INDOOR NAVIGATION SYSTEMS BASED ON iBEACON FINGERPRINTING

By

Meng Wang

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Computer Science

May, 2015

Nashville, TN

Approved:

Jules White
Douglas C. Schmidt

ACKNOWLEDGMENTS

I am very fortunate to have received help and inspiration from some people during my Master's thesis study, for which I feel always grateful.

First of all, I would like to thank my advisor, Dr. C. Jules White, who is extremely intelligent and supportive. This thesis would not have been possible without his guiding me in the right direction and contributing insightful discussions. His emphasis on perfectionism has constantly stimulated me to improve both in terms of sciences and my writing. Dr. White has also been very patient and supportive even when the project experienced technical drawbacks.

I would also like to thank our collaborator, in particular, Dr. Yu Sun of the California State Polytechnic University, Pomona, who is very knowledgeable in the field of web server and has been essential to the technical merits of the project by proposing insightful suggestions and creative solutions.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

1 INTRODUCTION

Indoor positioning systems (IPS) have become well-known and are a solution that use sensors, magnetic fields, or other signals, sensible by mobile devices, to locate objects inside buildings or specific regions. GPS technology is a mature and efficient positioning technology that has been widely used in mobile phones for outdoor localization and is based on the analysis of satellite signals to provide location information. However, indoors, GPS loses its ability to provide accurate navigation due to the inability to receive satellite signals. Indoor positioning systems are designed to overcome this drawback of the global positioning system (GPS) and provide precise indoor localization. Prior research on this topic has investigated several methods to provide indoor navigation, such as geomagnetic fingerprinting and inertial tracking.

GPS has become a ubiquitous technology for identifying the location of mobile devices outdoors. However, as discussed earlier, GPS does not provide accurate positioning indoors. This thesis investigates alternative approaches for providing indoor localization on mobile devices. In particular, this thesis investigates the feasibility of using Bluetooth Low Energy signals and machine learning for indoor localization.

One of the distinct challenges of indoor positioning on mobile devices is that many of the common positioning approaches are not possible to implement practically on a mobile device. In particular many indoor positioning technologies rely on radio frequency calculations that are not possible from a mobile app due to limitations placed by equipment manufacturers on what device hardware capabilities a mobile app is allowed access. For example, none of the major mobile platforms, such as android and iOS, allow apps access to the low-level interfaces of the underlying radios on the device, which prevent many common localization techniques from being used.

First, we explore existing localization techniques and evaluate their potential for use on mobile devices. For each technique, we discuss key limitations with respect to implementation on a mobile device. We perform this analysis from the perspective of localization techniques that are possible purely from the app layer of a mobile device.

After analyzing a variety of localization approaches, we introduce fingerprinting of Bluetooth signals, which is the focus of this thesis. Finally, we discuss empirical results from experiments fingerprinting Bluetooth signals in a 1.2M sqft test facility and how machine learning parameters affect fingerprinting performance.

1.1 Common Location Tracking Systems

There are four principal techniques that location tracking systems use for positioning, which are RF triangulation [1], RF/acoustic proximity [2], and fingerprinting [3]. Triangulation [4] uses the geometric properties of triangles to calculate object locations based on observed signals properties and is composed of angulation [5] and lateration [6]. RF angulation and lateration estimation are based on detecting the actual distance between a target location and a reference location based on signal analysis. In contrast to the "fine-grained" localization of RF triangulation, however, proximity detection provides a "coarse-grained" estimate of distance, but not position, from a fixed transmitter. Fingerprinting is a technique, which attempts to use machine learning to match a user's location against a predefined set of locations based on one or more characteristics of sensor signals at each of the locations.

1.1.1 Ranging Methods

A core component of many localization systems is ranging, which is the estimation of distance to known points. Two common methods are used to estimate range: monitoring signal characteristics from known RF transmitters or detecting signal reflection off of physical objects. RF transmitters can range from WiFi access points to detecting the closest tower in a cellular network. Reflection based approaches send out signals, such as ultrasound, that reflect off of physical objects. These reflections can then be used to estimate distance. These types of ranging approaches require clear lines of sight between the device and a known structure and can easily be interfered with by people in an area, making them unsuitable for mobile localization.

RF techniques are less impacted by people and other objects in an environment. Angle of Arrival (AOA) measurement [7,8,9] is one approach that analyzes the angle

between the direction of propagation of an RF signal wave and a reference orientation and can be used to aid in positioning. AOA localization is susceptible to measurement noise and also requires low-level access to a mobile device's radios that is not available to apps. Lateration techniques include time of arrival (TOA) [10,11,12] and time difference of arrival (TDOA). TOA, sometimes also called time of flight (TOF) [13,14], is the duration of time required to send a signal from a transmitter to receiver. The distance is calculated by the light speed in a vacuum. In arbitrary environments, there can be significant inaccuracy when there are lots of obstacles between transmitters and receivers. An additional challenge of TOA is that it requires the transmitter and receiver to be time synchronized, which can be hard on 1,000s of mobile devices. Another Lateration technique is time difference of arrival (TDOA). TDOA is an improved version of TOA and can reduce the complexity of time synchronization and reduce lost signals. It is a method that relies on each transmitter sending two signals to receiver. These two signals propagate at different speeds and the receiver can calculate the difference of arrival time between the signals to compute the distance.

1.2 Fingerprint-based Indoor Positioning Systems

Non-RF technologies of Indoor Positioning System (IPS) provide users with accurate indoor location by using ultrasound [15, 16, 17], inertial sensors [18], magnetic fields [19], etc. to estimate position through techniques, such as particle filtering. Some of these techniques are widely employed in non-mobile domains, such as sonic positioning in underwater vehicles and inertial navigation in autonomous ground vehicles. However, significant accuracy problems in the typical inertial and other sensor on mobile devices have limited their use in mobile apps.

Another type of indoor navigation approach that has been widely deployed in mobile is based on triangulation of WiFi signals [20, 21, 22, 23], which takes advantage of the wireless infrastructure in a building to localize a device. Wi-Fi-based positioning uses RSSI to estimate the distance to known WiFi access points. The primary drawback of WiFi triangulation is that its accuracy is dependent on the number of WiFi access points visible to the device. Due to cost, there are normally insufficient WiFi access points in a building to provide accurate positioning. Further, on iOS, apps cannot access the

lower-level information about WiFi signals needed to perform positioning.

Another approach that has been studied is the use of Bluetooth to estimate distances to pre-deployed Bluetooth transmitters. Previously, although these approaches showed promise, Bluetooth required too much power to be packaged into a low-cost long-lived device that could be cheaply deployed around a building for localization. However, the recent development of the Bluetooth Low Energy (BT LE) standard and Apple's iBeacon standard have allowed for the creation of small low-cost Bluetooth transmitters that can be deployed in a building for Bluetooth-based localization. These new Bluetooth transmitters, called beacons, have generated significant interest in the use of Bluetooth-based fingerprinting for localization. However, significant open questions about the best approaches for localizing device using Bluetooth signals have still not been answered. In particular, RF fingerprinting has been shown to be one of the most effective techniques for localization with beacons, but little research has looked at the fundamental questions related to how beacon signals should be fingerprinted and how machine learning parameters, which underlie these techniques, impact localization. This thesis investigates these critical research questions and provides concrete empirical data from a 1.2 million square-foot test facility.

1.2.1 RF Fingerprinting

RF Fingerprinting is an indoor navigation technology based on the collection of RSSI values and comparison of the values to a database of location fingerprints. To determine a device's location, the device records a sample of the signals currently visible to it and calculates RSSI values. These RSSI samples are then compared to previously recorded RSSI samples throughout the building in order to find the location that best matches the unique pattern of the sample that the mobile device currently sees. Many approaches have been used to perform this matching, with machine learning techniques, such as Random Forest, providing the best performance.

An advantage of this technology is that it does not suffer from many of the drawbacks of other localization approaches, such as requiring high-accuracy measurements, and is easily deployable at the app-layer on a mobile device. For example, RF triangulation requires accurate distance estimates to three known transmitters.

However, when a mobile device is in a user's pocket, the human body is mainly water and the resonance frequency of water is in the nearly of 2400 Hz, which can produce inaccurate distance measurements to transmitters, which adversely impacts triangulation. Fingerprinting can account for these types of inaccuracies in the database of location fingerprints. This ability for fingerprinting to account for many signaling, sensing, and other inaccuracies on mobile devices makes it a very attractive approach for indoor positioning.

Deployment of a fingerprinting system has two main steps: an offline phase to collect the initial samples to populate the fingerprint database and an online phase to predict a device's location. During the offline phase, beacons are deployed throughout the building and then sample data is collected at each location in the building. A model for estimating location based on a signal sample is built by running a machine learning algorithm on the data to learn the characteristics of the signals at each location. The sample data at each location includes the estimated distances to the deployed beacons, RSSI values to beacons, and location where the sample was taken. The collected data is stored in the database and is trained into a model by machine learning algorithms. During the online phase, the model is used by the device for localization. The device collects reference data, sends data to the server to make a location prediction using the model, and then receives the predicted location. Figure 1 depicts the fingerprinting approach.

