

SECURE GEOLOCATION FOR WIRELESS INDOOR NETWORKS

A Thesis
Presented to
The Academic Faculty

By

Yu-Xi Lim

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Electrical and Computer Engineering



School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2006

Copyright © 2006 by Yu-Xi Lim

SECURE GEOLOCATION FOR WIRELESS INDOOR NETWORKS

Approved by:

Dr. Henry Owen, Committee Chair
Professor, School of ECE
Georgia Institute of Technology

Dr. John Copeland
Professor, School of ECE
Georgia Institute of Technology

Dr. Randal T. Abler
Professor, School of ECE
Georgia Institute of Technology

Date Approved: April 7, 2006

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 ORIGIN AND HISTORY OF THE PROBLEM . .	2
2.1 IEEE 802.11	2
2.2 Location information and services	2
2.3 Current location systems	3
2.3.1 Ad hoc systems	4
2.3.2 Infrastructure systems	5
2.3.3 Radio map technique	5
2.4 Secure location services	8
2.5 Requirements of a secure location service	8
2.5.1 Accuracy	9
2.5.2 Integrity	10
2.5.3 Availability	10
2.5.4 Secrecy	10
2.6 Applications of a secure location service	10
2.6.1 Authentication	10
2.6.2 Billing	11
2.6.3 Intrusion response	12
2.6.4 Ubiquitous applications	12
CHAPTER 3 RESEARCH OBJECTIVES AND CHALLENGES .	13
3.1 Secure architecture	13
3.2 Functional differences	14
3.3 Challenges for secure architecture	15
3.3.1 Scalability	15
3.3.2 Signal Variations	16
3.4 Threat model	17
3.4.1 Physical access	18
3.4.2 Spoofing	18
3.4.3 Flooding	18
3.5 Algorithms	19
3.5.1 Location estimation	20
3.5.2 Support Vectors	23
3.6 Wireless network parameters	27

CHAPTER 4	IMPLEMENTATION	29
4.1	Implementation overview	29
4.2	Server	32
4.3	Sensors	32
4.4	Data collection	34
4.4.1	Site survey	34
4.4.2	Normalization	35
4.4.3	Round-trip time	36
4.5	Algorithms	37
4.5.1	Algorithm framework	37
4.5.2	Identification of mobile nodes	37
4.5.3	SVR training	37
CHAPTER 5	RESULTS AND DISCUSSION	39
5.1	Interpreting results	39
5.2	Signal Map	40
5.3	Hardware variations	40
5.4	Signal and noise levels	41
5.5	Environmental probes	42
5.6	Round-trip time	43
5.7	Comparison	43
CHAPTER 6	CONCLUSION	58
APPENDIX A	SERVER GUI CONSOLE	61
APPENDIX B	MAP OF SENSOR LOCATIONS	66
APPENDIX C	MAP OF LOCATIONS FOR SITE SURVEY	68
List of Terms		70
REFERENCES		74

LIST OF TABLES

Table 1	Kernel types and respective parameters.	25
Table 2	List of features to be investigated	28
Table 3	Summary of optimal SVR parameters and results.	44

LIST OF FIGURES

Figure 1	Typical architecture of location systems for IEEE 802.11.	6
Figure 2	Proposed system architecture.	14
Figure 3	KNN in two dimensions.	21
Figure 4	Simple regression for improved resolution.	22
Figure 5	SVM classification in two dimensions.	24
Figure 6	SVM classification with slack variables for classification errors.	26
Figure 7	Training data flow.	30
Figure 8	Online data flow.	31
Figure 9	Probe frame layout.	33
Figure 10	SVR predictions performing as classifiers.	39
Figure 11	Signal map for sensor in room 331 (marked by red “X”) late in evening.	46
Figure 12	Correlation between signal strengths of different wireless NICs.	47
Figure 13	Plot of cross-validation rate against training parameters C , γ , and p for x-coordinate for ε -SVR.	47
Figure 14	Plot of cross-validation rate against training parameters C , γ , and p for y-coordinate for ε -SVR.	48
Figure 15	Plot of cross-validation rate against training parameters C , γ , and p for x-coordinate for ν -SVR.	49
Figure 16	Plot of cross-validation rate against training parameters C , γ , and p for y-coordinate for ν -SVR.	49
Figure 17	Plot of predicted versus actual locations for ε -SVR for a) x-coordinate and b) y-coordinate.	50
Figure 18	Plot of estimated and training points.	51
Figure 19	Plot of cross-validation rate against training parameters C , γ , and p for x-coordinate with environmental probes using ε -SVR.	52
Figure 20	Plot of cross-validation rate against training parameters C , γ , and p for y-coordinate with environmental probes using ε -SVR.	53

Figure 21	Plot of cross-validation rate against training parameters C , γ , and ν for x-coordinate with environmental probes using ν -SVR.	54
Figure 22	Plot of cross-validation rate against training parameters C , γ , and ν for y-coordinate with environmental probes using ν -SVR.	55
Figure 23	Plot of predicted versus actual locations with environmental probes for a) x-coordinate and b) y-coordinate.	56
Figure 24	Plot of cross-validation rate against training parameters C , γ , and p for x-coordinate with RTT using ε -SVR.	57
Figure 25	Plot of cross-validation rate against training parameters C , γ , and p for y-coordinate with RTT using ε -SVR.	57
Figure 26	Sensor list.	62
Figure 27	Sensor filters.	62
Figure 28	SVR training settings.	63
Figure 29	Map visualization for survey, training, and testing data.	63
Figure 30	Sensor monitoring settings.	64
Figure 31	Raw sensor data table.	64
Figure 32	Correlated sensor data table.	65
Figure 33	Location of sensors on third floor of the College of Computing building.	67
Figure 34	Location of survey sample points on third floor of the College of Computing building.	69

CHAPTER 1

INTRODUCTION

The objective of the research is to develop an accurate system for indoor location estimation using a secure architecture based on the IEEE 802.11 standard for infrastructure networks. Elements of this secure architecture include: server-oriented platform for greater trust and manageability; multiple wireless network parameters for improved accuracy; and Support Vector Regression (SVR) for accurate, high-resolution estimates. While these elements have been investigated individually in earlier research, none has combined them to a single security-oriented system. Thus this research investigates the feasibility of using these elements together.

CHAPTER 2

ORIGIN AND HISTORY OF THE PROBLEM

2.1 IEEE 802.11

What is commonly known as IEEE 802.11 actually refers to the family of standards that include the original IEEE 802.11 itself, IEEE 802.11a, IEEE 802.11b, and IEEE 802.11g. Other common names include Wi-Fi and Wireless Local Area Network (WLAN).

IEEE 802.11 has become virtually ubiquitous as a wireless computer networking standard and is poised to enter even the mobile voice communications sector. With such significant market penetration, IEEE 802.11 will remain a leader even in the face of the existing competing and emerging standards.

For the purposes of this research, it is important to note an important dichotomy in IEEE 802.11 networks: infrastructure networks and ad hoc networks. IEEE 802.11 specifies these two network architectures[1], each with its own benefits and disadvantages[2]. Briefly, infrastructure networks rely on a fixed network infrastructure to facilitate communications between the mobile nodes (known as stations in the standard) and ad hoc networks have the mobile nodes communicate directly with each other. Infrastructure networks are more costly to set up, but offer generally better performance and manageability. They are vastly more popular than ad hoc networks.

2.2 Location information and services

In wired networks, the network address is usually strongly correlated to the physical location of the node. However, with wireless networks, the nodes tend to be highly mobile and this is no longer the case. In spite of this, it is often useful, and sometimes even necessary, to ground the network location in the physical world[3].

Location information is useful, and in both fixed and mobile networks, location

information is often used for management purposes. However, in a mobile context, the value of location information increases significantly as it provides novel functionality over previously fixed wired networks[4]. Many location-aware applications have already been proposed[5, 6] or even commercially deployed[7, 8]. Further applications are discussed in Section 2.6.

The IEEE 802.11 standard makes no provisions for location information[1]. This deficiency has to be addressed using additional systems to extract location information and present them to the user.

Location information needs to be placed within some frame of reference. For example, Global Positioning System (GPS) units generally operate using a terrestrial frame of reference, using the Greenwich meridian and the equator. Other systems with smaller coverage areas choose a more appropriate reference frame, such as the south-west corner of the region or even with respect to other users.

Location information is complex, not merely a set of Cartesian coordinates[9]. A location service collects and stores location information and provides access to such information. Other applications providing value-added services and information may then obtain the location information from the location service. The location service may even be integrated with other systems providing related data to form a Geographic Information System (GIS).

2.3 Current location systems

There exists a variety of systems that provide location information for outdoor, network and non-network use. Probably the most well known non-network system is the GPS[10]. Among the network-oriented systems, there is the well-publicized Enhanced 911 (E911) wireless services established by the Federal Communications Commission (FCC)[11]¹. However, these outdoor systems do not face the same unique technical

¹There are concerns for using the E911 system for Voice-over-IP (VoIP) phones, even for fixed lines. A location service for wireless networks can help.

challenges of similar indoor systems.

Indoor location systems face a somewhat harsher environment which results in complex radio propagation patterns in all but the simplest of set ups. These propagation patterns are generally site-specific and are characterized by poor Line of Sight (LOS) and severe multi-path conditions[4]. These greatly limit the use of simple triangulation algorithms that form the core of other systems like GPS.

Location systems use fixed reference points at known locations from which to determine the location of the user. If there is no existing wireless network infrastructure, most deployments use specialized hardware in the form of beacons or tags operating on RF or infrared. These may use triangulation if several beacons or sensors are deployed in a single small area. Alternatively, the systems may use one or more beacons or sensors to detect presence in a given area, rather than provide precise coordinates. For environments with an existing wireless network, it is generally more cost-effective to utilize the equipment that has already been deployed, be it the mobile nodes themselves or other network infrastructure. The use of mobile nodes or fixed infrastructure usually depends on whether the network is operating in ad hoc or infrastructure mode.

2.3.1 Ad hoc systems

Ad hoc wireless networks may be deployed indoors or outdoors. The outdoor scenario (e.g. for emergency services and military networks) is beyond the scope of this research. These focus mostly on sensor networks and provide coordinates using some form of triangulation using the Received Signal Strength (RSS)[12], Time of Arrival (TOA)[13, 14], and Angle of Arrival (AOA)[15] measurements, or provide relative location estimation based on connectivity with neighboring nodes[16]. Indoor scenarios are affected significantly by the presence of reflectors and attenuators[14]. Otherwise, little research has been done on location estimation for ad hoc networks in the more complex indoor environments.

2.3.2 Infrastructure systems

IEEE 802.11 infrastructure networks do not afford the same flexibility as ad hoc networks with regard to deployment and are generally in-building systems. Unlike ad hoc networks, infrastructure networks are at least partially fixed, and often the fixed access points are used as reference points for location estimation.

There are two possible approaches for location estimation when using the fixed infrastructure as reference points. Most commercial systems utilize the simpler method, which is to provide a very approximate guess based on the sensor or access point which receives the strongest signal or has connectivity[17, 18]. The mobile node is then assumed to be in the vicinity of that particular access point or sensor. This method has poor resolution and poor accuracy. It is able to resolve only as many zones as there are sensors. Its accuracy is questionable because it assumes that a strong signal indicates closer physical proximity, which is not always the case in an indoor environment.

The more complex method is using a radio map. This technique factors measurements from multiple sensors or access points and is discussed in greater detail in Section 2.3.3. It is able to provide greater resolution and accuracy than the naïve method described above. There are some commercial systems implementing this method[8, 19] and many research systems[20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34].

Most systems implementing the radio map technique use a user-centric architecture shown in Figure 1. The user’s device performs the measurements, and often the calculations, necessary to determine the user’s position. This may then be transmitted to a server for storage or tracking.

2.3.3 Radio map technique

The radio map technique is founded on the premise that each location can be uniquely identified by a radio “fingerprint” measured at the sensors. Most systems use the RSS measured at each sensor to form a tuple that serves as the fingerprint. Other possible metrics are connection “quality” (usually some measure of packet loss), and frame

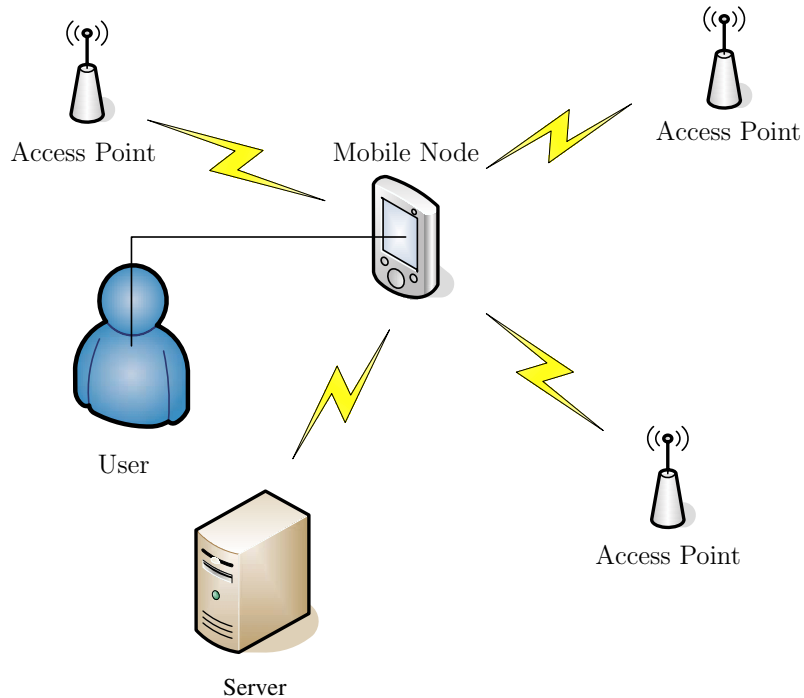


Figure 1. Typical architecture of location systems for IEEE 802.11.

Round Trip Time (RTT)[35].

