INFERRING SOCIAL NETWORKS FROM PASSIVELY COLLECTED WI-FI METADATA

A Thesis

by

MANDEL OATS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Jean-Francois Chamberland
Committee Members,	Krishna Narayanan
	Pierce Cantrell
	Natarajan Gautam
Head of Department,	Miroslav M. Begovic

December 2017

Major Subject: Electrical Engineering

ABSTRACT

The emergence of smartphones and other highly portable Wi-Fi enabled devices offers unprecedented amounts of information leaked through Wi-Fi metadata. The constantly connected nature of Wi-Fi devices together with the intimate relationship between users and their device presents an opportunity for using a user's device to gain information about the user themselves. Through passive data collection, without interference or the possibility of being detected, it is possible to harvest large datasets. This work looks at the possibility of inferring underlying social networks through the analysis of these metadata traces. Using spatiotemporal proximity as an indicator of friendship, findings demonstrate the ability to accurately predict underlying social structures in various simulated settings.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professors Jean-Francois Chamberland, Krishna Narayanan, and Pierce Cantrell of the Department of Electrical and Computer Engineering and Professor Natarajan Gautam of the Department of Industrial and Systems Engineering at Texas A&M University.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported by an assistantship from Texas A&M University.

NOMENCLATURE

MAC	Media Access Control
RSSI	Received Signal Strength Indicator
WLAN	Wireless Local Area Network
GHz	Gigahertz
MHz	Megahertz
BSS	Base Service Set
BSSID	Base Service Set Identifier
SSID	Service Set Identifier
ESS	Extended Service Set
HK	Holme-Kim
BA	Barabasi and Albert
PDA	Personal Digital Assistant
DTN	Delay Tolerant Network
MSN	Mobile Social Network
PNL	Preferred Netowrk List
USB	Universial Serial Bus
NUC	Next Unit Computing
ICT	Intercontact Time
GeSoMo	Generalized Social Mobility Model

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

1. INTRODUCTION

In recent years, the proliferation of Wi-Fi infrastructures paired with the ubiquitous use of mobile devices has created a near-constant use of this technology. Every active mobile device today is essentially connected to a wireless access point or searching for one. This reality, along with the intimate relationship between a user and their mobile device, suggests a pathway for leveraging Wi-Fi interactions to infer information about a user from their device. To demonstrate this new reality, we employ a distributed network of sensors to harvest metadata in a Wi-Fi setting. These sensors monitor Wi-Fi packets within the area, collecting time-stamped media access control (MAC) addresses and received signal strength indicator (RSSI) values. This monitoring is performed inconspicuously, without network access and without interrupting the user's Wi-Fi connection. After collecting metadata, we format samples into useful datasets to subsequently perform various inference tasks. The goals of this thesis are to develop a sensor prototype and methodology for the passive collection of Wi-Fi metadata, and to explore this information as a viable data source for the inference of underlying social networks. This is accomplished by using simulated data to test the effects of sensor network design on the performance of inference algorithms.

1.1 Background

The problem can be formulated as estimating the underlying social network of a group of active Wi-Fi users operating within a prescribed area as they interact with one another over time. The primary objective is to identify which observed pairs of users are connected socially and which are not, thusly identifying the underlying social structure of the users. We begin by monitoring an area of interest, and the users within, for a reasonable amount of time (on the magnitude of hours, days, or weeks). This is done using a distributed network of sensors. The collected data is then merged into a single dataset. From this dataset, we aim to identify relationships by observing the spatiotemporal proximity and movement of pairs of users. In the envisioned scenario, a matrix X of several features can be extracted from these observations and utilized to learn the dyadic relationships of observed users.

1.1.1 Wi-Fi Technology

A brief exposition of the Wi-Fi protocol and some of its components is helpful to understand the work presented in this thesis. The following section offers a review of pertinent points.

Wi-Fi is a wireless local area network (WLAN) technology defined in the 802.11 standard. Our problem setting is that of one or more Wi-Fi hotspots in an arbitrary area. This could be an office building, a laboratory, a college campus or even a city. Within this area there are three main components: the access points, client devices, and sensors. An access point is a piece of networking hardware that allows a Wi-Fi device to connect to a wired network. The access point facilitates the transfer of data from the device to the wired network via 2.4 GHz or 5 GHz radio bands. A client device can be any piece of networking hardware that connects to an access point. These can be mobile like a smartphone or laptop, or stationary like a desktop computer or printer. Access points are usually stationary while client devices are free to roam the covered area without losing network connection. A single access point and the devices connected to it form a base service set (BSS). A BSS is identified by a BSSID (base service set identifier). Yet, due to the range and connection limitations of access points, they are often deployed in groups. A group of

access points sharing the same service set identifier (SSID) and the associated client devices is called an extended service set (ESS). Mobile devices are free to roam the entire coverage area of an ESS, hopping between access points within the ESS without losing connection. An example of this is a campus wide Wi-Fi that covers an entire university within a single ESS. Our research requires a closer look at how information is passed between devices and access points within this Wi-Fi setting.

We focus on the data link layer of information transfer between the device and the access point. At this stage, information is transferred in what are known as frames. See figure 1.1. These frames have two parts: the header and the payload. The packet contains data and network layer routing information, but our focus lies within the frame header. This frame header contains the information required for the transfer of data between the device and access point within the BSS. This includes the network SSID, a source MAC address, a destination MAC address, signal strength, signal noise levels, and other available information depending on what is provided by the network card driver. The information contained in the header is often referred to as packet metadata.

The AP/device connection is dependent on this packet metadata. This metadata and its contents have several important properties that our data collection takes advantage of. First, metadata is typically unencrypted. This means that, given a frame, the metadata for the packet stored in the frame header can be easily an quickly interpreted. The metadata also contains two MAC addresses, one for the source (origin) of the packet and one for the destination of the packet within the WLAN. MAC addresses are 48 bit identifiers assigned to each piece of network hardware by the hardware manufacturer; they are often written in hexadecimal format, e.g., `08:00:08:15:CA:FE`. In the simplest case, when a device receives a packet, it checks if the destination MAC address matches its own. When these two addresses do not match, the device knows that the packet is intended for someone else and it therefore ignores the packet. MAC addresses are of interest to our work because they are globally unique. Every network capable device comes with a MAC address from the manufacturer, and that device can be identified by that MAC address universally. There is a caveat that a different non-unique MAC address can be locally assigned to a client device, however

this is detectable, and rare enough to not be of concern.

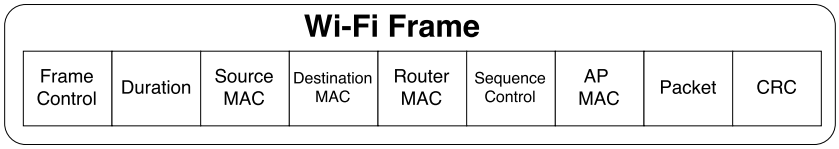

Figure 1.1: This figure shows the structure of a typical 802.11 frame.

Apart from typical frames employed to transmit data between an AP and device, there are several management frames used to establish and maintain connections in the WLAN network. To establish a connection, client devices can use one of two methods: passive or active scanning. In the passive method, a device passively listens for available access points to broadcast beacons. Beacons are frames broadcast to indicate network availability and contain information that the client device uses to make a connection. The drawback of this method is that the client device Wi-Fi radio interface must be always on and listening. The active method is when the client device periodically broadcasts a probe request frame. If an AP is in the area of the broadcast it responds with a probe response frame, and a connection is established. This method is common in power limited mobile devices because the Wi-Fi can be active in short periodic bursts rather than being on continuously. These probe request frames are periodically broadcast from the client regardless if a Wi-Fi network is available or not. This is particularly important to our research because it insures that devices in our monitored area will always be sending frames even if the device is not actively being used. In fact, because devices send probe requests regardless of network availability, our proposed technical approach is not limited to Wi-Fi hotspots. Still, due to the fact that probe requests only happen every 15–90 seconds when a device spends significant time without Wi-Fi access, the granularity and abundance of useful data may decline drastically under such circumstances.

In this initiative, we explore techniques based on the harvesting of metadata from frames in a Wi-Fi hotspot. If we were in control of said hotspot this would be as trivial as monitoring the traffic

through the access points. However, we aim to harvest frames without any previous knowledge or access to the Wi-Fi network using sensors. This approach generalizes our setting to any Wi-Fi network. Given that any network is potentially not ours to interfere with, we aim to do our data collection as inconspicuously as possible, without detection or network performance degradation.

1.1.2 Social Networks

At the same time, this research initiative relies heavily on social networks. Research into the way people interact within these networks influence the metrics by which we evaluate aspects of our work, and underlie several design decisions in building the system. Therefore a background in social networking theory is necessary.

Social networks are made up of individual people and social ties. Social ties are defined as any meaningful social relationship between two people. The social tie between two individuals can have varying levels of strength. Attempts to categorize these strengths result in terms like family, friendship, and acquaintance. A social neighbor in this research is two individuals for which any social tie exists. An individual in a network can have many social neighbors; however, for simplicity, it is assumed that the tie is symmetric, i.e., $u \in \text{neighbors}(v) \Rightarrow v \in \text{neighbors}(u)$. The neighborhood of node u is then the collection of all u in $\text{neighbors}(u)$. These social structures can naturally be represented as graphs, and as such many of the terms and techniques for their study are applied directly from graph theory as we will begin to see.