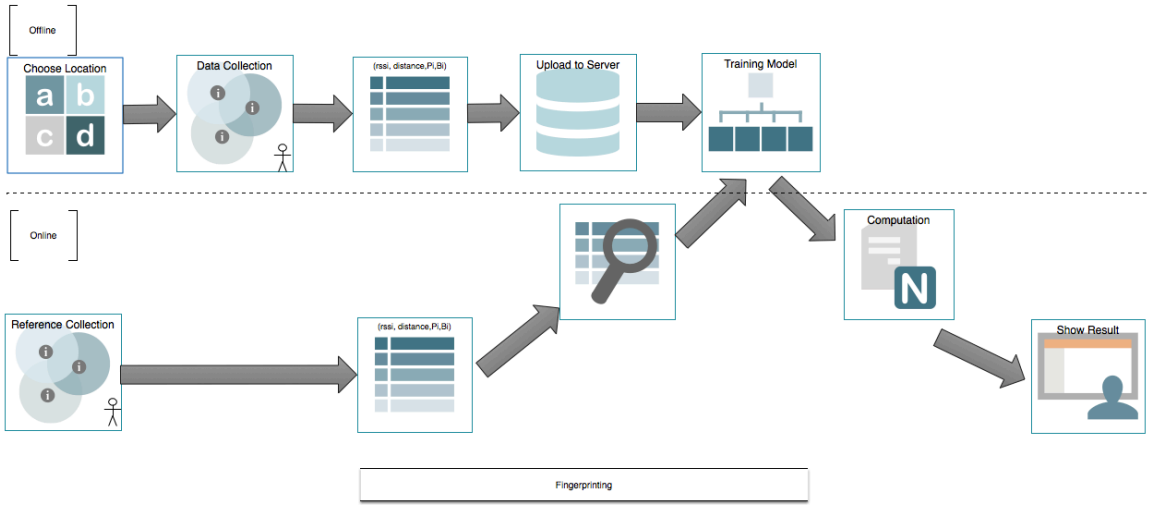


Figure 1: An overview of fingerprinting.

1.3 Received Signal Strength Indicator (RSSI) & Distance Estimation

RSSI is a radio receiver metric, which is described as a measurement of the power of the received signal at a receiver. It is a measurement of the power level being received by the device and produces higher RSSI values for more powerful transmitters. When the distance between the transmitter and receiver becomes larger, the signal power drops off and the RSSI value falls.

The advantage of RSSI ranging is that it is a low-cost ranging technology. In addition it can provide rough localization under the effects of multipath [24, 25] noise due to signal reflections in the surrounding environment. However, there exist some technical challenges of RSSI. First, When RSSI is used in an indoor environment; the signal strength is not linear and signal loss can sometimes happen due to multipath and shadowing [27, 28]. The structure of the rooms, walls, people, antenna orientation of device and doors will make the signal experience multipath effects. Human bodies are also comprised of water and affect the accuracy of RSSI. In addition, antenna gain has an impact on RSSI and this parameter is also influenced by the orientation of antenna. Further, as a room becomes smaller, non-linear signal loss can be serious and accuracy decreases. All of these effects make RSSI a challenging metric to work with when precise estimates are needed, such as for triangulation.

1.4 IBeacons

iBeacon is a new technology, developed by Apple, to provide location services in iOS apps. iBeacon is built on Bluetooth Low Energy and iOS devices can detect iBeacons, which are simple Bluetooth Low Energy transmitters that adhere to a specific transmission format, and estimate distance to them using RSSI values. iOS includes mechanisms for notifying apps when specific beacons are seen based on the identifiers of the beacons. In the iOS operation system, the core location framework has methods to register for notifications when a user enters or exits specific regions defined by the presence of an iBeacon [44]. Beacon region monitoring is in charge of detecting beacon signals to notify the listening app but each app is limited to listening to 20 regions or UUIDs (universally unique identifiers). iBeacon has two main configurations that can be

leveraged for indoor proximity detection, which are: region monitoring and ranging. Beacon region monitoring notifies an app when it enters an area defined by the device's proximity to Bluetooth low-energy beacons with a specific UUID. Ranging notifies an app continuously as its distance to beacons with a specific UUID changes. The distance is discretized into three different buckets: immediate (within centimeters), near (within few meters) and far (greater than 10 meters). The maximum distance of an iBeacon transmission is dependent on the placement of beacons and obstructions around the environment. In addition to a UUID, each beacon has a major value and minor value that are used for application-specific purposes.

1.5 Research Questions

Previous literature on fingerprinting of Bluetooth signals has not focused heavily or at all on Bluetooth low-energy signals. In particular, little analysis has been done on how deployment of eye beacons at scale and various properties of mobile devices impact localization performance in a fingerprinting based indoor positioning system. Finally, little research has answered key questions about how different machine learning techniques and parameters impacts the accuracy of the models used for fingerprint matching and localization. This thesis investigates the following key unanswered questions within this research space:

- **Research Question 1: How does beacon placement impact localization performance?** In many real-world scenarios, beacons cannot be easily placed in the building at arbitrary locations due to aesthetic, structural, or other issues. An important issue to analyze is understanding how variations in placement impact the performance of the localization system. For example, the angle between a beacon's antenna and a mobile devices antenna can impact the range and signal strength of our SSI measurements which can impact localization performance.

- **Research Question 2: How many beacons are needed to reach high accuracy localization in a real-world environment?** Without concrete data on exactly

how many beacons are needed and metrics for measuring Beacon coverage, it's difficult to deploy a beacons at scale in a real building. Concrete metrics for analyzing how Beacon deployment density impacts localization accuracy are critical for cost effective and high accuracy indoor localization.

- **Research Question 3: How do different machine learning algorithms perform for fingerprint matching and localization?** The few papers that exist on this topic analyze a single machine learning algorithm and report results on localization performance. However, to further the scientific understanding of indoor positioning, comparative analysis of different machine learning algorithms as needed.

- **Research Question 4: How do machine learning parameters impact indoor navigation performance?** Machine learning out rhythms can have widely varying accuracy based on parameter values that are provided to the machine learning algorithm. In order to produce a high accuracy indoor navigation system, understanding how these parameters impact fingerprinting performance is critical.

2 EMPIRICAL RESLUTS

To answer these key research questions, we conducted a number of experiments to determine how fingerprinting performs in real world environments and how beacon placement and other parameters impact localization performance. The experiments were conducted in a unique environment, the Nashville Music City Center, which is a convention center in downtown Nashville. As part of a research collaboration with Vanderbilt University, 63 beacons were deployed throughout the building to provide signaling for an indoor navigation system covering 1.2 million sqft.

2.1 Device And Platform

To conduct the experiments, we employed us a variety of iPhone smartphones for experimentation. All iPhones ran iOS 7 or later and had a Bluetooth 4.0 radio, which is required for detecting beacons. All applications were developed in Objective-C using Apple's XCode development environment.

2.2 Beacon Placement Experiments

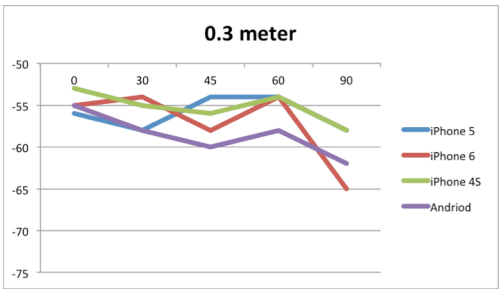
Beacons broadcast signals in a cone shape and for the best performance of detecting a signal, it is better to attach the beacon to the walls mounted on a metal wedge to aim the cone at the specific area. There is a major factor, which affects the accuracy of distance measurements: the orientation of the beacon's antenna. Based on the normal posture that users take when holding a phone (i.e., phone in hand held out in front of the body), we did experiments to find the relationship between relative angle of beacon mount orientation to the wall and RSSI values observed on the device.

Our experiments to measure the impact of beacon placement orientation on RSSI value were performed in a hallway of the Music City Center. In the hallway, we placed a beacon at varying linear distances of 0.3m, 0.5m, 1m, 2m, 3m, 4m, 5m, 6m, 10m and

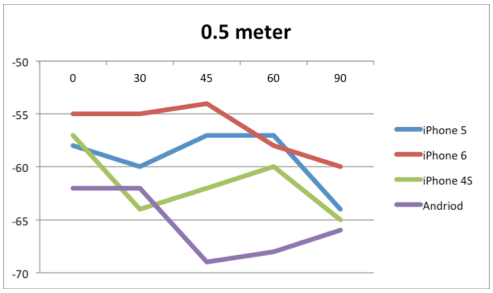
measured the RSSI value recorded on the phone while at orientations relative to the wall of 0 degrees, 30 degrees, 45 degrees, 60 degrees, and 90 degrees to the beacon with 4 different smartphones respectively. In each test, the device was moved slowly back and forth on a 30 cm line, maintaining orientation, remaining equidistant from the measuring device, and this operation was repeated for one minute [44]. After this phase, we gathered the RSSI values from the device and make an average of the collected RSSI values to obtain the measured power value for the orientation [44]. In each measured distance with the same angle the beacon was attached to metal bracket [46]. During the experiment, we measured the performance of RSSI on several different cellphones, such as the Nexus 5, iPhone 4S, iPhone5, and iPhone 6 Plus. The hardware and Bluetooth parameters of the different smartphones are shown in Table 1.

