A Measurement Study for Understanding Wireless Forensic Monitoring

Yongjie Cai and Ping Ji
Doctoral Program in Computer Science, Graduate Center &
John Jay College of Criminal Justice
City University of New York
{*ycai@gc.cuny.edu, pji@jjay.cuny.edu*}

Abstract: *Monitoring Wireless network activities for cybercrime investigation purpose is getting to be an important task in Digital Forensic field. While malicious network users can easily camouflage their true identities through various anonymity techniques, randomly hopping on a wireless communication channel and conduct ominous activities without leaving any traces is one of the most convenient and efficient network masquerading tactic. In a related work [7], we outlined a wireless forensic monitoring system (SMoWF), which aims to establish a forensic database based on encrypted (or hashed) wireless trace digests, and to answer the critical investigation questions: which wireless device appeared at where during what time. To answer these questions, we need to accomplish two challenging tasks successfully, namely Device Identification and Device Localization. In this paper, we explore the challenges of device localization schemes through a measurement study conducted in a metropolitan area of New York City. We gathered frame level wireless traces by using three different equipment settings over four different time periods, and initiated the research by exploring the WiFi deployment characteristics in a densely populated urban neighborhood. The signal strength information from the gathered traces is then used for a particular device to perform localization. We also evaluated the performance of two localization schemes - the K-Nearest-Neighbors (KNN) approach and the Log-distance Path Loss Modeling approach. Our experiment and evaluation results show that it is a challenging task to offset the inconsistency between a signal strength fingerprinting database and the real position coordinates caused by measurement time and device differences. Further, we observe that for the KNN approach, when $k = 3$, with k being the number of Access Points used in the localization algorithm, an optimal tradeoff between "the number of APs" and "the location variance" is most likely to achieve.*

1 INTRODUCTION

As Wireless Networks and mobile devices claiming dominant roles in modern communication technology, the manifestations of digital crimes have started integrating *Wireless* element as well, which makes digital forensic investigation unprecedentedly difficult. For example, a malicious hacker may walk on the street, randomly pick an open Access Point (AP) such as one from a coffee shop, conveniently connect to the AP, upload and/or download malicious files or send commands through the AP to Zombie machines compromised for a distributed attack, then close the communication session and take off freely. The whole process may only take minutes to accomplish. No one would even notice what has happened until the victim device(s) detects an intrusion. When an attack has been identified, the investigators will easily run into the situation where the best *point-of-interest* that can be traced back is the benign Access Point, through which the true attacker conducted the malicious activity. Since most WiFi users do not regularly keep security logs and monitor the activities of their Wireless network, it is almost certain that the hacker is off the hook.

In our work on designing a Security Monitoring system for Wireless Network Forensics (SMoWF) [7], we have outlined the infrastructure of a Wireless forensic monitoring system that aims to keep encrypted trace digests of WiFi activities and to answer the important investigation question: *which device appeared at where during what time*. With a deeper consideration, it is easy to realize that to answer this investigation question successfully, we have to handle two challenging tasks, namely Device Identification and Device Localization. In this paper, we focus on exploring the latter challenge

- Device Localization through a measurement study on the wireless networks of a Metropolitan area of New York City. We started the work by exploring the depolyment characteristics of WiFi Networks in a densely populated urban neighborhood. We then compare and contrast the measurement and evaluation results of three different equipment settings and two localization algorithms. The three equipment settings are: 1. a MacBook Pro laptop running Mac OS X10.7.4, its integrated wireless card, and a BU353 GPS receiver; 2. a MacBook Pro laptop running Backtrack system in a virtual box, combining with an Alfa wireless card, and a BU353 GPS receiver; 3. an HTC Jetstream Android tablet that subscribed to the data plan provided by a large Cellphone carrier. The two localization techniques that we explored are: the K Nearest Neighbor [6] algorithm and a channel characterization technique using the Log-distance Path Loss Model. From our experiments, we observe that a common GPS receiver (e.g. BU353 GPS receiver) does not work well in the Metropolitan area. Its performance degraded into the accuracy provided by Cellular and WiFi Networks. The integrated GPS in tablet provides much more accurate locations.

