considered unoccupied if levels are less than 20%.

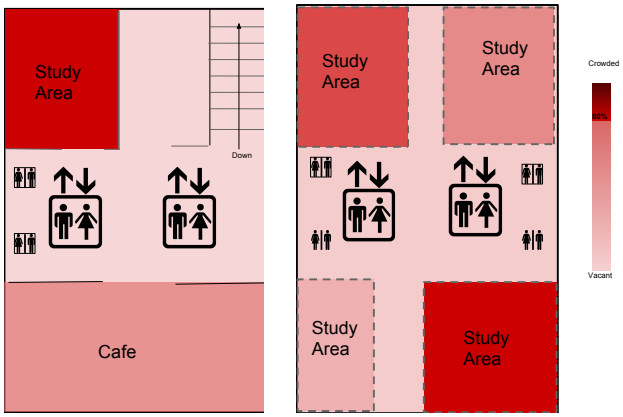


(a) (b)

Figure 6.3: Spatial differences in occupancy inside an academic and library building. (a) Normalized floor-wise occupancy of an academic building (b) Floor wise occupancy within the Library

The figure shows that occupancy patterns vary significantly by building type. The administrative building shows an 8am-5pm occupancy pattern. The academic building, which has student labs/offices has higher evening use as well and shows an 8am-8pm pattern. The occupancy patterns of the classroom building closely follow the lecture schedule. The research labs show a 9am-7pm schedule, while dining halls have peak occupancy during meal hours (e.g., breakfast, lunch and dinner hours). These results show that HVAC schedules need to be aligned with occupancy patterns and building type. The occupancy will vary depending on how the building is used.

Spatial Differences Within Buildings: Next we analyze whether different parts of a building can exhibit different occupancy patterns and by how much. Our analysis of the 112 buildings showed that buildings do exhibit spatial differences in occupancy patterns. This is illustrated in Figure 6.3, where Figures 6.3(a) and 6.3(b) depict occupancy patterns of two buildings on our campus - an academic department and the library. In Figure 6.3(a), the academic building has one floor comprising of administrative staff, offices, and classrooms, while two other floors comprise faculty offices and research labs. Not surprisingly, occupancy patterns for the floor with staff



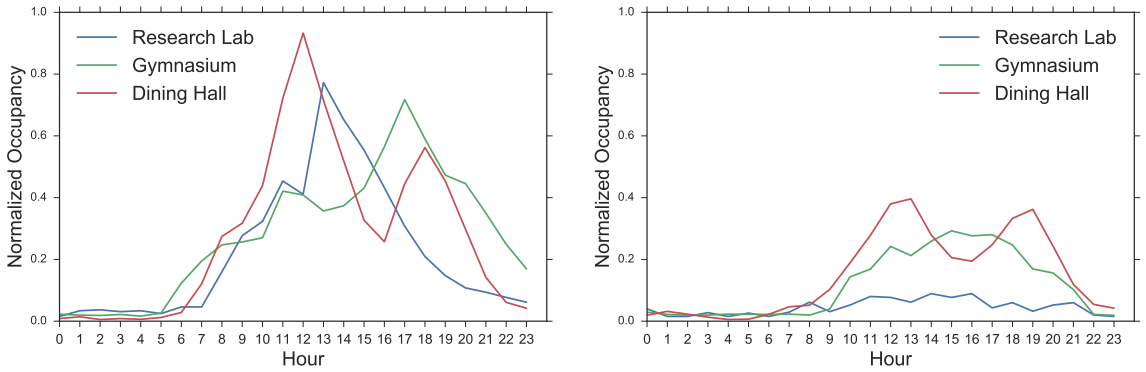
(a) Library Floor 1 (b) Library Floor 5

Figure 6.4: Library Occupancy depicted as a heatmap on a Tuesday from 10:00 - 11:00 AM shows the differences in occupancy across and within two different floors and motivates the use of unique schedules for different floors and zones.

offices and classrooms is markedly different than those of floors with faculty offices and research labs.

Similarly, the different floors of our campus library house different functions, as shown in Figure 6.3(b). One floor consists of the learning commons (LC) that are open 24×7 with student study desks and small breakout rooms for group study. Other floors contain library staff offices (LO), Quiet Study Area (QSA), Resource Center (RC) and checkout desks (Cafe/SC) or aisles of books (BS). As a result, we see very different patterns for these floors.

Finally, we show how occupancy across zones within a given floor of a library differ in Figure 6.4. In this figure, normalized occupancy is represented as a heatmap and demonstrates how different floors within a building can see different occupancy patterns. In these plots, we show the occupancy of the first and fifth floors from 10:00 - 11:00 AM on a Tuesday. We note that the top left side of floor 1 (panel a) has a study area that is crowded, while the lower side of floor 1, which has a cafe is less occupied. On Floor 5 (panel b) we see the study areas are occupied but with different occupancy percentage. Zone-based scheduling can further optimize HVAC usage – the top-left portion of Floor 1 has more than 10% occupancy from 9:00 AM



(a) Weekday occupancy

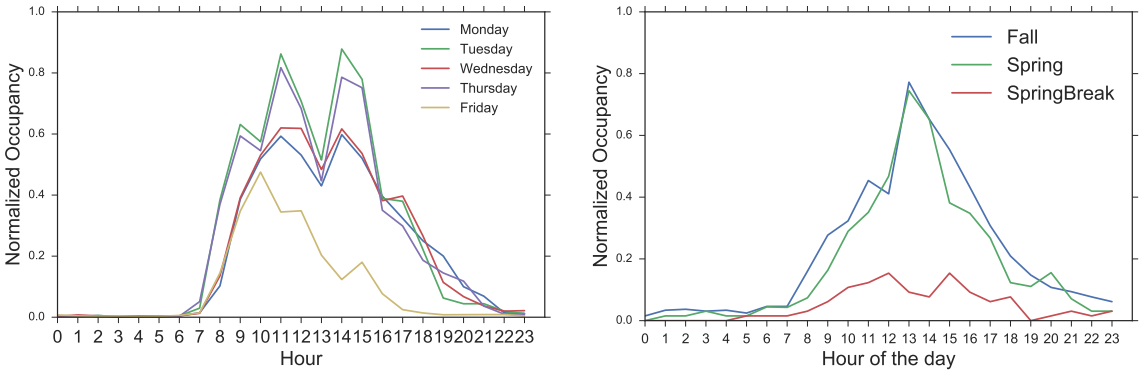
(b) Weekend occupancy

Figure 6.5: Weekday versus weekend occupancy within a research lab building, gymnasium and a dining hall.

- 11:45 PM on the day the heatmap was plotted, while the bottom portion of Floor 1 was more than 10% occupied from 9:30 AM - 5:45 PM. In contrast, the bottom right region of floor 5 sees similar occupancy levels from 9:00 AM - 7:00 PM, while the top most region was occupied from 9:45 AM - 5:15 PM. These observations all motivate the ability to set independent HVAC schedules across both floors and zones.

Overall, our results show that whenever a single building has different types of occupants or houses different types of users in different floors, spatial occupancy patterns will vary.

Temporal Variations: The previous results analyzed spatial occupancy patterns across different types of buildings on a campus and also within a building. Next, we analyze how occupancy patterns vary on a temporal basis over the course of a week and across seasons. Figure 6.5(a) depicts the normalized occupancy patterns seen in a research lab building, a gymnasium, and a dining hall on a weekday, while Figure 6.5(b) depicts occupancy patterns for the same buildings on a weekend. Not surprisingly, weekend patterns vary from weekday occupancy; the research lab building sees low weekend occupancy, the gymnasium sees occupancy at different hours on the weekend, while the dining halls see a later "start" to the day on weekends.



(a) Weekday Classroom Occupancy

(b) Lab Occupancy across semesters

Figure 6.6: a) Occupancy of the first floor of a classroom building during different days of the week b) Occupancy of a student research lab during Spring, Fall and Spring Break

Figure 6.6(a) shows changes in occupancy patterns over the course of a week for the same building on a particular floor. Since this is a classroom building containing a series of lecture halls, the occupancy patterns are highly correlated to lecture schedules, which vary by the day of the week. Finally, Figure 6.6(b) shows the occupancy patterns of a research lab in the Fall, Spring, and Spring break. Note that there are seasonal differences in the Fall and Spring semester. Unsurprisingly, we see lower occupancy during breaks when compared to occupancy observed during Fall and Spring.

Due to space constraints, this section depicts only a summary of key results from an extensive analysis of 112 buildings. These results collectively validate our hypothesis that there can be significant differences in occupancy patterns within a building, across buildings, and that the occupancy patterns can change over longer periods (e.g. seasons).

6.3 Learning HVAC Schedules

In this section, we describe iSchedule's data-driven learning algorithm that automatically learns HVAC schedules from occupancy data across campus buildings.

We assume that our system receives a raw log of smartphone association and disassociation information to each access point in the wireless network. Practically every commercial enterprise wireless network product routinely logs such information (e.g. Cisco, HP Aruba). The location of each access point within each building is assumed to be known.[2] Given this data, iSchedule learns schedules as follows:

Step 1: *Compute Temporal Occupancy Per Access Point*

Our system processes the raw WiFi logs to partition the logs on a per-access point basis. It then computes the number of active devices (i.e. users) connected to the AP in each time interval. This is done by incrementing the number of active users upon each new device association and decreasing it for each disassociation event. Doing so yields the number of active users in the vicinity of that AP during each time interval (e.g., every 15 minutes or hourly) over the duration of the log.

Step 2: *Derive Spatial Occupancy within a Building*

Since the location of each AP in a building is known, we can group all AP's spatially to obtain observed occupancy within each part of a building. Any spatial grouping can be chosen (depending on how fine-grain the HVAC control can be). The default grouping is on a per-floor basis—by aggregating the temporal occupancy seen by all APs on each floor, we obtain the number of users that are present on that floor in each time interval over the duration of the WiFi log. This yields a spatial distribution of users across the building and the change in spatial occupancy over time.