1.1.2.1 Definitions

To discuss the properties of social networks, some basic graph metrics need to be defined. First and foremost, graphs consist of a set of vertices V (or nodes) and a set of edges E. Each edge $e_{u,v} \in E$ connects two nodes $u, v \in V$. In the remainder of this work, each vertex can be assumed to represent a user; and each edge, a social tie. The first graph measurement we define is the average path length. The average path length can be described as the average length of the shortest

path between every pair of nodes in a graph. More formally, the average path length of graph G is

$$L(G(V,E)) = \frac{1}{|V|(|V|-1)} \sum_{u,v \in V} \text{dist}(u,v) \tag{1.1}$$

The next metric is the local clustering coefficient. It is a measure of the extent that a users friends are also friends. Given a neighborhood N_u for node u, the local clustering coefficient captures how close the neighborhood is to being a clique (complete graph). This is calculated as follows:

$$C(v) = \frac{2|e_{u,w} : u, w \in N_v, e_{u,w} \in E|}{\deg(v)(\deg(v)-1)}, \tag{1.2}$$

with the clustering coefficient for the entire graph G just being the average over all local clustering coefficients,

$$C(G(V,E)) = \frac{1}{|V|} \sum_{v \in V} C(v). \tag{1.3}$$

1.1.2.2 Properties of Social Networks

The following properties capture some commonly observed aspects of social networks.

- **Small World Property:** Social networks are observed to exhibit the small world property. This property states that a network has a small average path length relative to the size of the network. This translates to the graph having many cliques or near cliques leading to a high clustering coefficient. This property was made famous by the six degrees of freedom result of Milgram's 1967 experiment [1]. In this experiment he passed out letters to random participant with the goal of delivering the letter to a recipient in Boston. The constraint was that you could only forward the letter to people with whom you were on a first name basis with. While only a fraction of the letters made it to the destination, of the ones received it was observed that it only took 5.5 forwards on average. This result was later repeated in several different forms and popularized under the moniker of "six degrees of separation" found throughout popular culture. Small-worldness can be quantified through a small-coefficient. This small-coefficient σ is calculated by comparing the average path length and clustering

coefficient of a graph to that of a random graph of equivalent average degree,

$$\sigma_G = \frac{\frac{C_G}{C_r}}{\frac{L_G}{L_r}}. \tag{1.4}$$

If $\sigma_G > 1$, the network is said to be small world.

- **Scale Free Networks:** Social networks are also frequently observed to exhibit the scale free property. This property states that the degree distribution of nodes tends to follow a power law distribution as opposed to the poisson distribution expected from a random network. What this means is that few nodes known as hubs have many neighbors while the majority of nodes have much smaller degrees.

- **Community Structure:** Social networks often consist of many communities. A community within this context is a highly connected clustering of nodes that are closely linked to one another either through direct ties or ties between other nodes. Communities can be identified by the social relationship or common interest that connects the users. For example, within a social network a family may be a community, or a book club, sports team, or college alumni can be a community. Communities within a social network are often hierarchical with several communities making up a larger community. The community structures are theorized to be what leads social networks to the high clustering coefficients, which are common.

1.1.2.3 Social Network Models

The first step in simulating our system is the generation of social networks. We desire to test networks of various sizes and structures to observe the effects that graph topology has on the performance of our sensing infrastructures and proposed inference algorithms. We are unlikely to find empirical data on social networks for every simulation we perform and, as such, it makes sense to generate these networks according to abstract models. Ideally, these generated networks can retain the essential properties of social networks and, hence, provide realistic results. The following social network models are explored for this purpose.

The *connected caveman model* is the simplest model explored. It starts as a caveman model, which consists of a graph of several separate cliques of fully connected nodes. These cliques are referred to as caves drawing the comparison to groups neanderthals living in caves separate from other communities. The caveman graph is generated by providing two inputs: the number of cliques and the clique size with the total number of nodes in the graph obviously being the product of these inputs. To convert the caveman graph to a connected caveman graph, one edge is chosen at random from each clique and is connected to a node in a different clique. This creates a ring structure of connected cliques. These graphs clearly have a high clustering coefficient, however the average path length is linear with the number of nodes. This results in not displaying the small world property.

As the name suggests, the *power law cluster model* aims to output a network conforming to the scale free property. This is done using the Holme-Kim (HK) model [2]. The Holme-Kim model is itself an extension of the Barabasi-Albert (BA) model for growing scale free networks [3]. The BA model grows networks using the following steps:

- **Initial Step:** The model begins with m_0 vertices and no edges.
- **Growth:** One vertex v with m edges is added at each time step t.
- **Preferential Attachment:** Each edge of v is attached to an existing vertex with a probability proportional to its degree. The probability that vertex w is attached to v is then

$$P_w = \frac{\deg(w)}{\sum_{u \in V} \deg(u)}. \tag{1.5}$$

In this model, the growth step is repeated $|V|$ times and, as such, the attachment step is repeated m times for each growth step. This results in a scale free network that exhibits the small world property. However, it was observed that, as $|V| \to \infty$, the clustering coefficient for the graph decays to zero. To combat this phenomenon, the HK model adds an additional Triad Formation step to ensure a high clustering coefficient:

- **Triad Formation:** If an edge between v and w was added in the previous PA step, then add one more edge from v to a randomly chosen neighbor of w. If there remains no pair to connect, i.e., if all neighbors of w are already connected to v, do a preferential attachment step instead.

The *Toivonen model* grows a scale free network in a very similar way to the Holme-Kim model. However the networks developed by the Toivonen network differ significantly from those of the HK model in terms of community structure and assortativity [4]. The steps for generating Toivonen networks are as follows:

1. Start with a seed network of N_0 vertices.
2. Pick on average $m_r \geq 1$ random vertices as initial contacts.
3. Pick on average $m_s \geq 0$ neighbors of each initial contact as secondary contacts.
4. Connect the new vertex to the initial and secondary contacts.
5. Repeat steps 2 to 4 until the network has grown to the desired size.

This results in networks with strong assortative mixing, a high clustering coefficient, and a slow growing average path length. The networks retain some scale free properties, as the degree distribution can be approximated by a power-law followed by an exponential decay. The Toivonen network has been shown to approximate real social networks well.

1.1.3 Machine Learning

Once we have collected our Wi-Fi metadata, our goal is to use machine learning techniques to identify the underlying social structure of device users through analysis. This will be covered in greater detail in the methods section. However, at this point, the following section is useful to establish a basic background and language used throughout this work.

Machine Learning is a broad field encompassing the development and application of different inference and clustering algorithms. Our research focuses on the subset known as classification

algorithms. Classification is the problem of identifying which sub-population an observation belongs to. We look to classify the dyadic relationships of observed devices in order to identify the underlying social structure of the users. Two different approaches to this classification can be taken depending on the data: supervised and unsupervised. In both instances you start with a dataset X. This dataset is a collection of m observations containing n features. A feature can be a continuous measured value (Temperature) or a discrete categorical indicator (Color). This data can be labeled or unlabeled. In a labeled dataset, the sub-population to which the observation belongs is known. In an unlabeled dataset, the subpopulation for each observation is unknown. Supervised learning is done using algorithms that iteratively improve upon or train a model based on labeled data. After training, unobserved examples can be classified based on the learned model. However, if the dataset is unlabeled this approach is not possible. Instead unsupervised methods attempt to separate or cluster observations into discrete, and ideally separable groups based on their features. In our work, we focus on using supervised methods on labeled simulated datasets as a first step.

In a typical deployment of our system, often in depth information about the users is not available, and thus a ground truth for classification purposes is unknown or ambiguous at best. This requires the use of unsupervised methods for social network discovery. Specific methods used will be covered in further sections.

1.2 Literature Review

Mobile devices, including cell phones, PDA's, laptops, and smartphones, have been used as mobile sensors from their inception. These mobile devices are often equipped with an array of technologies that can be used for sensing purposes. This includes cellular, Bluetooth, and Wi-Fi technologies. There has been extensive research into using one or many of these functions for sensing to create useful data sets [5]. The primary inference task of this initiative is to use data collected from mobile devices to accurately infer underlying social networks between users.

The premise beneath this inference goal is that spatiotemporal proximity is an indicator of social relationship [6]. In other words, if users are often in the same place at the same time, they are likely to know one another. This inference is not as straightforward as it seems; at the core of

a social network is the character of the relationship. What defines this relationship is not always consistent or uniform across individuals. While behavioral context, e.g., spending time at work vs meeting for dinner, has been shown to be a good indicator of social ties [7], in-depth information about behavior is not often available to establish a ground truth. This leads to data based on user reports or from specific scenarios where underlying social networks are known [8, 7]. Despite this ambiguity, there have been several research initiatives aimed at tying mobile device data to social networks, with some success [8, 6, 5]. However, these initiatives often use customized software on the devices, bluetooth, or network access to accumulate data. Our research is different in that it aims to gather data opportunistically from devices using Wi-Fi technology, and without user interaction or prior knowledge about the network itself. This is detailed in Chapter 2. Using data collected in this fashion, we aim to explore machine learning techniques and assess their ability to accurately identify social relationships.