Even in the common case of RSS, the measured values do not vary linearly with location. Obviously, in an indoor environment, attenuation is a significant factor. The complex indoor environment also presents a host of reflectors, scatterers, and diffractors, making multi-path effects such as phase cancellation and delay spread a major concern.

Unfortunately, most current IEEE 802.11 hardware are only able to measure such effects indirectly using the RSS or vague qualitative measures such as signal quality (hence the predominance of applications using the RSS). In the future, as the radio hardware becomes more complex, such as in IEEE 802.11n (Wi-Fi Multiple In/Multiple Out (MIMO)), it may be possible to obtain more accurate measures of these phenomena.

The wavelengths at which IEEE 802.11 operates (approximately 12.5 cm at 2.4 GHz and 5.5 cm at 5 GHz) may result periodic fading (due to phase cancellation) with

frequencies on a similar physical scale. Under ideal conditions, it may be possible to exploit this periodic fading to pinpoint the location of the mobile node[27].

The radio map technique usually utilizes empirical measurements obtained through a site survey[20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]. However, the values may also be computed using a mathematical model of the radio environment, a technique known as radio propagation modeling[32, 33, 34]. Both methods have been used in various other implementations to varying degrees of success, however the empirical method is more popular. This is because it is difficult to accurately model the environment, with all details such as building materials and furnishings, and this is made worse by dynamic influences such as doors, electrical appliances, and people.

Given the non-linear variations in the RSS, the small signal variations, and numerous other factors which may affect each measurement, it is nearly impossible to simply match results from a previously obtained radio map against current measurements to determine a user's present location. A certain amount of tolerance to errors in measurement is required. Furthermore, there is a significant limit to the granularity of the map obtained through the empirical method and the values at points between those actually surveyed are not easy to predict. All these mean that matching a new set of measurements to the map has to be done "intelligently."

Various algorithms have been used to do the matching, from reasonably straightforward approaches like k -Nearest Neighbor (KNN) to more complex statistical methods like Hidden Markov Model (HMM) and other techniques that involve machine learning. Similarly, improving the granularity of the map is usually done by interpolation or time averaging[36, 37].

2.4 Secure location services

The primary goal of this research is to develop a location estimation system suitable for security-related applications and other applications that require trusted and accurate location data. Examples of such applications are location-based authentication[3, 38, 39] and location-based billing.

2.5 Requirements of a secure location service

[40] lays out a set of requirements for location-aware applications. These are:

- Scalability
- Accuracy
- User privacy
- Flexibility
- Fault tolerance
- Resource requirements
- Response time

These requirements are broad and not intended specifically for IEEE 802.11-based systems. Nevertheless, these are still useful in constructing a new architecture and determining its feasibility. The attributes tend to be closely related and, generally, providing for one would also support the other attributes.

The four main aspects that this proposed research specifically focuses on are:

Accuracy The deviation of the estimates from the true location must be minimal and the resolution of the system must be sufficiently fine-grained.

Integrity The data (both radio measurements and location information) must be protected from tampering, or if tampering occurs, it must be evident.

Availability The location information must be available when it is needed and as often as the need arises.

Secrecy The data can only be accessed by authorized users.

These can be viewed as a simplification of [40] and encompass the same areas. Current research on location estimation in IEEE 802.11 networks focus on the first aspect, accuracy, and overlook the security aspect entirely.

2.5.1 Accuracy

The requirement for accuracy is not difficult to understand. The general definition of accuracy actually quantifies two properties: the Euclidean distance between the actual and estimated physical locations (the error distance), and the probability of each error distance. Often, for ease of comparison, these two attributes are condensed to a single value by taking the Root-Mean-Square (RMS) at the expense of some details[41]. For a majority of applications, this measure of accuracy is the only reasonable one.

There are still many situations where accuracy is only a concern in terms of the ability to discern between the various zones, i.e. the resolution of the system. Zones are physical areas in which different policies (i.e. security or billing) are enforced. They may be as small as one room in a building or as large as a metropolitan area. For the purposes of most indoor networks, it is assumed that the smallest zone would be a room. However, the system developed will operate on continuous coordinates with respect to some origin. The ability of the system to differentiate between locations in different rooms will be derived from these coordinates.

The accuracy of various approaches found in literature may vary significantly with the experimental parameters, thus it is not possible to compare published results from different research groups. Given the constraints of the proposed research, only the approach using SVR discussed in Section 3.5.2 will be used. However, the research will investigate the use of various parameters and adaptive techniques to improve the

accuracy of the estimates.

2.5.2 Integrity

For integrity, the entire flow of information from the raw measurements to the final processed location must be tamper-proof, or at least any tampering must be detectable. Common means of ensuring integrity include data encryption and signing.

2.5.3 Availability

The location information also needs to be available as and when the need arises. If the user requires a location-based service at a certain time, availability of the location information would usually entail data collection and processing before and up to that time. Availability is usually addressed through the use of managed redundancy. Certain applications, like health-care or prison and reform, may required continuous tracking of the mobile nodes and not merely intermittent location information. In such cases, the location system must be able to deliver continuous availability with short response times.

2.5.4 Secrecy

Confidentiality is a secondary concern in location systems. In many deployments, location can be easily observed by third parties, therefore it is less imperative to protect the confidentiality of the location information or the raw data before processing. However, standard procedures can and should still be observed to protect the data, such as ensuring the physical security of the systems, using proper access controls, and using secure tunnels to encrypt data in transit.

2.6 Applications of a secure location service

2.6.1 Authentication

Since the coverage of a wireless access point is usually an ill-defined area that often extends beyond the physical boundaries of the region being serviced, it is often useful

to deny access to users beyond the defined physical boundaries. Thus, instead of relying on the natural attenuation of the signal to limit access, administrators can take an active approach by defining precise physical regions in which access is allowed.

Even within a building, it may sometimes be necessary to further partition the coverage regions. Access by visitors in a public lobby can be limited, while authorized users within the offices may be granted more permissions.

Location information should be seen as an additional aspect by which to provide authentication. Unlike passwords or security tokens, location is an aspect of “what you are” and cannot be faked or stolen. However, like other systems that use complex data to authenticate, such as biometrics, there is some uncertainty and a potential for error, so it should not be the only means for authentication.

Unlike other authentication methods, location-based authentication can be done automatically and transparently[42, 3]. This means it can be done continuously and would be an additional guarantee against session hijacking attacks.

2.6.2 Billing

Mobile communications providers currently advertise service plans that charge different rates depending on when and where the connection is initiated. Time-based billing is currently done at a very fine granularity. However, with regards to where the connection was initiated, service providers usually differentiate at a coarse granularity between “local” and “long-distance”. With a location system in place, it may be possible to implement finer-grained billing schemes, should service providers see the need for this in future.

However, without a secure location system, it would be possible for a customer to circumvent the imposed zones by modifying the location information that the service provider receives. Hence, such a method of billing would also require a secure location system.

2.6.3 Intrusion response

Unlike attackers on a wired local network, attackers compromising a wireless network cannot easily be traced to a specific physical location from which they connect from. By using a secure wireless location service, it would be possible to pinpoint the attacker and take the necessary steps to stop the attacks, such as physical detention or arrest.

Intrusion detection and response would be impossible if the location service was utilizing the conventional insecure architecture since it would require that the attacker inform the system of his or her location based on the radio measurements that the attacker makes. Requiring such cooperation[20] from the attacker is absurd.

Conversely, location information can also be used as evidence for the innocence of users in cases where their accounts have been compromised[42, 3].

2.6.4 Ubiquitous applications

In ubiquitous computing applications, the user may be required to authenticate repeatedly with the numerous services he or she may encounter. A location-based authentication system would improve usability by making such authentications transparent[39].

There are other uses for location information in ubiquitous applications, mostly in the form of user tracking to provide a personalized or optimized service or to facilitate interactions between users in close physical proximity.

CHAPTER 3

RESEARCH OBJECTIVES AND CHALLENGES

Current systems, though varied, do not meet the needs of applications requiring a secure location service. This research addresses this deficiency through the use of:

Secure architecture A different architecture is proposed and will address the issues of integrity, availability and secrecy.

More varied wireless network parameters A larger range of network parameters will be utilized to improve the accuracy and integrity of the estimates.

Improved algorithm A more advanced machine learning algorithm will utilize the additional parameters to produce more accurate estimates. Resolution will also be improved by use of regression instead of simple classification.

3.1 Secure architecture

Architectures for location information systems have been proposed in [40, 9]. However, these were designed for large-scale ubiquitous computing applications which employ multiple technologies such as Radio-Frequency IDentification (RFID) badges, infrared sensors, and pressure-sensitive mats.

For this research, the system architecture is a simplification of these large-scale architectures (Figure 2). It is based on the wireless geolocation system architecture described in [4] and is similar to the conceptual system design presented in [43]. Unlike [43], this architecture was deliberately chosen for its security possibilities.

The architecture itself does not depend on the network infrastructure, though the utilization of existing infrastructure is an added advantage for deployment. As such, it is easy to adapt the system for both infrastructure and ad hoc networks.

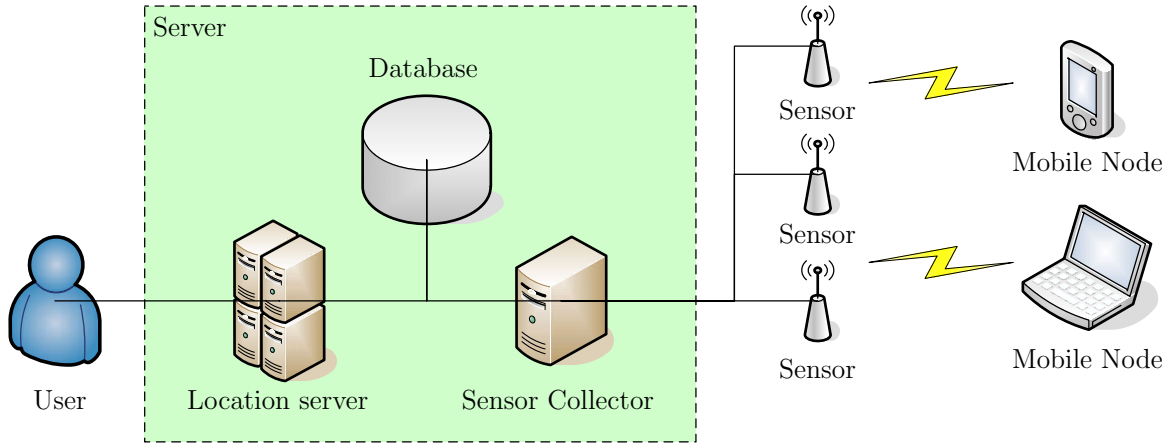


Figure 2. Proposed system architecture.

3.2 Functional differences

The architecture differs from the majority of IEEE 802.11 location estimation systems using the radio map technique in two key respects:

Measurements at sensors The measurements of the radio environment take place at fixed sensors rather than by the mobile nodes themselves.

Algorithms at location server The location estimation algorithms run at the location server than at the mobile nodes. This approach is less common than the previous.

By eliminating the mobile nodes from the data path of the system, the location information can be more easily secured. Mobile nodes are in the hands of users and cannot be trusted as they are easily tampered, but the fixed infrastructure in the proposed architecture can be physically protected against such tampering. Using measurements obtained from the mobile node for security applications, such as in [20], is unacceptable.

The data can be further protected from tampering by using a secure (encrypted and authenticated) tunnel during transmission and physically securing the sensors. These details are deployment decisions and entirely optional.

Relying on the fixed infrastructure to execute the location estimation algorithms means the algorithms are no longer constrained by the generally poor performance of the mobile nodes. The mobile nodes are typically battery-powered devices designed for operational longevity rather than raw computational power. This system also eliminates the possibility that location information would be unavailable because the mobile node went offline before it could transmit its results.

This research was limited to the use of Commercial off-the-Shelf (COTS) hardware for IEEE 802.11b networks, utilizing existing infrastructure where possible. Nevertheless, it should be possible to extend the applicability of the results to other similar networking standards.

Part of the research includes the implementation of the secure architecture (Section 3.1) to evaluate its feasibility and as a platform for the next part of the proposed research—the investigation of suitable algorithms (in particular Support Vector Regression[25, 44]) for secure location estimation.

3.3 Challenges for secure architecture

3.3.1 Scalability

The proposed secure architecture does not scale in the same way as a conventional architecture. In a conventional architecture, each mobile node performs the measurements and calculations necessary to determine its position. Thus, adding additional mobile nodes places no additional load on the original system. However, with the secure architecture, the measurements and calculations are performed centrally. This potentially leads to problems of scalability as more nodes are added to the system.

One way to overcome this issue is by clustering the central server. Clustering will also simultaneously improve the availability of the system.

Clustering is practical in this architecture because each mobile node can be tracked independently of the others. Thus clustering can be done on a per-mobile node basis. Another alternative for physically expansive deployments is to divide the geographic

region being covered into overlapping sectors, taking into account the fact that signals from distant mobile nodes will be detectable at only a fraction of the available sensors. This process is similar to what was described in [45] but the term “clustering” was used differently.

Other than scaling with the number of mobile nodes being tracked, the system also has to scale with the area being monitored. The conventional architecture has each mobile node monitoring its own small region. The secure architecture would require the number of sensors deployed to vary with the size of the area being monitored. This is likely to be a linear scaling.

Scalability and fault tolerance are difficult to verify experimentally in a small set up that is the scope of this research. However, with modest resource requirements, most location systems should exhibit linear scaling of processing and storage requirements with the number of nodes being tracked and the number of requests—hardly unreasonable.¹ The secure architecture will also exhibit this characteristic.

3.3.2 Signal Variations

Various wireless network interfaces exhibit different transmission powers. These may range from 15 dBi to 25 dBi. The system must compensate for this large variance in transmission power when using the RSS for location estimation.

Currently, values are normalized to eliminate the effects of transmission power variances. Preliminary tests indicate that after normalization, the system is reasonably tolerance to hardware variations. Additional testing is needed to determine the effectiveness of normalization over the entire range of transmission powers for consumer hardware.