The rest of this paper is organized in the following: Section 2 explores the related work; Section 3 illustrates the measurement experiment set up in detail, and discusses the experiment results of the traces that were gathered through four different time periods; Section 4 presents two localization schemes, and evaluates their performance on our traces; Section 5 concludes this paper.

2 RELATED WORK

Device localization via WiFi has stayed a popular research topic in recent decades. The main task of the localization problem is to map the received signal strength (RSS) to the physical location coordinates of a particular device. The existing methods can be summarized into two categories, fingerprinting-based and model-based. There are two steps in fingerprinting-based localization. The first step is to build a radio frequency (RF) map [6] by dense measurements to obtain sufficient training data so that each position has a unique fingerprint put in a database. Usually, the set of received signal strength from nearby access points is used as the fingerprint of a position. The second phase is to estimate the position coordinates of a device by comparing its signal strength signature with those in the established database. K nearest neighbor (KNN)[6] provides a simple frame work for radio frequency (RF) fingerprinting. In KNN, the average of distance of the K closest neighbor positions is used to obtain an estimated position. Further, Euclidean distance is used to calculate the similarity between two fingerprints. Another group of fingerprinting uses probabilistic techniques to estimate the signal strength distributions on the RF map[15][16][10][11]. Horus system [16] captures the device location in a RF map, which has the maximum probability with a given fingerprint. In Horus, a clustering module, where a cluster is defined as a set of locations sharing a common set of access points, is used to reduce the computation complexity and save energy. The estimation phase starts by searching potential positions from which the first access point in the fingerprint is observed on the RF map with certain probability. The search continues to find the positions where the first and second access points are both observed, and a corresponding probability is computed. Finally, the search ends probability exceeds a threshold.

The fingerprinting methods involve massive human labor to build the RF map, which may also introduce human errors. An alternative way for localization is to adopt radio signal strength models [6] [8]. In [14], the log-distance path loss model is used to describe the energy level of the received signal strength which decreases with the increasing distance between access points and clients. Based their received signal strength, a position is then determined by using triangulation[8]. However, in reality, many physical obstacles such as buildings, trees, walls, etc., exist during radio transmission.

The radio would reflect, diffract and refract on these objects. Therefore, it is difficult to accurately model the propagation path of radios due to multi-path and shadowing problems.

3 Experiment and Data Analysis

We conduct a measurement study on the activities of wireless networks in a metropolitan area of New York. The equipments used in our measurement experiments consist of a MacBook Pro laptop, an external Alfa Awuso36nh USB 802.11 wireless card, two BU353 GPS receivers and an HTC Jetstream android tablet. Kismet [3] is installed on the Macbook Pro to gather wireless network traces. Kismet sets a supported wireless card into raw monitoring mode, passively collects packets and saves them in pcap format files. It also logs the coordinates of packets where it is observed when it is connected to a GPS receiver. We enabled Kismet's channel hopping function to explore activities across the entire radio spectrum. As a result, only partial trace were collected from each channel. To compare the localization results provided from satellite network (i.e. through the GPS receiver), we deployed a similar set of experiment on an HTC Jetstream tablet that is supported by the data plan from a large cellphone carrier. We wrote a small Android application named WiFum to actively scan available WiFi networks as well as measurement point location coordinates read from the internal GPS/aGPS that is built in the tablet. We compared two location providers in Android tablet, the GPS Provider and Network Provider [1]. GPS Provider determines location using satellites. Network Provider determines location based on availability of cell tower and WiFi access points. Our testbed covers a three-block metropolitan area of the mid-west side in NYC, which is around 260m*260m. In each run of the experiment, we walked along the street in a zigzag path to cover the entire area. The path we took is shown in Figure 1.

We conducted our measurement study by using three different combinations of the available devices. The first combination contains a MacBook Pro laptop with built-in wireless card and a BU353 GPS receiver. In the second experiment setting, we install Backtrack [2] in a virtual box on the MacBook Pro, and combine it with an Alfa wireless card, an Antenna and a BU353 GPS receiver. The third experiment setting uses the HTC tablet with internal wireless card and its integrated GPS/aGPS. With the three experiment settings running simultaneously, we walked around the neighborhood along the path from A-H or H-A shown in Figure 1 in four time periods on Dec 26, 2011. The four periods are 14:06-14:35 (29 mins, A-H), 14:36-15:05 (29 mins, H-A), 15:07-15:35 (28 mins, A-H) and 15:36-15:58 (22 mins, H-A).