1.2.1 Delay Tolerant Networks

Research exploring the indicators of underlying social networks has several applications. One initiative we draw from is focused on the diffusion of information in delay tolerant networks (DTN). Delay tolerant networks are composed of mobile-nodes which, as the name suggests, move about an area. Information in delay tolerant networks is disseminated between the mobile-nodes when they meet at the same time and space. This spatiotemporal collocation in this context is known as an encounter. In the simplest protocol, known as epidemic routing, mobile nodes transfer information to all other mobile-nodes it encounters. The underlying social network of the mobile nodes is of high priority. The rate at which information is disseminated and the final percentage of informed nodes is dependent on this structure. Therefore, extracting this network from collected data can provide valuable information about users in real-world delay tolerant network. Social network indicators can then be explored as possible features for our learning algorithms. Basic indicators include the following items.

- **Friendship Index Based on Encounter Time:** The ratio of total time nodes a and b are encountered to the total time.

- **Friendship Index Based on Encounter Count:** The ratio of the number of encounters of nodes a and b to the total number of encounters for a.

- **Friendship Index Based on Encounter Locations:** The ratio of the number of locations a encounters b compared to the total locations a travels.

We should point out that the authors in [5] also explore more complex indicators like eigen-behavior-vectors [9]. It is also notable in [5], through the examination of empirical WLAN traces collected on various college campuses, that these basic indicators follow exponential distributions, i.e. most mobile nodes pairs do not have large friendship indexes, and are asymmetric $\mathcal{F}(a,b) \neq \mathcal{F}(b,a)$. This influenced both our feature extraction and data cleaning later explored in Chapter 3.

1.2.2 Mobile Social Networks

The advent and ubiquitous use of the Internet has led to the forming of online social networks. What started as simple chatrooms has grown into a huge industry consisting of billions of users and household names such as Facebook, Twitter, and Instagram. These systems are based on social relationships and allow users with similar interests or backgrounds to come together and form countless virtual communities. With the popularity of mobile devices, one of the key initiatives of these companies is an advancement into Mobile Social Networks, a technology based on physical connection as opposed to online connection. Significant research into a distributed MSN architecture similar to delay tolerant networks is being done. In this distributed architecture, recognizing and quantifying the underlying social network is of crucial importance because the dissemination and collection of information is done through social contact. Many methods and algorithms for community detection and information routing within mobile social networks make use of graph based measures like centrality and clustering coefficients. These require that a graph of the social structure be quantifiably known.

In an overview of the architecture and key research challenges of mobile social networks, the authors of [10] compile the major strength indicators of a dyadic relationship between two people

in a mobile social network.

- **Frequency:** The more frequently persons interact with one another, the stronger the social tie.
- **Intimacy:** The time invested into a contact. This is connected to the amount of time since a social tie was created.
- **Recency:** The time elapsed since the last interaction between two people.
- **Regularity:** The consistency and repeatability of interactions.
- **Social Homogeneity:** The degree to which preferences between people are alike, sharing common interests.

These indicators, while slightly more general, are very similar to the Friendship Indexes cited in [5]. Also, we note that the **Social Homogeneity** is driven by the context of the relationship. This is also supported by the findings of [8], where the context of an interaction is vital in allowing relationships to be inferred from mined data with high accuracy.

In addition to the strength indicators, [10] also defines multiple categorizations of social neighbors:

- **Community Member:** This attribute describes two nodes belonging to a common social group. It is characterized by a high number of contacts and long contact duration. Contact patterns are in most cases periodical. For example, the colleagues in an office have a very strict schedule of contacts, since they meet every day while working.
- **Familiar Stranger:** This attribute is characterized by a high number of contacts, but with short contact duration. Contact patterns are again periodical, since this is the underlying cause of the high number of contacts. For example, consider two people that use the train to go to work every day at the same time.

- **Stranger:** This attribute is characterized by a low number of contacts and short contact duration. In this category, there is no periodicity; the patterns could be viewed as purely random.

- **Friend:** This attribute is characterized by a lower number of contacts with longer contact duration. High periodicity is encountered in this category. For instance, two friends tend to meet similar times per week, or per month, depending on their social relationships. The low number of interactions is what differentiates friends from community members, since members of different communities, e.g., living in different cities, could share a friendly relationship.

While our work focuses on the binary classification of dyadic relationships, these categoric definitions are enough that the work presented could be extended to a multiclass problem. Also, these categories of social neighbors could be used to extend a social mobility model, possibly resulting in more realistic traces and results.

1.2.3 Mobility Models

There is also extensive research involving social networks and their effects on human mobility patterns on both small and large scales. While there are a few real data sets available, in order to better study new protocols and various networking strategies there is a need for synthetic traces. Originally these traces were simply modeled as random walks. However, it became evident that human mobility depends heavily on users individual and social behavior as well as the environment. This resulted in an abundance of research into different strategies for accurately modeling human mobility traces. Research into the characteristics that make synthetic models accurate is valuable to our research as it is natural for the same characteristics to mirror themselves as we try to infer social relationships. The authors in [11] break these characteristics into spatial, temporal, and connectivity categories; and present the different ways models adhere to them. The relevant sections of this will be discussed further in the simulation chapter.

1.2.4 Antenna Design

A secondary goal of this initiative is to explore the effects of sensor antenna design on the performance of Wi-Fi monitoring and social network inference. Initially, we intended to incorporate RSSI information into our machine learning algorithms resembling inference work previously done in [12]. However, there were several issues with Wi-Fi inconsistencies (explained here [13]) that we were unable to overcome. Instead, we research the way in which antenna design affects the sampling of Wi-Fi packets in the area. At the limits, a high gain isotropic antenna will sample many packets over a large area, whereas a low gain or directional antenna will sample fewer packets over a confined area. While the results from [6] suggests that the size of sampling area has little effect on the probability of friendship compared to the number of encounters under the proposed Bayesian model, the effects of the sampling size on other potential features remains unseen. These effects of sampling size on the performance of the inference algorithms could lead to application specific antenna designs, dependent on variables like the size of the area and the density of the users. This also opens the door for research into tailoring inference algorithms to incorporate information about antenna radiation patterns.

1.2.5 Other Relevant Works

While most of the relevant research of social network inference is usually in the context of Delay Tolerant Networks or human mobility models, there is some work that is very similar to ours. Barbera's *Signals From the Crowd* paper addresses the same problem of reconstructing social networks based on passively collected Wi-Fi metadata. His work focuses on harvesting Wi-Fi probe request frames, and using these probe requests to build a preferred network list (PNL) of service set identifiers (SSIDs) for each device. After six weeks of data collection, the author analyzed the PNLs of the harvested devices by creating a bipartite graph structure, with users as V_1 and preferred network SSIDs as V_2, and extracting a social graph by accounting for PNL intersections while penalizing popular SSIDs. Instead of spatiotemporal proximity as an indicator of relationship, the analysis relies on the idea that friends will have similar preferred Wi-Fi networks [14].

2. PROBLEM FORMULATION

As previously stated, our engineering problem can be formulated as estimating the underlying social network of a group of active Wi-Fi users operating within a prescribed area as they interact with one another over time. The primary objective is to identify which observed pairs of users are connected socially and which are not, thusly identifying the underlying social structure of the users. In this section, we describe the problem in more detail, highlighting the key elements within the setting, what particular observations are made, as well as end goal of the initiative.

2.1 Setting and Elements

Our proposed setting requires several key elements. One is the area itself, which in the case of our project is assumed to be on the scale of a single Wi-Fi Extended Service Set. While this could be very large, for our purposes we assume one or more neighboring buildings. The users of interest are moving within this area. A user in this context is a person with an active Wi-Fi enabled mobile device which is assumed to be on or near their person. These users are then identifiable due to their associated MAC address. These users are intrinsically connected to one another in a social structure. The connections or ties that make up this structure, while in reality are complex and vary in intensity, are assumed to be binary for our purposes. This means that either two users share a social tie or they do not. The users move about about an area at a constant velocity that is uniform across the entire setting. This is one of the reasons our area is assumed to be within the space of a couple building, as all the users assumed to be walking as their primary means of movement, without accounting for varying methods of transportation and velocities. The users however do not move aimlessly about the area. It is assumed that within the area users move between certain locations spending a considerable amount of time at a location before moving to the next one. This is akin to someone going to their desk for a period, stopping by the break room, and then to a co-workers desk. Assuming this non-random motion is one of the driving factors behind the particular observations we make, as discussed next.

2.2 Observations

The observations we aim to collect are abstractly fairly simple, and are defined on a pairwise level. The first of which is total time spent together. As previously stated, we are assuming that spatiotemporal proximity is an indicator of social relationship. Under this assumption, naturally the total amount of time two users spend within each others company should be collected. Next is the total number of times that two users meet. A meeting (or encounter) in this context is defined as two users coming together after not having been together during the previous collection period. This is separate from total time in that spending more time together does not imply more encounters as the users could just stay together, however they are obviously not uncorrelated. Finally the number of locations at which two users were collocated is also collected, as intuitively two users who sit at a desk together and also go to the break room together are more likely tied that two users who only sit together.