Step 3: *Use Predictive Model to Infer Floor/Zone Occupancy*

Next, our system predicts the occupancy of each floor/zone. We use a supervised training technique to predict occupancy. A Gradient Boosting Regressor Ensemble model is trained using the occupancy data computed in the previous step. In the case of university campus buildings, the following features are a strong indicator of

[2]Since a user may own multiple mobile devices, we avoid double counting by only counting mobile phones connected to an AP and ignoring other device types.

occupancy and form our feature set: (i) building name, (ii) building floor or spatial region, (iii) day of the week, (iv) time interval (e.g. hour of day or a 15 minute interval , 9:00 AM to 9:15 AM, etc), (v) semester of the year (vi) month, (vii) holiday and (viii) year. The floor/zone occupancy forms the label set.

The first two features capture building-specific information, the day of the week captures occupancy variations driven by working versus non-working days, while the time interval captures occupancy at a certain time of that day (e.g. 9:00 AM to 9:15 AM on Mondays). The semester (spring/fall/summer) captures seasonal effects, while holidays capture whether classes are currently in session (e.g. weekends, spring breaks and winter breaks). Note that most of these features are general and can be applied more broadly to any commercial building; specific features, such as semester, can be replaced by a more generic feature, such as the current month.

Our predictive model is based on a regression-based learning approach, which uses a gradient boosting regressor. We use a boosting ensemble method that incrementally builds base estimators so that each sequential estimator is trained to reduce the bias of the earlier estimators. For the model, we used the least squares regression loss function, which is optimized by each estimator. For parameter selection, a 10-fold cross validation technique was used.

Step 4: *Classify Intervals as High/Low Occupancy*

In this step, our system performs a binary classification of each time interval as high or low occupancy based on the occupancy predicted in the previous step. To do so, we first compute a probability distribution of the number of users observed in each part of a building (e.g. probability distribution of users seen on a floor). We define the maximum occupancy of a floor or zone as the high percentile of this distribution. Next, we select a threshold value τ, that represents a fraction of this maximum occupancy – fractional occupancies above or below this threshold are marked as high (H) or low (L) occupancy respectively. This step yields a trace for each portion of a

building where each interval is marked H or L over the entire duration of the WiFi trace. As an example, if the max occupancy of a floor is 100 and $\tau = 10\%$, then any interval with the floor occupancy exceeding 10 users is marked H and others marked L.Parameter τ can be tuned by the facility manager to chose a suitable tradeoff between building occupant comfort and energy saving. If a high value of τ is chosen then the model becomes aggressive by turning off HVAC equipment more frequently, while a low value of τ causes the model to become less aggressive and leaves HVAC equipment on for longer periods.

Step 5: *Learning a Schedule from the Predictive Model*

In the final step, we derive the actual HVAC schedule. To do so, we consider all seven days of the week and use the model to predict the occupancy for each floor of each building for every time interval of a day. Then, we convert the predicted occupancy in H or L occupancy periods as described in step 4.

We consolidate each sequence of H periods into a single interval where the HVAC must be turned on and consolidate each sequence of L periods into intervals where the HVAC should be turned down. Smoothing can be used to eliminate small periods of H or L periods. This yields a schedule, which gives periods for each day of the week on how the HVAC should be operated on each floor and building (e.g., turn on HVAC from 8:30 AM-5:45 PM on the 3rd floor of the library on Monday and again from 8:00 PM-10:00 PM).

Such a schedule is automatically learned and uses the precise occupancy pattern in each part of a building to compute a custom schedule for different parts of a building.

6.4 Dynamic Adaptation of Learned Schedules

The previous section described our learning algorithm, which automatically learns a customized HVAC schedule for each spatial region of a building based on the occupancy patterns observed in that region. However, occupancy patterns are not

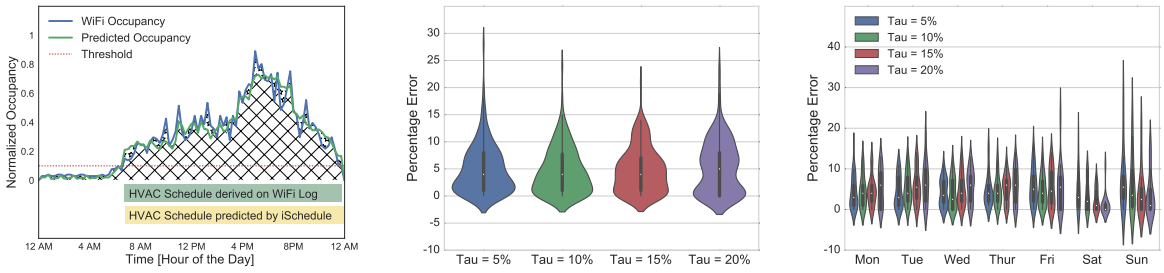
stationary and will slowly (or abruptly) change over time. These changes in occupancy patterns may occur for a number of reasons: the building or floor may get re-purposed for a different class of users. For example, an academic building may become administrative space with new types of users moving in or there may be subtle changes in occupancy patterns with different types of users over time (e.g., due to changing of class schedules or different user patterns).

Regardless of the cause, the learned schedules cannot remain static—they must adapt and evolve with changing occupancy patterns. In other words, once learned, the HVAC schedule must be dynamically and periodically recomputed and adjusted. The algorithm presented in the previous section can be enhanced in one of two ways to support adaptation.

Continuous Adaptation: In this method, WiFi activity data is ingested every day and spatial occupancy observed within each building during that day is added to the historical trace. The predictive model is re-learned using all data, including the newly ingested information, and the HVAC schedule (step 5) is re-computed. The frequency with which the schedule is recomputed is configurable (e.g., daily, weekly, monthly, etc).

On-demand Adaptation: A limitation of the continuous adaptation approach is that it wastes computational resources when no significant changes to occupancy are observed, as the model is re-trained periodically, regardless of whether it is necessary. On-demand adaptation is an alternate approach that triggers re-training only when the prediction deviates from observed occupancy.

As before, new WiFi activity data arrives continuously and is added to the historical data repository. The system then periodically invokes the previously learned predictive model to predict high and low occupancy labels for a recent time interval. The model predictions are compared to the actual occupancy levels observed in the newly captured WiFi-based occupancy data. If the model predictions match the ob-



(a) (b) (c)

Figure 6.7: Accuracy of iSchedule's learning algorithm (a) Comparison with ground truth (b) Violin Plot of error for different values of τ (c) Violin plot of Day-wise error for different τ

served levels, then the occupancy patterns are same as before and neither the model nor the HVAC schedules need to be adjusted. On the other hand, if the recently observed occupancy levels begin deviating from model predictions, then our system triggers a re-training of the predictive model and uses the new model to recompute the HVAC schedules.

Thus, a new model is learned only when needed and only for those buildings (or parts of a building) where significantly different occupancy patterns are observed. The threshold error ϵ between model predicted and actual observations that trigger a re-learning is configurable: a smaller ϵ triggers more frequent re-computations and schedule adjustments and vice versa.

6.4.1 Discussion

Selecting τ ***:*** The value of τ can be selected by the facility manager to choose a suitable tradeoff between building occupant comfort and energy saving. Since our algorithm learns building occupancy independent of a particular schedule, our model supports any value of τ. Selecting high values of τ results in more aggressive schedule since HVAC is turned off for periods of occupancy lower than the threshold. Such schedules have a high energy saving but may not necessarily have high comfort. On

the other hand, lower values of τ results in a non-aggressive schedule. Therefore, we allow this parameter to be adjusted based on building occupant feedback.

Residential Buildings: While our approach works well for a broad range of office buildings, any campus building that sees "residential" usage requires special handling. Specifically, our technique assumes that *lack of WiFi activity corresponds to a lack of occupants*. In campus buildings with residential environments such as student dorm or campus hotel, students or hotel guests sleep at night and, thus, are present in the environment despite a lack of WiFi activity. Thus, turning down heating and cooling overnight due to lack of observed activity will result in incorrect schedules for these buildings. A simple enhancement can be made to our algorithm by adding a new binary feature called "Sleeping Zones". The computed schedules are then adjusted to keep the HVAC system operational during night hours (e.g. 11 PM - 6 AM) in all areas marked as sleeping zones.

6.5 Experimental Evaluation

In this section, we experimentally evaluate the efficacy of our learning-based algorithm and its ability to dynamically adjust schedules based on changing occupancy patterns. We use data from 112 buildings (2015-2016) on our university campus for our experimental evaluation. We compare the schedule derived from iSchedule against those derived from WiFi-based occupancy and static pre-set schedules. As a baseline for comparison, we assume the set of static schedules that are shown in Table 6.1; these static schedules are based on a facility manager's expectation of how different buildings are used.