Table 1 Hardware Configuration of Test Devices.

Phone	Processor	RAM	Bluetooth
IPhone4s	1 GHz (under-clocked to 800 MHz) dual-core ARM Cortex-A9 Apple A5 (SoC)	512 MB LPDDR2	Bluetooth 2.1 + EDR (Broadcom 4325)
IPhone5	1.3 GHz dual-core Apple-designed ARMv7s Apple A6	1 GB LPDDR2	Bluetooth 4.0
IPhone5s	1.3 GHz dual-core Apple-designed ARMv8-A 64-bit Apple A7 with M7 motion coprocessor	1GB LPDDR3	Bluetooth 4.0
IPhone6plus	1.4 GHz dual-core Apple-designed ARMv8-A 64-bit Apple A8 with M8 motion coprocessor	1GB LPDDR3	Bluetooth 4.0
Nexus 4	1.5GHzquad-core Krait Adreno 320	2GB of LPDDR2 RAM, clocked at 533MHz	Bluetooth 4.0 with A2DP

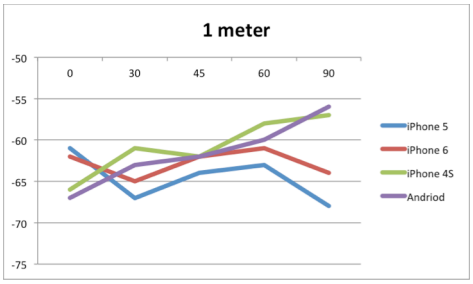


(a) Correlation between RSSI and orientation angle at a distance of 0.3 m.

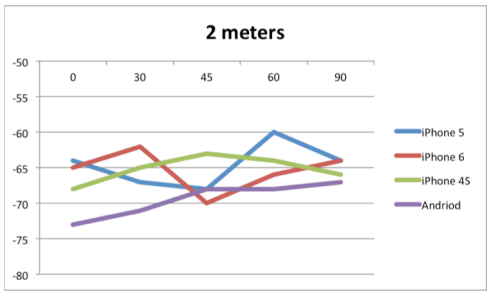


(b) Correlation between RSSI and orientation angel at a distance of 0.5 m.

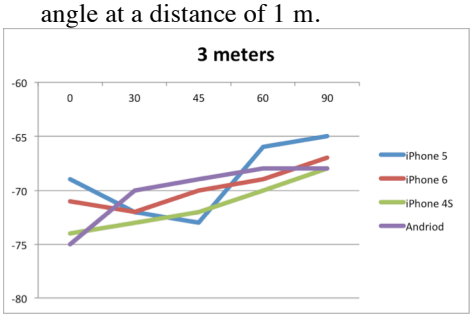
Figure 2 Correlation between RSSI and orientation angle at an intermediate distance (less than 1 meter).



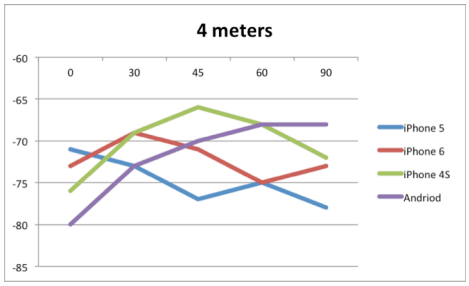
(a) Correlation between RSSI and orientation angle at a distance of 1 m.



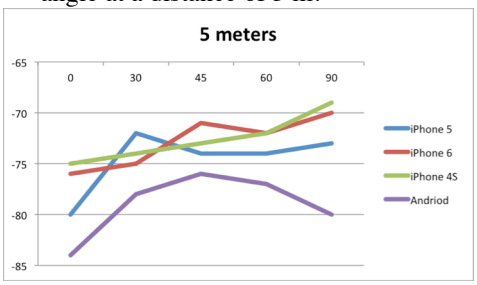
(b) Correlation between RSSI and orientation angle at a distance of 2 m.



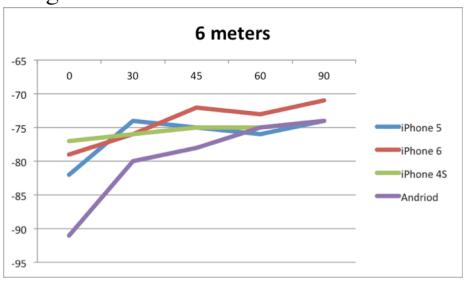
(c) Correlation between RSSI and orientation angle at a distance of 3 m.



(d) Correlation between RSSI and orientation angle at a distance of 4 m.

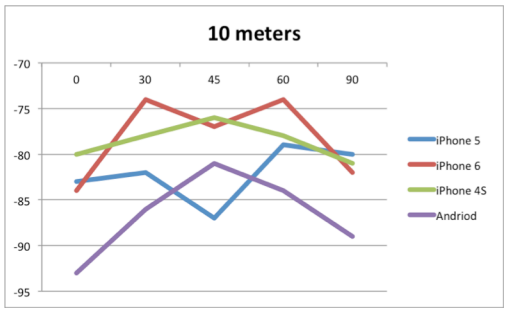


(e) Correlation between RSSI and orientation angle at a distance of 5 m.



(f) Correlation between RSSI and orientation angle at a distance of 6 m.

Figure 3 Correlation between RSSI and angles at the near distance (less than 8 meters).



Correlation between RSSI and orientation angle at a distance of 10 m.

Figure 4 Correlation between RSSI and angle to the beacon at the far distance (more than 8 meters).

Figure 2, 3 and 4 show the correlation between orientation angle of the beacon and RSSI value with different devices. The distances are in the ascending order, and were measured at 0.3 meters, 0.5 meters, 1 meters, 2 meters, 3 meters, 4 meters, 5 meters, 6 meters, and 10 meters.

In Figures 2, 3 and 4, the vertical axis represents the RSSI value measured on the device and the horizontal axis is the angle of orientation that the beacon was mounted on the wall. When the orientation angle is close to zero; the average measured RSSI value on the devices is higher. Based on the results shown in these figures, we can draw some conclusions that when the distance between the beacon and the phone is less than 1 meter, it is better to mount the beacon at 0 degree; For expected distances to the device of 1-6m, a mount orientation of 90 degrees is most appropriate. Beyond 6m, no single mount orientation provides a consistently stronger signal. The iPhone 6 Plus has best overall performance of detecting the RSSI value based on this test.

This data on beacon orientation for deployment can be used to supplement beacon placement optimization algorithms, such as the G1 algorithm [45]. The G1 algorithm is a simply greedy algorithm that optimizes beacon placement location by selecting a minimal set of hyperplanes that cover an entire indoor space and assigning beacons to the hyperplanes [45]. Adding orientation data to the algorithm could help to improve its positioning decisions and simultaneously optimize beacon installation orientation.

3 FINGERPRINTING EXPERIMENTS

For the fingerprinting experiments with machine learning, we collected data from the beacons in the Music City Center, including RSSI value, distance, major value, and minor value of beacons. Table 2 shows the different configurations of devices we used to collect data and Table 2 illustrates the format of the collected data. In the process of collecting data, the experimenters walked with an average speed of 1.0m in random directions within specific region to try to collect a random sample of data. The recording application sampled the beacon signal data twice per second. In the Music City Center, we used 126 locations and collected 9756 samples for the dataset.

Table 2 Format of data in the dataset

Beacon1 Major value	Beacon1 Minor Value	Beacon1 Distance	Beacon1 RSSI	…	BeaconN Major value	BeaconN Minor Value	BeaconN Distance	BeaconN RSSI	Region name
0	1	3.23	-69	…	0	78	11.32	-73	Room A
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	5	1.34	-59	…	0	76	12.98	-70	Room N

The key question we wanted answer in this research was how different machine learning algorithms and their parameters impact localization accuracy. We used the data set that we collected in the Music City Center to train very machine learning algorithms and then measure their performance when localizing a mobile device using data sampled within the Music City Center. The Music City Center provided a large-scale testbed -- the larger than any testbed that has been presented in published literature -- to investigate these critical questions. In the remainder of this chapter, we explore varying machine learning algorithms and the impact of their parameters on localization accuracy. In the experiments, we compared the C4.5 [29, 30, 31, 32], Random Forest [33, 34, 35], and Bayesian Networks [36, 37, 38] algorithms.

In this chapter there are two main experiments: the first one is aimed to find best parameter configuration of machine algorithms and the other one is aimed to check how many neighboring beacons are needed for a given level of accuracy. The dataset, which is used in the experiments includes 576 random samples from within the Music City Center and is analyzed against models produced from a training dataset containing 9756 data points from the Music City Center.