The orientation of the mobile devices have been known to affect the RSS[46]. Two

¹An example where resource requirements increase faster than linearly are when the algorithms rely on the positions of the neighboring nodes, and the number of neighboring nodes increase linearly with the total number of nodes. Such a system would have $O[N^2]$ complexity but rarely occurs in practice.

factors contribute to this observation: the directionality of the antenna used, and the presence of the user's body.

Most mobile devices utilize either a plug-in wireless network card or a built-in wireless interface. In the case of a plug-in card, these typically use a microstrip patch antenna integrated directly on the PCB. These antennae are directional in nature. Built-in systems may use an antenna which resembles a dipole, usually in the form of a wire running around the edge of the screen. Though these dipole antennae may be omnidirectional in the horizontal plane (the vertical axis is usually a lesser concern), the implementation in mobile devices tends to exhibit directionality due to the shielding effects of the device's case and other components in the device. In larger and newer devices, built-in antennae may resemble patch antennae instead of dipoles.

The human body, though not a perfect occluder, does pose as a measurable attenuator to RF signals. In a typical set up, the user's body would attenuate signals in a small but non-negligible sector.

Another concern is the use of antenna diversity. Fortunately, most current IEEE 802.11 mobile systems currently utilize only one antenna. If, in future, multi-antenna configurations become more popular, the performance of the current system may suffer. The use of antenna diversity may result in sudden changes in the RSS as the antenna is switched. However, the algorithms can be adapted to make use of antenna diversity to improve measurements.

3.4 Threat model

Various attacks are possible on a IEEE 802.11 location system[47, 48]. The secure architecture proposed above should be able to address most of these threats if properly implemented. The possible threats are discussed below.

3.4.1 Physical access

A physical compromise of the system may result in the location data being modified. Fortunately, the physical security of the proposed architecture is enhanced with respect to the conventional user-centric architecture. The only data from the user is the radio signal itself (threats concerning the radio signal are discussed in the following sections) so the entire data path can be easily secured against physical access.

3.4.2 Spoofing

Given the numerous factors which affect the measured radio signal and the relatively small measurement space of each signal characteristic, there is a possibility that signals from one location may be misinterpreted as being from another location. This has been called signal aliasing[46].

This may occur unintentionally or, more importantly, intentionally as a result of a conscious attempt to manipulate the signal and to create a fake presence (spoofing)[29]. A system relying on only one metric for measuring the signal would be easily vulnerable to signal aliasing. However, it is suggested that as the number of measured parameters increase, the likelihood of aliasing decreases; from an attacker’s perspective, the complexity of simulating a signal increases. With a good selection of parameters, an attempt at faking a location would require nothing short of gaining physical access to the sensors themselves and directly generating a signal at each one.

3.4.3 Flooding

Another less subtle way of affecting the radio signal is through flooding. Flooding attacks are Denial of Service (DoS) attacks that result from the system being overloaded with large amounts on “nonsense” data that it is unable to process legitimate data and requests.

There are two primary methods of attacking a IEEE 802.11 network using flooding attacks. The first is by jamming the radio signals using a high-powered transmitter

operating on the same frequency. The second is by exploiting the unauthenticated management frames in the IEEE 802.11 standard in what is known as a deauthentication attack[49].

Unfortunately, all current IEEE 802.11 networks are susceptible to jamming attacks and this means that a location system based on such technology will also be vulnerable. However, this form of attack is not commonly used and the IEEE 802.11 standards committee has not made any plans to adopt physical standards that would be resistant to such attacks.

The proposed architecture is not susceptible to the deauthentication attack (and related variants such as Request-To-Send (RTS)/Clear-To-Send (CTS) flooding) since it is passively listening to signals and not processing them using the full IEEE 802.11 standard. Being invulnerable to the attack means the system can actually be used to localize the source of the attack.

3.5 Algorithms

Much current research in IEEE 802.11 location systems focus on improving the accuracy of the system through the use of complex algorithms. However, as observed earlier, these systems utilize an architecture in which the measurements are done by the mobile node itself. This research focuses on a secure server-centric architecture where measurements are done using distributed sensors and algorithms executed at a centralized server. Furthermore, little research has been done on the efficiency of these algorithms on the secure architecture.

There are several requirements for the algorithms in this architecture:

Accuracy The algorithms must be able to provide satisfactory resolution and good accuracy.

Robustness The algorithms should be able to function even in the typically dynamic conditions of real-world wireless networks.

Additionally, for the purpose of this research, the algorithms should rely on minimal hardware specifications. This would allow them to be extended to other wireless networking technologies.

Even though the system is designed to test various algorithms, this research will focus only on one algorithm, leaving the other possibilities as avenues for further research.

3.5.1 Location estimation

As mentioned in Section 2.3.3, the radio map technique used here creates a map of the environment, with each location having a radio fingerprint—a unique set of measurements such as signal strength, RTT, etc (discussed in greater detail in Section 3.6). When presented with a radio fingerprint, the system should be able to determine the location from that.

However, as was also mentioned, the matching is not straightforward since there are variations in the fingerprints for any given location due to a variety of factors. Furthermore, the radio map is usually obtained through empirical measurements, thus limiting the practical resolution of the map.

With regards to the first problem, where the measured fingerprint does not exactly match any previously measured fingerprint, common techniques using algorithms such as KNN, HMM, or Support Vector Machine (SVM) try to obtain the closest matching fingerprint by attempting to minimize the error. In effect, each measurement that comprises the fingerprint represents a dimension (or a feature).

KNN algorithm that finds an arbitrary k nearest neighbors to a given point using the Gaussian distance and determines what class (the location) the majority of these points belong to. In Figure 3, the four nearest neighbors to the red point are found, and it is determined that majority class is black, though the absolute nearest neighbor is actually gray. Unfortunately, nearest neighbor searches scale exponentially with each additional dimension. Furthermore, KNN weights each dimension equally when

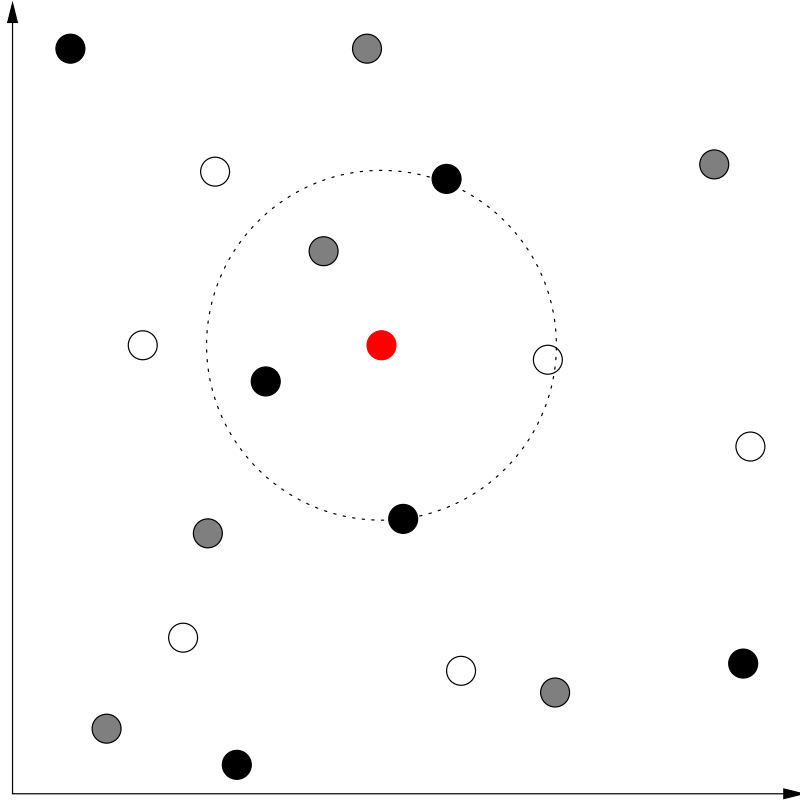


Figure 3. KNN in two dimensions.

calculating the Gaussian distance. Thus a proper solution requires a more complex algorithm.

The second problem is the limited resolution of the map due to the practical limitations of obtaining samples. A possible solution would be to perform a regression on the data to obtain a curve (or rather, a hyperplane, since there are more than two dimensions) that represents the variations of the radio map in relation to spatial coordinates. Figure 5 shows an example of how regression would help with “filling in” the missing parts of the map. Also note the many-to-one mapping of signal strength to location, which occurs often in empirical measurements. Differentiating each location would thus require the consideration of other measurements, such as signal strengths from other sensors, or the RTT, etc—in other words, the regression and subsequent evaluation has to occur in the full multi-dimensional space. In the next section, we examine support vectors, which were used for this regression calculation.

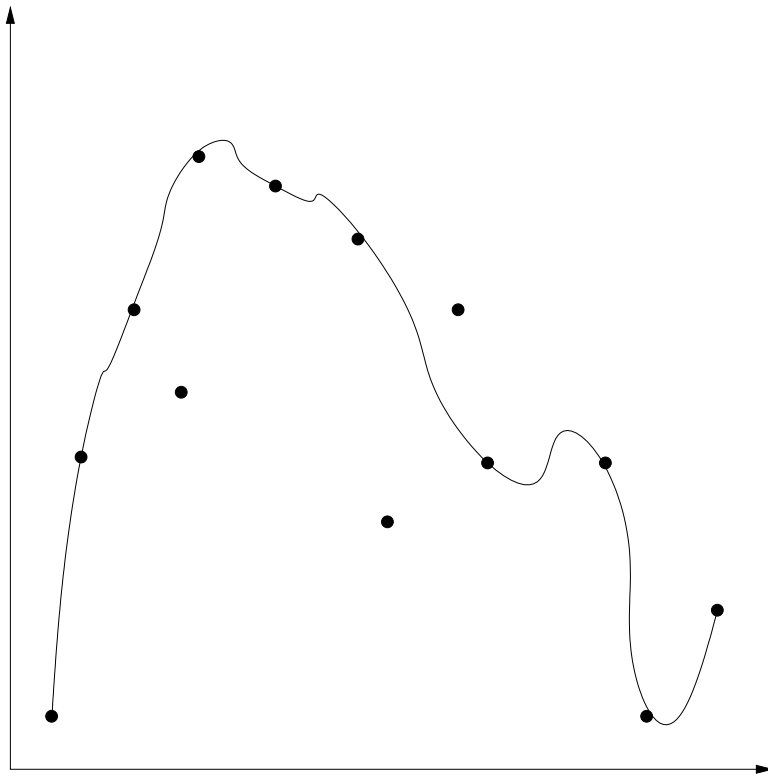


Figure 4. Simple regression for improved resolution.

3.5.2 Support Vectors

The Support Vector method of machine learning was selected for the estimation algorithm. This algorithm is general-purpose and has been used for various purposes such as image recognition and data mining. The Support Vector method was chosen because the models it generates encompass the functionality of a significant proportion of neural networks, Radial Basis Function (RBF) networks, and polynomial classifiers[50]. The simplicity and flexibility afforded by the Support Vector method thus makes it ideal when exploring the myriad parameters available in the location estimation system.

Support Vectors can be used to either classify data or perform regression analysis. Using Support Vectors as classifiers (often called SVM) would limit the granularity of the results to only those points surveyed. However, with regression (the technique used in this research), it is possible to interpolate between the measured values, potentially providing a higher resolution than the empirical measurements[36]. Unfortunately, there would still be limitations to the regression given the small signal variations. Extensive analysis of the use of SVRs for location estimation can be found in [25].

[44, 30] suggest using different radio maps to account for the changing radio environment at various times of the day. The maps were selected based on environmental probes. This system will also use environmental probes. However, instead of explicitly selecting the particular map to use based on the probes, the system will combine the measurements of the the mobile nodes and probes into a single tuple for regression.

The Support Vectors are complex algorithms with a variety of parameters. As mentioned earlier, the Support Vector method is being used for regression to estimate the x- and y-coordinates independently. The learning algorithm in SVR is used to minimize a convex function, but in this case, the goal is to fit the function to the data points. The final result from the SVR method would be an equation of the

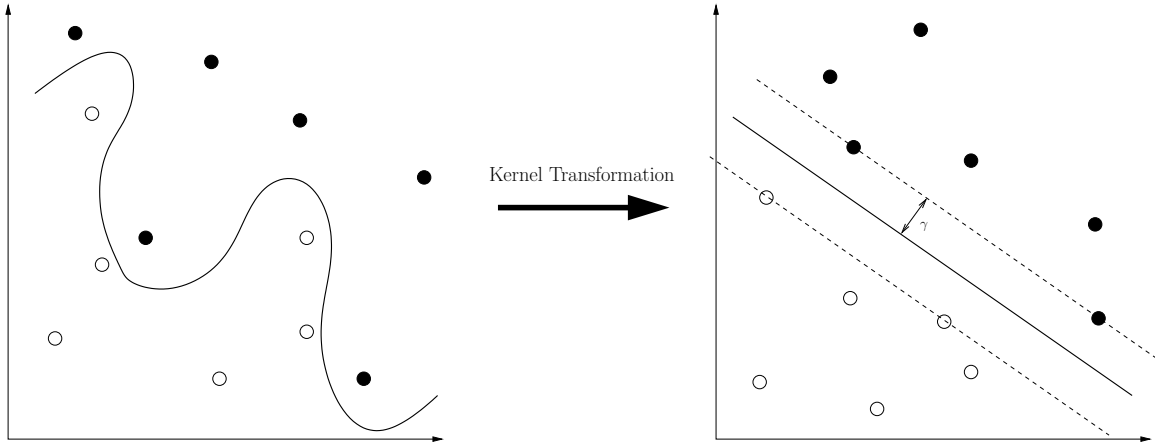


Figure 5. SVM classification in two dimensions.

form $f(x) = \sum_{i=1}^n \phi(\mathbf{x}_j)\mathbf{w} + b$, where n is the dimension of the input space, ϕ is the mapping to feature space, \mathbf{w} is the weight factor, and b is the bias.