3.1 Passive Scanning vs Active Scanning

We compare the average number of observed access points at each position, the average number of access points observed per trip and the average percentage of encrypted access points observed overall. The results are showed in Table 1. From Table 1, we observe that HTC with WiFum discovered about twice the number of access points at one position as of the other two sets. The average encryption percentage of the third setting is 10% more than those of the other two. But the total number of access points observed per trip using WiFum is 16% less than it of the other two settings. We believe this result is due to different scanning mechanisms between WiFum and Kismet. WiFum works in an active scanning way, in which the device sends probe requests to nearby access points and waits for their response. Whenever WiFum broadcasts a probe request, all nearby access points reply probe responses. So it can detect more access points at one position. However, the access points with hidden SSID won't reply the probe request unless WiFum manually specifies its SSID. On the other hand, Kismet performs scanning passively. It listens to the beacon signals sent from access points periodically and does not produce any additional traffic. Kismet is able to detect all access points including

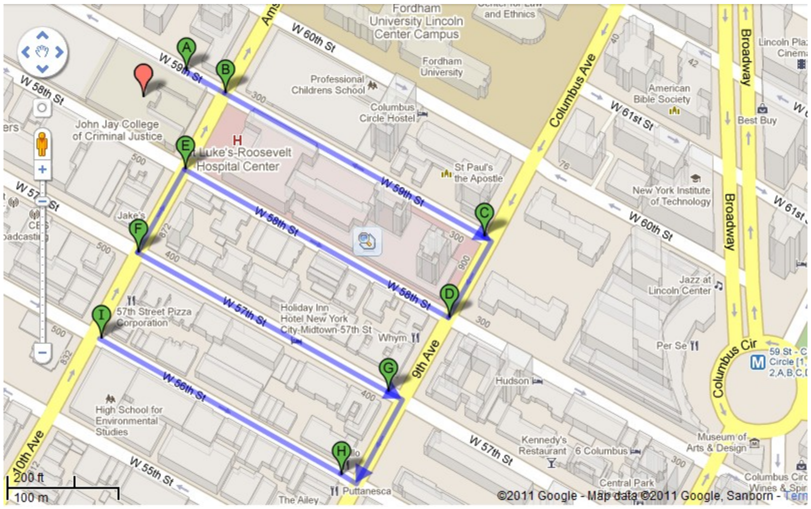
A
B
C
D
E
F
G
H
I
W 59th St
W 60th St
W 58th St
W 57th St
W 56th St
W 55th St
W 61st St
9th Ave
10th Ave
Columbus Ave
Broadway
Columbus Cir
Fordham University Lincoln Center Campus
John Jay College of Criminal Justice
St Luke's-Roosevelt Hospital Center
Professional Childrens School
Columbus Circle Hostel
St Paul's the Apostle
American Bible Society
New York Institute of Technology
Jazz at Lincoln Center
57th Street Pizza Corporation
Holiday Inn Hotel New York City-Midtown-57th St
High School for Environmental Studies
Museum of Arts & Design
200 ft
100 m
©2011 Google - Map data ©2011 Google, Sanborn

Figure 1: Experiment Path

those with hidden SSID when these APs are transmitting packets. Therefore, Kismet detected more access points in one trip. Notice that the set of access points observed at the same position may be different when using different tools.

Table 1: Measurement comparison using different tools

Tool	Airport+Kismet	Alfa+Kismet	HTC+Wifum
Avg AP Number/Position	15	11	24
Avg AP Number/Trip	1823	1829	1520
Avg Encryption Percentage	0.706	0.6835	0.8205

3.2 RSS sensitivity

We also compared the distributions of the average received signal strength from above three experiment settings and show them in Figure 2. From Figure 2, we observe that the major ranges of RSS for Airport, Alfa, and HTC are [-40,-100], [-20,-70], and [-50,-100] in the unit of dBm respectively. We can see that the average RSS of Alfa is stronger than that of Airport which is stronger than HTC.