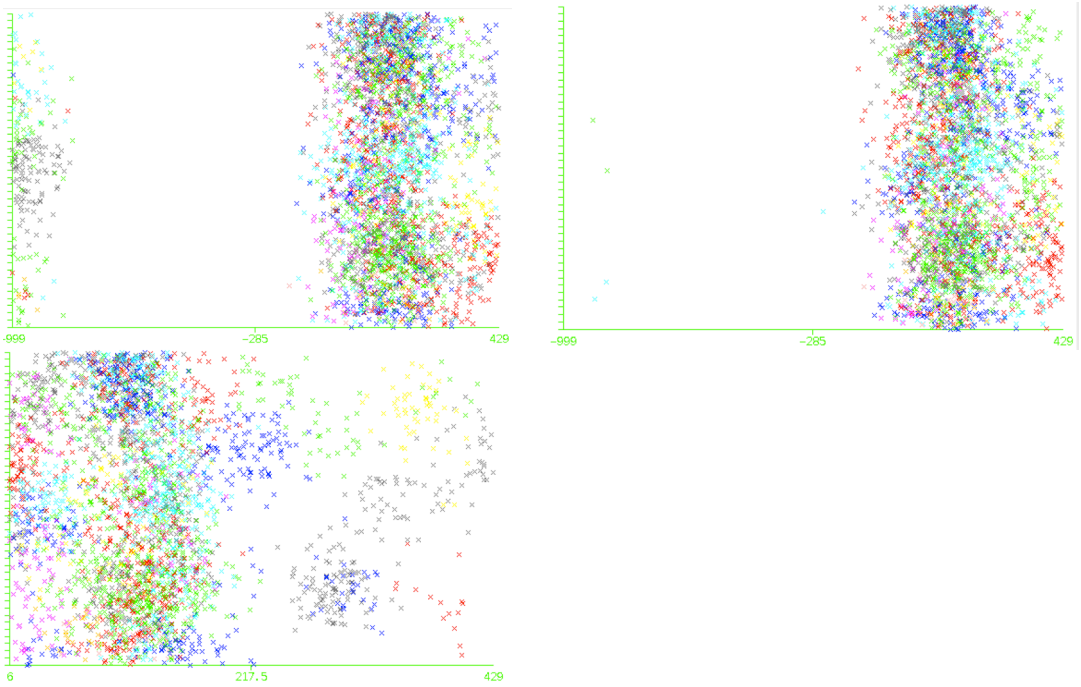


Figure 5 Correlation between minor value of beacon and location id.

Figure 5 shows a visualization of the test dataset that correlates location with beacon identity. The top left of the Figure is the correlation between minor values of the third nearest beacon and location where the sample was observed. The top right figure shows the correlation between the minor values of the second nearest beacon and the location where the sample was obtained. The bottom left shows the correlation between minor values of the nearest beacon and the location where the sample was obtained. The x-axis indicates the minor value of beacons and y-axis is the location. A negative minor value indicates that the device did not find a beacon for that scan.

3.1 Results Of Experiments With The C4.5 Algorithm

The first set of experiments that we ran were with the C4.5 algorithm [29, 30, 31, 32]. These experiments tested the performance of this algorithm on localization using randomly sampled locations from within the Music City Center. The decision trees produced by the algorithm were based on the entire training data set from the Music City Center. Various experiments were performed to understand how parameters and other aspects of the algorithm impacted localization accuracy.

3.1.1 Confidence Factor Impact On Accuracy

The first experiment with this algorithm investigated how the algorithm's confidence parameter [39], which controls how aggressively the tree is pruned, impacted its location prediction accuracy. We wanted to understand how tree pruning due to the confidence parameter influenced the real-world accuracy of the algorithm.

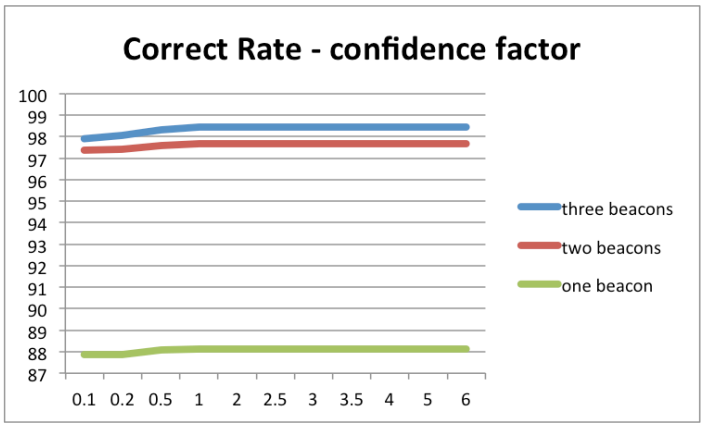


Figure 6 Correlation between correct localizaton rate and confidence factor in the C4.5 algorithm.

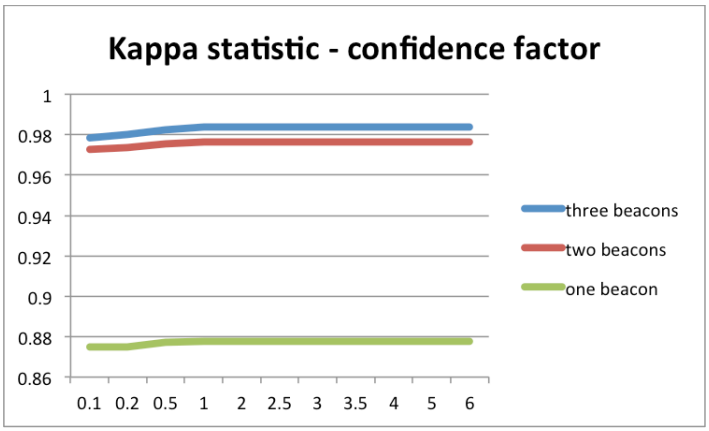


Figure 7 Correlation between kappa statistic and confidence factor in the C4.5 algorithm.

Figure 6 above shows the relationship between correct localization rate and the confidence factor provided to the C4.5 algorithm when varying numbers of beacons were visible. The horizontal axis indicates the confidence factor provided to the algorithm and the vertical axis indicates the correct localization rate. From the data shown in the figure, we can draw some conclusions:

1. The common feature of these three different datasets is that there is an increase in accuracy as the confidence factor increases from 0.1 to 1.0. Beyond a confidence value of 1.0, the accuracy rate does not change.
2. When the dataset includes more neighboring beacons, the localization rate is higher. There is an obvious reduction in accuracy from the two beacon dataset to the one beacon dataset. This is an interesting note because many

commercial approaches attempt to rely on a single visible beacon for localization, which is not optimal. However, the difference between having two and three beacons visible is less than 1%.

Figure 7 shows how the kappa statistic [40] changes based on the confidence factor provided to the C4.5 algorithm with different number of beacons present. The horizontal axis indicates the confidence factor and the vertical axis is the Kappa statistic value. The closer to the one Kappa statistic is; the more accurate the model is. From the figure, we could draw some conclusions:

1. As with the prior dataset, the Kappa statistic value does not improve when the confidence value is greater than 1.0. Beyond a confidence value of 1.0, the Kappa statistic keeps the same value.
2. When more beacons are visible, the Kappa statistic value is higher. However, the difference in Kappa value between the two beacon and three beacon datasets is less than 0.008.

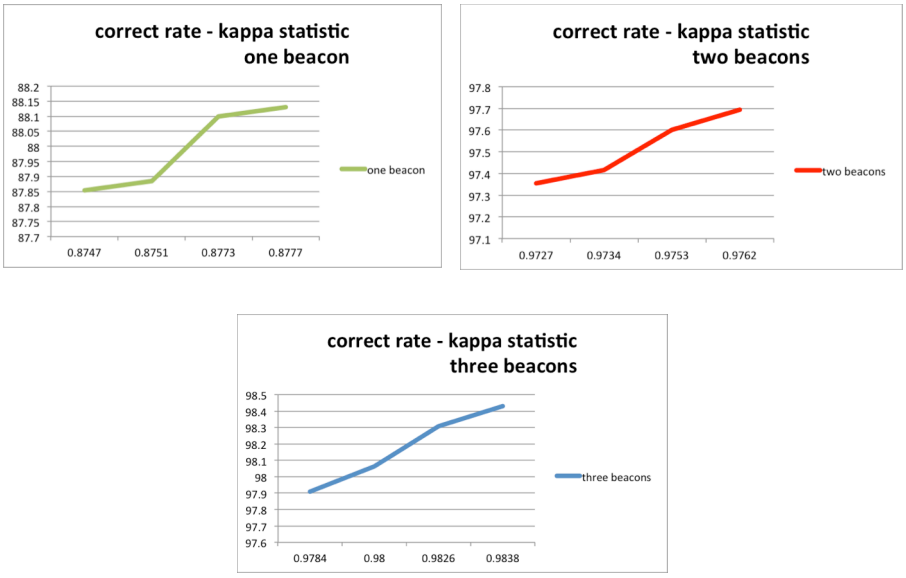


Figure 8 Correlation between correct instances rate and kappa statistic from one beacon dataset, two beacons dataset and three beacons dataset in C4.5 algorithm.

The above Figure 8 shows the correlation between correct localization rate and kappa statistic for the one beacon, two beacon, and three beacon datasets respectively. All of the figures show that the kappa statistic is relatively positive. The correlation between correct localization rate and Kappa Statistic is more positive as the number of neighboring beacons increases.