The Support Vector method transforms a high-dimensional feature space with a kernel function to a lower-dimensional space suitable for linear classification or regression as seen in Figure 5. There are several commonly used kernel functions: linear, polynomial, sigmoid, and RBF. Though the choice of kernel would directly affect the function form of the final estimate and the performance of the SVM, practical results from various researchers have indicated the final support vectors generated by each kernel overlap significantly. Thus the choice of kernel based is less affected by functional form than it seems[50]. Furthermore, the nature of SVM training makes it computationally expensive to use kernels with more than a few parameters.

Many systems use the RBF kernel because of its generality (the behavior of the linear, polynomial, and sigmoid kernels are similar to the RBF kernel for certain parameters) and because it experiences less numerical difficulties (large degree polynomial kernels have asymptotic cases and sigmoid kernels are invalid for certain parameters)[51]. Each kernel has a several parameters that describe it (see Table 1). From there, another benefit of using the RBF kernel can be observed: only one parameter, the Gaussian width (represented by γ in Figure 5), is necessary, thus

Table 1. Kernel types and respective parameters.

Kernel type	Parameters
Linear	None
Polynomial	γ, d
Sigmoid	γ, r
RBF	γ

making its application relatively easier.

It is necessary to define a loss function to accommodate small errors (noise in the samples, possibly from experimental errors) in order to obtain a suitable generalization, as depicted in Figure 6. This is the ε -insensitive loss function[52]. The deviation of the training samples is represented by the slack variable ξ . However, minimizing ξ alone is insufficient to obtain a suitable generalization. It is also necessary to reduce the model complexity, and thus increase its ability to predict novel values. This is done by adjusting two parameters, C and ε which in turn affect the “flatness” of the regression curve. It is important to reduce C as far as possible as in has a prominent effect on the execution time of the SVR.

The RBF kernel was chosen for this analysis, given its generality. All factors considered, SVR tuning requires only the adjustment of C , ε , and the RBF kernel parameter γ [53, 54].

Manual adjustment of these parameters will yield fair results. The best values would be obtained through an exhaustive search through all parameters with cross-validation using several subsets of the entire training data set. Obviously this is far from practical. [51] recommends a grid-search approach with the possibility of enhancing accuracy by adaptively adjusting the grid resolution based on the resultant cross-validation rate. While this is also a brute force approach, it has the benefit

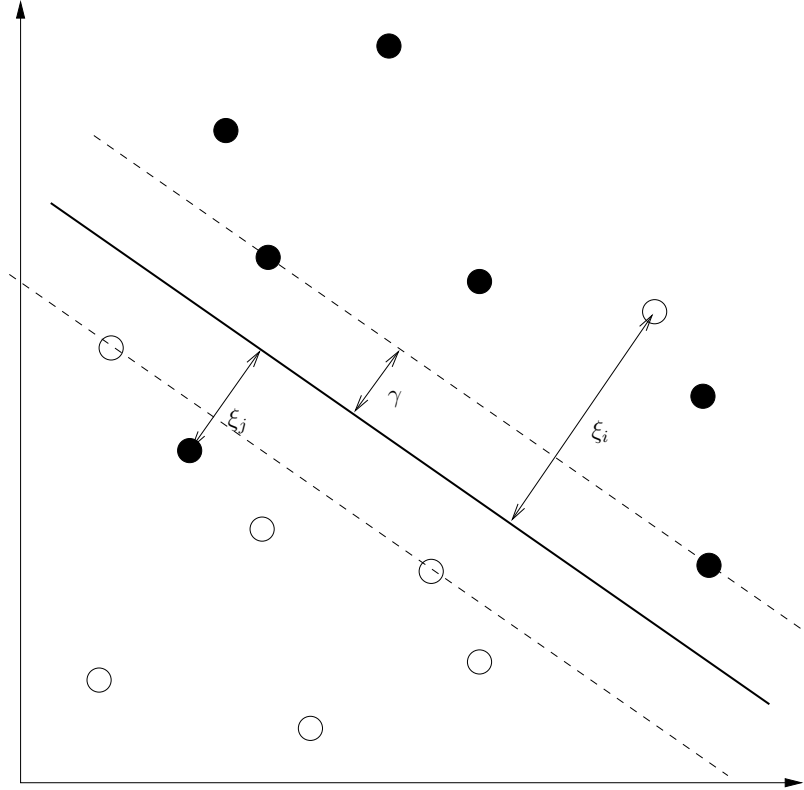


Figure 6. SVM classification with slack variables for classification errors.

of being easier to parallelize. There is less concern that it may miss key values unlike other more efficient but also more complex methods. [54] discusses an analytic approach to parameter selection given the properties of the input data and [55] describes tuning methods using evolutionary algorithms. The analytic approach will be examined briefly in the discussion (Section 5.7).

The training process itself requires the solution of a very large Quadratic Programming (QP) optimization problem. Various methods have been proposed to reduce this large QP problem and its corresponding computational complexity. The most popular of these approaches is Sequential Minimal Optimization (SMO)[56, 57] and its variants. SMO partitions the original QP problem to several smaller ones which can then be solved analytically. This avoids expensive operations on large matrices, resulting in linear scaling of memory requirements and, at worst, quadratic scaling of computational time.

Fortunately, most of the required functionality can be found in the LIBSVM[53]. LIBSVM implements ε -SVR and ν -SVR, uses a variant of SMO, and provides tools for cross-validation and grid-search. Both SVRs were evaluated.

Feature selection is another aspect of the Support Vector method that bears consideration. The list of features is discussed in greater detail in Section 3.6. Obviously, the number of features used would affect the speed of the SVR and ideally, the SVR should use an absolute minimum number of features. However, finding the minimum set of these features which can be used to reliably estimate location requires searching over all possible subsets.

To avoid the brute force approach, various standard search techniques can be applied. Fortunately, given configuration of the system, there are only a relatively small number of features available, compared to the thousands which may be used in applications like pattern recognition. Hill climbing is probably the easiest to implement with minimal heuristics but it is grossly inefficient in certain landscapes. More advanced algorithms utilizing forward selection or backward elimination would be more efficient but the complexity offsets most potential speed gains. Discussion of such algorithms lie beyond the scope of this research.

3.6 Wireless network parameters

Current location estimation systems rely only on a limited subset of parameters to aid location detection. This research combines the various parameters in an attempt to improve the prediction accuracy of the system.

The use of COTS hardware in implementing the architecture would restrict the use of advanced radio measurement techniques to the lowest common denominator available. This would ensure the portability of the algorithms to other wireless networking standards such as IEEE 802.11a, 802.11g and Bluetooth.

For the purpose of location estimation using radio measurements, parameters may

Table 2. List of features to be investigated

Feature	Source
Signal strength	Mobile nodes
Signal strength	Sensor probes
Noise level	Environment (Global)
Frame RTT	Mobile nodes

include the RSS measured at each sensor, the probe RSS, and the environment noise levels (Table 2 has a comprehensive list of features that are considered).

CHAPTER 4

IMPLEMENTATION

This research is concerned as much with the feasibility of the proposed secure architecture as with the performance of the algorithms.

4.1 Implementation overview

The training phase requires several steps. Figure 7 shows the data flow of the training phase. Similarly, after training, the online phase when the system is in operation also requires several levels of processing (Figure 8).

In the training phase, data is collected from the sensors and the surveyor. Given the Media Access Control (MAC) address of the sensors and surveyor, the system then filters the sensor data to obtain the appropriate sensor readings. In the correlation step, the various sensor readings are recombined to form the necessary features (environmental probes, RTT, etc) for the training. The data is also scaled and normalized to give remove bias and better results with the SVR. Finally, training of the SVR is done using values derived from the grid-search algorithm and this produces a model for the SVR, comprising of the necessary support vectors that describe the data.

For the online phase, readings from the sensors are collected. The system is also given the MAC addresses of the mobile nodes being tracked. As before, the readings will be combined and filtered to produce streams of data representing the sensor measurements of the mobile nodes and of the environmental probes. The correlate and scale step is also similar to the training phase, selecting the appropriate features and scaling them to match the scaling used in the training phase. These values and the SVR model are presented to the SVR algorithm to produce a prediction. If location history is necessary, this new location prediction may be fed back to the

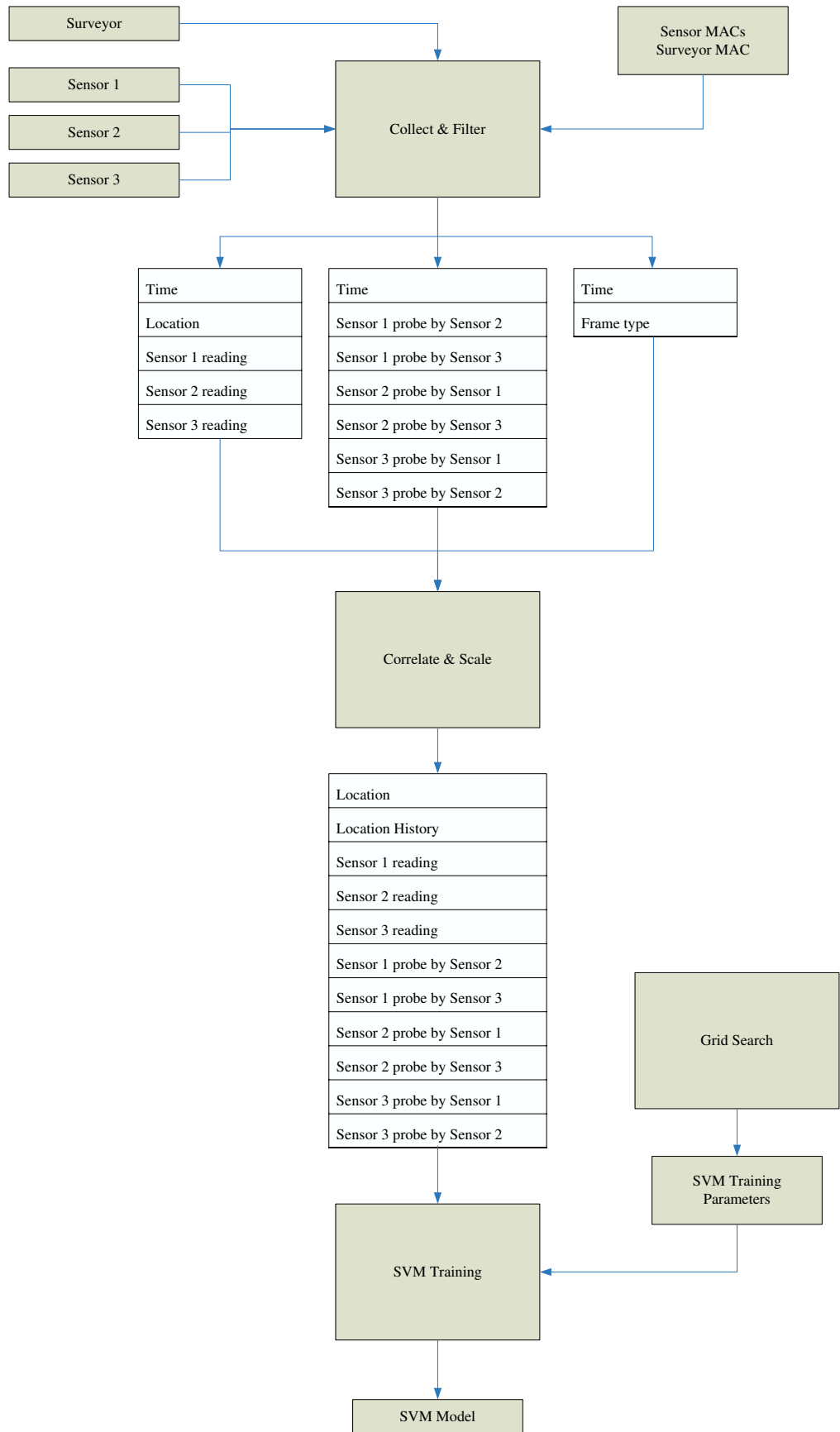


Figure 7. Training data flow.

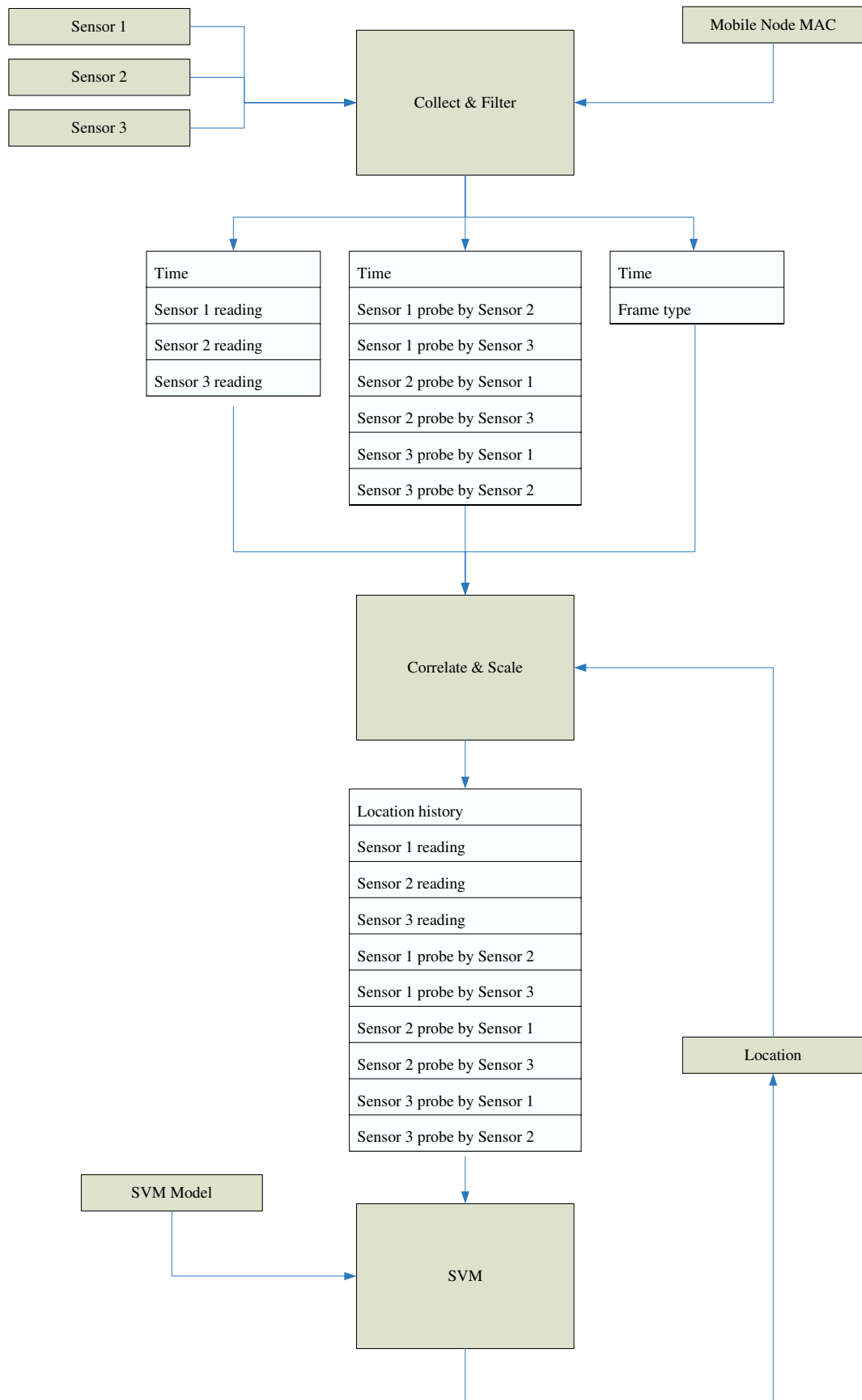


Figure 8. Online data flow.

system.