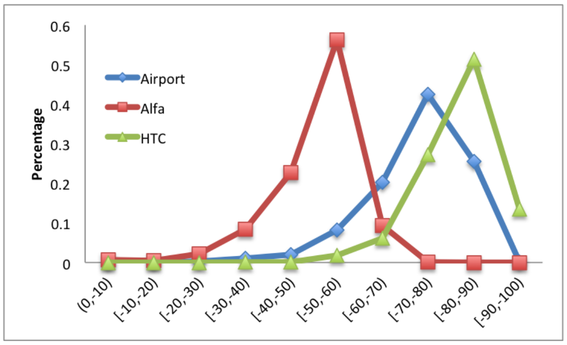

Figure 2: RSS Distribution

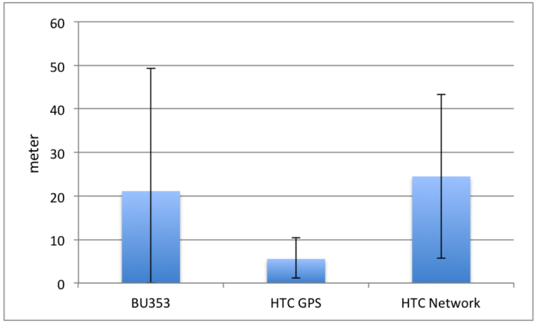

Figure 3: Average error distance of location provider with standard deviation

3.3 Location Accuracy

We also conducted a second set of static measurements to compare the accuracy of localization schemes from the GPS provider, the Network Provider (via WiFi and Cell Tower) on HTC tablet and the BU353 GPS receiver. We chose 10 static locations in this experiment. At each location, we performed scanning, and recorded GPS coordinates from three location providers continuously for about two minutes. In each location measurement, we define the center point, whose GPS coordinates (latitude and longitude) are the average latitude and longitude of empirical data (the GPS coordinates read in two minutes), as the *ground truth*, and calculate the *error distances* that is defined as the difference between each empirical GPS reading and the center point. The average error distances derived from the three experiment settings and their standard deviations are showed in Figure 3. The result is surprising. BU353 GPS receiver has a larger variance than expected. Its accuracy level is similar to the HTC Network Provider. To the opposite of the conclusions in [17], the performance of integrated GPS is substantially better than the GPS receivers in our experiments. This indicates that a common GPS receiver does not work well in the Metropolitan environment.