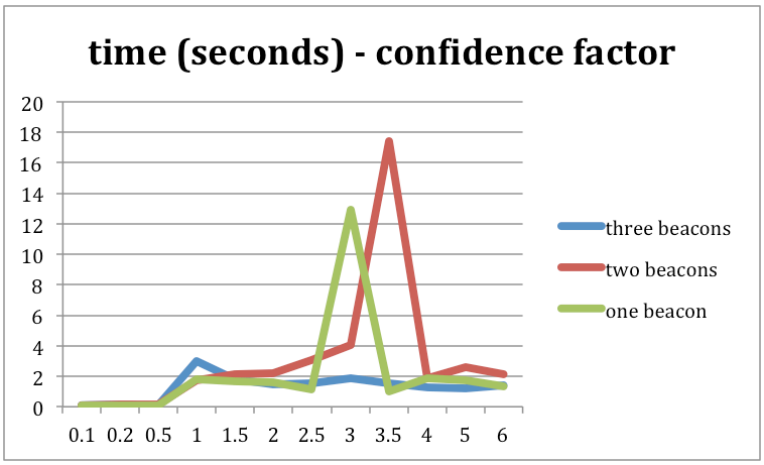


Figure 9 Correlation between time taken to make model and confidence factor in C4.5 algorithm.

Figure 9 shows how the confidence factor impacts model building time for the various datasets. The time required to build the model is less than 1 second when the confidence factor is from 0.1 to 0.5. When the confidence factor is within the interval from 1 to 2.5 and after 4, the time spent building the model does not change much. In the confidence factor interval between 2.5 and 4, the time to build the model increases dramatically. An interesting result is that the algorithm can build a model much more quickly for the three beacon dataset, regardless of the confidence factor.

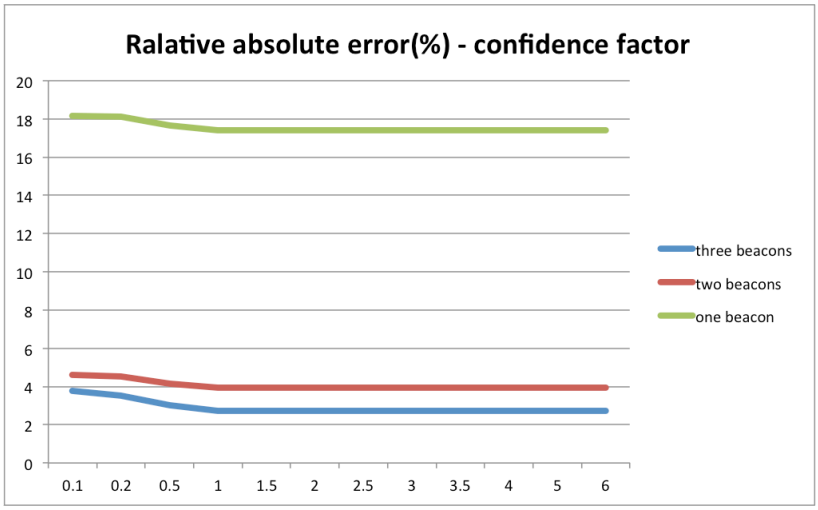


Figure 10 Correlation between relative absolute error and confidence factor in C4.5 algorithm.

Figure 10 shows the relationship between absolute error and the confidence factor. The x-axis represents the confidence factor provided to the algorithm and the y-axis indicates the relative absolute error. As was expected given the prior results, the relative absolute error remains roughly constant above a confidence factor of 1.0. Furthermore, increasing the number of neighboring beacons reduces relative absolute error.

Figure 11 Different correlations of training data and test data between correct instances rate and confidence factor in C4.5 algorithm.

Figure 11 shows an unexpected result. Although the overall accuracy of the trained model has an improvement in accuracy when validated against the training data, increases in confidence factor have little impact on the localization accuracy of the model with the real world sample data. The result shows that the biggest difference between correct locatization rate for the randomly captured test data samples is less than 0.1% regardless of the confidence factor value. This result indicates that the model will not break down with unknown data, or when future data is applied to it.

Combined with prior conclusions, to ensure perfect agreement, correct instances rate, relative absolute error and time cost, the best choice is to ensure that a minimum of three beacons are visible at every location within a building. Further, a confidence factor of 1.0 is a reasonable value to produce high accuracy and lower model creation times.

3.1.2 Test Of Number Of Folds

The following results show the impact the algorithms parameter that determines the number of folds [42] and the accuracy of the trained model.

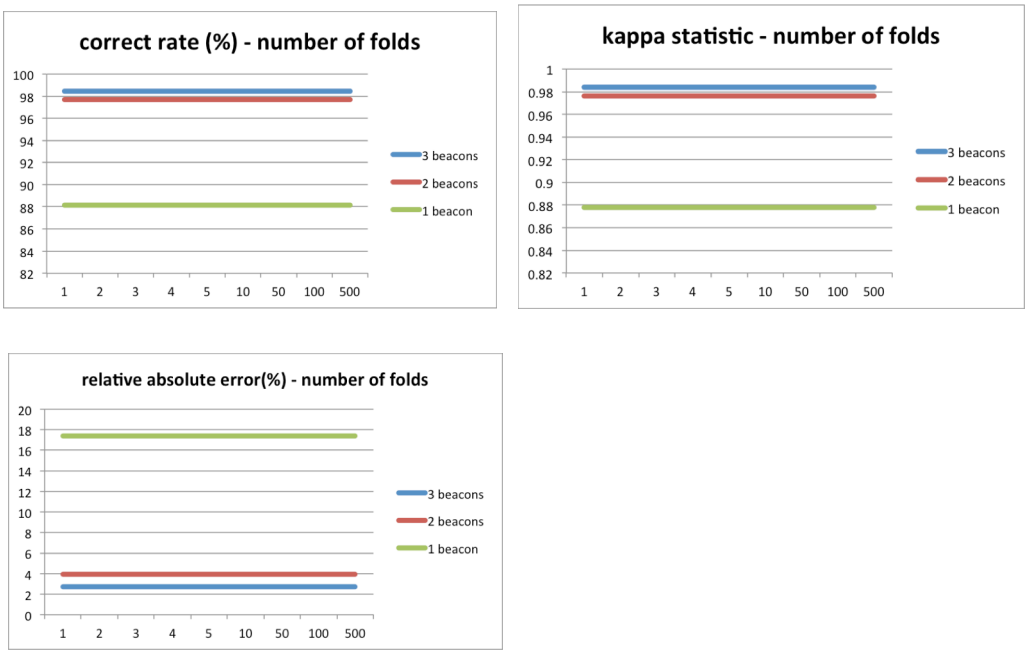


Figure 12 Correlation between number of folds and other main evaluation standards in C4.5 algorithm.

In Figure 12, we can see that the number of folds does not influence the accuracy of the model, kappa statistic, or relative absolute error. Further, regardless of the number of folds, the 3 beacons dataset produces the highest accuracy.

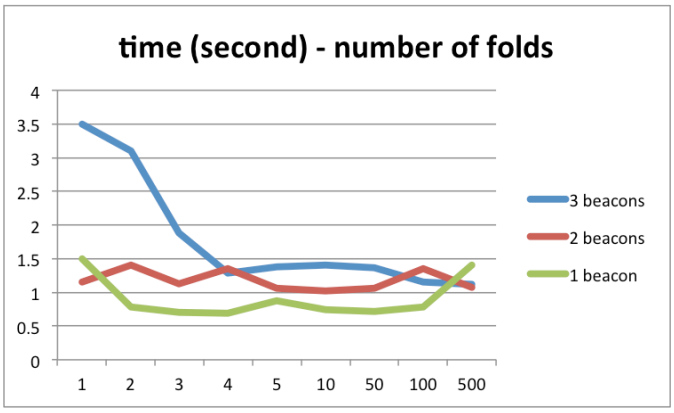


Figure 13 Correlation between time and number of folds in C4.5 algorithm.

Figure 13 shows the time to construct the model as a function of the number of folds. The x-axis is number of folds and the y-axis is the time consumed to build the model. In this Figure, we find the number of folds does influence the time consumed to build the model and its overall size. Therefore, when the number of folds is 2, 3 or 4, the time and space size are relatively balanced. To build a model of the 3 beacons dataset, choosing 4 folds for the C4.5 algorithm produces fast build times with high accuracy. For the model with 2 beacons, choosing 5 folds produces the best balance of accuracy and modeling building time.

3.2 Results Of Experiments With Random Forest Algorithm

We next investigate the Random Forest algorithm, which is another high-performance tree-based classification approach. We focus on the influence of one key parameters: the number of sub-trees. Typical values for the number of sub-trees are 10, 30, 50 and 100. We did tests to compare their performances.

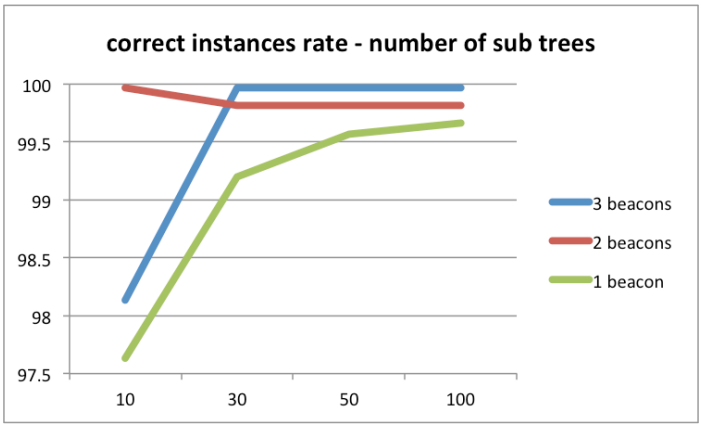


Figure 14 Correlation between correct instances rate and number of sub trees in Random Forest.