It should be noted that when selecting what to observe, the assumption of non-random movements is of key importance. With the collection strategies explored, there is a chance of incidental contact between users. This would be like two strangers passing each other in the hallway. While this means very little in a social perspective, it is still reported by our system. If motion is assumed random in nature, then all records of collocation are inherently incidental and mean little in terms of social structure.

2.3 Goal

The goal of this project is to evaluate the possibility of using Wi-Fi sensors as a means of data collection for the purpose of inferring social networks in our setting. Explicitly we want to be able to monitor an area using Wi-Fi sensors, make the previously stated observations, and use them to identify the social structure of the observed users. In order to evaluate our approach quantitatively, we also implement a Bluetooth collection strategy which has been previously used for social network estimation, and compare the results of the two methods.

In further detail, since we define our social structure in a binary manner, our evaluations are

done on a pairwise level. In reality social network estimation would be done using clustering methods. This is due to the fact that in depth information about the users is not available, and the resulting datasets are therefore unlabeled. We instead focus on classifying labeled datasets and operate on the assumption that good results from binary classification implies separable data. We believe that clustering based on these same features would trend similarly. Our goal is then to classify each potential edge in the social network as existent or nonexistent, thereby allowing us to estimate a social structure based on the observations. We do this for both collection strategies and analyze the results under various conditions.

3. SIMULATION

We begin by developing a platform consisting of several components to simulate both our Wi-Fi collection strategy and the more common Bluetooth strategy in order to compare the results. Firstly, a mobility model is selected to dictate how nodes move around the area of interest. Because we do not want this motion to be completely random as previously stated, we look to simulate human motion as accurately as possible. This section describes the pertinent characteristics of human mobility and the categories of models which serve to mimic different characteristics. Our selected model is then covered in detail. We discuss how we use the output mobility traces to develop datasets simulating both the Bluetooth and Wi-Fi collections. We go on to explain our classification method before finally we compare the results of the two strategies.

3.1 Simulation Model Selection

3.1.1 Human Mobility Characteristics

Human mobility is a topic of interest in many fields including Sociology, Geography, and Engineering. Due to this widespread interest, there are a variety of empirical datasets available that have been previously collected for research purposes. These records are referred to as mobility traces. These traces log a persons location over time allowing researchers to recreate or analyze their movements. However, these datasets are usually collected for a certain purpose within a particular context; typically, they do not generalize well. This has driven the development of mobility models. These models aim to generate synthetic mobility traces that mimic real human mobility. Through research, statistical features of generic human mobility have been quantified and used to predict user movement. These features are classified into three different categories: spatial, temporal, and connectivity characteristics.

3.1.1.1 Spatial

It has been shown in virtually every quantitative study on the topic that there exists a close relationship between mobility, space, and distance. Often the mobility of users is compared to that

of random motion. It can be observed that human movement is not random, but instead displays many spatial regularities. While there is currently no way of reliably estimating the correlation between space, distance, and mobility, several important features have been discovered in that regard.

3.1.1.2 Temporal

Analysis of human mobility has revealed two major temporal characteristics as well. The first is short return time. It can be observed that throughout time users often visit a small number of locations regularly. For example, a student might go to class, the gym, the library and back home on a regular basis, leading to short predictable return times. Another characteristic is pause time. Instead of constant motion, it is common for users to spend periods of time at rest. These periods are known as pause times, and are prevalent in human mobility.

3.1.1.3 Connectivity

Connectivity naturally plays an important role in human mobility. Still, the extent to which these features characterize mobility in humans has yet to be proven. Regardless, many mobility models take them into account. Prevalent measures of connectivity are described below.

- **Contact Time:** Contact time is the amount of time people spend together.
- **Intercontact Time (ICT):** This measure represents the amount of time between successive contacts between people.
- **Aggregated Intercontact Times:** Since analyzing ICTs on a pairwise level is often difficult due to variability and lack of sufficient data on a given pair, the aggregate distribution of ICTs is often observed instead.
- **Graph-based Analysis:** This type of analysis seeks to link mobility to the properties of underlying social networks or connectivity graphs.

3.1.2 Popular Mobility Models

There are several popular mobility model categories used by researchers in a variety of fields.

- **Map Based Models:** These mobility models use extracted data from real-world mobility traces in an effort to produce similar traces synthetically.

- **Location Based Models:** Location-based models rely on the relationship between a set of preferred locations and users to drive movement within an area.

- **Community Based Models:** This popular model category uses the identification of subnetworks of strongly connected users to support spatial and graph based movement dependencies.

- **Sociological Based Models:** Such models Leverage real observations of human interactions to characterize movement.

Every category of models aims to capture and produce certain characteristics. In this sense, it remains a challenge be develop a model that simulates all the measures of human movement. As such, we used the succinct description of popular models found in [11] in order to select a model that best fit our problem based on the previously described characteristics. The selected model, GeSoMo, is described in more detail in the next section.

3.2 GeSoMo

GeSoMo is a Sociological Based Model that pays most attention to spatial and temporal regularity, and also the relationships between locations and other users. This section summarizes how this model operates.

3.2.1 Basics

The primary input for GeSoMo is a social network. This social network represents the users whose movements are to be simulated as well as the social ties between them. These ties directly influence the movement of users. For each node within the input social network, a home anchor is created. Anchors are the abstract locations in which a high volume of meetings take place. These anchors are then evenly distributed across the 2D grid on which the simulation takes place. For instance, anchors can represent tables, classrooms, or building depending on the size of the

simulation. Nodes traverse across the area between anchors, simulating people moving about an area. Once a node arrives at an anchor, it pauses for short period of time before moving to a new destination. User movement can occur individually or in groups. The next destination anchor is selected probabilistically based on the current state of the simulation. Each anchor exerts an attraction towards a node based on the location of the anchor and the nodes currently dwelling at that location. When a node (or group of node) is selecting which anchor to go to next, the more attractive locations are selected with higher probability.

3.2.2 Attractions

Every anchor $a \in A$ exerts the sum of three forms of attraction on a node v: location attraction, node attraction, and node repulsion.

- **Location Attraction:** Location attraction A_{loc} depends on time and the owner of anchor a such that nodes prefer to visit a small subset of anchors and return frequently exhibiting the temporal regularity seen in real mobility traces.

- **Node Attraction:** Node attraction A_{node} depends on time and the nodes connected to v located at anchor a.

- **Node Repulsion:** Node repulsion A_{rep} depends on time and the nodes located at a, which are not connected to v. This can be seen as negative attraction and is employed to prevent unwarranted triadic closures.

After a node pauses at an anchor for a specific dwell time, it prepares to leave for the next node. If other nodes located at the same anchor are nearing their time to leave as well, the nodes select join into a group G and travel together with probability proportional to their remaining dwell time. Traveling individually can be seen as a group of one. The overall attraction an anchor a exhibits on the group G at time t can be calculated as

$$A_o(G, a, t) = \sum_{v \in G} [A_{\text{loc}}(v, a, t) + A_{\text{node}}(v, a, t) + A_{\text{rep}}(v, a, t)] \tag{3.1}$$

If the overall attraction A_o is negative, i.e., the node repulsion outweighs the other attractions, A_o is set to zero. Once the overall attractions for every anchor $a \in A$ are calculated, the next destination is selected in proportion to the other anchors according to

$$\Pr(\text{New Destination} = a) = \frac{A_o(G, a, t)}{\sum_{a' \in A} A_o G, a', t}. \tag{3.2}$$

3.2.3 Control Loop

While this explains the basis of how nodes move in GeSoMo, the model also incorporates a control loop to ensure robust performance. GeSoMo operates this control loop on the idea that the number of meetings between two nodes should be proportional to the input social network. The model also stipulates that the total number of meetings experienced by a node be proportional to the size of the network as well. To ensure the proportion of meetings between two nodes is accurate, a correction factor is calculated into the attractions. This correction factor is updated periodically, and serves to increase or decrease the probability of two nodes meeting. The intuition behind this correction factor is that people often try to balance their social interactions. If they have not seen a friend recently, they put more effort in meeting up with them. To ensure the proportion of total meetings, nodes are allowed to enter an isolation state. When a node begins to select another anchor, if it has a disproportionate number of meetings, it can instead enter an isolation state. An isolation state induces a mobility pattern where, instead of traveling to an anchor, the node goes to a random location in the area and pauses for a short time before traveling to a new anchor. The intuition here is that people often seek out alone time with no social interaction, and less social people would obviously do this more often.

As the nodes move, there positions are recorded periodically and used to create synthetic mobility traces. These mobility traces are then combined and output to a file. This file contains the position of every node as a function of time, and serves as both the output of the GeSoMo simulation and the input for our model.

3.2.4 Extending GeSoMo

Using GeSoMo as a mobility model, we tailor this simulation framework to match our needs.