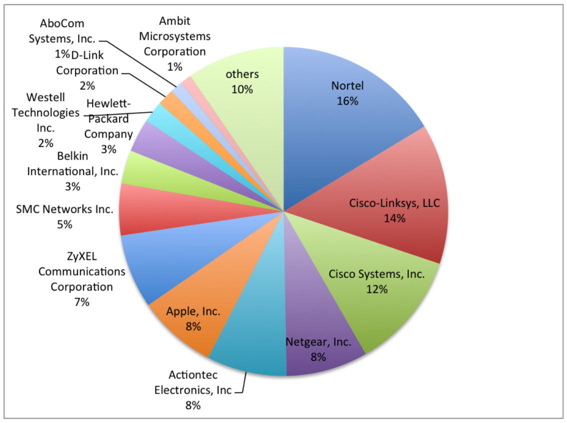

Figure 4: Access Points

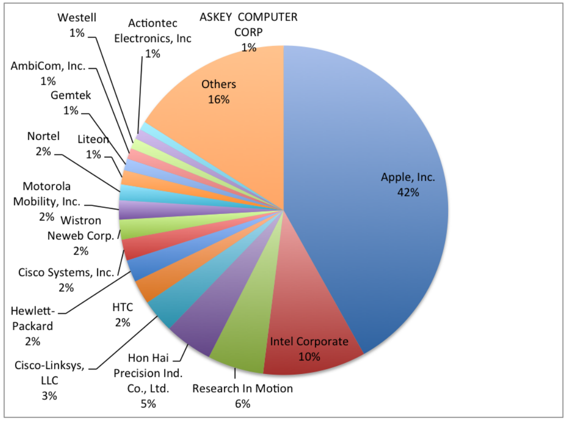

Figure 5: Clients' Wireless Card

3.4 Device Identification

Although the facus of this work is on device localization, we also did simple analysis on our traces for device identification. WiFum only collected probe response from access points, while Kismet saved all the packets it listened. In other words, Kismet not only identifies the existence of access points, but also those devices communicating with these access points. Intuitively, since MAC address is unique to a wireless card which is usually integrated to a mobile device, it can be used as a basic identification for a device. If MAC spoofing is used for malicious activities, other fields in the packets can be considered in further identification steps in order to eliminate the innocence and help to find the true criminal. The fields include the set of $< ipaddress, port >$, *SSID probes*, *broadcast packets sizes* and *MAC Protocol Fields*[12]. Furthermore, RF signature can be used to identify a wireless device by exploiting the imperfection of commercially used RF transmitter and amplifiers, which is difficult for attackers to modify [9][5][13]. We extract the MAC address by Kismet, and count them based on the Organizationally Unique Identifier [4] which is the first 24-bit in MAC address. We consider that the MAC addresses observed from BSSID fields are from access points, and others never appearing in BSSID fields are from client devices. In total, we observed 6615 client devices and 3807 access points. Figure 4 and 5 show the statistics of the manufacturers of the access point and client wireless cards. In the access points, Cisco-Linksys and Cisco Systems occupy 25%, Nortel covers 16%, Netgear, Apple, Actiontec Electronics covers 8% respectively. Regarding the client wireless card manufacturers, Apple occupies 42% and the next two most popular manufactures are Intel and *Research in Motion* known as Blackberry in smartphone world.

3.5 Trace Storage

In our *moving* experiment discussed in the beginning of Section 3, we collected 362,305 packets using Kismet, which is about 210MB in data size. These traces are only from four short trips with two trace gathering tools. If we extend our experiment into much longer periods, the data size would easily increase to 10GB - 100GB per day. It is a heavy load for our repository. To reduce the load, we analyze the packets types in 802.11 protocols. As shown in Figure 6, half of the packets are the beacons sent from access points. These beacons make little contribution to our system because they indicate nothing about mobile device and we want to focus on the activities from mobile device. Therefore, the beacon packets would be only preserved periodically. On the other hand, when a user is uploading/downloading a/an image/document/video, there are bunches of data packets in the session. Several of these packets in the session are enough for the identification and trace backing processes. We can use flow/session identifying techniques (e.g. sequence number) to identify the beginning and ending of a session, then select part of these packets to preserve in the repository as digital evidence.

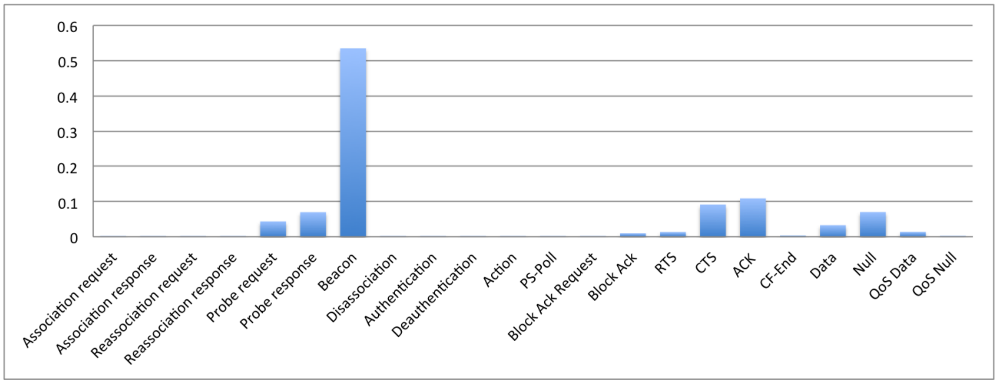

Figure 6: 802.11 Packet Type

4 Two Device Localization Approaches

In this section, we describe two widely used device localization approaches, present their performance based on our data set and discuss their pros and cons. The two approaches are K Nearest Neighbor (KNN) and a signal characterization approach using Log-distance Path Loss model.

4.1 K Nearest Neighbor

K nearest neighbor method, as a typical fingerprinting algorithm in localization, was first introduced in [6], is a relatively simple but effective algorithm. The fingerprinting of one position refers to the set of observed access points at that location and their associated signal strengths.