Many of the processing stages are similar between both the training and online phases. All these stages will occur in the “Server” module in the architecture (Figure 1). The stages will be described in detail in the following sections.

4.2 Server

The server encapsulates most of the functionality and is the central point of control for the system. To facilitate easy management, a GUI administrative console was created for the server. It provides the following capabilities:

- Monitor sensor availability and data
- Collect site survey data
- Filter and manipulate the data
- Visualize the data
- Adjust algorithm parameters
- Train algorithms
- Test algorithm accuracy

The interface of the GUI console can be seen in Appendix A.

4.3 Sensors

The sensors are low-end Pentium PCs with an IEEE 802.11b card and running Linux. Their sole purpose is to obtain measurements of the radio environment. The sensor program, written in C++, together with the Host AP drivers[58], capture all wireless traffic on a specified channel.

The drivers are able to provide a wealth of information through the PRISM monitoring header (identified as ARPHRD_IEEE80211_PRISM in various source code).

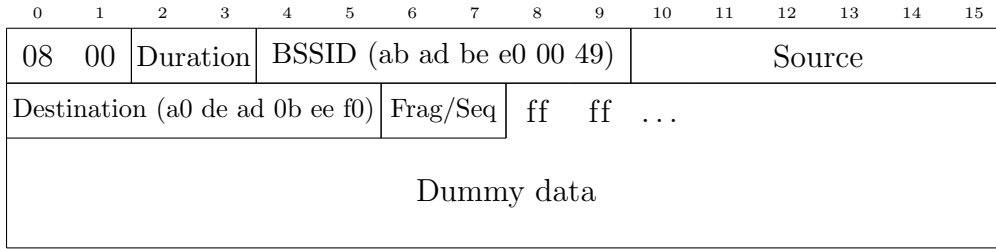


Figure 9. Probe frame layout.

Of concern is the signal strength, noise level, and MAC time. These three values, together with the source MAC address and destination MAC address are parsed by the software and sent to the sensor collector.

There are three sensors deployed on the third floor of the College of Computing Building (CCB) in Georgia Tech (Refer to Appendix B). While three sensors are deployed, it is possible to filter the measurements in software to simulate the use of only one or two of the sensors.

While it is possible to configure the sensors for completely passive monitoring, the current set up has each sensor broadcast a probe frame. These probe frames are IEEE 802.11 frames that can be easily identified by their contents. The layout of each probe frame is shown in Figure 9.

The duration and fragment/sequence fields are calculated and set automatically by the firmware. The source field, as is normally the case, contains the MAC address of the sensor broadcasting the probe. The other fields (Basic Service Set Identifier (BSSID), destination, and data) are used to uniquely identify the frame as a probe.

Since the sensors are at fixed locations, the fluctuations in signal characteristics of the probes originating from each sensor are an indication of changes in the radio environment. This data will be utilized by the algorithm later for refining the location estimations.

The rate at which the probes are sent is tunable, but is currently set to 120 seconds.

As noted earlier, it is possible to secure the data by providing physical security at the sensors and information security through the use of secure tunnels. Because of Information Technology policies at the deployment location, the data was secured using an Secure SHell (SSH) tunnel and the sensors themselves were placed in rooms with restricted access. This set up provides a reasonable level of security by encrypting the data and providing limited authentication at the end-points (sensors and server).

4.4 Data collection

4.4.1 Site survey

The datasets for both training and evaluation are subsets of a larger dataset comprising of multiple measurements accumulated over several days. The data collection process will also be referred to as the site survey.

A laptop and a PDA were used for the site survey, each using a different wireless card. This variety was used to evaluate the ability of the system to handle the significantly different radio characteristics of each device. The site survey software, written in Java, runs on the laptop.

The survey process comprises of the user going to fixed locations and indicating these locations on the survey software. The survey software will then convey this location to the database and the information will be correlated with radio measurements of the survey system at that instant in time. When this is repeated over several locations at different times of the day, thus building a map which relates the location of the device to a particular set of radio measurements.

The locations at which measurements were taken is indicated in Appendix C.

Both laptop and PDA ran ping with a 1 KB payload (`ping -s 1024 www.google.com`) to ensure that the sensors will have adequate opportunity to detect their signal. Since the survey software would not run on the PDA, the PDA was placed close to the laptop for each measurement, so that the location of the laptop indicated in the survey software would also reflect the location of the PDA.

As mentioned earlier, the site survey was done at different times of the day. This was to obtain measurements of the same location when the radio environment is potentially different due to the presence of human and other environmental factors. Measurements were taken during the middle of the day when the building was most crowded, during the evening when it was minimally occupied, and in the middle of the weekend when it is nearly deserted.

Survey readings were taken at various orientations. However, the software does not take note of the orientation of the user, so no data is available as to how orientation affects the measurements. Instead, readings from the different orientations will be used together to see if the system can be made insensitive to the orientation of the user.

In all, 3,267,197 sensor readings were collected, which were filtered and correlated to form 12,724 data points for the next stages.

4.4.2 Normalization

Normalization of the data eliminates the dependence of the algorithms on certain units of measure. Furthermore, the units used in reporting signal strength and noise levels by the PRISM monitoring header are ambiguous.

However, even with normalization, there is the issue of representing the lack of a data point, i.e. when a sensor is unable to detect any signal from the mobile node at a certain location. It was decided that the values would be normalized to a range of $[0, 1]$, with 0 representing the lowest detected level. If there were no readings available, it is represented with -1 to differentiate that from the case where a minimum reading was available.

Different wireless network interfaces have different characteristics such as transmission power and directionality. Despite this, it is expected that the different network cards will still exhibit similar variations in signal strength with respect to location. This hypothesis was tested in the implementation.

4.4.3 Round-trip time

Unlike signal and noise levels, the RTT of the nodes are not directly reported by the wireless hardware. [35] proposes a method using the high-resolution MAC timestamps on RTS/CTS frame pairs as a measure of RTT, since these are handled rapidly by the wireless hardware, thus eliminating other potential sources of variation.

Since the MAC timestamps have microsecond resolution, one would reasonably conclude that the smallest distance measurable using the RTT delay is $3 \times 10^8 \text{ m/s} \times 1\mu\text{s}/2 = 150 \text{ m}$. However, [35] suggests that stochastic resonance may result from the clock quantization and thus allow a greater resolution (less than 150 m) than this.

Stochastic resonance is commonly applied to biological and other “chaotic” systems. It relies on a known noise distribution which is added to the signal. This added noise will cause the measured signal to alternate between two quantized measurements. The distribution of the alternating measurements is dependent on the measured signal (in terms of fractions of a quantum). Thus observation of distribution of alternating measurements can aid in the resolution of the measurements.

The RTS/CTS frame exchanges in an infrastructure network happen between the mobile node and access point. To be able to record both frames for RTT estimation requires that the sensor be able to receive signals from both mobile node and access point. Conventional radio-fingerprinting techniques relying on signal and noise levels only require the mobile nodes to be within detection range. This method thus limits the sensor deployment, or would suffer from having much less useful data than conventional radio-fingerprinting techniques. It may be possible to reduce the constraints somewhat, i.e. a rapid string of consecutive data frames, initiated by a RTS, or a string of ACK frames initiated by a CTS. However, this requires that the fragments being transmitted are all of identical size. While this is most often the case, it is not guaranteed by the IEEE specifications[1].

[35] utilized RTT measurements both in- and outdoors and noted that the indoor set up was far less accurate, probably because of multi-path effects. It is likely that this issue would also manifest itself in this research.

4.5 Algorithms

4.5.1 Algorithm framework

The proposed algorithm framework is designed as a test platform for various estimation algorithms. It prepares the data from both the database and sensors for processing by the algorithms and then interprets the results from the algorithm for the visualization system.

The proposed algorithm framework eliminates the dependence on variables such as the specific identity of the mobile nodes being tracked, the physical units of position and radio signal quality.

4.5.2 Identification of mobile nodes

The proposed system identifies and distinguishes mobile nodes using the IEEE 802.11 MAC address in the header of each and every frame transmitted by the nodes[1, 2]. This has the weakness of being susceptible to spoofing attacks. Future research may address this issue through the use of a unique identifying key in each frame, possibly derived from a cipher stream, or by using unique identifying characteristics of the hardware such as sequence numbering patterns or clock drift. In the mean time, it may be possible to detect spoofing attempts as any other wireless spoofing attack—by detecting the abnormal variation in the physical location of the actual node and the attacking node.

4.5.3 SVR training

SVR training is a computationally intensive process, especially when the grid-search algorithm is used to optimize the SVR training parameters. As such, it was necessary to extensively exploit the implicit parallelism of the grid-search algorithm to obtain

results quickly.

The grid-search module in the server dispatches training jobs to a pool of clients. Since the LIBSVM[53] code is easily ported, the server could utilize a wide range of systems for the pool of clients. The experimental set up used had up to four PCs in the pool, each averaging the performance of 2.5 GHz Intel PC. The dispatch code was constructed robustly so as to recover from client failures and allow clients to join and leave the pool during training.

Even with these resources, each data set took about one day to train and optimize. Each data set was optimized first using a coarse grid covering a medium-sized region of space. Depending on the results, a finer, smaller grid or coarser, larger grid may be selected for further iterations. The process is repeated until the optimal training parameters can be determined, i.e. the optimal parameters and optimal correlation coefficient is stable at different granularities. The parameters for each iteration were configured through the server GUI console.

LIBSVM provides a simple means of doing n -fold cross validation. All results were obtained using two-fold cross validation. Unfortunately, the LIBSVM scripts only performs grid-search on SVMs, not SVRs. A grid-search on an SVR has the added complexity of an extra parameter, resulting in a search in 3D space and had to be accomplished via a custom script. Visualization of the grid-search multi-dimensional results was done using modified data-mining software[59].

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Interpreting results

The following results may be represented by the Mean Squared Error (MSE). The MSE is simply the square of the RMS of the error.

The training data points and testing data points are taken from the same grid of survey points. The n -fold cross validation described previously selects a fraction of the survey data to use for training and testing. Thus the location of the test points would coincide with the training points—effectively the SVR in this simple set up would only have to determine which point on the grid a given set of test data falls on, thus work as a classifier (Figure 10a). However, given the ability for the SVR to interpolate between the training points, it may produce predictions that are not aligned to the grid (Figure 10b). Furthermore, the predicted points tend to be close to the original grid. This results in very small MSEs.

In some cases, the SVR will make an entirely wrong prediction that places the predicted point nearer to a completely different point on the grid—in other words, misclassifying the point. Because of the number of test points used, the MSE will

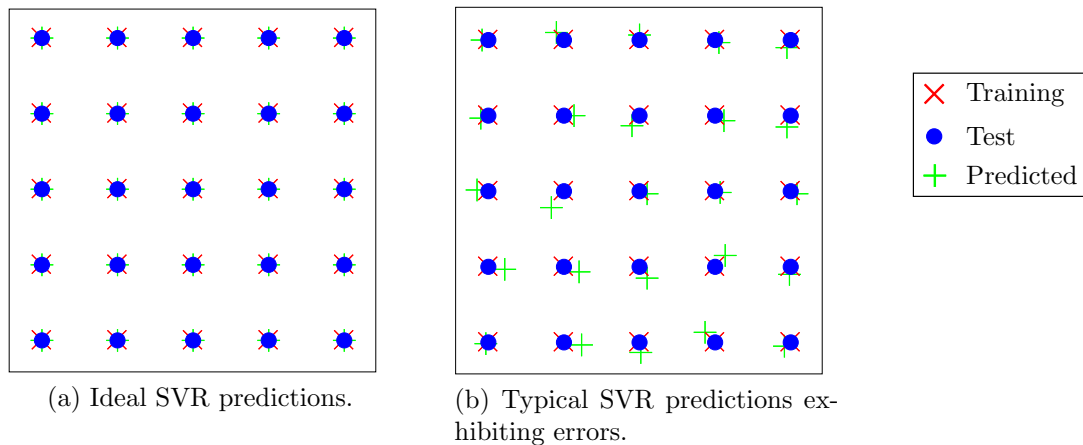


Figure 10. SVR predictions performing as classifiers.

still be relatively small because of the averaging.