3.2.4.1 Bluetooth Approach

While our initiative focuses on the collection of data using our Wi-Fi sensors, a common approach for empirical data collection relies on using mobile devices directly. In this context, data collection is often performed with the use of a custom application utilizing Bluetooth for the detection of nearby devices. To compare our approach to this one, we begin by implementing this collection framework. Conceptually, this is accomplished by having each node *carry* a sensing antenna, enabling the node to monitor nearby devices. This implementation is done by calculating the pairwise distance matrix $D(t)$ at each time-step, where $d_{i,j}(t)$ is the distance between nodes i and j at time t. Using this matrix, we record all the occurrences where $d_{i,j} < r$, with r representing the sensing radius of the Bluetooth antenna. At each of these occurrences, we check if $d_{i,j}(t-1)$ is also less than r. If not, we add the encounter to the records as well. The intuition here is that either the nodes were already together, or they are meeting after being apart which results in an encounter. Based on these calculations, we aggregate the data pairwise such that each pair of nodes has a recording of the total number of time-steps they took together, and the total number of encounters between them. This aggregate dataset contains key features used to make predictions on social ties.

3.2.4.2 Wi-Fi Approach

To implement our Wi-Fi sensor approach, we make some small changes to the previous Bluetooth approach. Instead of the sensing antennae being carried by nodes, they are instead placed in a gridded pattern across the simulation area. As a node passes through or pauses within the antenna footprint, its presence is recorded. Any interactions between nodes outside of the antenna patterns are not observed. This is implemented in a manner akin to the Bluetooth approach. However, in this latter setting, the distance matrix $D(t)$ is instead the pairwise distance between every node and every antenna. That is, $d_{i,j}$ is the distance between node i and antenna j. Whenever $d_{i,j} < r$, we

again check for encounters in the same way as before. Moreover, when an observation takes place, we also make record of the antenna at which the node was detected. Aggregating the data pairwise results in the total number of time-steps, the total number of encounters, and the total number of different locations (antennae) where a node was detected. Both this dataset and the Bluetooth dataset are subsequently employed to make predictions.

3.3 Classification Method

Our goal is to use the simulated data to make predictions on social ties between users. This can be seen as a binary classification problem where, given the data for each pairwise set of nodes, we want to predict whether a social tie exists in the original network input into the simulation model. We do this by first transforming our data, and then using Regularized Logistic Regression to make our predictions.

3.3.1 Data Transformation

We begin where virtually all machine learning problems do: transforming the data into useful features. At this stage, the data output from our simulation contains $\binom{N}{2}$ rows where N is the number of nodes simulated. Each of these rows contains a node A column and a node B column, representing one possible edge in the original graph. For both models, there is also a column containing the total number of time-steps detected together, and another column for the total number of encounters between node A and node B. Additionally, for the Wi-Fi model there is a column for the total number of locations (or antennae) where an encounter took place, ranging from zero to the total number of antennae. This information is summarized in a compact form below:

- **Node A:** The first node of the potential edge,
- **Node B:** The second node of the potential edge,
- **Total Time:** The total number of time-steps A and B took together,
- **Total Encounters:** The total number of times A and B met,
- **Total Locations:** The total number of different antennae A and B were observed together.

We emphasize that the last piece of information is not available for the Bluetooth model.

To scale the data, to reflect the increased sociability of certain nodes, and to prevent extremely connected nodes from dominating the features; these columns are instead used to calculate the Friendship Indices seen previously. Furthermore, a column is added to indicate whether or not an edge exists in the input social network. The indices are summarized as follows:

- **Friendship Index Based on Encounter Time:** The ratio of total time nodes A and B are encountered to the total time,
- **Friendship Index Based on Encounter Count:** The ratio of the number of encounters of nodes A and B to the total number of encounters for A,
- **Friendship Index Based on Encounter Locations:** The ratio of the number of locations A encounters B compared to the total locations A travels,
- **Edge:** One if the edge exists in the input graph, and zero otherwise.

This data is then fed into the logistic regression algorithm.

3.3.2 Logistic Regression

Logistic Regression is a widely used classification algorithm which uses the sigmoid function as a hypothesis h. Specifically,

$$\Pr(y = 1|x) = h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{3.3}$$

$$\Pr(y = 0|x) = 1 - \Pr(y = 1|x) = 1 - h_\theta(x). \tag{3.4}$$

Our goal is to find a θ such that the hypothesis is large when $x = 1$ and small when $x = 0$. This can be done by minimizing the loss function

$$J(\theta) = -\frac{1}{m} \sum_i \left(y^{(i)} \log \left(h_\theta \left(x^{(i)} \right) \right) + \left(1 - y^{(i)} \right) \log \left(1 - h_\theta \left(x^{(i)} \right) \right) \right). \tag{3.5}$$

3.3.3 Cross Validation

To prevent overfitting, a regularization is often performed changing the cost function to

$$J(\theta) = -\frac{1}{m}\sum_{i}\left(y^{(i)}\log\left(h_\theta(x^{(i)})\right) + \left(1-y^{(i)}\right)\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right) + \frac{\lambda}{2m}\sum_{j}^{n}\theta_j^2. \tag{3.6}$$

To tune this regularization constant to yield the best results, we simply do a search over a logarithmic set of variables, testing each regularization constant and implementing the one that works best. We use stratified k fold cross validation with $k = 10$, following convention, to compare the results of each test. This is a method where the data set is divided into k partitions with each fold containing roughly the same proportion of labels, after which $k-1$ folds are used for training with the remaining fold used for testing. Each fold is used as the test set once, and k scores are averaged together. This method results in significantly less variance than a simple train test split.

3.4 Results

The results presented herein correspond to simulations a forty node Toivonen network on a one hundred meter square grid. See the appendix for more information regarding the simulation parameters. The anchors and antennas are arranged in a square gridded pattern with the locations depending on the number of nodes. The following graphs show the results of the two models tested for antenna count c, antenna radius r in meters, frequency of sample s in seconds, and the effects of zero-mean Gaussian noise with a standard deviation of σ meters.

3.4.1 Number of Antennas used for Detection

We begin by varying the number of sensors used for detection. In the Bluetooth model, this is equal to the number of nodes and, as such, it cannot be varied without changing the social structure of the input network. On the other hand, for the sensor network approach, we observe in figure 3.1 that a high number of sensing antennas leads to generally better results. Also, there is an increase in performance when the antenna grid matches up with the grid of anchors. This intuitively makes sense; if the antennas are located at meeting points, they are more likely to

observe actual meetings as opposed to incidental contact between nodes. This emphasizes the importance of sensor location.

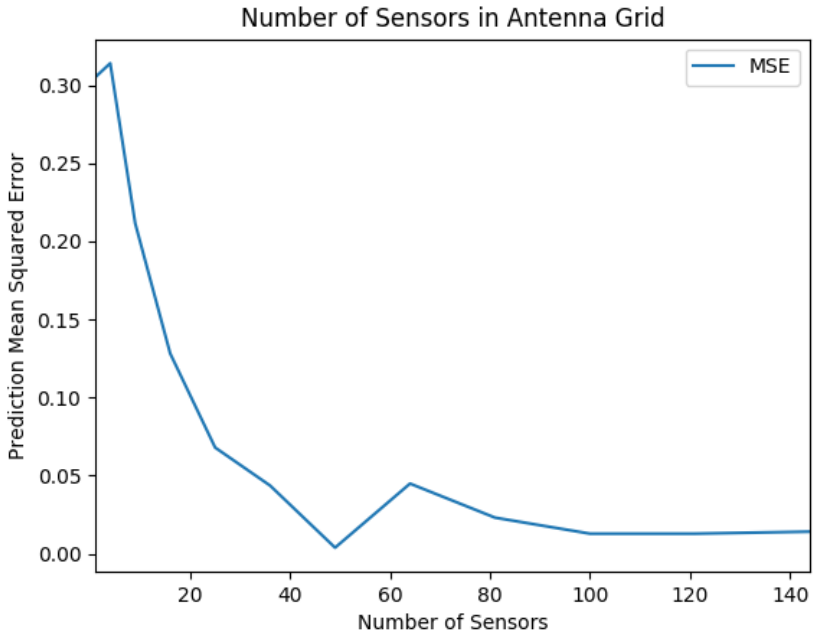

Figure 3.1: This figure shows the results of varying the number of sensors in the sensor grid; $r = 10$, $s = 10$, $\sigma = 0$.

3.4.2 Range

We begin by varying the range of individual nodes, both the Bluetooth and Wi-Fi antenna. At one extreme, if the antenna radius is too large, then every node is considered together at every time-step. At the other extreme, if the antenna is too small, nodes interactions are unlikely to be observed. For instance, we are unlikely to find two users with their phones within 10 cm of one another. In the simulation, the antenna footprint is varies by changing the value of r when combining the distance matrices $D(t)$. This is done at the minimum sampling period of 10 s provided by our implementation of GeSoMo. We can observe in figure 3.2 and figure 3.3 a similar cutoff between both methods at around 12 m, which is conveniently within the range of the Bluetooth and Wi-Fi

standards. In an applied setting, a threshold on signal strength can be employed to approximate the observation of devices within a certain radius of interaction.