For the purposes of our evaluation, we use the WiFi occupancy data and our learning-based algorithm to learn a schedule for each building and various floors of each building. We compare the learned schedules, which are based on actual

Building Type	Weekday	Weekend
Classroom	8:00 AM to 8:00 PM	Off
Administrative	8:00 AM to 6:00 PM	Off
Academic	8:00 AM to 6:00 PM	10:00 AM to 4:00 PM
Dining	7:00 AM to 10:00 PM	7:00 AM to 10:00 PM
Research Lab	8:00 AM to 7:00 PM	10:00 AM to 4:00 PM
Library	24 hours	8:00 AM to 8:00 PM
Student dorm	24 hours	24 hours
Student Union	8:00 AM to 8:00 PM	10:00 AM to 6:00 PM

Table 6.1: Statically determined schedules for when the HVAC should be turned on in different types of buildings

Floor Type	$\tau = 5\%$	$\tau = 10\%$	$\tau = 15\%$	$\tau = 20\%$
Office & Lab	8:00 to 23:59	9:00 to 19:59	9:00 to 18:59	9:00 to 18:29
Faculty Office	9:30 to 19:29	9:30 to 19:29	9:30 to 19:29	9:30 to 18:44
Classroom & Discussion Rooms	7:45 to 22:14	8:00 to 20:59	8:00 to 19:44	8:15 to 19:29
Admin & Lab	7:45 to 22:14	8:45 to 21:14	8:45 to 21:14	8:45 to 18:14

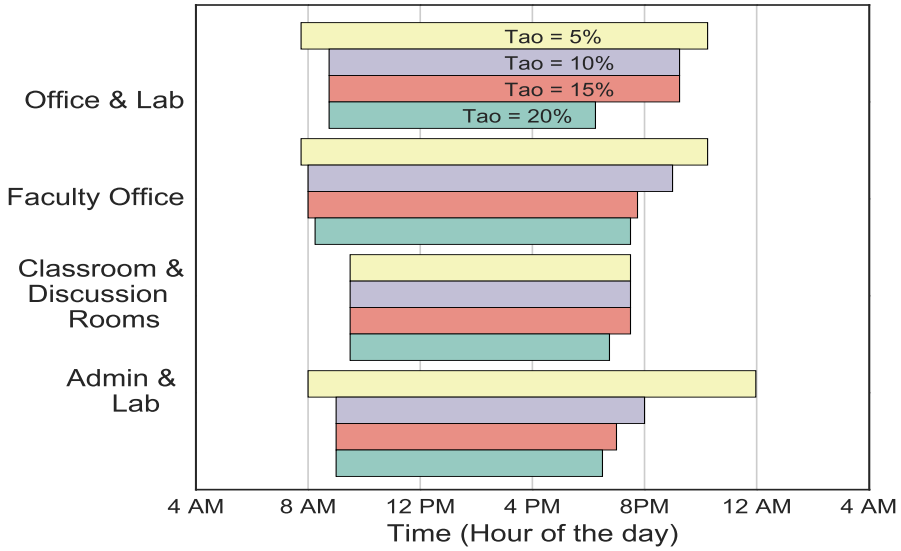


Figure 6.8: Comparison of Predicted HVAC schedule for each floor of an academic building, derived for $\tau = 5\%$,10%,15%,20%.

occupancy data, to the statically chosen schedules, and also quantify whether the learned schedule yields increased energy savings and user comfort.

Energy savings are computed as the reduction in *waste time* (WT) when compared to the static schedule, where waste time is defined as the duration for which the HVAC system is turned on even though an area is in a low occupancy period. The increase in user comfort is computed as the reduction in *miss time* (MT) when compared to the static schedule, where miss time is defined as the duration for which the HVAC system is off while an area is in a high occupancy period (causing user discomfort). We formally define waste and miss time in the Appendix.

6.5.1 Accuracy of our algorithm

In Figure 6.7(a) we compare the generated HVAC schedule to the actual occupancy. We see that based on the WiFi occupancy detected we find the low and high occupancy periods as marked by H or L Occupancy. This string of H or L Occupancy generated by the system is then smoothed to remove any short intervals of H or L and the smoothened HVAC schedule is obtained. We find that the HVAC schedule generated by iSchedule closely matches the HVAC schedule generated from WiFi data. Further, Figure 6.7(b) shows the model error computed for a wide range of building types for different values of τ. We trained our model on the historic training dataset and predicted the HVAC schedule for the next 15 days with the adaptation feature disabled. The error was computed against the WiFi building occupancy. We find that our model has a high accuracy of 95.35% with a coefficient of variation of 3.15%. Finally, Figure 6.7(c) shows the error computed for a wide range of building types for different values of τ for each day of the week. We find that highest variation in error occurs on Sunday for all values of τ. Also, for all weekdays the mean error range was 0 - 7% for different types of buildings and different values of τ.

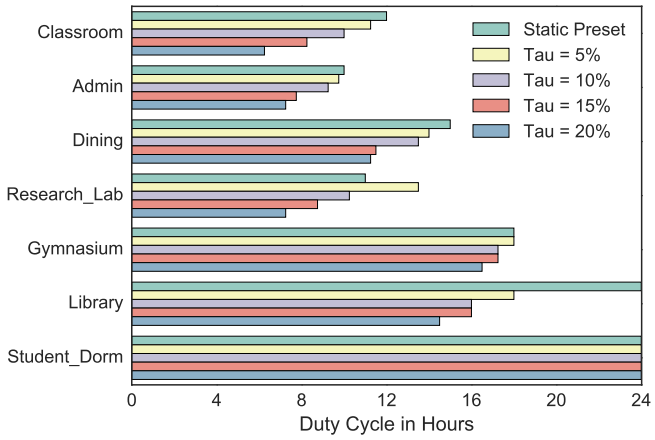


Figure 6.9: Duty Cycle of learned weekday HVAC schedules for different types of buildings for different thresholds.

Building Type	Weekday	Weekend
Classroom	8:15 AM to 6:14 PM	off
Administrative	8:00 AM to 4:44 PM	off
Dining	8:30 AM to 9:14 PM	11:15 AM to 7:59 PM
Research lab	8:00AM to 6:59PM	off
Library	8:45 AM to 11:59 PM	1:00 PM to 10:59 PM
Student dorm	24 hours	24 hours
Gymnasium	6:45 AM to 11:44 PM	11:15 AM to 7:59 PM

Table 6.2: Learned weekday and weekend HVAC schedules for different types of buildings computed with $\tau = 5\%$.

6.5.2 Efficacy of Learned Schedules

Table 6.2 shows weekday and weekend schedules learned for several different buildings on our campus. These schedules correspond to the observed occupancy and we observe several differences between the learned schedules and the statically set ones, which demonstrates the ability of our approach to capture fine grain occupancy differences. Figure 6.9 shows the weekday duty cycle for the same set of buildings, as shown in Table 6.2, for different values of τ. These schedules are derived by iSchedule and we observe several differences between the duty cycles of the learned and static schedules. The duty cycles for derived schedules are highest for $\tau = 5\%$ and are lowest for $\tau = 20\%$; as τ increases, the model is more aggressive in turning off HVAC equipment. We also observe that the duty cycle of static schedules are highest for non-residential buildings, which demonstrates that there is energy wastage by conditioning buildings when occupancy is low.

Building Type	Monday Schedules
Classroom	8:29 AM to 6:14 PM
Administrative	7:30 AM to 5:14 PM
Dining	7:45 AM to 9:14 PM
Research Lab	9:45 AM to 6:59 PM

Table 6.3: Learned HVAC schedules for a Monday.

Next, Table 6.3 shows the schedules learned for a specific weekday (Monday) by our algorithm for several types of buildings on our campus – a threshold of $\tau = 20\%$ was used to compute these schedules. The table also reveals differences from the static schedules, which imply that they incur either more waste or miss time.

Table 6.8 and Figure 6.10 show schedules learned for different floors of an academic building and library. Since there are spatial differences across floors of each building, a manual schedule that uses a single schedule for the entire building is sub-optimal. Our approach can exploit the observed differences in spatial occupancy and choose

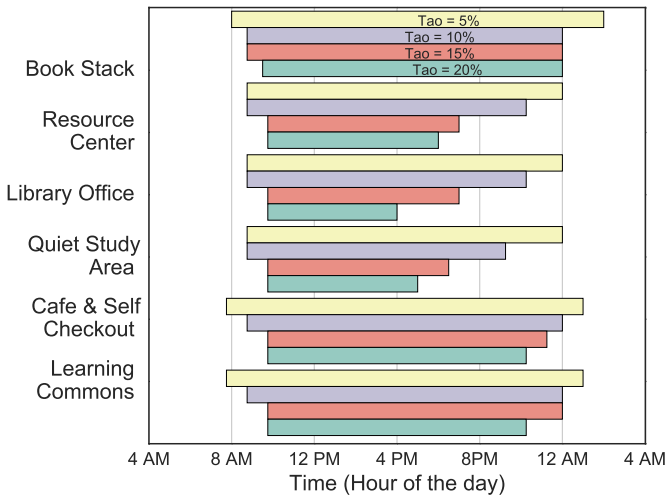


Figure 6.10: Predicted weekday HVAC schedule derived with different thresholds for a library building with different types of occupancy on each floor.

different schedules for each floor of a building. We again observe the floor-specific schedules are in line with observed occupancy shown in Figure 6.3.

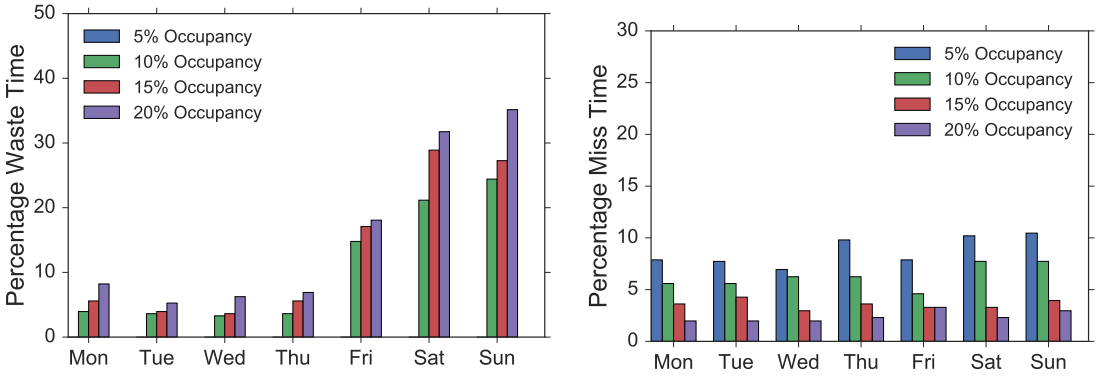
Handling residential areas on campus: For buildings such as student dorms with sleeping residents, our basic algorithm will turn off the HVAC equipment 12 AM to 8 AM on weekdays and 2 AM to 10 AM on weekends in student dorms due to lack of WiFi activity in the night. However, our enhanced algorithm can handle sleeping zones and, as shown in Table 6.2, leaves HVAC equipment on for 24 hours on weekdays and weekends when the semester is in session, while reverting to a normal schedule during summer breaks when the residence halls are vacant.