Figure 14 shows the impact of the number of sub trees on accuracy. The x-axis is the correct localization rate (%) and the y-axis indicates the number of sub trees. As the number of sub trees increases, accuracy of both the 3 beacon and 2 beacon datasets increases and stays relatively constant. The correct localization rate of the model of the two beacon dataset drops a little and at 30 sub trees. In this case, performance of the model of the 3 beacons dataset does the best when the number of sub trees was equal or beyond 30 sub trees. The accuracy of model of the 2 beacons dataset was highest when the number of sub trees was equal to 10.

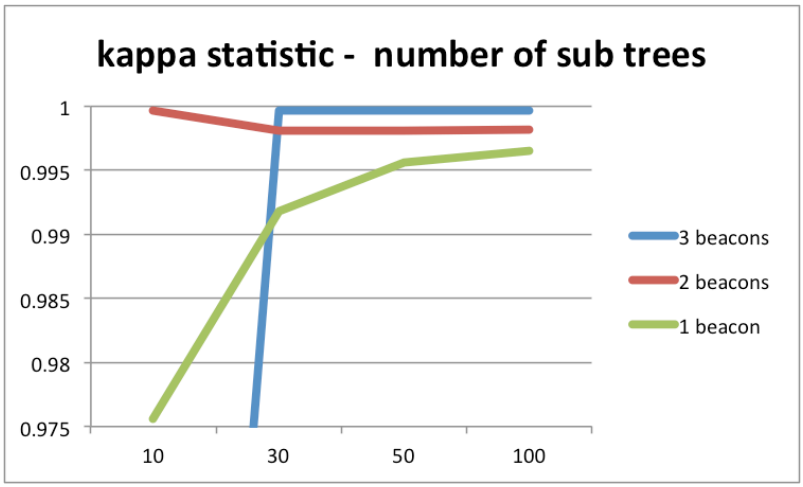


Figure 15 Correlation between kappa statistic and number of sub trees in Random Forest.

Figure 15 shows the impact of the number of sub trees on the Kappa Statistic. The x-axis indicates the kappa statistic and the y-axis is number of sub trees. The results mirror what was seen for the impact of sub trees on localization accuracy.

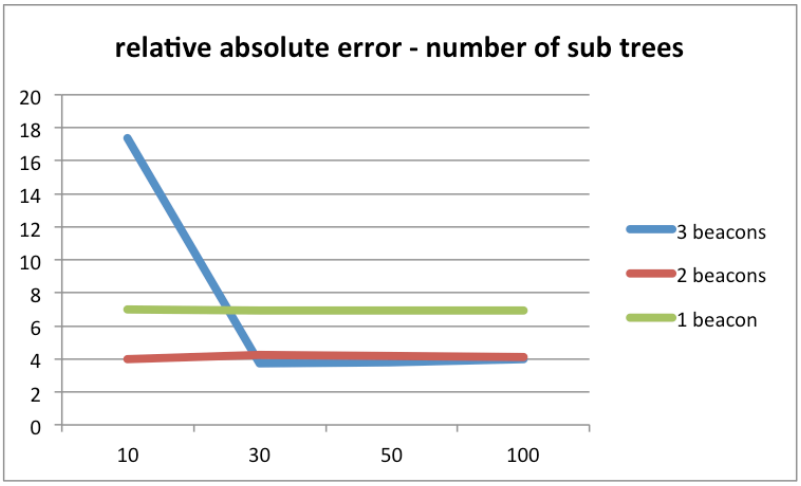


Figure 16 Correlation between relative absolute error and number of sub trees in Random Forest.

In Figure 16, the vertical axis is the relative absolute error and the horizontal axis is the number of sub trees. In this Figure, we can see that when the number of sub trees is equal to 30, the relative absolute error of the model of the 3 beacons dataset is minimized as expected. Furthermore, the relative absolute error of both the model of the 2 beacon dataset and one beacon datasets stays relatively constant as the number of sub trees increases. Overall, the most accurate predictions are achieved with the 3 beacon dataset and 30 sub trees.

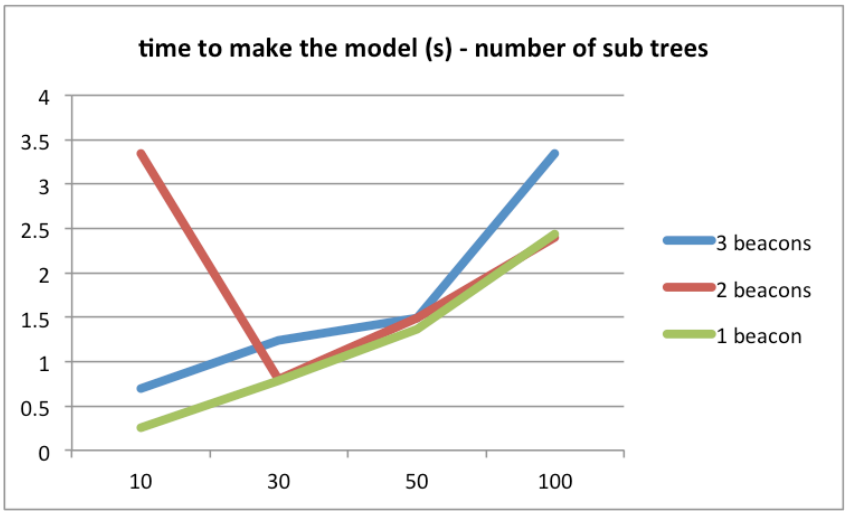


Figure 17 Correlation between time and number of sub trees in Random Forest.

Figure 17, shows the correlation between the time to build the model and the number of sub trees. The time to construct a model with the 3 beacon dataset and the one beacon dataset increases as the number of sub trees increases. The model of the 2 beacons dataset is unique because the time consumed decreases when the number of sub trees is less than 30, reaches the minimum at 30 sub trees, and increases when the number of sub trees is more than 30.

We used the same test dataset with 478 instances to test the model accuracy with different number of sub trees. The result shows that the accuracy of all models approaches 100%. The maximum difference between correct localization rate for the training data and that of test data is not beyond 2%. This result indicates that the model of will not break down with unknown data, or when future data is applied to it.

Based on the results, we can draw the following conclusions:

- The overall accuracy of all datasets is above 97.5%.
- The kappa statistic is very high and all reach beyond 0.97.
- When the number of sub trees reaches 30, all models take relatively less time to get relative higher accuracy and an improved kappa statistic.
- For the model of the 3 beacon data, it is better to choose 30 sub trees in Random Forest to get the best performance.
- For the model of the two beacons data, choosing 30 sub trees in Random Forest

maybe the best choice. Although accuracy and kappa statistic is not best, time consumed in 30 trees could save about 2 seconds computation time and the difference with best accuracy is not beyond 0.2%.

- For the model of the three beacon data, choosing 10 sub trees in Random Forest maybe the best choice. Though accuracy and kappa statistic is not best, time consumed in 10 trees could save about 2.5 seconds computation time and the difference with best accuracy is not beyond 2%.
- In the Random Forest method, both the three beacon dataset and the one beacon dataset are a good choice to make an accurate model.

3.3 Results Of Experiments With Bayesian Networks

The experiment with Bayesian Networks [36, 37, 38] focus on the values for the alpha parameter [42, 43]. The alpha is an important parameter in the estimator of the Bayesian Networks and used for estimating the probability tables that the algorithm relies on. We conducted tests to see how different alpha values affected the localization accuracy.

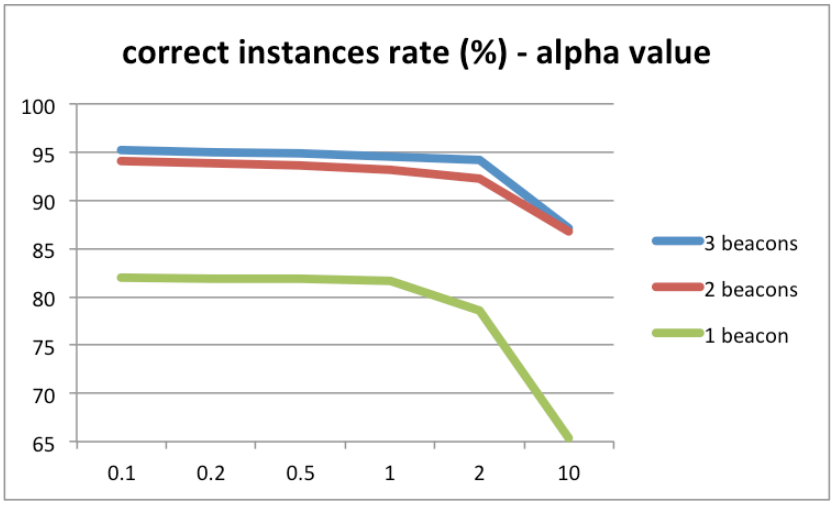


Figure 18 Correlation between correct instances rate and alpha value in Bayesian Networks.