The first step in KNN approach is to establish an RF fingerprint database through experiments. At each physical location, a pair of position coordinates and a set of access points with their RSS levels are recorded. In the second step, the localization step, a new fingerprint with unknown coordinates are observed for a device, KNN then estimates the location of a device based on the distance between the newly observed fingerprint and the old ones in the database. Specifically, it selects K nearest points, and average their coordinates as the new fingerprint's position. The Euclidean distance is used as the distance between two fingerprints, shown in Equation (1). Figure 8 in our experiments described in Section 4.3 shows the median error distance is smallest when K equals 3.

$$d(newF, oldF) = \sqrt{(newF_1 - oldF_1)^2 + (newF_2 - oldF_2)^2 + ... + (newF_m - oldF_m)^2} \quad (1)$$

$newF_i, oldF_i$ are the two received signal strengths of the ith access point. When the ith AP is not observed in the location, its RSS would be set to -100dBm which is the signal strength below the threshold receivers could sense.

4.2 Log-distance Path Loss Model

To reduce the pre-deployment calibration effort, modeling methods are proposed to describe the radio propagation. Log-distance Path Loss Model (LDPM) [14][8] describes the average received signal strength decreases logarithmically with distance whether in outdoor or indoor radio channels, shown in Equation (2).

$$s_{ij} = S_i - 10\gamma_i logd_{ij} + X_\sigma \quad (2)$$

Put aside X_σ. To build models for access points, we need to determine four parameters $< S_i, \gamma_i, x_i, y_i >$ for each access point. Theoretically, four measurements of $< s_{ij}, x_j, y_j >$ are sufficient. In the positioning phase, one unknown location can be determined by three access points using triangulation. In current WiFi blanket coverage, Both conditions are easily satisfied. Instead, we have to solve a set of over-determined equations. There are several solutions to this problem. For example, Paper [8] proposed to find solutions to minimize the least mean absolute error of equations. To simplify the implementation, we used trust-region-reflective optimization approach to minimize the least square error which is defined in Equation (3).

$$J_i = \Sigma_j (s_{ij} - S_i + 10\gamma_i logd_{ij})^2 \quad (3)$$

4.3 Experiments and Results

Since Kismet monitors the entire spectrum by hopping through channels, the coverage of access points and mobile clients in our testbed is incomplete. We included 12 more data traces gathered through a similar measurement study conducted during a week of April, 2011 using Kismet. Adding the four traces gathered from the same device setting and four traces gathered from the alfa wireless card, we have a total number of 20 rounds of measurements.

4.3.1 KNN

We conducted experiments on the performance of average KNN and weighted KNN algorithms on the whole dataset. The difference between the two algorithms is that, during the position step, weighted KNN adds K closest coordinates p_i with a weight w_i as the estimated coordinates $\hat{p}$ shown in Formula (4). w_i is proportional to the Euclidean distance between the new and old fingerprint. The sum of w_i equals 1. In average KNN, each weight equals $1/K$. The result shown in Figure 7 indicates weight KNN (green line) achieves 1 meter less error distance than average KNN (blue line). Hence, we apply weighted KNN to the followed KNN experiments.

$$\hat{p} = \Sigma_i w_i p_i, (i = 1, 2, ..., K) \quad (4)$$

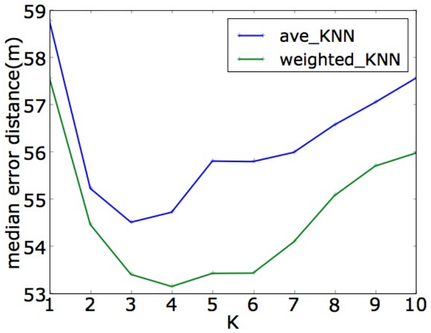

Figure 7: ave/weighted KNN

Table 2: Feature Number

Dataset	Feature number			
	all	all ap	ave ap	max ap
p-airport	7525	6249	3432	1529
airport, airport+alfa	10550	7813	4289	1889

In all measurements, we detected 12034 unique devices, among which there are 8521 access points. The signal signature from mobile devices can also be included in the fingerprint. But, it would require intensive computation if all the received signal strength from these devices are involved in the fingerprinting process. Furthermore, mobile clients may change their locations frequently. The RSSs from them vary while they move, while access points as infrastructure devices would be fixed at one place for a long time. Therefore, we compare the performance of including all devices named *all* in the above table and the performance of only access points is labeled as *all ap*. In addition, we apply two naive method to select access points. One named *max ap* is to select APs which show the maximum RSS at more than one location in the measurement. The other one called *ave ap* is to select those whose RSS is above the average at more than one location.