6.5.3 Impact on Energy Use and User Comfort

While the previous results highlight the ability of our approach to automatically derive schedules that closely match observed occupancy, we now quantify the benefits of these derived schedules in terms of energy saving and user comfort. We vary the threshold τ that determines the low occupancy period for each building on our campus and use our algorithm to generate a schedule for that τ. We compare the derived

schedule to the static schedule and compute the increased energy savings and user comfort.

Figure 6.11 depicts the percentage of waste time for the entire campus (112 buildings) for different days of the week for varying values of τ (τ = 5%, 10%, 15% and 20%).



(a) Reduction in waste time across the campus for varying values of τ

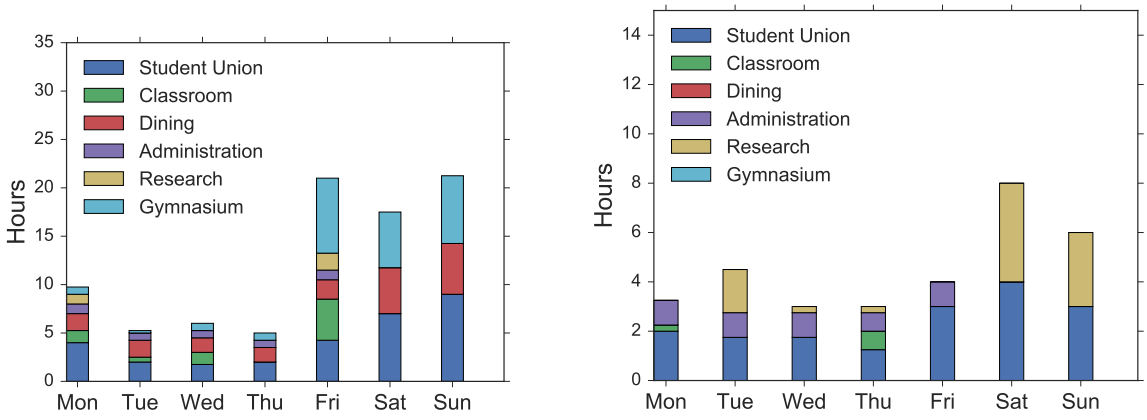
(b) Reduction in miss time across the campus for varying values of τ

Figure 6.11: Waste Time and Miss Time

Figure 6.11(a) shows an increasing reduction in waste time with increasing value of τ; the figure depicts the average reduction in waste time across all 112 campus buildings for different days of the week. This occurs because, as the threshold τ is increased, our algorithm is more aggressive in turning off the HVAC equipment via the learnt schedules at higher levels of occupancy. The percentage reduction in waste time is around 3-20% for τ = 10% and increases to 15-37% for τ = 20%.

Figure 6.11(b) shows the schedules computed by our approach are also able to increase user comfort, which is achieved by reducing miss times. The figure depicts the average reduction in miss time across all 112 campus buildings for different days of the week. Unlike energy savings, user comfort shows a *decreasing* trend with increases in τ. This occurs because, with higher τ, the HVAC equipment is on for fewer hours, which reduces the opportunity to simultaneously increase user comfort. The average reduction in miss times is around 17% for τ = 10% and around 9% for τ = 15%.

Together the results show that across 112 buildings with varied use, automatically learning HVAC schedules using occupancy data yields energy savings while also providing a more comfortable ambient environment to users.



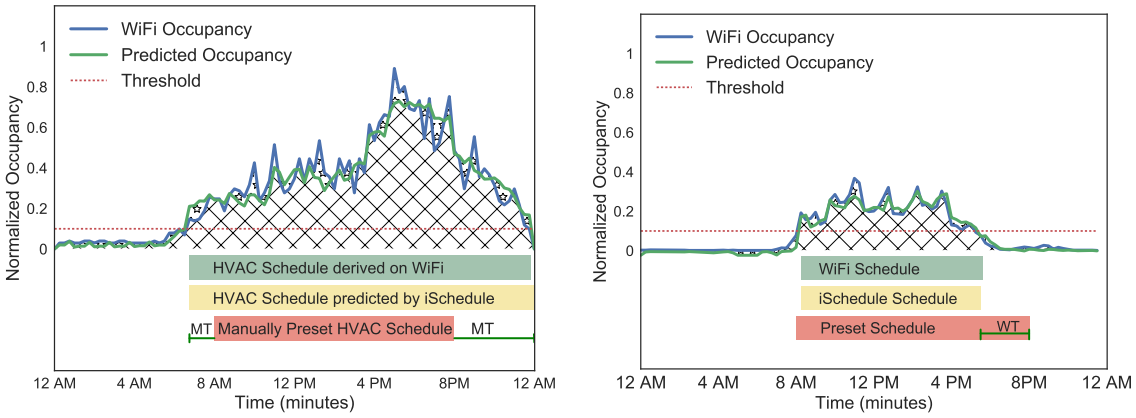
(a) Waste time for $\tau = 0.10$

(b) Miss time for $\tau = 0.10$

Figure 6.12: Daily reduction of waste and miss time (in number of hours) for a selection of campus buildings

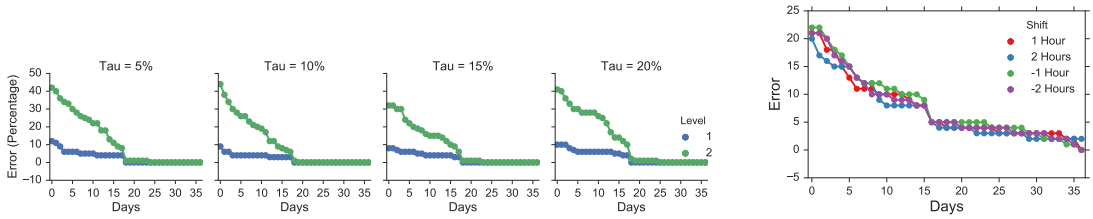
Figure 6.12(a) and (b) show a breakdown of energy saving and comfort for different types of building for $\tau = 10\%$. The greatest gains are observed where learned schedules and actual occupancy varies the most with the static manual schedules. For example, more energy savings are seen on weekends than weekdays. On weekdays, the student union building sees the most savings. Classroom buildings see more savings on Fridays than other days due to a shorter lecture schedule on Fridays. Dining shows high energy savings on Fridays and weekends. Administrative buildings show increased comfort on weekdays, while research labs show an increase in comfort on weekends by following the dynamic schedule.

Finally, Figure 6.13(a) depicts the normalized occupancy of one illustrative campus building on our campus (the student union) on Friday. As can be seen, the learned schedule is better aligned with observed occupancy on that day and the ground truth occupancy derived schedule. Also, we can see that the derived schedule results in



(a) Miss Time (MT) (b) Waste Time (WT)

Figure 6.13: Comparison of Schedule derived from WiFi Occupancy, iSchedule and Static schedule, where MT is Miss Time and WT is Waste Time



(a) (b)

Figure 6.14: Adaptability of iSchedule's learning algorithm for change in occupancy pattern (a) System Error across different types of floors of a building for 1 Hour Shift (b) System Error for different shifts

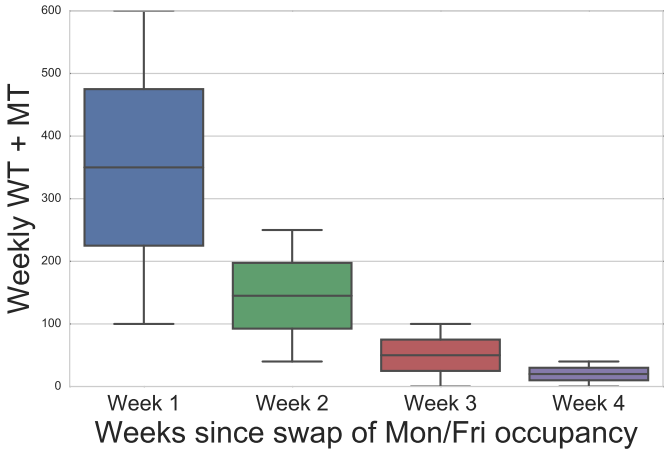


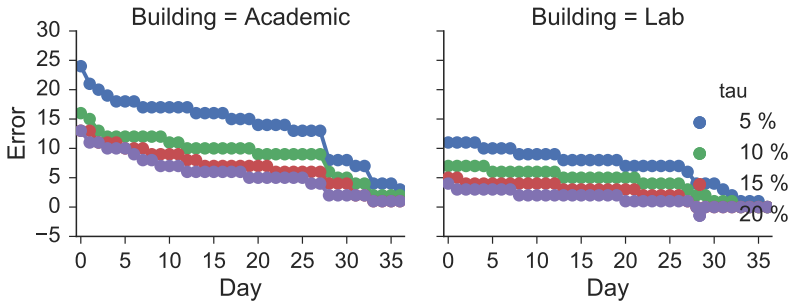
Figure 6.15: Weekly Waste Time + Miss Time of iSchedule compared to WiFi when Monday and Friday occupancy is swapped for an academic building and a dining hall.

improved user comfort during the evening hours (8 PM-12 AM); the static schedule turns down the HVAC even though the building is at 30% occupancy, while the learned schedule keeps the HVAC system running to maintain comfort for building users. Finally, figure 6.13(b) depicts the normalized occupancy of one illustrative campus building on our campus (Learning Center, with Classrooms) on Friday – the learned schedule shows substantial savings over a static schedule by reducing waste time. It shows the energy savings during the evening hours (where the static schedule keeps heating or cooling the building later than necessary).