In Figure 18, the vertical axis is the correct localization rate and the horizontal axis is the alpha value of the estimator [42, 43]. We can see in this figure that as the value of alpha increases, the correct localization rate decreases. This reduction in accuracy is

particularly apparent when the value of alpha is beyond 2. In addition, both the 3 beacon and 2 beacon datasets have relatively good performance when the value of alpha is near 2. The accuracy of the model produced with the one beacon dataset is poor.

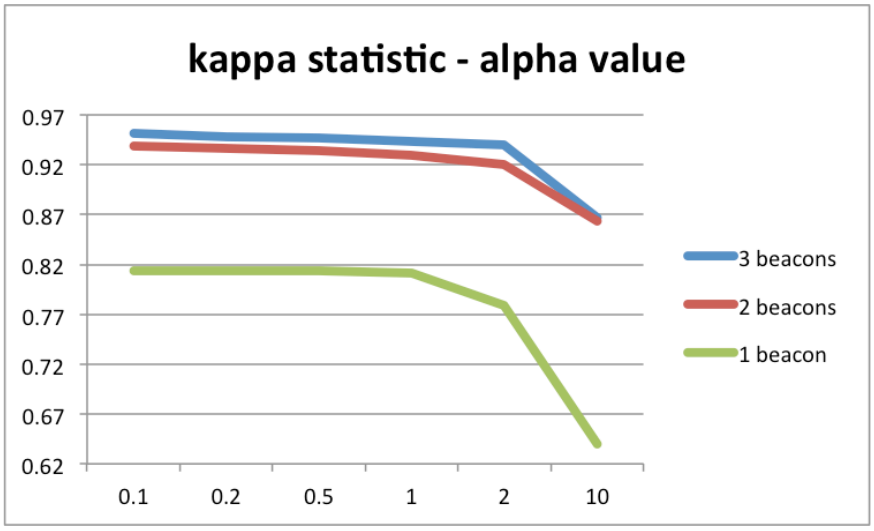


Figure 19 Correlation between kappa statistic and alpha value in Bayesian Networks.

The vertical axis in Figure 19 is the kappa statistic and the horizontal axis in this Figure is the value of alpha. The kappa statistic of all the models decreases when the value of alpha increases. The models built with the 3 beacon and 2 beacon datasets have a relatively better kappa statistic than the model built from the one beacon dataset.

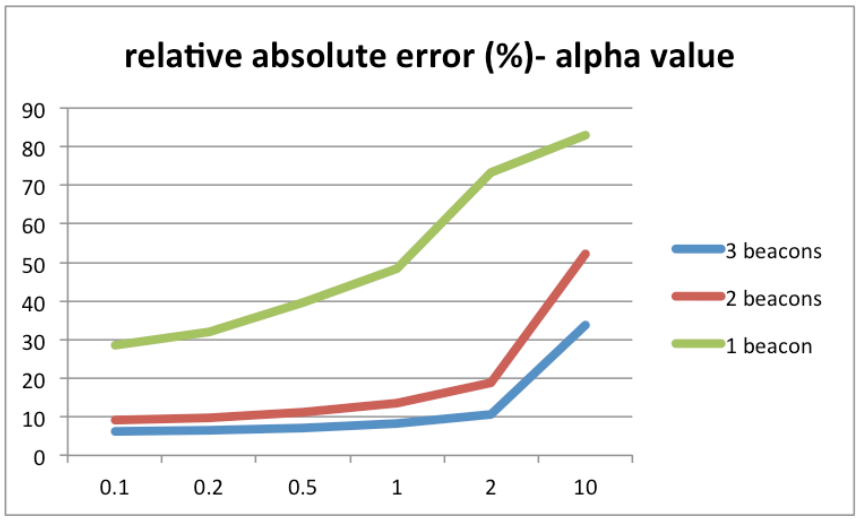


Figure 20 Correlation between relative absolute error and alpha value in Bayesian

Networks.

Figure 20 shows the correlation between relative absolute error and value of alpha. The vertical axis is the percentage of relative absolute error and the horizontal axis is the value of alpha. We can learn from the Figure that as the value of alpha increases, the percentage of relative absolute error increases. When the value of alpha is lower than 0.5, the relative absolute error of the model lower. The models from the 3 beacon and 2 beacon datasets have the lowest relative absolute error when the value of alpha is less than 0.5.

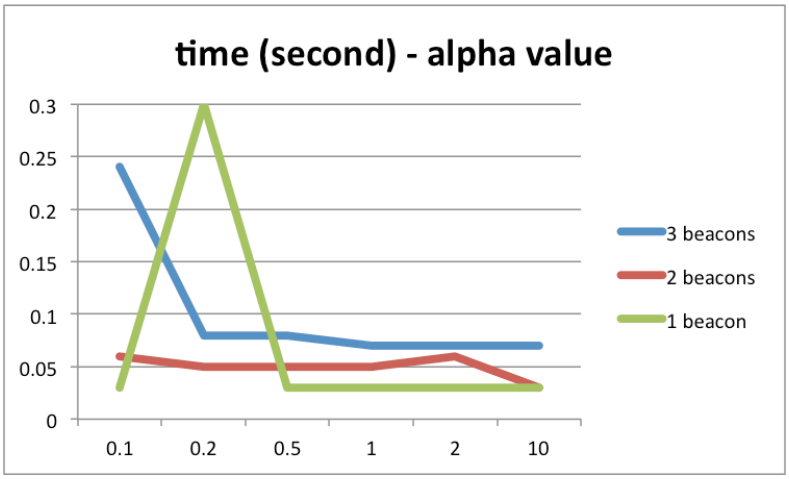


Figure 21 Correlation between time and alpha value in Bayesian Networks.

Figure 21 shows the correlation between the time to build the model and the value of alpha. The horizontal axis is the value of alpha in increasing order and the vertical axis is the time to build the model. The models of the 3 beacons dataset and the 1 beacon dataset take less time to produce when the value of alpha increases.

We used the same test dataset with 478 instances to test the model accuracy with different values of alpha. The results show that the accuracy of the model is 95.6597%. The maximum difference between correct localization rate for the training data and that of test data is not beyond 0.1%.

Due to the results above, we can draw the following conclusions.

- Both of the models of the 3 beacon and 2 beacon datasets have over 93% accuracy.
- The overall performance of the 1 beacon dataset model is not ideal and the accuracy

is less than 82%.

3.4 Discussion

Finally, we compare the accuracy of each of the models from the machine learning techniques to each other. The figures below show the localization accuracy of the models.

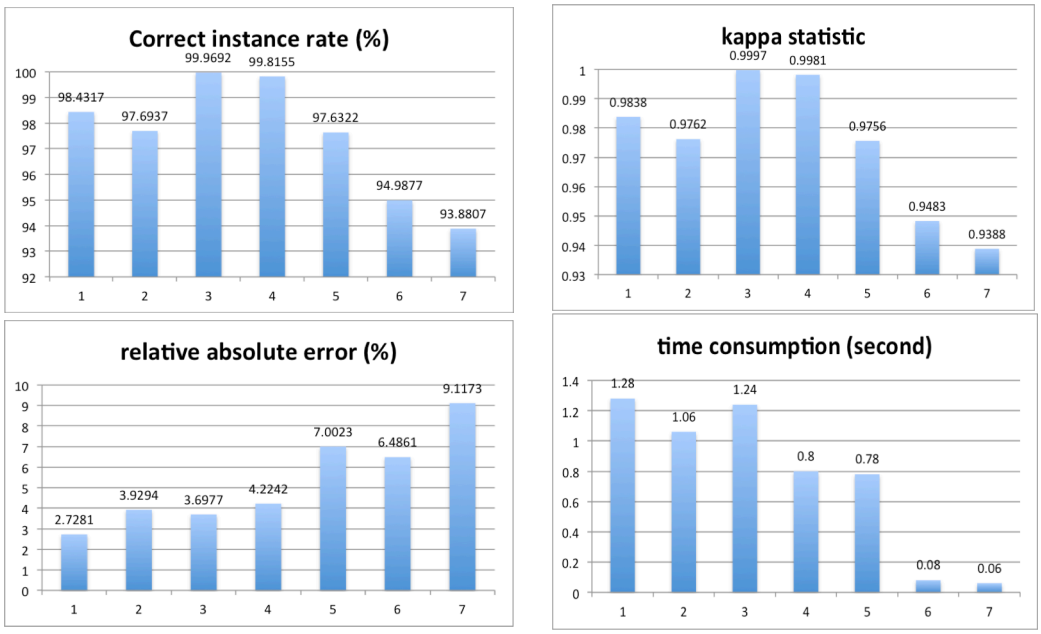


Figure 22 Performances of different machine learning algorithm.