It is important to note that the MSE does not indicate the resolution of the system when used this way, only how far the predicted points vary from the original test and training data (i.e., the accuracy). Since the test data is given at approximately 3-foot intervals (Section 4.4.1), it cannot be concluded if the SVR can resolve (i.e., differentiate) distances smaller than this because the predicted locations will cluster around each of the original training points.

5.2 Signal Map

The signal maps collected for this research represent a large amount of data. Each map would cover the 245 sample points at different times and be different for each sensor. However, to provide an idea of how the signal varies in the environment, a signal map for the sensor in room 331 (marked by an “X”) late in the evening is show in Figure 11.

5.3 Hardware variations

Variations in the radio performance of the laptop and PDA can be see in Figure 12, which plots their measured RSS values against each other to show the correlation. The data plotted are the signal strengths of the laptop and PDA measured at the sensor whenever values for both devices are simultaneously available. As mentioned earlier (Section 4.4.2), the PRISM monitoring header uses unspecified units. Various unofficial sources indicate that the units are on the order of dBi, but the absolute values may vary depending on the firmware version. Fortunately, since the measurements are done by the same sensors, the units are unimportant and only the relative values matter. Furthermore, these values will be normalized before being used with the SVR.

The correlation coefficient for the two datasets is large (approximately 0.9957),

which indicates that the signals from the different wireless cards vary in a similar fashion. With the appropriate normalization, it would be possible for the estimation algorithm to function with different wireless cards without recalibration. Future work may investigate a larger selection of wireless network cards to confirm this.

5.4 Signal and noise levels

The estimation algorithms were initially trained with only the sensor data for the surveyor, i.e. no environmental probes or RTT, using both ε - and ν -SVRs. The ν -SVR was extremely sensitive to the training parameters, and execution time would increase abruptly for certain values, making grid-search difficult.

The results of the ε -SVR are shown in Figures 13 and 14 for the x- and y-coordinates. The results of the ν -SVR are shown in Figures 15 and 16. Some graphs are similar but are plotted over different regions of search space. The final training parameters are listed in Table 3. If several parameters give the same results, the one with the lowest C , and thus the fastest, was selected. In the plots, the correlation coefficient is indicated by the size (larger cubes indicate higher correlation) and color (color legend is alongside the graph) of the cubes, and small black cubes indicate where data was unavailable, usually because the training process took too long. Also, the parameter values with the highest correlation coefficient are indicated with translucent red cubes. Note that the axes are \log_2 , which corresponds directly with the grid-search algorithm, and that the values for each of the parameters are unitless, since the input data has been normalized to be likewise unitless.

Plots of the predicted versus actual normalized (unitless) locations for the ε -SVR are shown in Figure 17 for the best training parameters as determined by the grid-search. The plots indicate a fair correlation (MSE of 0.00573 and 0.00537 for the x- and y-axis respectively) between the predicted and actual locations with little error and agrees well with a similar, more extensive test performed in [37]. Unfortunately,

the excessive training times of the ν -SVR ruled out its practical use.

While Figure 17 shows good accuracy of approximately 1.11 feet, this was the result of testing all the training data against an SVR model that was produced likewise. However, the regression method was chosen to improve the apparent resolution of the radio map. To this end, it appears to be successful. To test this, certain data points were removed from the training data to create a coarser survey grid. Figure 18 plots selected estimates from the algorithm given a coarse grid of survey data. The \circ points are those used to train the algorithm, while the \times points are those predicted by the algorithm. \triangle indicates the actual location. Locations marked with \triangle were not used in training the SVR, but only in the testing phase. Deviations are larger than in Figure 17 because less data points were used in the training.

5.5 Environmental probes

The second set of tests were performed with the inclusion of environmental probe information. In contrast to methods in [26], the probes were not used to select a specific radio map, but were included as part of a general map. This would potentially allow the estimation algorithms to interpolate between the surveyed radio conditions. The training results of the ε -SVR with environmental probes are presented in Figures 19 and 20, while those for the ν -SVR are shown in Figures 21 and 22. Figure 23 shows the accuracy of the ε -SVR predictions and the improvement in accuracy is immediately apparent. The MSE of the x- and y-axis are 1.02×10^{-6} and 0.189×10^{-6} respectively, which translate to a virtually negligible distance of 0.000125 feet. As noted earlier, the MSE is obtained by averaging the error distances over the large number of sample points. This does not necessarily imply that the system may resolve such tiny distances, only that virtually all predicted locations coincided with the training points. In fact, the MSE would undoubtedly be larger if the system was tested with samples from locations that did not coincide with the 3-foot training grid,

as was tested in Section 5.4.

5.6 Round-trip time

Given the additional constraints imposed when obtaining the RTT for radio-fingerprinting (Section 4.4.3), it comes as no surprise that there is little suitable RTT data in the experimental dataset. A total of 31 data points were extracted from the entire dataset, far too little for any meaningful purpose.

The problem was compounded by the small ping payload used (Section 4.4.1). The default RTS threshold for PRISM cards on Linux is 2,432 bytes [58]. Frames below this size are sent without requiring RTS/CTS exchange [1, 2]. The ping payload of 1,024 bytes, even with IP and IEEE 802.11 headers, is less than 2,000 bytes, and thus, few RTS/CTS pairs were generated by the survey equipment.

In an attempt to increase the RTT data available for the location estimation algorithm, the constraints were relaxed to include frames sent in rapid succession. 1,053 data points were extracted using this criteria and used with the location estimation algorithm.

The grid search results for RTT is shown in Figures 24 and 25. The final values and correlation coefficients are in Table 3.

5.7 Comparison

Table 3 compares the performance of each set of features and the SVR types.

Using the ν -SVR requires much a longer training time than the ε -SVR but yields similar accuracy. Thus, it is recommended that only the ε -SVR be used.

Environmental probes prove to have a significant positive impact on accuracy. Given the different environmental conditions in which the data was collected, some variation in the sensor readings was expected. With the use of environmental probes, the prediction algorithm was able to compensate for these variations to produce more

Features		SVR	Axis	Parameters (\log_2)			Correlation coefficient
Probes	RTT			C	γ	ε or ν	
No	No	ε	x	2	11	-12	0.8635
			y	3	9	7	0.9272
		ν	x	10	10	0	0.8656
			y	4	9	0	0.9244
Yes	No	ε	x	7	5	-10	0.9969
			y	13	5	-13	0.9981
		ν	x	6	5	0	0.9969
			y	5	4	-2	0.9979
No	Yes	ε	x	-2	8	-6	0.6196
			y	0	8	-4	0.5266

Table 3. Summary of optimal SVR parameters and results.

accurate predictions.

It can be seen from these results that using the RTT does little to improve the quality of the predictions. This is expected given the limited quality of the data. Further testing with better data would probably be needed before the utility of RTT in an indoor environment can be concluded. However, based on the data collected for this research, the use of RTT cannot be recommended.

Also, it is interesting to note that the optimal parameters for the SVRs obtained by the grid-search method differ significantly from those predicted by the analytic method from [54]. For example, using the analytic method on the x-axis data, we obtain:

$$C = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|) = 1.2124$$

$$\varepsilon = \tau\sigma\sqrt{\frac{\ln n}{n}} = 0.02166$$

for the ε -SVR without environmental probes or RTT, where \bar{y} and σ_y are the mean and standard deviation of the training responses respectively, τ is an empirically derived constant [54], σ is the input noise level, and n is the number of training

samples.



Figure 11. Signal map for sensor in room 331 (marked by red "X") late in evening.

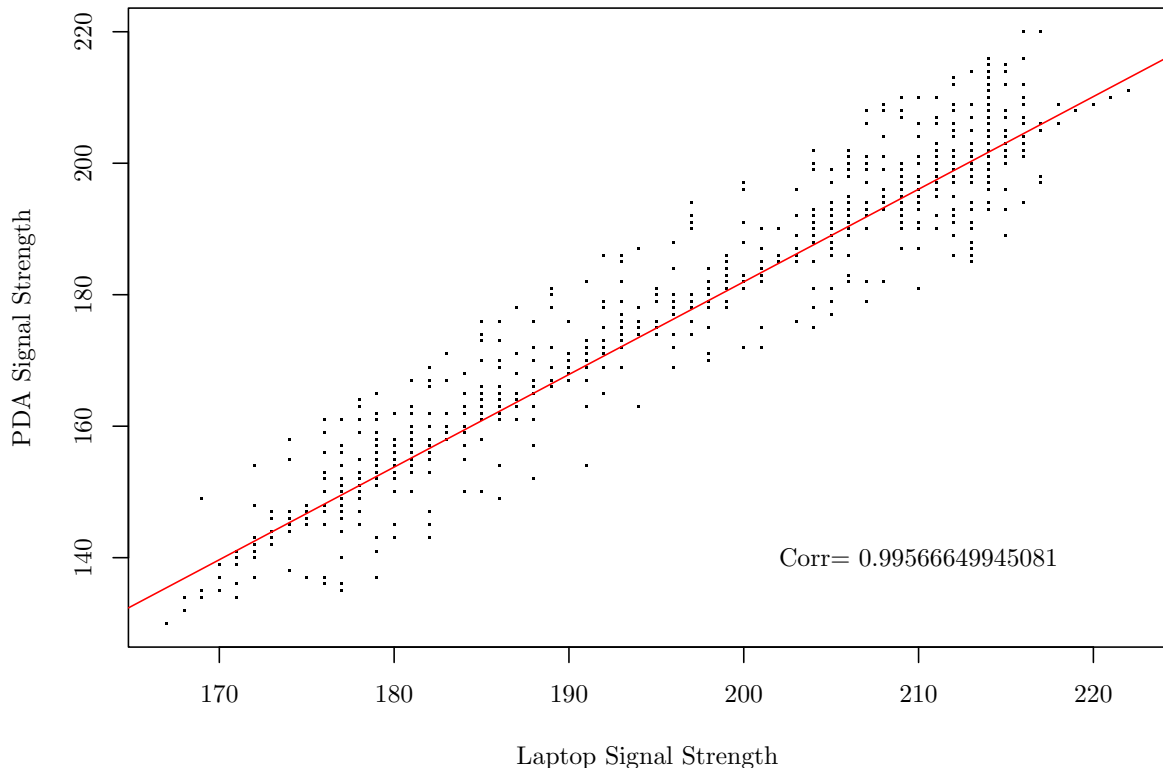


Figure 12. Correlation between signal strengths of different wireless NICs.

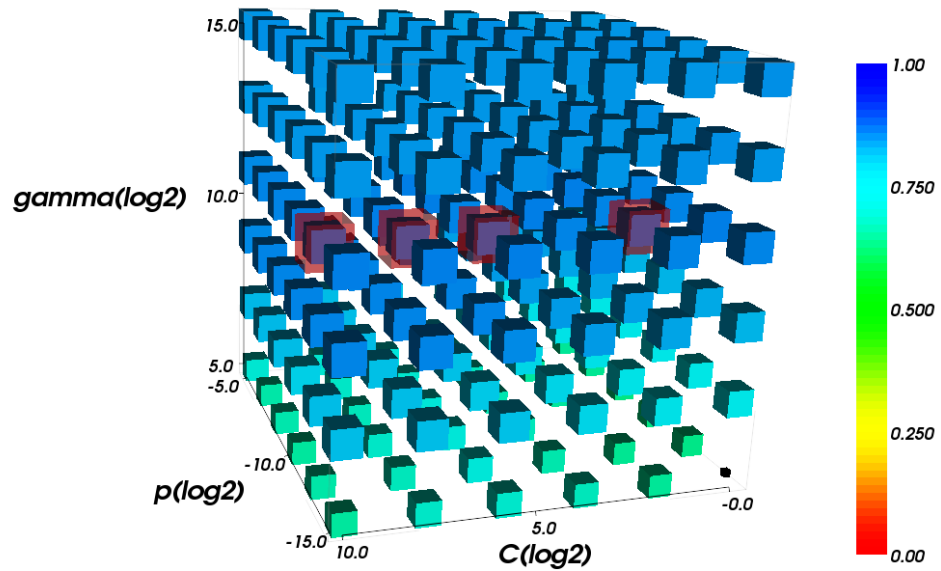
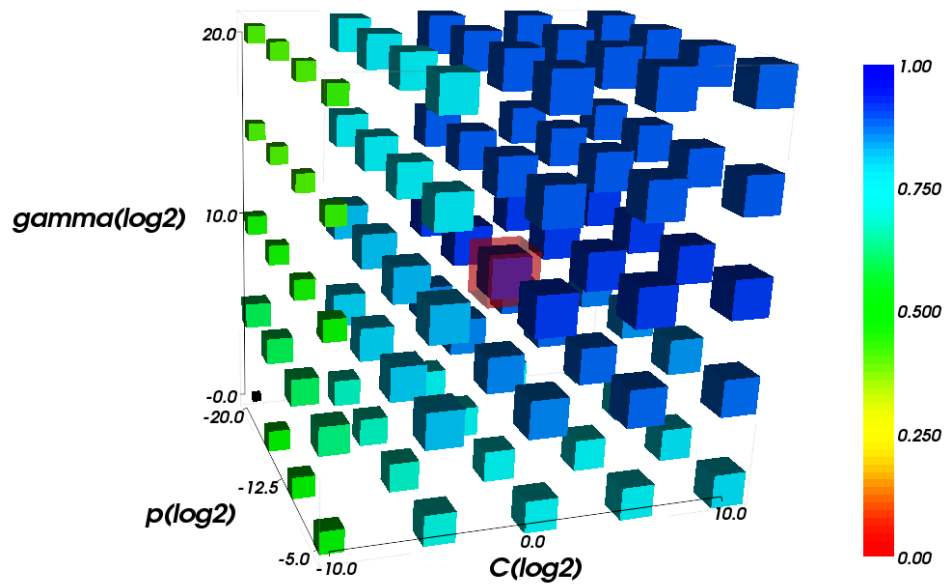
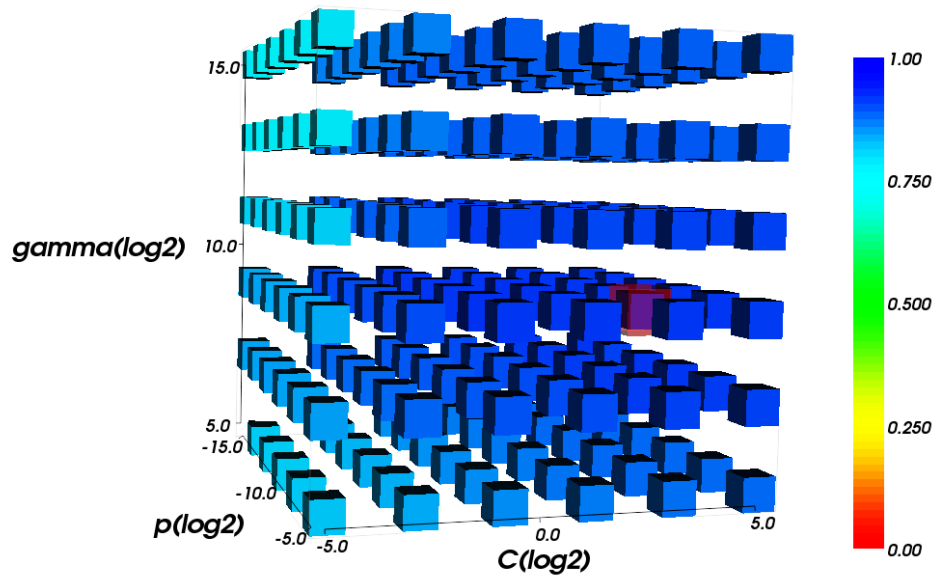


Figure 13. Plot of cross-validation rate against training parameters C , γ , and p for ϵ -SVR.