6.5.4 Efficacy of Dynamic Adjustments

Finally, we evaluate the efficacy of our technique to adjust to dynamic changes in occupancy that may occur in a building. We use WiFi activity data from an academic building and synthetically modify the trace data to emulate two types of changes. First, we shift the observed occupancy to earlier hours, which reflects users arriving to the office earlier than previously observed data. We study the impact of users arriving 1 hour and 2 hours earlier than usual and leaving proportionately sooner, as well as the impact of users arriving 1 - 2 hours later than usual and leaving proportionately later. Second, we swap every Monday and Friday for a set of different building types to simulate a change in working hours, since Monday and Friday occupancy patterns are very different.

Figure 6.14 and Figure 6.15 depict our results. In Figure 6.14(a), we show the change in error for the first 35 days of a building with 2 levels each having a different type of floor occupancy. Level 1 has an almost stable occupancy schedule due to administrative offices while Level 2 has a very dynamic occupancy pattern that differs each day of the week due to the presence of Classrooms and Discussion Rooms. We observe that Level 1 converges quickly as compared to Level 2. Also, we see high error in the first week after the change in occupancy; this triggers re-training



(a) Academic (b) Research Lab

Figure 6.16: Model Error for learning HVAC schedules of a newly constructed building.

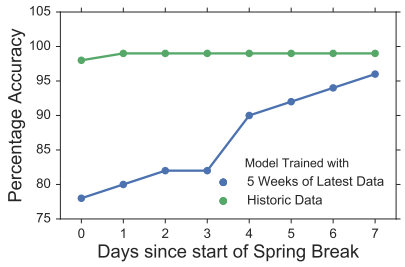


Figure 6.17: Utility of historic data

of the model each day which reduces error. The model learns the new occupancy pattern over time and achieves accuracy improvements by the end of the second week – this demonstrates the ability of our approach to adapt to non-transient changes in occupancy patterns for different types of floor dynamics. This experiment also demonstrates that our model adapts to occupancy changes for buildings that have nearly identical occupancy throughout the week or a highly fluctuating occupancy across each day. Figure 6.14(b) shows that for varying values of shift in schedules the model error converges by the end of week 2 resulting in accuracy of more than 90%. Figure 6.15 shows that for the first week the error is highest resulting in a high MT + WT value and decreases as the model retrains with new data.

Figure 6.16 depicts how quickly our algorithm can compute HVAC schedules for two new buildings that lacks historical occupancy patterns. In particular, we consider a newly built academic building and research lab.

In this case, as new WiFi data arrives, the model is retrained and HVAC schedules updated daily. An academic building has multiple floors and each floor has a varying schedule over weekdays and weekends, while the research lab has an almost fixed schedule across weekdays and weekends. As can be seen, our techniques can still compute schedules with only a few days of occupancy data and its accuracy improves gradually as more occupancy data becomes available – 10 to 28 days of data seems to be sufficient to converge to good schedules, which shows the agility of our technique.

Finally, Figure 6.17 shows the model accuracy during a change of semester (start of Spring Break) for a dining hall. The model, when trained on only five most recent weeks of data, shows lower accuracy than the model trained on historic data. From the start of Spring Break, the model trained on historic data has a high accuracy of the predicted HVAC schedule, whereas the model trained on the latest 5 weeks of data shows lower accuracy but gradually learns.

6.6 Related Work

Occupancy-driven versus schedule-driven HVAC control: Efforts such as Sentinel and others [16, 7, 8] have shown how occupancy sensors can *directly control* HVAC systems. The basic approach, depicted in Figure6.1(b), uses observed periods of high and low occupancy to directly control HVAC systems and save energy during off-peak periods. This approach, while novel, is not compatible with most existing BMSs that employ *schedule-driven* control (Figure 6.1(a)). In the latter approach, occupancy data is first used to learn a repeating schedule, which then is then set in the BMS to control the HVAC system. Thus, occupancy information only indirectly, rather than directly, influences HVAC operation.

While direct occupancy-driven control approaches may be appropriate for buildings with local (e.g., room-specific) HVAC units [62], they are not viable for the majority of centralized commercial HVACs controlled through BMS schedules. In addition, given their experience with schedule-based control, many facility managers may be uncomfortable with ceding direct HVAC control to software. Thus, by deriving repeating occupancy-based schedules, we enable facility managers to retain some control over HVAC usage.

Residential versus Commercial: In residential settings, efforts such as smart thermostat [77], iProgram [53], as well as products such as Nest, Ecobee, and Lyric, have been used to improve HVAC energy-efficiency. Such smart thermostats, as well as all "dumb" programmable thermostats, use schedule-based HVAC control, where occupancy information (from onboard sensors, phone GPS, or even electricity meters [53]) is analyzed to automatically learn a custom schedule. Occupancy sensors may occasionally turn on "away" mode, but they do not exercise direct control. User feedback has also been used to optimize HVAC use [43, 70]. While homes need only binary temporal occupancy larger commercial buildings need spatial occupancy data. Thus, our work can be seen as analogous to these residential efforts but applied to commercial buildings—a more complex problem.

Inferring Occupancy: There has been significant work in deriving occupancy information both for residential and office buildings. Prior work on deriving occupancy information falls into three categories: (i) design of novel occupancy sensors, (ii) use of existing soft sensors [92], and (iii) use of energy analytic methods to learn occupancy [91, 21, 70, 43, 27, 8, 81, 34, 66, 65]. However, most approaches only derive occupancy and do not apply it for HVAC control. As shown in Figure 6.1(a), deriving occupancy data is only a necessary first step for smart HVAC control and is not sufficient for addressing the broader control problem. One closely related technique combines soft sensing with HVAC scheduling [11]; human occupancy is sensed by

monitoring human-induced HVAC heat loading and is used as feedback to modify an existing schedule. While our system, iSchedule, also derives HVAC schedules, it instead leverages WiFi-based soft sensors for predicting occupancy. WiFi-based soft sensors can more explicitly derive occupancy counts since there is a direct mapping between numbers of device associations and numbers of occupants. In contrast, our system is easily deployed across an entire campus rather than relying on more specific feedback from advanced HVAC functionality, which could be limited to more recently constructed buildings with newer HVAC units. Furthermore, HVAC-based soft sensors can only operate at the granularity of already defined zones; WiFi access points are typically deployed at a higher spatial density, enabling a building manager with data needed to potentially redefine zones in the future.

6.7 iSchedule summary

In this chapter, we presented a system for campus-scale HVAC scheduling using mobile WiFi data. Our campus-scale analysis showed spatial and temporal variations in occupancy within and across buildings and motivated the need for an automated approach for learning HVAC schedules in campus buildings. We presented iSchedule's supervised learning algorithms and show its efficacy and accuracy with extensive evaluations using a real world large university campus dataset.

CHAPTER 7

SUMMARY AND FUTURE WORK

7.1 Thesis Summary

This thesis has investigated across 3 main areas - mobility characterization, mobility modeling, and design of mobility-aware applications through passively sensed WiFi logs. I have demonstrated how user and device mobility can be passively sensed from WiFi syslogs and presented many new insights on mobility as observed at multiple spatio-temporal granularity through empirical characterization of human and device mobility (Chapter 3). Using these insights I proposed and investigated a different take on human and device mobility modeling (Chapter 4) with the use of machine learning, and NLP techniques. Further, I presented two mobility aware applications (Chapter 5,6) that have practical real world use.The main contributions of this thesis are as below:

- *Empirical Characterization of Mobility :* First, I discussed how the current mobility research largely assumed device mobility of modern Internet users owning multiple mobile devices to be independent and analyzed mobility at the spatial scale of the underlying mobile dataset. I challenged both these beliefs and presented empirical analysis for mobility of modern Internet users owning multiple devices at multiple spatial scales using a large campus WiFi dataset resulting in three-fold contributions. First, I demonstrated that mobility of multiple devices belonging to a user needed to be analyzed and modeled as a group, rather than independently, and that there are substantial differences in the correlations exhibited by device trajectories across users that also need to

be considered. Second, the mobility of users showed different characteristics at different spatial scales such as within and across buildings. Third, we found that mobility is related to device type—phones have 3.5X greater mobility than laptops. Despite these differences, devices belonging to the same user show moderate to strong correlations in mobility for the majority of the users, and the type of building has a significant influence on the frequency and timing of mobility patterns observed. More broadly, our empirical results pointed to the need for new modeling research to fully capture the nuances of mobility of modern multi-device users.

- *Mobility Modeling:* Second, I proposed a Transformer-based, data-driven approach that models indoor human mobility at multiple spatial scales using WiFi system logs. WiFiMod takes as input enterprise WiFi system logs to extract human mobility trajectories from smartphone digital traces.

- *Mobility-aware Applications:* Third, I discussed the importance of designing mobility-aware applications and presented applications that broadly fall into 2 categories- (i) applications where we backtrack the past observed mobility after an event happens in time to infer attributes, causes, or preventive actions associated to the event, (ii) applications based on predicting future mobility or aggregated mobility (such as occupancy prediction) over a horizon. I presented one application per category:

  - *Network-centric Contact Tracing :* WiFiTrace, a network-centric approach for contact tracing of infectious diseases using passive WiFi sensing. We backtrack observed user trajectories for infectious disease containment.
  - *Mobility-aware HVAC Scheduling :* iSchedule, a machine learning-driven technique to automatically learn custom occupancy-based Heating, Ventilation, and Air Conditioning (HVAC) schedules for buildings across a large

campus. An application based on predicting future building occupancies by observing past mobility and occupancy of each building.

7.2 Future Work

This dissertation covers a broad range of areas from characterization and modeling to designing applications in the area of human mobility and gives rise to several promising directions to enable the next generation of smart applications. Here, I will outline some directions for future work that has emerged from this thesis.

- *Mobility-aware HVAC scheduling :* The machine learning approach in iSchedule can be extended to focus on shifting HVAC schedules to account for a building's thermal inertia, predicted weather conditions, and thermal communication across zones; these optimizations will further improve user comfort beyond the presented results. We have also worked on using mobile phone batteries to measure indoor ambient temperature and use crowd sensing to increase accuracy of model. The usage of on-phone models for temperature measurement along with fine-grain fingerprint- based WiFi occupancy detection, building thermal inertia, space usage, and human activity to further improve the accuracy of zone level scheduling for higher user comfort and energy saving will be a novel extension of the work.