The top left of Figure 22 compares the localization accuracy, the chart on the right top is the kappa statistic, the chart on the bottom left is relative absolute error, and the chart on the right bottom is time to build the models. The numbers on the horizontal axis from left to right indicate the model constructed using the 3 beacons dataset and the C4.5 algorithm, with confidence factor is 2 and number of folds equal to 4, the model with the 2 beacon dataset in C4.5 algorithm, of which confidence factor is 2 and number of folds is 5, the model with 3 beacons dataset in Random Forest algorithm, of which number of sub trees is 30, the model with 2 beacons dataset in Random Forest algorithm, of which number of sub trees is 30, the model with one beacons dataset in Random Forest algorithm, of which number of sub trees is 10, the model with 3 beacons dataset in Bayesian Networks algorithm, of which alpha is 0.2 and the model with two beacons

dataset in Bayesian Networks algorithm, of which alpha is 0.1, respectively.

The following conclusions can be drawn from the comparison:

- The overall accuracy and kappa statistic performance of the models built using Random Forest are best and Random Forest produces the highest real-world accuracy.
- C4.5 algorithm has the advantage of relative absolute error and it has the best classification on the training data but not the real-world test data.
- Bayesian Networks takes the least time to build the models and runs well in the larger datasets.
- In the case of limited beacons in a very large space to cover, Random Forest is the best choice because its model of one neighboring beacon provides high localization accuracy.
- Overall, having at least three neighboring beacons at each location is best.

4 CONCLUSION

In this paper, we first introduce some basic indoor positioning concepts and then focus on one method that relies on fingerprinting Bluetooth iBeacon signals using machine learning. In addition, we study the influence of beacon mounting positions on the received signal strength indicator and find it has some correlation between received signal strength and the mounting angle of the beacons. In further experiments with the machine learning algorithms, we found that both Random Forest and C4.5 provide high real-world localization accuracy. When fewer beacons are required to cover a very large space, the Random Forest is the best choice because its localization accuracy with a single beacon is still high.

REFERENCES

[1] Hightower J, Borriello G. Location systems for ubiquitous computing[J]. Computer, 2001, 34(8): 57-66.

[2] Rosenfeld A, Thurston M. Edge and curve detection for visual scene analysis[J]. Computers, IEEE Transactions on, 1971, 100(5): 562-569.

[3]Quate C F, Khuri-Yakub B T, Akamine S, et al. Near field acoustic ultrasonic microscope system and method: U.S. Patent 5,319,977[P]. 1994-6-14.

[4]Kaemarungsi K, Krishnamurthy P. Properties of indoor received signal strength for WLAN location fingerprinting[C]//Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on. IEEE, 2004: 14-23.

[5]Kemper J, Walter M, Linde H. Human-assisted calibration of an angulation based indoor location system[C]//Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on. IEEE, 2008: 196-201.

[6]Yang J, Chen Y. Indoor localization using improved rss-based lateration methods[C]//Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE. IEEE, 2009: 1-6.

[7] Xiong L. A selective model to suppress NLOS signals in angle-of-arrival (AOA) location estimation[C]//Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium on. IEEE, 1998, 1: 461-465.

[8] Peng R, Sichitiu M L. Angle of arrival localization for wireless sensor networks[C]//Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on. IEEE, 2006, 1: 374-382.

[9] Kułakowski P, Vales-Alonso J, Egea-López E, et al. Angle-of-arrival localization based on antenna arrays for wireless sensor networks[J]. Computers & Electrical Engineering, 2010, 36(6): 1181-1186.

[10] Chan Y T, Tsui W Y, So H C, et al. Time-of-arrival based localization under NLOS conditions[J]. Vehicular Technology, IEEE Transactions on, 2006, 55(1): 17-24.

[11] Qi Y, Kobayashi H, Suda H. On time-of-arrival positioning in a multipath environment[J]. Vehicular Technology, IEEE Transactions on, 2006, 55(5): 1516-1526.

[12] Cheung K W, So H C. A multidimensional scaling framework for mobile location using time-of-arrival measurements[J]. Signal Processing, IEEE Transactions on, 2005, 53(2): 460-470.

[13] Lanzisera S, Lin D T, Pister K S J. RF time of flight ranging for wireless sensor network localization[C]//Intelligent Solutions in Embedded Systems, 2006 International Workshop on. IEEE, 2006: 1-12.

[14] Bevilacqua A, Di Stefano L, Azzari P. People tracking using a time-of-flight depth sensor[C]//Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on. IEEE, 2006: 89-89.

[15] Medina C, Segura J C, De la Torre A. Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy[J]. Sensors, 2013, 13(3): 3501-3526.

[16] Fischer C, Muthukrishnan K, Hazas M, et al. Ultrasound-aided pedestrian dead reckoning for indoor navigation[C]//Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments. ACM, 2008: 31-36.

[17] Knauth S, Kaufmann L, Jost C, et al. The iloc ultrasound indoor localization system at the EvAAL 2011 competition[M]//Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking. Springer Berlin Heidelberg, 2012: 52-64.

[18] Mandal, Atri, et al. "Beep: 3D indoor positioning using audible sound."Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE. IEEE, 2005.

[19] Mautz, Rainer. Indoor positioning technologies. Diss. Habilitationsschrift ETH Zürich, 2012, 2012.

[20] Gu, Yanying, Anthony Lo, and Ignas Niemegeers. "A survey of indoor positioning systems for wireless personal networks." Communications Surveys & Tutorials, IEEE 11.1 (2009): 13-32.

[21] Evennou, Frédéric, and François Marx. "Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning." Eurasip journal on applied signal processing 2006 (2006): 164-164.

[22] Zandbergen, Paul A. "Accuracy of iPhone locations: A comparison of assisted GPS,

WiFi and cellular positioning." Transactions in GIS 13.s1 (2009): 5-25.
[23] Bose, Atreyi, and Chuan Heng Foh. "A practical path loss model for indoor WiFi positioning enhancement." Information, Communications & Signal Processing, 2007 6th International Conference on. IEEE, 2007.
[24] Tayebi A, Gomez J, Saez de Adana F M, et al. The application of ray-tracing to mobile localization using the direction of arrival and received signal strength in multipath indoor environments[J]. Progress In Electromagnetics Research, 2009, 91: 1-15.
[25] Vanheel F, Verhaevert J, Laermans E, et al. Automated linear regression tools improve rssi wsn localization in multipath indoor environment[J]. EURASIP Journal on Wireless Communications and Networking, 2011, 2011(1): 1-27.
[26] Obayashi S, Zander J. A body-shadowing model for indoor radio communication environments[J]. Antennas and Propagation, IEEE Transactions on, 1998, 46(6): 920-927.
[27] Kara A, Bertoni H L. Effect of people moving near short-range indoor propagation links at 2.45 GHz[J]. Communications and Networks, Journal of, 2006, 8(3): 286-289.
[28] Liberti J C, Rappaport T S. Statistics of shadowing in indoor radio channels at 900 and 1900 MHz[C]//Military Communications Conference, 1992. MILCOM'92, Conference Record. Communications-Fusing Command, Control and Intelligence., IEEE. IEEE, 1992: 1066-1070.
[29] Quinlan J R. C4. 5: programs for machine learning[M]. Elsevier, 2014.
[30] Quinlan J R. Bagging, boosting, and C4. 5[C]//AAAI/IAAI, Vol. 1. 1996: 725-730.
[31] Quinlan J R. Improved use of continuous attributes in C4. 5[J]. Journal of artificial intelligence research, 1996: 77-90.
[32] Ruggieri S. Efficient C4. 5 [classification algorithm][J]. Knowledge and Data Engineering, IEEE Transactions on, 2002, 14(2): 438-444.
[33] Liaw A, Wiener M. Classification and regression by randomForest[J]. R news, 2002, 2(3): 18-22.
[34] Díaz-Uriarte R, De Andres S A. Gene selection and classification of microarray data using random forest[J]. BMC bioinformatics, 2006, 7(1): 3.
[35] Breiman L. Random forests[J]. Machine learning, 2001, 45(1): 5-32.
[36] Friedman N, Geiger D, Goldszmidt M. Bayesian Networks classifiers[J]. Machine

learning, 1997, 29(2-3): 131-163.

[37] Cheng J, Greiner R. Comparing Bayesian Networks classifiers[C]//Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1999: 101-108.

[38] Jensen F V. An introduction to Bayesian Networks[M]. London: UCL press, 1996.

[39] Drazin S, Montag M. Decision tree analysis using WEKA[J]. Machine Learning-Project II, University of Miami, 2012: 1-3.

[40] Carletta J. Assessing agreement on classification tasks: the kappa statistic[J]. Computational linguistics, 1996, 22(2): 249-254.

[41] Bouckaert R R. Bayesian Networks classifiers in weka[M]. Department of Computer Science, University of Waikato, 2004.

[42] Bouckaert R R. Bayesian Networks classifiers in Weka for version 3-5-7[J]. Artificial Intelligence Tools, 2008, 11(3): 369-387.

[43] Hall M, Frank E, Holmes G, et al. The WEKA data mining software: an update[J]. ACM SIGKDD explorations newsletter, 2009, 11(1): 10-18.

[44] Developer A. Getting Started with iBeacon[J]. 2014.

[45] David Heckerman. March 1995. A Tutorial on Learning With Bayesian Networks.

[46] LLAMADIGITAL (2014 August 9th) Mounting Estimote beacons on brackets. Retrieved from http://www.thetalkingllama.com/2014/08/mounting-estimote-beacons-on-brackets/