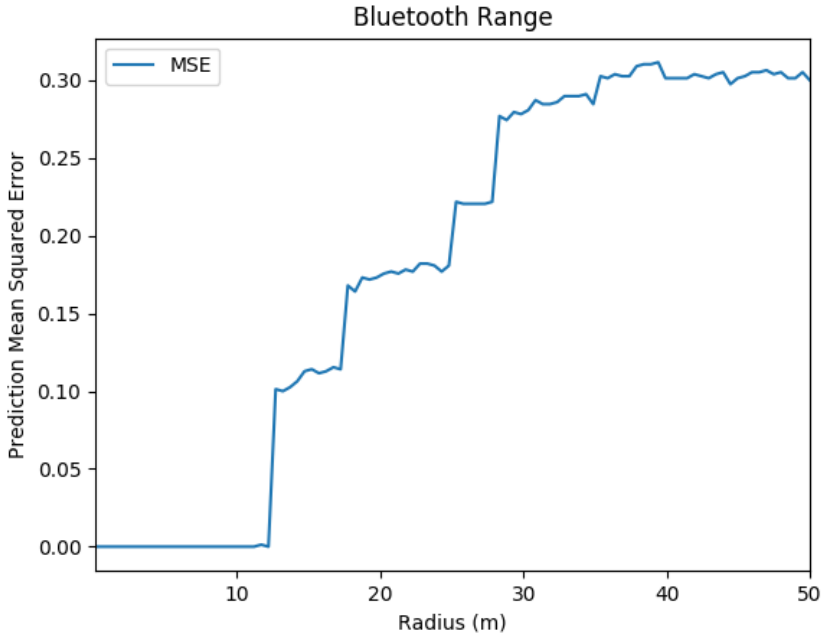

Figure 3.2: This figure shows the results of varying the radius of the Bluetooth antennae footprint for $s = 10$ and $\sigma = 0$.

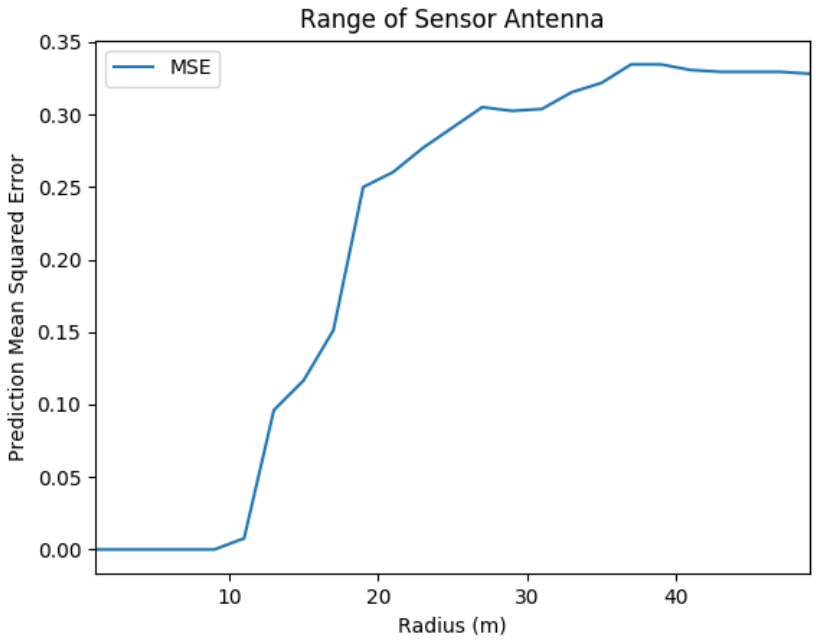

Figure 3.3: This figure shows the results of varying the radius of the sensor antennae footprint for $c = 49$, $s = 10$, and $\sigma = 0$.

3.4.3 Sampling Rate

Next, we vary the sampling rate at which observations are detected in the two network models. This is especially important for the Bluetooth model (and the Sensor model depending on implementation), as this could affect the frequency of antenna activity in what is most likely a power-constrained mobile device. It also dictates the storage requirements for the gathered information as well as the overall algorithm runtime. It can be observed in figure 3.4 and figure 3.5 that the error rate increases linearly with sampling rate in both models.

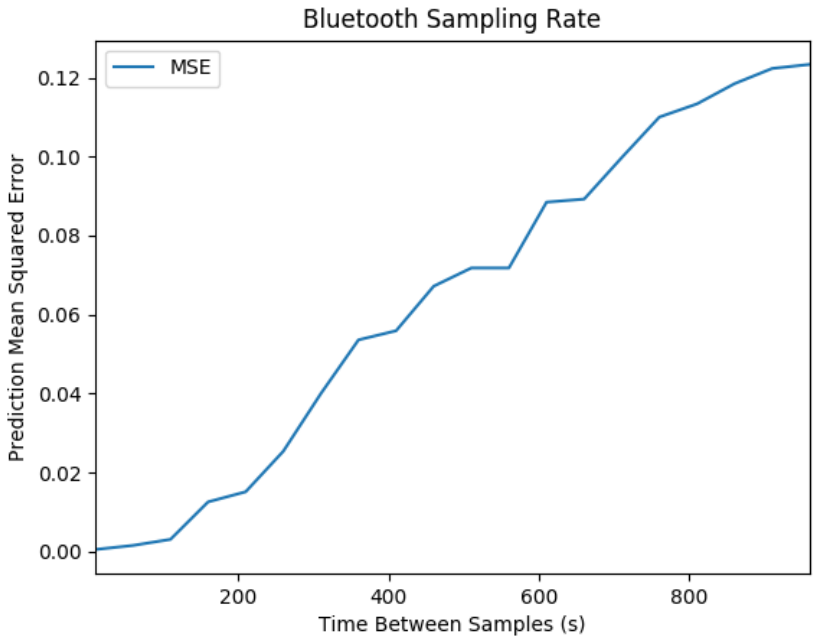

Figure 3.4: This figure shows the results of varying the number of seconds between each successive Bluetooth sample for $r = 5$ and $\sigma = 0$

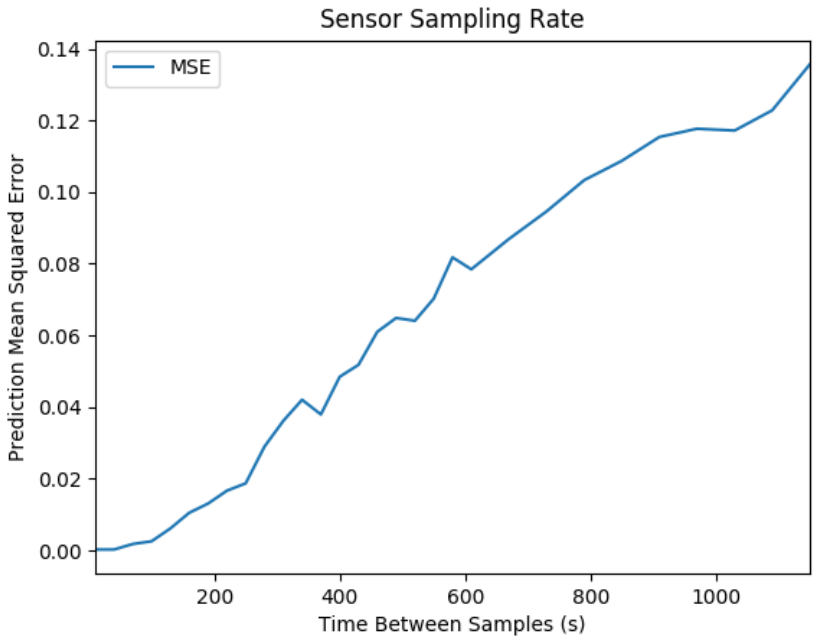

Figure 3.5: This figure shows the results of varying the number of seconds between each successive sensor sample for $c = 49$, $r = 5$, and $\sigma = 0$.

3.4.4 Noise

In a realistic system, there is only a noisy connection between wireless signals and distance. That is, one will not know the exact distance between nodes or users. Antenna patterns, even with strict power level thresholds, are inherently noisy. We simulate this reality by adding a zero-mean Gaussian noise matrix to the distance matrix,

$$D_{\text{noisy}}(t) = D(t) + \mathcal{N}(0, \sigma). \tag{3.7}$$

The results in figure 3.6 and figure 3.7 support our previous antenna range graphs as when the noisy radius gets above a threshold the error rate grows rapidly.

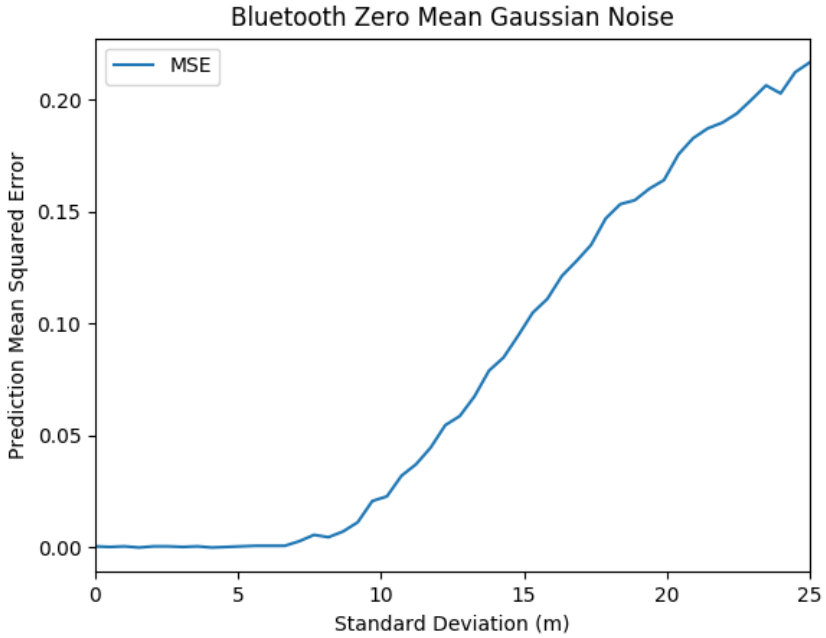

Figure 3.6: This figure shows the results of varying the standard deviation of the zero mean Gaussian noise in the Bluetooth model for $s = 10$ and $r = 5$.