Considering device difference in RSS sensitivity described in Section 3, we split the dataset into three sets: *p-airport, airport, alfa+airport*. Here, *p-airport* consists 12 rounds of measurements which were made in one week using intergrated wireless card on the MacBook Pro laptop. The *airport* dataset contains all of 16 rounds using the airport. The *alfa+airport* dataset contains of all 20 rounds. We use 16 airport-made rounds as training data, alfa-made 4 rounds as testing data.

Figure 8(a) and (b) show the median and the standard deviation error of running feature filter methods on the three datasets. *p-airport-all* represents the performance of using RSSs from all devices as features in *p-airport* dataset. *p-airport ap* represents the performance of using RSSs from all access points as features in *p-airport* dataset. The corresponding number of applicable features are listed in Table 2. Although only about half number of APs in *ave ap* are used, the accuracy are quite similar to those involve all APs. More interestingly, when we include about a quarter number of APs with *max ap*, the median error distances achieve as small as using all devices. *max ap* is the best feature filter as it uses the least number of access points but delivers close to the minimum median error in all three datasets (36 meters, 38 meters and 47 meters respectively).

In Section 3, we have compared RSS sensitivity of three devices. Figure 8 suggests that using different measurement devices achieves different localization accuracy. When we use two sets from different devices in the experiments of *alfa+airport* shown in the figure with dash lines, the median error increase about 10 meters compared to other two experiments. The other two sets shown with dotted and solid lines have much smaller median error distance.

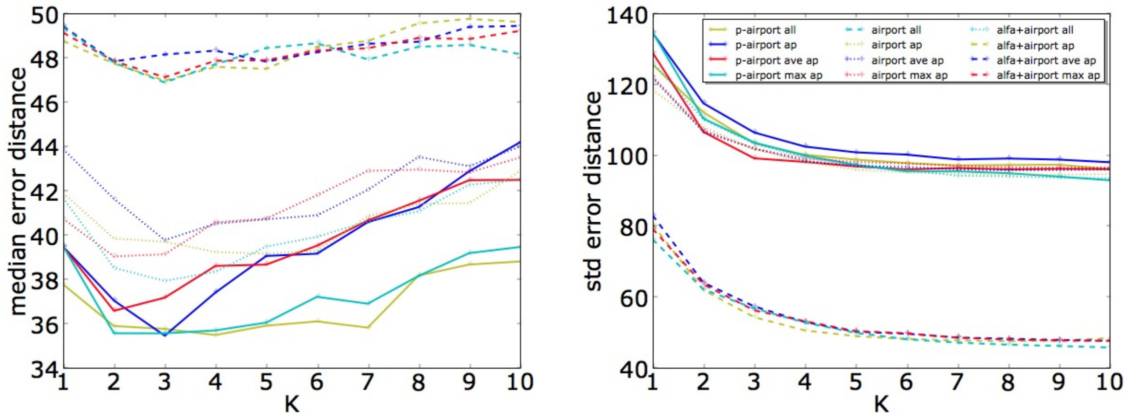

Figure 8: Error distance (meter) in KNN: (a) median error, (b) std error.

4.3.2 Log-distance Path Loss Model

We built models for access points filtered by *max ap* on the datasets of all airport measurements using the methods described in the previous section. The median error distance is 55m and standard deviation is 148m when we randomly select 20% data as testing data. The accuracy is a little lower than KNN and the variance is much larger. But, since it requires only 4 RSS from the device which need to be located, this method might be more applicable in cyber crime detection.