(a)



(b)

Figure 14. Plot of cross-validation rate against training parameters C , γ , and p for ϵ -SVR.

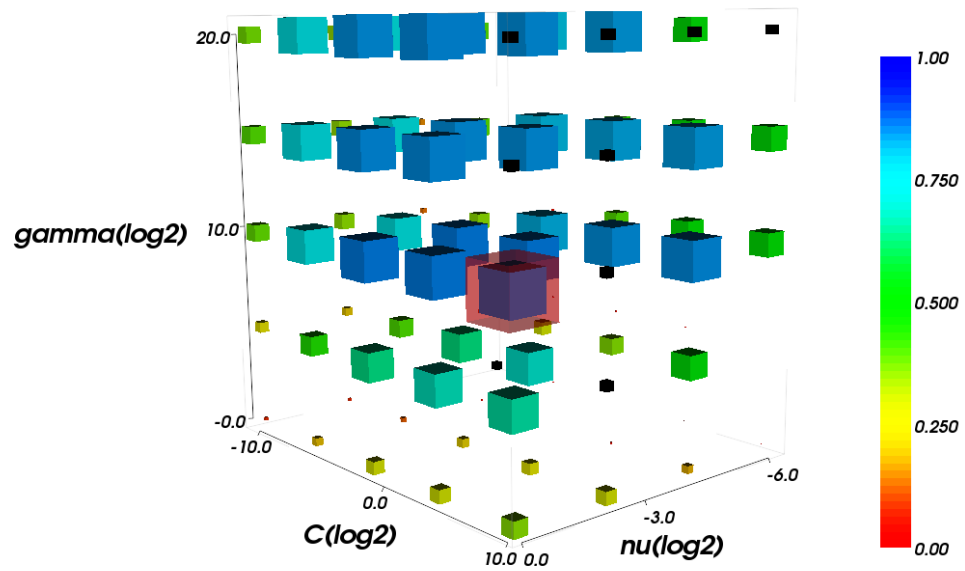


Figure 15. Plot of cross-validation rate against training parameters C , γ , and p for x-coordinate for ν -SVR.

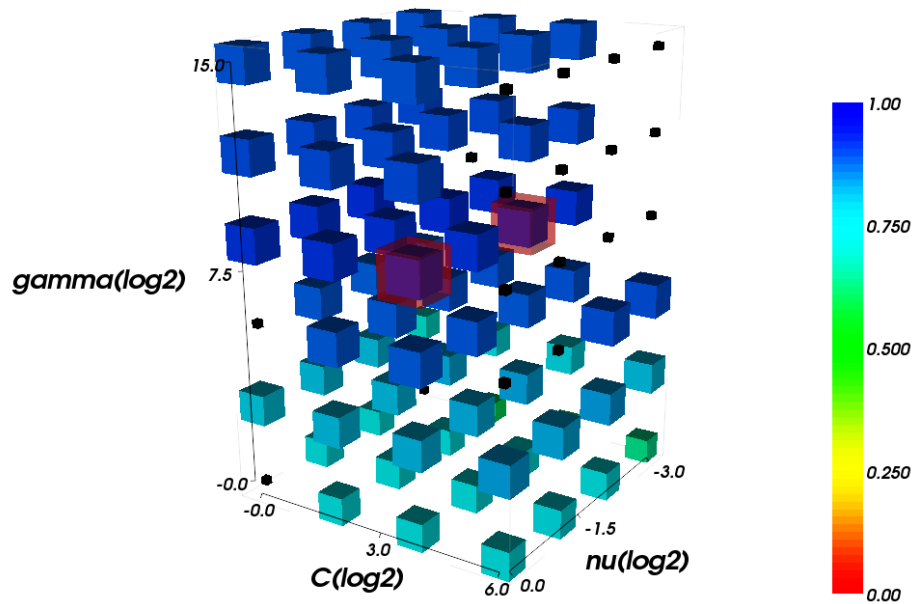
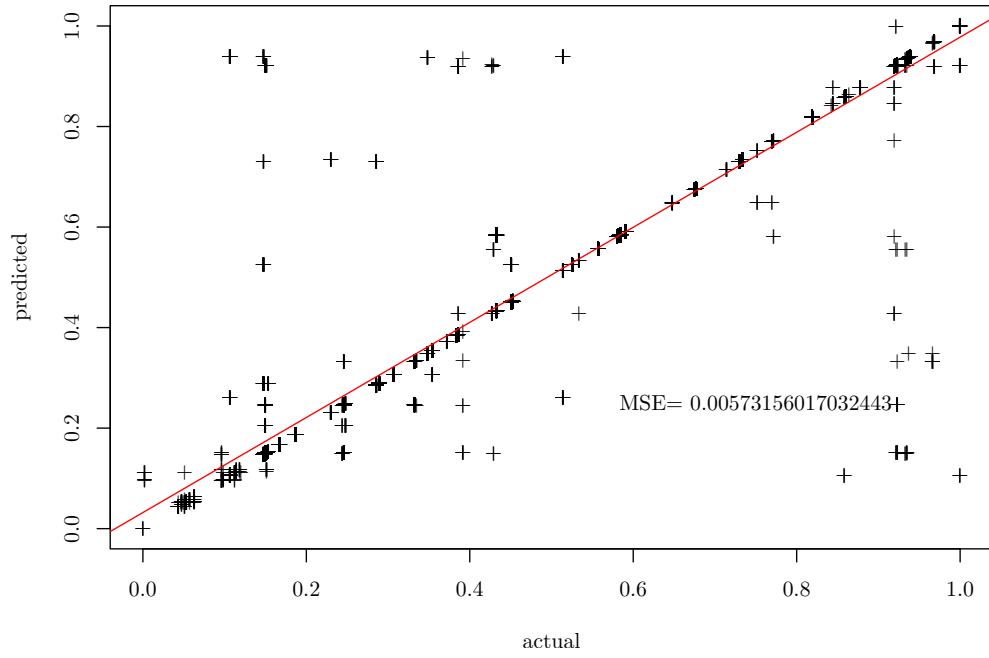
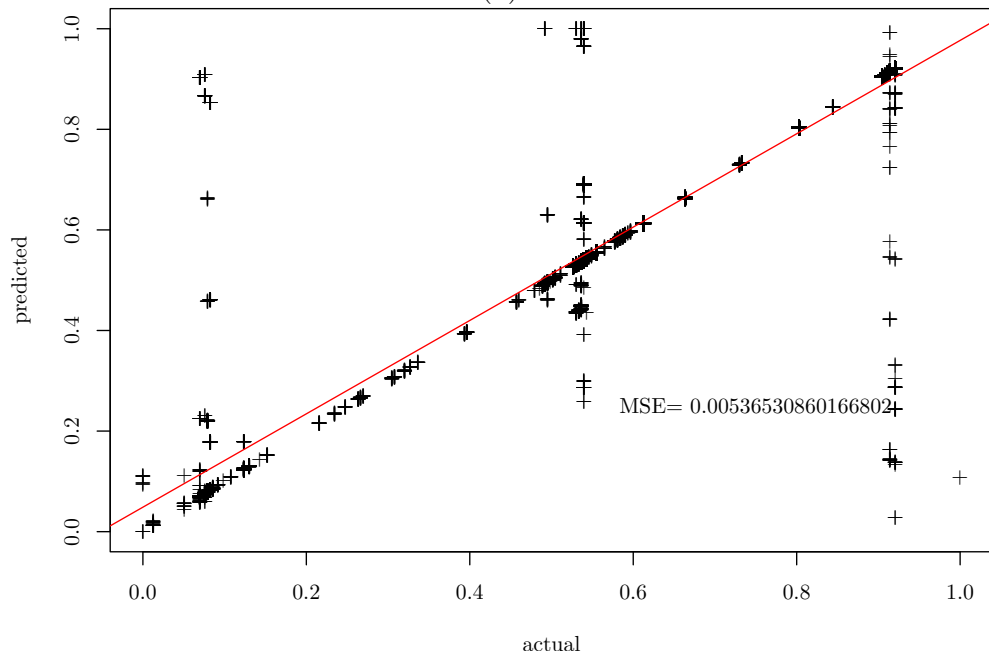


Figure 16. Plot of cross-validation rate against training parameters C , γ , and p for y-coordinate for ν -SVR.



(a)



(b)

Figure 17. Plot of predicted versus actual locations for ε -SVR for a) x-coordinate and b) y-coordinate.

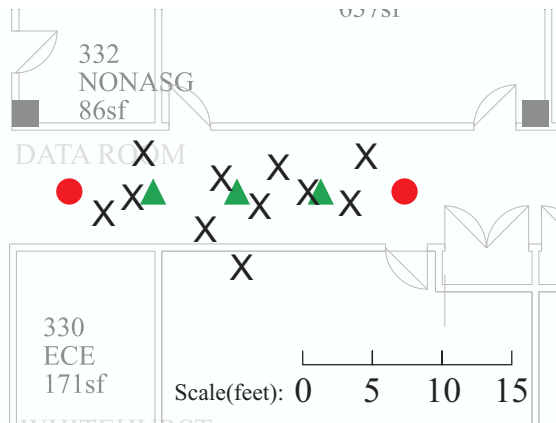
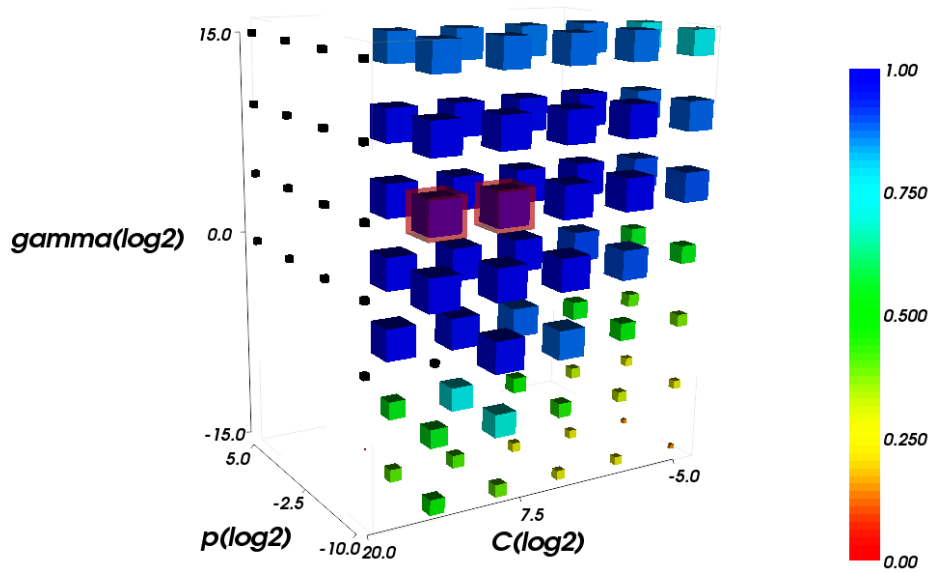
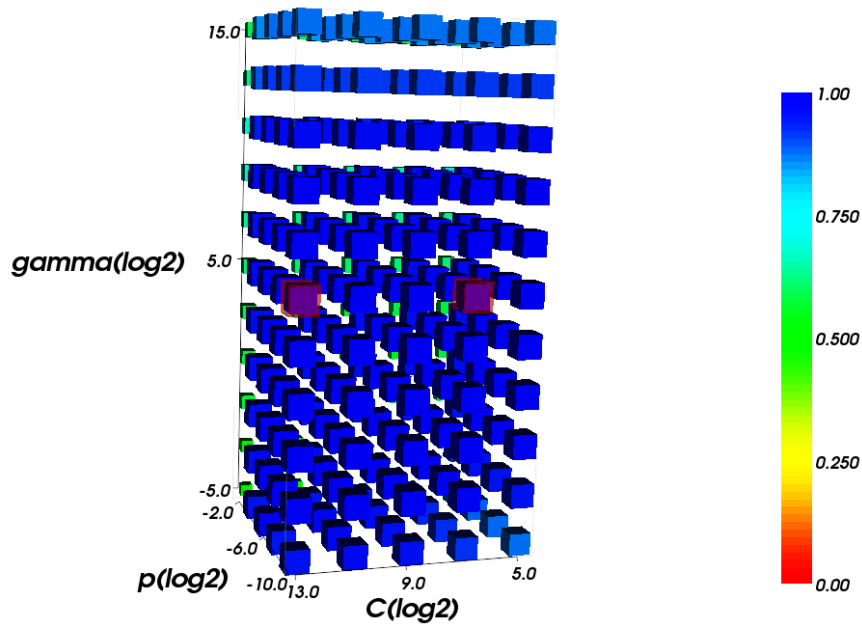


Figure 18. Plot of estimated and training points.