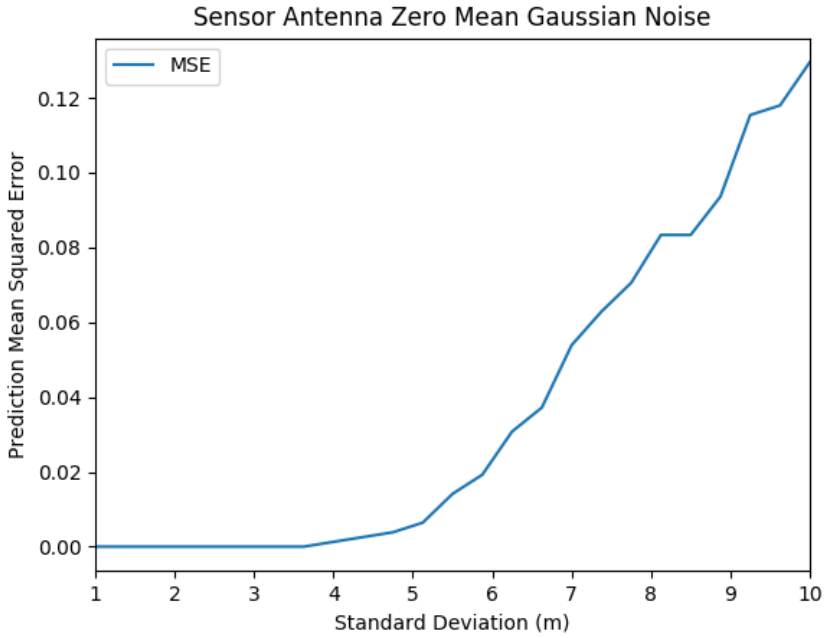

Figure 3.7: This figure shows the results of varying the standard deviation of the zero mean Gaussian noise in the sensor model for $c = 49$, $s = 10$, and $r = 5$.

4. EXPERIMENT

The next stage of our initiative was to deploy a sensor network in an area on campus and use the collected data to gauge our ability to accurately estimate social networks. We intended on doing this in a somewhat controlled setting, where a sort of social structure was known, e.g., a laboratory course with assigned lab groups that meet regularly. However, as the simulation results show, a large number of antennas are required to obtain meaningful results. While we were able to develop a working sensor prototype proving it was possible to harvest metadata in our envisioned scenario, the sheer number of sensors required was prohibitive to actually collecting sufficient data. As such, without putting significant effort into a more efficient sensor deployment, we are unable to show results for a real data experiment at this time. The rest of this section serves to detail the sensor prototype design efforts.

4.1 Sensor Prototype

The sensor prototype design began as part of a class project, with the goal of detecting devices in a lab area on campus. It was determined that due to the prevalence of smartphone use, along with the growing ubiquity of Wi-Fi, one can exploit the network connection of devices in order to detect their locations in a broad sense. A naive approach is to have users connect to a Wi-Fi access point, and log these connections. Under such circumstances, one can directly employ device credentials and login information to identify users. While this approach is appropriate whenever the network is under the control of the monitoring team, most students connect to the campus Wi-Fi, and not to an open Wi-Fi access point. Consequently, packet sniffing is a more practical solution for the purpose of passive monitoring.

Packet sniffing is often done for network analysis and, by definition, consists in monitoring packets within one's own network. In wireless settings, sniffing can be performed without network access and, therefore, is commonly associated with the hacking community. Packet sniffing is a fundamental way in which hackers gather information on a wireless network in order to begin an

attack. It is virtually impossible to detect or prevent under current standards, making it ideal for attackers. It also enables legitimate network troubleshooting and optimization. There are many readily available and free applications that make packet sniffing straightforward, e.g., Wireshark.

Using Wireshark, one can gain the ability to monitor any packets within range of our network antenna. This is accomplished by switching the network card into *monitor mode*. This mode basically enables the card to view, process, and log all Wi-Fi packets, without outputting any signals. It should be noted this this is different from *promiscuous mode*, which serves virtually the same purpose, except that promiscuous mode requires network access and can be used in a wired network. Unfortunately, not every network card can be put into monitor mode, as it is seldom used in normal circumstances. Suitable external USB network cards can be identified through Internet forums and help pages.

With the necessary hardware, Wireshark makes it simple to perform packet sniffing and capture (logging). For instance, it becomes straightforward to to output such logs to timestamped files. After collecting MAC addresses from lab devices, and developing a program to filter the output files based on a whitelist, the course project achieved its goal of detecting when a user is within the lab area. Other uses of this technology were explored in class projects, and served as a stepping off point into both this research and research seen in [12].

4.1.1 Current Iteration

While this thesis leverages some of the components developed within the context of a course, many sensing components have received an upgrade to better suit the needs of this project and its deployments. The current version of the sensing platform is discussed below.

4.1.1.1 Current Iteration: Hardware

The current iteration of the sensor's hardware is focused around a four inch by four inch computer known as the Intel Next Unit of Computing (NUC). This is a full fledged desktop computer based on the Intel Core processor, with expandable memory, high capacity SSD, networking, and graphics support. Its relatively small form factor paired with its ability to run desktop operating

systems make it easy to prototype with, yet convenient enough to deploy as a sensor.

The NUC supports both wired and dual band wireless networking. For consistency in simultaneously evaluating multiple antenna designs we use two identical external USB wireless networking adapters. While these external adapters are used for data collection, the internal networking hardware is still required in order to retrieve data from the device. Currently, this is done via a wired ethernet connection. We selected the ALFA AWUS051NH as the external adapter because it supports both monitor mode and the ability to collect data on the 2.4 GHz and 5 GHz bands. This allows the sensor to monitor devices across the entire Wi-Fi spectrum.

Connected to these external USB network adapters are Wi-Fi antennas. There are were two main antenna designs explored: a patch directional antenna and omnidirectional antenna. These two antennas are identical across all sensors. We implemented two separate antennae in order to observe the effect antenna design has on both sampling frequency and inference results. At the limits, an omnidirectional antenna will sample a wider area while a directional antenna will sample a smaller area. This was of much greater importance to the project before we ran into the previously state RSSI challenges.

The sensor prototype is designed to be mounted through a ceiling tile. This is done because ceiling tiles are cheap, easily modifiable, and easy to install/replace. Using several purpose built and 3D printed components, we are able to inconspicuously mount the sensor such that the only visible components are the antennae. In order to be able to aim the directional antenna, we attached the antenna to the end of a bendable arm allowing us to point the antenna in virtually any direction, as seen in Figure 4.1.

4.1.1.2 Current Iteration: Software

Running on our Intel NUC we have the rolling release of Kali Linux. Kali Linux is a popular Debian based linux distribution designed and packaged with penetration testing and hacking in mind. This obviously made Kali our first choice of distribution, as many of the tools and libraries used during the prototyping of the sensor (like Wireshark) come prepackaged within Kali. Apart from the operating system, several other major software components are required for our sensor to