5 Discussion and Future Work

Our ultimate goal for this project is to establish a Security Monitoring system for Wireless Network Forensics [7], and we are particularly interested in designing such a system for densely populated urban neighborhood. However, in a metropolitan area such as New York City, it is not always easy to deploy monitoring points at a large number of APs. This makes it difficult to conduct study on evaluating the performance of different localization algorithms. Nonetheless, when Access Points receive signals from mobile devices, presumably the devices also receive signals from the Access Points. Intuitively, the signal from device to AP and the signal from AP to device should carry strongly correlated strength information. Therefore, in this work, we conducted a measurement study on the accuracy of localization algorithms in an environment with densly deployed WiFi networks through traces gathered by *mobile devices*.

Our study first shows some insight regarding the WiFi deployment characteristics, such as the density of WiFi networks. We then adopt three different equipment settings in the measurement for evaluating localization algorithms. In our experiments, signal strength information are gathered between testing devices and neighboring Access Points. Radio Frequency fingerprintings are thereafter established and used for later device localization. From our research, we observe that common GPS receivers (in our study the BU353 GPS receiver) do not work well in identifying the location of a mobile device in a metropolitan environment. Its performance degraded into the accuracy provided by Cellular and WiFi Networks. A tablet PC with an integrated GPS provides much more accurate location estimation for the mobile device. We also applied two localization schemes on our traces and realized that, the error distances derived from the two algorithms in the outdoor environment of a Metropolitan area are both much larger than those derived from indoors environments. It is particularly interesting to see that when KNN algorithm is used, the "performance optimal" is most likely to be achieved when $k = 3$. This indicates that in our future measurement study, in which AP monitoring will be involved, we should know it is possible that three APs which gather the strongest signals from a particular device should be enough to provide good localization results. Using more APs does not necessarily improve the location accuracy.

In the future, we will conduct measurement research at the Access Points in a controlled environment (indoor or/and outdoor), and combine the gathered traces with those observed from mobile devices to handle the investigation questions that we outlined in the very beginning: *Which device has appeared during what time at where!*

References

[1] Android location provider:
http://developer.android.com/reference/android/location/locationmanager.html.

[2] Backtrack http://www.backtrack-linux.org/.

[3] Kismet http://www.kismetwireless.net/.

[4] Oui: http://en.wikipedia.org/wiki/organizationally_unique_identifier.

[5] S. Dolatshahi A. Polak and D. Goeckel. Identifying wireless users via transmitter imperfections. *IEEE Journal on Selected Areas in Communications- Special Issue on Advances in Digital Forensics for Communications and Networking*, 2011.

[6] P. Bahl and V.N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000*, volume 2, pages 775 –784 vol.2, 2000.

[7] Yongjie Cai and Ping Ji. Security monitoring for wireless network forensics (smowf). *Systematic Approaches to Digital Forensic Engineering (SADFE)*, 2012.

[8] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. MobiCom '10, pages 173–184, New York, NY, USA, 2010. ACM.

[9] S. Dolatshahi, A. Polak, and D.L. Goeckel. Identification of wireless users via power amplifier imperfections. *ASILOMAR*, 2010.

[10] Azadeh Kushki, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Kernel-based positioning in wireless local area networks. *IEEE Transactions on Mobile Computing*, 6:689–705, June 2007.

[11] D. Madigan, E. Einahrawy, R. P. Martin, Wen-Hua Ju, P. Krishnan, and A. S. Krishnakumar. Bayesian indoor positioning systems. volume 2, pages 1217–1227, 2005.

[12] Jeffrey Pang, Ben Greenstein, Ramakrishna Gummadi, Srinivasan Seshan, and David Wetherall. 802.11 user fingerprinting. MobiCom '07, pages 99–110, NY, USA, 2007. ACM.

[13] A.C. Polak and D.L. Goeckel. Rf fingerprinting of users who actively mask their identities with artificial distortion.

[14] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[15] Teemu Roos, Petri Myllymki, Henry Tirri, Pauli Misikangas, and Juha Sievnen. A probabilistic approach to wlan user location estimation. *IJWIN*, 9(3):155–164, 2002.

[16] Moustafa Youssef and Ashok Agrawala. The horus location determination system. *Wireless Networks*, 14:357–374, 2008.

[17] Paul A Zandbergen. Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning. *Transactions in GIS*, 13:1467–9671, 2009.