- *Impact of COVID19 on Human mobility :* Our campus-wide WiFi logs have been collected since 2013 and with COVID19, a major pandemic that crippled human mobility for a long time, changing the course of human mobility behavior, it would be very interesting to perform a longitudinal analysis to understand the impact of COVID-19 on human mobility once the world has moved past the pandemic and user activities return to the new norm.

- *Predicting Influenza Like Illness (ILI) patients* : Influenza is a very contagious disease and a major contributor to morbidity. We could use trajectory back-tracking as used in WiFiTrace along witth changes in mobility as observed passively using WiFi log to predict the likelihood of a person falling sick with a contagious ILI. Such a model will help with predicting the number of expected patients at Health services for better allocation of resources as well as serve as an early indicator of the onset of ILI wave on a campus like environment.

APPENDIX

CHAPTER 6 APPENDIX

A.1 Formal definition of Waste and Miss Time

Formally, we define the miss time and waste time in terms of *conditioning period* (CP) below for N time periods with normalized occupancy N(t) for threshold τ.

$$O(t) = \begin{cases} 0, & \text{N(t)} < \tau, \\ 1, & \text{N(t)} \geq \tau. \end{cases} \tag{A.1}$$

$$CP(t) = \begin{cases} 1, & \text{if the zone is conditioned at time t,} \\ 0, & \text{otherwise.} \end{cases} \tag{A.2}$$

Given CP(t), we then define the average daily miss time and waste time over a time period N, as shown below.

$$MT = \frac{\sum_t (O(t) - CP(t))}{N} \quad \forall\, t \text{ where } O(t) = 1 \tag{A.3}$$

$$WT = \frac{\sum_t (CP(t) - O(t))}{N} \quad \forall\, t \text{ where } CP(t) = 1 \tag{A.4}$$

BIBLIOGRAPHY

[1] *Safe Paths*.

[2] *Apple Google Partner Covid-19 Contact Tracing*, 2020.

[3] *COVIDSAFE APP*, 2020.

[4] *Privacy Preserving Proximity Tracing*, 2020.

[5] Singapore built a coronaviris app but it hasnt worked so far, April 22 2020.

[6] *TraceTogether App Covid-19 Contact Tracing*, 2020.

[7] Agarwal, Y., Balaji, B., Dutta, S., Gupta, R.K., and Weng, T. Duty-cycling Buildings Aggressively: The Next Frontier in HVAC Control. In *IPSN* (April 2011).

[8] Agarwal, Y., Balaji, B., Gupta, R., Lyles, J., Wei, M., and Weng, T. Occupancy-Driven Energy Management for Smart Building Automation. In *BuildSys* (November 2010).

[9] Al-Molegi, Abdulrahman, Jabreel, Mohammed, and Ghaleb, Baraq. Stf-rnn: Space time features-based recurrent neural network for predicting people next location. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (2016), IEEE, pp. 1–7.

[10] Altuwaiyan, Thamer, Hadian, Mohammad, and Liang, Xiaohui. Epic: Efficient privacy-preserving contact tracing for infection detection. In *2018 IEEE International Conference on Communications (ICC)* (2018), IEEE, pp. 1–6.

[11] Ardakanian, Omid, Bhattacharya, Arka, and Culler, David. Non-intrusive techniques for establishing occupancy related energy savings in commercial buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments* (2016), ACM, pp. 21–30.

[12] Arif, Mohammed, Katafygiotou, Martha, Mazroei, Ahmed, Kaushik, Amit, Elsarrag, Esam, et al. Impact of indoor environmental quality on occupant well-being and comfort: A review of the literature. *International Journal of Sustainable Built Environment 5*, 1 (2016), 1–11.

[13] Ashbrook, Daniel, and Starner, Thad. Using gps to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Comput. 7*, 5 (Oct. 2003), 275–286.

[14] Axhausen, Kay W., Zimmermann, Andrea, Schönfelder, Stefan, Rindsfüser, Guido, and Haupt, Thomas. Observing the rhythms of daily life: A six-week travel diary. *Transportation 29*, 2 (May 2002), 95–124.

[15] Bai, Fan, and Helmy, Ahmed. A survey of mobility models. *Wireless Adhoc Networks. University of Southern California, USA 206* (2004), 147.

[16] Balaji, B., Xu, J., Nwokafor, A., Gupta, R., and Agarwal, Y. Sentinel: Occupancy Based HVAC Actuation Using Existing WiFi Infrastructure Within Commercial Buildings. In *SenSys* (November 2013).

[17] Balan, Rajesh Krishna, Nguyen, Khoa Xuan, and Jiang, Lingxiao. Real-time trip information service for a large taxi fleet. In *Proceedings of the 9th international conference on Mobile systems, applications, and services* (2011), ACM, pp. 99–112.

[18] Balazinska, Magdalena, and Castro, Paul. Characterizing mobility and network usage in a corporate wireless local-area network. In *Proceedings of the 1st international conference on Mobile systems, applications and services* (2003), ACM, pp. 303–316.

[19] Bay, Jason, Kek, Joel, Tan, Alvin, Hau, Chai Sheng, Yongquan, Lai, Tan, Janice, and Quy, Tang Anh. Bluetrace: A privacy-preserving protocol for community-driven contact tracing across borders. *Government Technology Agency-Singapore, Tech. Rep* (2020).

[20] Becker, Richard, Cáceres, Ramón, Hanson, Karrie, Isaacman, Sibren, Loh, Ji Meng, Martonosi, Margaret, Rowland, James, Urbanek, Simon, Varshavsky, Alexander, and Volinsky, Chris. Human mobility characterization from cellular network data. *Communications of the ACM 56*, 1 (2013), 74–82.

[21] Beltran, Alex, Erickson, Varick L., and Cerpa, Alberto E. Thermosense: Occupancy thermal based sensing for HVAC control. In *BuildSys* (2013), ACM, pp. 11:1–11:8.

[22] Berke, Alex, Bakker, Michiel, Vepakomma, Praneeth, Larson, Kent, and Pentland, Alex 'Sandy'. Assessing disease exposure risk with location data: A proposal for cryptographic preservation of privacy, 2020.

[23] Bettstetter, Christian, Hartenstein, Hannes, and Pérez-Costa, Xavier. Stochastic properties of the random waypoint mobility model. *Wirel. Netw. 10*, 5 (Sept. 2004), 555–567.

[24] Camp, Tracy, Boleng, Jeff, and Davies, Vanessa. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing 2*, 5 (2002), 483–502.

[25] Cho, Eunjoon, Myers, Seth A., and Leskovec, Jure. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2011), KDD '11, ACM, pp. 1082–1090.

[26] Cho, Hyunghoon, Ippolito, Daphne, and Yu, Yun William. Contact tracing mobile apps for covid-19: Privacy considerations and related trade-offs, 2020.

[27] Clear, Adrian, Friday, Adrian, Hazas, Mike, and Lord, Carolynne. Catch my drift?: Achieving comfort more sustainably in conventionally heated buildings. In *Proceedings of the 2014 Conference on Designing Interactive Systems* (New York, NY, USA, 2014), DIS '14, ACM, pp. 1015–1024.

[28] Dar, Aaqib Bashir, Lone, Auqib Hamid, Zahoor, Saniya, Khan, Afshan Amin, and Naaz, Roohie. Applicability of mobile contact tracing in fighting pandemic (covid-19): Issues, challenges and solutions. Cryptology ePrint Archive, Report 2020/484, 2020. `https://eprint.iacr.org/2020/484`.

[29] Davies, Vanessa Ann, et al. Evaluating mobility models within an ad hoc network. Master's thesis, Citeseer, 2000.

[30] Division, Development Finance. *MOEF: Korea Contact Tracing*, 2020.

[31] Do, Trinh Minh Tri, and Gatica-Perez, Daniel. Contextual conditional models for smartphone-based human mobility prediction. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (New York, NY, USA, 2012), UbiComp '12, ACM, pp. 163–172.

[32] D'Silva, Krittika, Noulas, Anastasios, Musolesi, Mirco, Mascolo, Cecilia, and Sklar, Max. If i build it, will they come?: Predicting new venue visitation patterns through mobility data. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2017), ACM, p. 54.

[33] Eames, Ken TD, and Keeling, Matt J. Contact tracing and disease control. *Proceedings of the Royal Society of London. Series B: Biological Sciences 270*, 1533 (2003), 2565–2571.

[34] Erickson, V., Carreira-Perpinan, M., and Cerpa, A. OBSERVE: Occupancy-based System for Efficient Reduction of HVAC Energy. In *IPSN* (April 2011).

[35] et. al., Ramesh Raskar. Apps gone rogue: Maintaining personal privacy in an epidemic, 2020.

[36] Farrahi, K., and Emonet, R. andCebrian, M. Epidemic contact tracing via communication traces. *PloS one 9*, 5 (2014).

[37] Feng, Jie, Li, Yong, Zhang, Chao, Sun, Funing, Meng, Fanchao, Guo, Ang, and Jin, Depeng. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference* (Republic and Canton of Geneva, Switzerland, 2018), WWW '18, International World Wide Web Conferences Steering Committee, pp. 1459–1468.

[38] Ferretti, Luca, Wymant, Chris, Kendall, Michelle, Zhao, Lele, Nurtay, Anel, Abeler-Dörner, Lucie, Parker, Michael, Bonsall, David, and Fraser, Christophe. Quantifying sars-cov-2 transmission suggests epidemic control with digital contact tracing. *Science 368*, 6491 (2020).

[39] for Disease Control, Center. *Contact Tracing: Using Digital Tools*, 2020. `https://www.cdc.gov/coronavirus/2019-ncov/downloads/digital-contact-tracing.pdf`.

[40] for Disease Control, Center. *Coronavirus Disease 2019 (COVID-19) Appendices*, 2020. `https://www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing/contact-tracing-plan/appendix.html`.