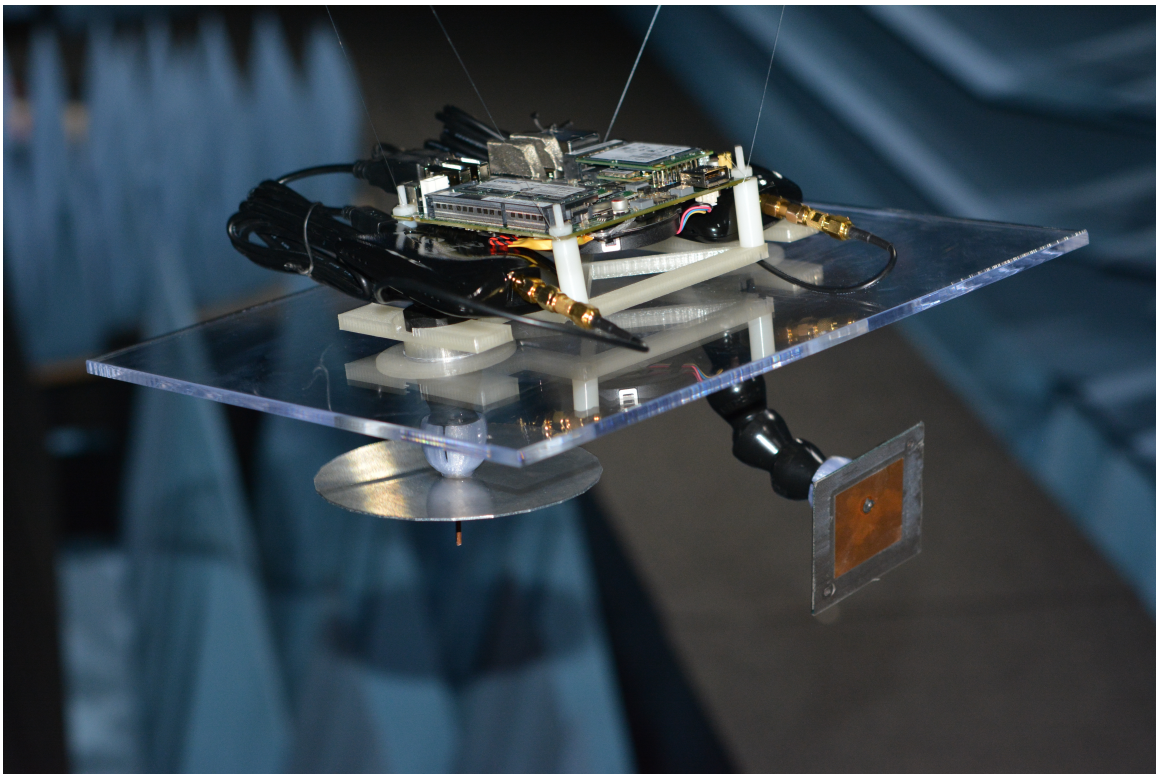

Figure 4.1: This picture shows the working sensor prototype.

function, the first of which is the our main monitoring software.

Our monitoring program is a C based software utilizing the `libpcap` library. This is the same library that the popular tcpdump command line tool is based on. Using this library we are able to switch the external network interfaces into monitor-mode to begin our packet collection. Once in monitor mode, the program operates on layer two of the OSI network model, collecting and decoding Wi-Fi frames within monitoring distance of our sensor. During testing in congested environments it became evident that our program is only able to process a limited number of frames per second, dropping a significant amount of potential data. However, given that many frames are not relevant to our data set, we are able to use a `pcap` filter program to filter frames based on MAC address, protocol being used, type of frame (probe request, acknowledgement, etc), and various other indicators. This not only allows us to avoid dropping relevant data, it also controls the growth rate of our dataset. Without filtering the dataset size can grow at a rate of tens of millions

per day. This can be reduced by several magnitudes depending on how strict the program is set to filter. As the filtered frames are collected, they are processed and the source MAC address as well as the destination MAC address are timestamped and written to a file. These files are regulated in size to a user set number of entries, after which a new file is seamlessly created and the logging continues.

Before discussing one of the limitations of this software, we must explain what a Wi-Fi channel is. Wi-Fi can be separated into two frequency bands: The 2.4 GHz band (ranging from 2400–2500 MHz) and the 5 GHz band (5725–5875 MHz). Each of these bands are then separated into channels. A channel is a 20 MHz section of the band. These channels are spaced 5 MHz or 12 MHz apart, and as such overlap. If a user sends data across Channel 1 (2412 MHz) and another user concurrently sends data across Channel 2 (2417 MHz), then this produces interference and can affect throughput drastically. To combat this phenomenon, networks typically send information over non-overlapping channels. There are three non-overlapping channels (1, 6, and 11) for the 2.4 GHz band, and twenty-four non-overlapping in the 5 GHz band. Data can be simultaneously transferred over these channels without interfering with one another. Often to increase the bandwidth available for a transfer, these channels are combined into a single medium at the cost of non-overlapping channel count. For example, with a 40 MHz channel width in 802.11n you only have two non-overlapping channels centered around 2422 MHz and 2462 MHz (3 and 11). There is a balance between the number of device connections and channel width that optimizes networks for different scenarios.

The use of the `libpcap` library in our program creates a limitation regarding Wi-Fi channels. While the program, as written, is able to passively monitor Wi-Fi users proximity, it only does so on a single channel. This is not ideal for our sensor because devices are automatically assigned to any one of the non-overlapping channels in the network across the 2.4 GHz and 5 GHz bands. By listening only on one channel out of twenty seven bands, we capture only a fraction of the packets within our sensing area. The `libpcap` library does not support programmatically changing the monitoring channel at the time of writing. Luckily, it is possible to change the wireless interfaces

channel through the `iwconfig` command line tool. Even more beneficial is that this can be done without having to take the interface down, which is time consuming and would have halted our monitoring program. Thus, the prototype includes a simple script that cycles through the non overlapping channels, stopping and monitoring on each one for a variable amount of time. For our project, we assumed at least 40 MHz channel width and switched through all the non-overlapping channels in under ten seconds. This parameters can easily be modified based on context. This implies that, if a user is in the footprint area of our sensor for less than ten seconds, there is a chance no frames for that user would be recorded due to sensing on the wrong channels.

Once a collection of sufficient time is performed, the data must be retrieved from the sensor. To keep the sensor undetectable by the network, we opted to do this by manually connecting an ethernet cable between the sensor and a laptop. By enabling SSH access, we are able to use the ethernet connection to access the sensor for data retrieval or to execute commands directly on the sensor.

4.1.2 Improvements

While we were able to develop a working sensor prototype, there are several improvements that could be made given more time. For a small to mid level deployment, the sensor could be much more inconspicuous. By minimizing the necessary hardware platform and using project specific antennae, the form factor of the device could be much smaller. This would also cause the device to be more energy efficient and possibly battery powered, enabling it to be deployed much more discretely. Also, in the current form, the sensors are accessed manually on an individual basis. As such updating the sensors or retrieving data can be logistically challenging. Due to time, these potential improvements have not found their way into the prototype developed in the course of this work.

5. CONCLUSION

Wi-Fi devices are constantly sending observable metadata which can be captured by sensing devices. This sensing can be done passively and without interference or network performance degradation. Using this harvested metadata, it is possible to estimate social ties between users. We observe that in a simulated setting, similar results to a common collection strategy implementing Bluetooth can be achieved, with similar trends appearing in both methods. Therefore, Wi-Fi data collection for social network inference becomes an option if access to individual Bluetooth device data is not possible. However, an efficient sensor design is needed due to the number of sensors required to exhibit meaningful results. We also observed that the sensor antenna pattern affects the sensors ability to collect meaningful data, with smaller more discriminating patterns yielding better results. Also shown was that sampling data more often leads to better results, but only to a point. The simulation implied that sampling could be done on the order of minutes and still yield very good results. This implication is especially important for future applications where power consumption or data storage becomes an issue. While in the simulation our measures are exact, in the real world antenna power values that are tied to distance are inherently noisy. By adding noise to our measures to better mimic real world measurements, we were able to observe that the Wi-Fi model for collecting data was less resilient to these inconsistencies, but trended similarly to the Bluetooth model.

REFERENCES

[1] J. Travers and S. Milgram, "The small world problem," *Phychology Today*, vol. 1, pp. 61–67, 1967.

[2] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Physical Review E*, vol. 65, no. 2, p. 026107, 2002.

[3] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[4] R. Toivonen, J.-P. Onnela, J. Saramäki, J. Hyvönen, and K. Kaski, "A model for social networks," *Physica A: Statistical Mechanics and its Applications*, vol. 371, no. 2, pp. 851–860, 2006.

[5] W.-J. Hsu and A. Helmy, "On nodal encounter patterns in wireless lan traces," *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1563–1577, 2010.

[6] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg, "Inferring social ties from geographic coincidences," *Proceedings of the National Academy of Sciences*, vol. 107, no. 52, pp. 22436–22441, 2010.

[7] N. Eagle, A. Pentland, and D. Lazer, "Infering social network structure using mobile phone data," *Proceedings of National Academy of Sciences*, 2006.

[8] N. Eagle and A. S. Pentland, "Reality mining: sensing complex social systems," *Personal and ubiquitous computing*, vol. 10, no. 4, pp. 255–268, 2006.

[9] W.-J. Hsu, D. Dutta, and A. Helmy, "Mining behavioral groups in large wireless lans," in *Proceedings of the annual ACM international conference on Mobile computing and networking*, pp. 338–341, ACM, 2007.

[10] N. Vastardis and K. Yang, "Mobile social networks: Architectures, social properties, and key research challenges," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1355–1371, 2013.

[11] P. Pirozmand, G. Wu, B. Jedari, and F. Xia, "Human mobility in opportunistic networks: Characteristics, models and prediction methods," *Journal of Network and Computer Applications*, vol. 42, pp. 45–58, 2014.

[12] P. K. Eedara, "Classification of wireless device location based on Wi-Fi metadata," Master's thesis, Texas A&M University, 2016.

[13] M. B. Kjærgaard and P. Nurmi, "Challenges for social sensing using WiFi signals," in *Proceedings of the ACM workshop on Mobile systems for computational social science*, pp. 17–21, ACM, 2012.

[14] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa, "Signals from the crowd: Uncovering social relationships through smartphone probes," in *Proceedings of the 2013 conference on Internet measurement conference*, pp. 265–276, ACM, 2013.