[41] Gambs, Sébastien, Killijian, Marc-Olivier, and del Prado Cortez, Miguel Núñez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility* (2012), ACM, p. 3.

[42] Ganti, Raghu, Srivatsa, Mudhakar, Ranganathan, Anand, and Han, Jiawei. Inferring human mobility patterns from taxicab location traces. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (New York, NY, USA, 2013), UbiComp '13, ACM, pp. 459–468.

[43] Gao, Peter Xiang, and Keshav, Srinivasan. SPOT: a smart personalized office thermal control system. In *e-Energy* (2013), ACM, pp. 237–246.

[44] Ghai, Sunil Kumar, Thanayankizil, Lakshmi V, Seetharam, Deva P, and Chakraborty, Dipanjan. Occupancy detection in commercial buildings using opportunistic context sources. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on* (2012), IEEE, pp. 463–466.

[45] Gidófalvi, Győző, and Dong, Fang. When and where next: individual mobility prediction. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems* (2012), ACM, pp. 57–64.

[46] Gonzalez, Marta C., Hidalgo, Cesar A., and Barabasi, Albert-Laszlo. Understanding individual human mobility patterns. *Nature 453*, 7196 (June 2008), 779–782.

[47] Gupta, Peeyush, Mehrotra, Sharad, Panwar, Nisha, Sharma, Shantanu, Venkatasubramanian, Nalini, and Wang, Guoxi. Quest: Practical and oblivious mitigation strategies for covid-19 using wifi datasets, 2020.

[48] Hang, Mengyue, Pytlarz, Ian, and Neville, Jennifer. Exploring student check-in behavior for improved point-of-interest prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018), ACM, pp. 321–330.

[49] Hasan, Samiul, Zhan, Xianyuan, and Ukkusuri, Satish V. Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing* (2013), ACM, p. 6.

[50] Henderson, Tristan, Kotz, David, and Abyzov, Ilya. The changing usage of a mature campus-wide wireless network. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2004), MobiCom '04, ACM, pp. 187–201.

[51] Huang, Lian, Li, Qingquan, and Yue, Yang. Activity identification from gps trajectories using spatial temporal pois' attractiveness. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on location based social networks* (2010), ACM, pp. 27–30.

[52] Isaacman, Sibren, Becker, Richard, Cáceres, Ramón, Martonosi, Margaret, Rowland, James, Varshavsky, Alexander, and Willinger, Walter. Human mobility modeling at metropolitan scales. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (2012), Acm, pp. 239–252.

[53] Iyengar, Srinivasan, Kalra, Sandeep, Ghosh, Anushree, Irwin, David E., Shenoy, Prashant J., and Marlin, Benjamin. iprogram: Inferring smart schedules for dumb thermostats. In *BuildSys* (2015), ACM, pp. 211–220.

[54] Jaisinghani, Dheryta, Balan, Rajesh Krishna, Naik, Vinayak, Misra, Archan, and Lee, Youngki. Experiences & challenges with server-side wifi indoor localization using existing infrastructure. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous New York City, NY* (2018), Henning Schulzrinne and Pan Li, Eds., pp. 226–235.

[55] Jayarajah, K., Lee, S Youngki, Misra, A., and Balan, Rajesh Krishna. Need accurate user behaviour?: pay attention to groups. In *Proceedings of the 2015 ACM Conference on Pervasive and Ubiquitous Computing (UbiComp15), Osaka, Japan* (September 2015).

[56] Jayarajah, K., and Misra, A. Predicting episodes of non-conformant mobility in indoor environments. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Volume 2 Issue 4* (December 2018).

[57] Jayarajah, Kasthuri, and Misra, Archan. Predicting episodes of non-conformant mobility in indoor environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2*, 4 (2018), 1–24.

[58] Jee, Charlotte. Is a successful contact tracing app possible? these countries think so., 2020.

[59] Jiang, Renhe, Song, Xuan, Fan, Zipei, Xia, Tianqi, Chen, Quanjun, Chen, Qi, and Shibasaki, Ryosuke. Deep roi-based modeling for urban human mobility prediction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2*, 1 (2018), 14.

[60] Jiang, Shan, Ferreira, Joseph, and González, Marta C. Clustering daily patterns of human activities in the city. *Data Mining and Knowledge Discovery 25*, 3 (Nov 2012), 478–510.

[61] Jurdak, Raja, Zhao, Kun, Liu, Jiajun, AbouJaoude, Maurice, Cameron, Mark, and Newth, David. Understanding human mobility from twitter. *PloS one 10*, 7 (2015), e0131469.

[62] Karmakar, G, Kabra, A, and Ramamritaham, K. Maintaining thermal comfort in buildings: feasibility, algorithms, implementation, evaluation. *Real-Time Systems 51*, 5 (2015), 485–525.

[63] Kim, Minkyong, Kotz, David, and Kim, Songkuk. Extracting a mobility model from real user traces. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings* (2006), IEEE, pp. 1–13.

[64] Kingma, Diederik P., and Ba, Jimmy Lei. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015), 1–15.

[65] Kleiminger, W., Beckel, C., Staake, T., and Santini, S. Occupancy Detection from Electricity Consumption Data. In *BuildSys* (November 2013).

[66] Kleiminger, Wilhelm, Santini, Silvia, and Mattern, Friedemann. Smart heating control with occupancy prediction: How much can one save? In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (New York, NY, USA, 2014), UbiComp '14 Adjunct, ACM, pp. 947–954.

[67] Kleinrock, Leonard. Nomadicity: Anytime, anywhere in a disconnected world. *Mob. Netw. Appl. 1*, 4 (Dec. 1996), 351–357.

[68] Klepeis, Neil E, Nelson, William C, Ott, Wayne R, Robinson, John P, Tsang, Andy M, Switzer, Paul, Behar, Joseph V, Hern, Stephen C, and Engelmann, William H. The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environmental Epidemiology 11*, 3 (2001), 231–252.

[69] Kotz, David, and Essien, Kobby. Analysis of a campus-wide wireless network. *Wireless Networks 11*, 1-2 (2005), 115–133.

[70] Lam, Abraham Hang-Yat, Yuan, Yi, and Wang, Dan. An occupant-participatory approach for thermal comfort enhancement and energy conservation in buildings. In *e-Energy* (2014), ACM, pp. 133–143.

[71] Lee, Jong-Kwon, and Hou, Jennifer C. Modeling steady-state and transient behaviors of user mobility: Formulation, analysis, and application. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New York, NY, USA, 2006), MobiHoc '06, ACM, pp. 85–96.

[72] Lee, Vernon J, Chiew, Calvin J, and Khong, Wei Xin. Interrupting transmission of COVID-19: lessons from containment efforts in Singapore. *Journal of Travel Medicine* (03 2020). taaa039.

[73] Liao, Dongliang, Liu, Weiqing, Zhong, Yuan, Li, Jing, and Wang, Guowei. Predicting activity and location with multi-task context aware recurrent neural network. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (2018), IJCAI'18, AAAI Press, pp. 3435–3441.

[74] Lin, Miao, Hsu, Wen-Jing, and Lee, Zhuo Qi. Predictability of individuals' mobility with high-resolution positioning data. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (New York, NY, USA, 2012), UbiComp '12, ACM, pp. 381–390.

[75] Lin, Ziheng, Yin, Mogeng, Feygin, Sidney, Sheehan, Madeleine, Paiement, Jean-Francois, and Pozdnoukhov, Alexei. Deep generative models of urban mobility. *IEEE Transactions on Intelligent Transportation Systems* (2017).

[76] Liu, Qiang, Wu, Shu, Wang, Liang, and Tan, Tieniu. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI* (2016), pp. 194–200.

[77] Lu, J., Sookoor, T., Srinivasan, V., Gao, G., Holben, B., Stankovic, J., Field, E., and Whitehouse, K. The Smart Thermostat: Using Occupancy Sensors to Save Energy in Homes. In *SenSys* (November 2010).

[78] Mahmood Khan, U., Kabir, Z., and Hassan, S. A. Wireless health monitoring using passive wifi sensing. In *Proc. 13th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)* (2017), pp. 1771–1776.

[79] Mathew, Wesley, Raposo, Ruben, and Martins, Bruno. Predicting future locations with hidden markov models. In *Proceedings of the 2012 ACM conference on ubiquitous computing* (2012), ACM, pp. 911–918.

[80] Melfi, R., Rosenblum, B., Nordman, B., and Christensen, K. Measuring building occupancy using existing network infrastructure. In *Proceedings of the 2011 International Green Computing Conference and Workshops* (Washington, DC, USA, 2011), IGCC '11, IEEE Computer Society, pp. 1–8.

[81] Milenkovic, Marija, and Amft, Oliver. An opportunistic activity-sensing approach to save energy in office buildings. In *Proceedings of the Fourth International Conference on Future Energy Systems* (New York, NY, USA, 2013), e-Energy '13, ACM, pp. 247–258.

[82] Mok, Esmond, and Retscher, Günther. Location determination using wifi fingerprinting versus wifi trilateration. *Journal of Location Based Services 1*, 2 (2007), 145–159.

[83] Networks, Aruba. *IT Analytics for Operational Intelligence*, 2020. `https://www.arubanetworks.com/products/location-services/analytics/`.

[84] Newsham, G.R., and Birt, B.J. Building-level Occupancy Data to Improve ARIMA-based Electricity Use Forecasts. In *BuildSys* (2010).

[85] Nguyen, Cong T., Saputra, Yuris Mulya, Huynh, Nguyen Van, Nguyen, Ngoc-Tan, Khoa, Tran Viet, Tuan, Bui Minh, Nguyen, Diep N., Hoang, Dinh Thai, Vu, Thang X., Dutkiewicz, Eryk, Chatzinotas, Symeon, and Ottersten, Bjorn. Enabling and emerging technologies for social distancing: A comprehensive survey, 2020.

[86] Nishiura, Hiroshi, Oshitani, Hitoshi, Kobayashi, Tetsuro, Saito, Tomoya, Sunagawa, Tomimasa, Matsui, Tamano, Wakita, Takaji, COVID, MHLW, and Suzuki, Motoi. Closed environments facilitate secondary transmission of coronavirus disease 2019 (covid-19). *MedRxiv* (2020).

[87] Noulas, Anastasios, Scellato, Salvatore, Lambiotte, Renaud, Pontil, Massimiliano, and Mascolo, Cecilia. A tale of many cities: universal patterns in human urban mobility. *PloS one 7*, 5 (2012), e37027.

[88] Oliver, Nuria, Letouzé, Emmanuel, Sterly, Harald, Delataille, Sébastien, Nadai, Marco De, Lepri, Bruno, Lambiotte, Renaud, Benjamins, Richard, Cattuto, Ciro, Colizza, Vittoria, de Cordes, Nicolas, Fraiberger, Samuel P., Koebe, Till, Lehmann, Sune, Murillo, Juan, Pentland, Alex, Pham, Phuong N, Pivetta, Frédéric, Salah, Albert Ali, Saramäki, Jari, Scarpino, Samuel V., Tizzoni, Michele, Verhulst, Stefaan, and Vinck, Patrick. Mobile phone data and covid-19: Missing an opportunity?, 2020.

[89] pandas development team, The. pandas-dev/pandas: Pandas, Feb. 2020.

[90] Pappalardo, Luca, and Simini, Filippo. Modelling individual routines and spatio-temporal trajectories in human mobility. *CoRR abs/1607.05952* (2016).

[91] Pisharoty, Devika, Yang, Rayoung, Newman, Mark W., and Whitehouse, Kamin. Thermocoach: Reducing home energy consumption with personalized thermostat recommendations. In *BuildSys* (2015), ACM, pp. 201–210.

[92] Prakash, A., Prakash, V., Doshi, B., Arote, U., Sahu, P., and Ramamritham, K. Fusing Sensors for Occupancy Sensing in Smart Buildings. In *ICDCIT* (February 2015).

[93] Qiao, Yuanyuan, Si, Zhongwei, Zhang, Yanting, Abdesslem, Fehmi Ben, Zhang, Xinyu, and Yang, Jie. A hybrid markov-based model for human mobility prediction. *Neurocomputing 278* (2018), 99–109.

[94] Radford, Alec, Wu, Jeffrey, Child, Rewon, Luan, David, Amodei, Dario, and Sutskever, Ilya. Language Models are Unsupervised Multitask Learners. *arXiv* (2019).

[95] Rhee, Injong, Shin, Minsu, Hong, Seongik, Lee, Kyunghan, Kim, Seong Joon, and Chong, Song. On the levy-walk nature of human mobility. *IEEE/ACM transactions on networking (TON) 19*, 3 (2011), 630–643.

[96] Salathé, Marcel, Althaus, Christian L, Neher, Richard, Stringhini, Silvia, Hodcroft, Emma, Fellay, Jacques, Zwahlen, Marcel, Senti, Gabriela, Battegay, Manuel, Wilder-Smith, Annelies, et al. Covid-19 epidemic in switzerland: on the importance of testing, contact tracing and isolation. *Swiss medical weekly 150*, 11-12 (2020), w20225.

[97] Song, Chaoming, Koren, Tal, Wang, Pu, and Barabási, Albert-László. Modelling the scaling properties of human mobility. *Nature Physics 6*, 10 (2010), 818.

[98] Song, Xuan, Kanasugi, Hiroshi, and Shibasaki, Ryosuke. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *IJCAI* (2016), vol. 16, pp. 2618–2624.

[99] Systems, Cisco. *Cisco DNA Spaces*, 2020. `https://www.cisco.com/c/en/us/solutions/enterprise-networks/dna-spaces/index.html`.

[100] Tang, Jinjun, Liu, Fang, Wang, Yinhai, and Wang, Hua. Uncovering urban human mobility from large scale taxi gps data. *Physica A: Statistical Mechanics and its Applications 438* (2015), 140–153.

[101] Tang, Qiang. Privacy-preserving contact tracing: current solutions and open questions, 2020.

[102] Tapia, E., Intille, S., and Larson, K. Activity Recognition in the Home Setting using Simple and Ubiquitous Sensors. In *Pervasive* (2004).

[103] Ting, K., Yu, R., and Srivastava, M. Occupancy Inferencing from Non-intrusive Data Sources. In *BuildSys* (November 2013).

[104] Trivedi, Amee, Gummeson, Jeremy, and Shenoy, Prashant. Empirical characterization of mobility of multi-device internet users, 2020.

[105] Van Doremalen, Neeltje, Bushmaker, Trenton, Morris, Dylan H, Holbrook, Myndi G, Gamble, Amandine, Williamson, Brandi N, Tamin, Azaibi, Harcourt, Jennifer L, Thornburg, Natalie J, Gerber, Susan I, et al. Aerosol and surface stability of sars-cov-2 as compared with sars-cov-1. *New England Journal of Medicine 382*, 16 (2020), 1564–1567.

[106] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Lukasz, and Polosukhin, Illia. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[107] Veloso, Marco, Phithakkitnukoon, Santi, and Bento, Carlos. Urban mobility study using taxi traces. In *Proceedings of the 2011 international workshop on Trajectory data mining and analysis* (2011), ACM, pp. 23–30.

[108] Wang, Haozhou, Su, Han, Zheng, Kai, Sadiq, Shazia, and Zhou, Xiaofang. An effectiveness study on trajectory similarity measures. In *Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137* (2013), Australian Computer Society, Inc., pp. 13–22.

[109] Weppner, Jens, Bischke, Benjamin, and Lukowicz, Paul. Monitoring crowd condition in public spaces by tracking mobile consumer devices with wifi interface. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (2016), pp. 1363–1371.

[110] Wolf, Thomas, Debut, Lysandre, Sanh, Victor, Chaumond, Julien, Delangue, Clement, Moi, Anthony, Cistac, Pierric, Rault, Tim, Louf, Rémi, Funtowicz, Morgan, Davison, Joe, Shleifer, Sam, von Platen, Patrick, Ma, Clara, Jernite, Yacine, Plu, Julien, Xu, Canwen, Scao, Teven Le, Gugger, Sylvain, Drame, Mariama, Lhoest, Quentin, and Rush, Alexander M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Online, Oct. 2020), Association for Computational Linguistics.

[111] WorldOmeter. *Covid19 User infected worldwide*, 2020.

[112] Wu, Hao, Chen, Ziyang, Sun, Weiwei, Zheng, Baihua, and Wang, Wei. Modeling trajectories with recurrent neural networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17* (2017), pp. 3083–3090.

[113] Xi, Wei, Zhao, Jizhong, Li, Xiang-Yang, Zhao, Kun, Tang, Shaojie, Liu, Xue, and Jiang, Zhiping. Electronic frog eye: Counting crowd using wifi. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications* (2014), IEEE, pp. 361–369.

[114] Xia, Feng, Wang, Jinzhong, Kong, Xiangjie, Wang, Zhibo, Li, Jianxin, and Liu, Chengfei. Exploring human mobility patterns in urban scenarios: A trajectory data perspective. *IEEE Communications Magazine 56*, 3 (2018), 142–149.

[115] Xia, Ye, and Lee, Gwendolyn. How to return to normalcy: Fast and comprehensive contact tracing of covid-19 through proximity sensing using mobile devices, 2020.

[116] Yang, Dingqi, Zhang, Daqing, Zheng, Vincent W, and Yu, Zhiyong. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems 45*, 1 (2015), 129–142.

[117] Yuan, Jing, Zheng, Yu, and Xie, Xing. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (2012), ACM, pp. 186–194.

[118] Zakaria, Camellia, Balan, Rajesh, and Lee, Youngki. Stressmon: Scalable detection of perceived stress and depression using passive sensing of changes in work routines and group interactions. *Proc. ACM Hum.-Comput. Interact. 3*, CSCW (Nov. 2019).

[119] Zeng, Yunze, Pathak, Parth H., and Mohapatra, Prasant. Analyzing shopper's behavior through wifi signals. In *Proceedings of the 2nd ACM Workshop on Workshop on Physical Analytics* (2015), p. 13–18.

[120] Zhang, Desheng, Huang, Jun, Li, Ye, Zhang, Fan, Xu, Chengzhong, and He, Tian. Exploring human mobility with multi-source data at extremely large metropolitan scales. In *Proceedings of the 20th annual international conference on Mobile computing and networking* (2014), ACM, pp. 201–212.

[121] Zhang, Hanyuan, Wu, Hao, Sun, Weiwei, and Zheng, Baihua. Deeptravel: a neural network based travel time estimation model with auxiliary supervision. *arXiv preprint arXiv:1802.02147* (2018).

[122] Zhao, Kai, Tarkoma, Sasu, Liu, Siyuan, and Vo, Huy. Urban human mobility data mining: An overview. In *Big Data (Big Data), 2016 IEEE International Conference on* (2016), IEEE, pp. 1911–1920.

[123] Zheng, Yu, Li, Quannan, Chen, Yukun, Xie, Xing, and Ma, Wei-Ying. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing* (2008), ACM, pp. 312–321.

[124] Zheng, Zimu, Wang, Feng, Wang, Dan, and Zhang, Liang. Buildings affect mobile patterns: Developing a new urban mobility model. In *Proceedings of the 5th Conference on Systems for Built Environments* (New York, NY, USA, 2018), BuildSys '18, ACM, pp. 83–92.

[125] Zhou, Mengyu, Pei, Dan, Sui, Kaixin, and Moscibroda, Thomas. Mining crowd mobility and wifi hotspots on a densely-populated campus. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers* (2017), ACM, pp. 427–431.

[126] Zhou, Z., Wu, C., Yang, Z., and Liu, Y. Sensorless sensing with wifi. *Tsinghua Science and Technology 20*, 1 (2015), 1–6.