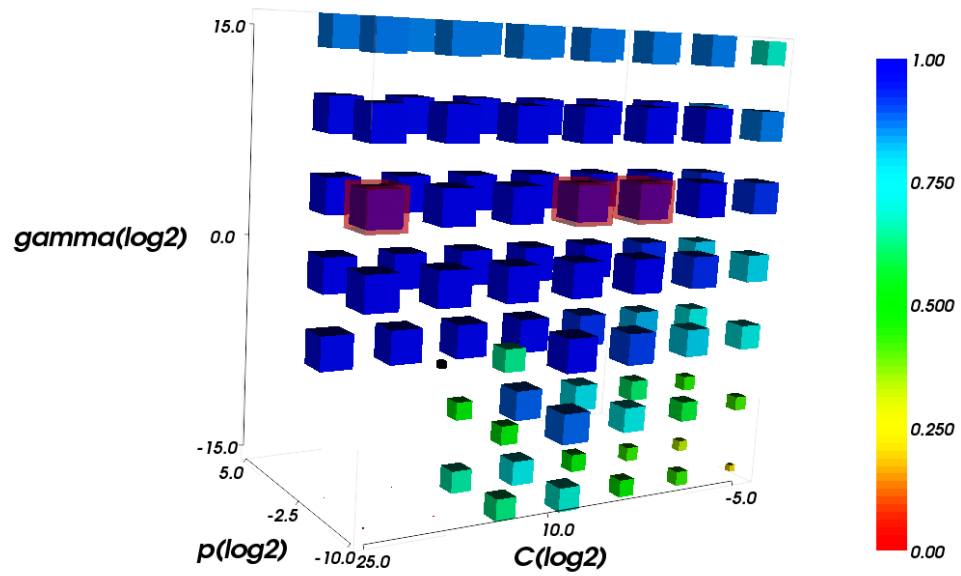


(a)

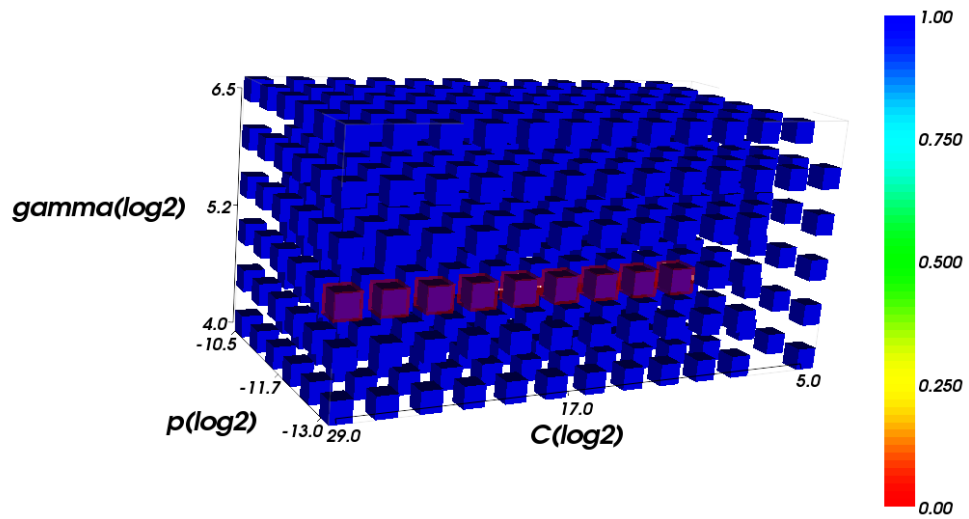


(b)

Figure 19. Plot of cross-validation rate against training parameters C , γ , and p for x-coordinate with environmental probes using ϵ -SVR.

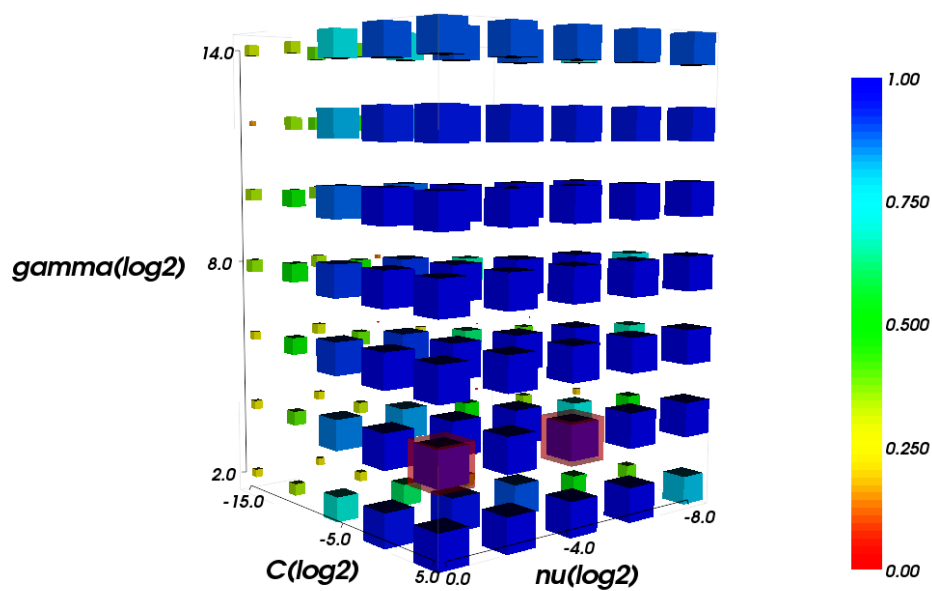


(a)

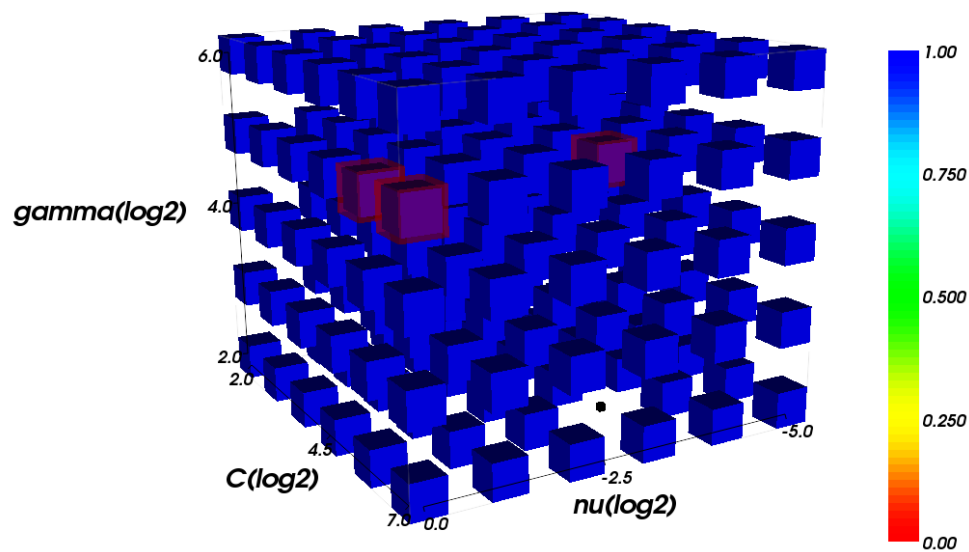


(b)

Figure 20. Plot of cross-validation rate against training parameters C , γ , and p for y -coordinate with environmental probes using ϵ -SVR.



(a)



(b)

Figure 21. Plot of cross-validation rate against training parameters C , γ , and ν for x-coordinate with environmental probes using ν -SVR.

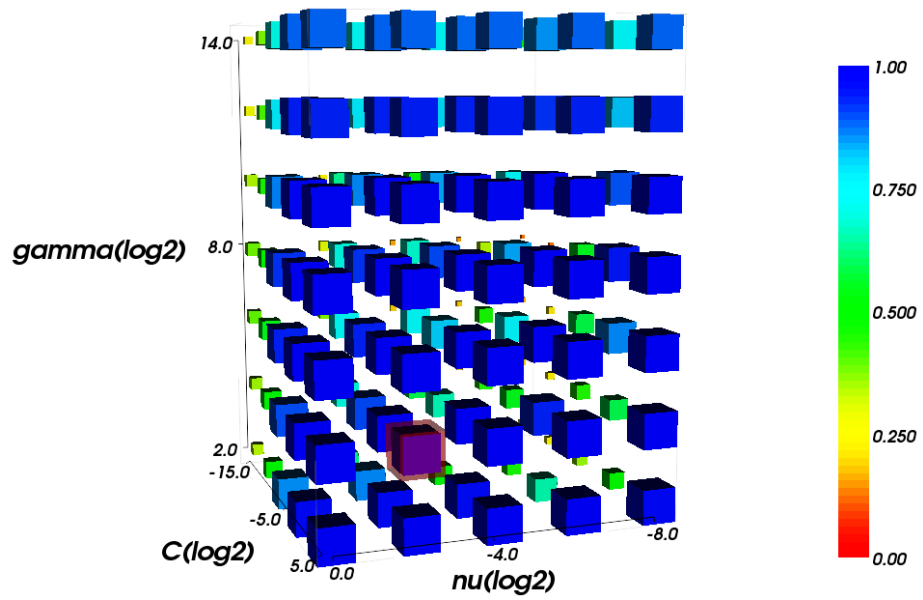
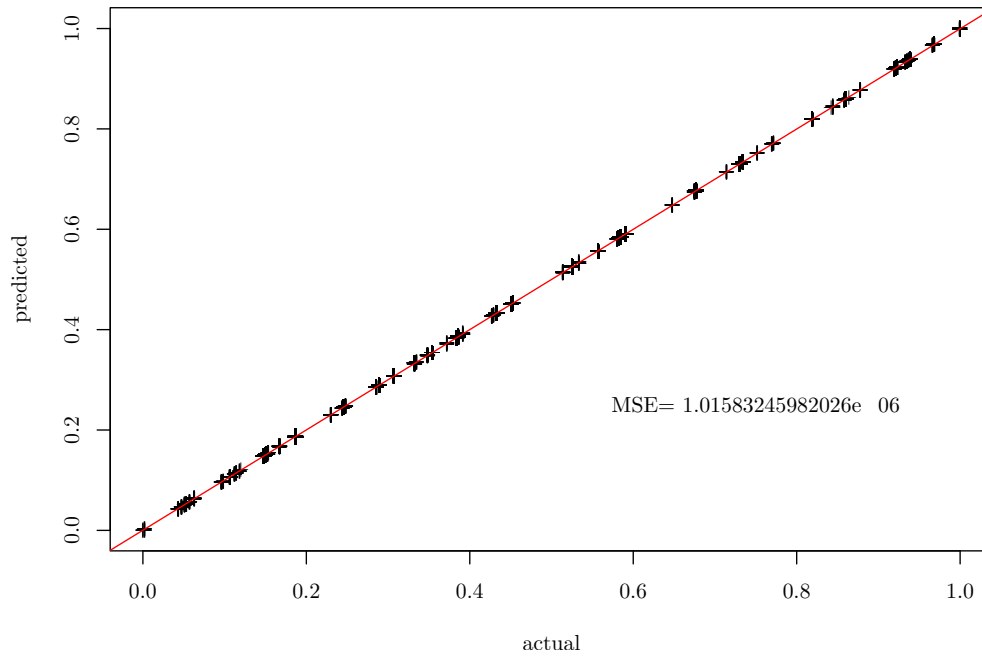
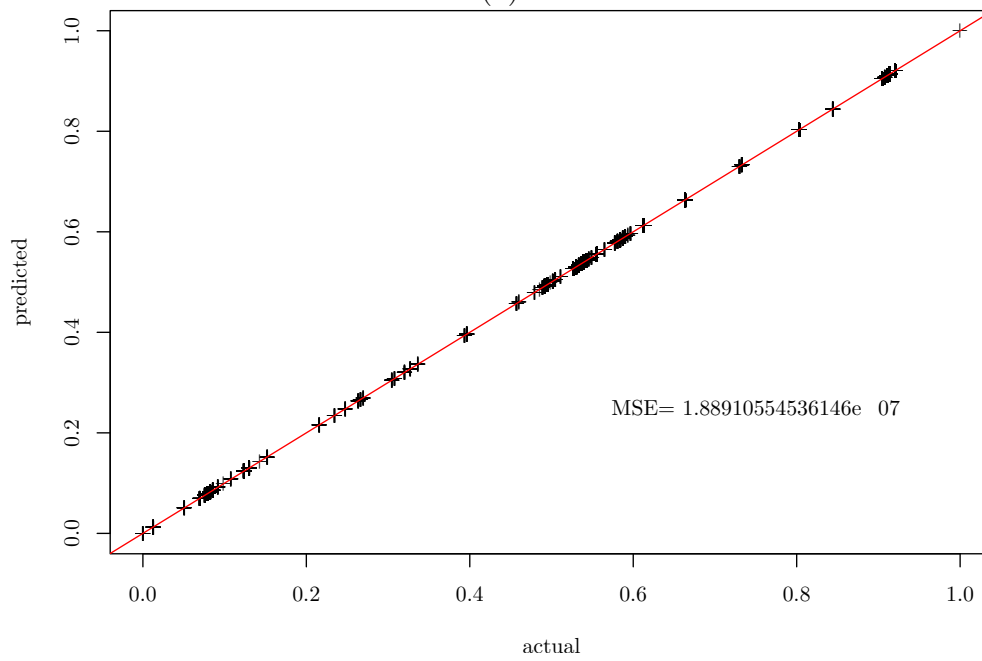


Figure 22. Plot of cross-validation rate against training parameters C , γ , and ν for ν -SVR.



(a)



(b)

Figure 23. Plot of predicted versus actual locations with environmental probes for a) x-coordinate and b) y-coordinate.

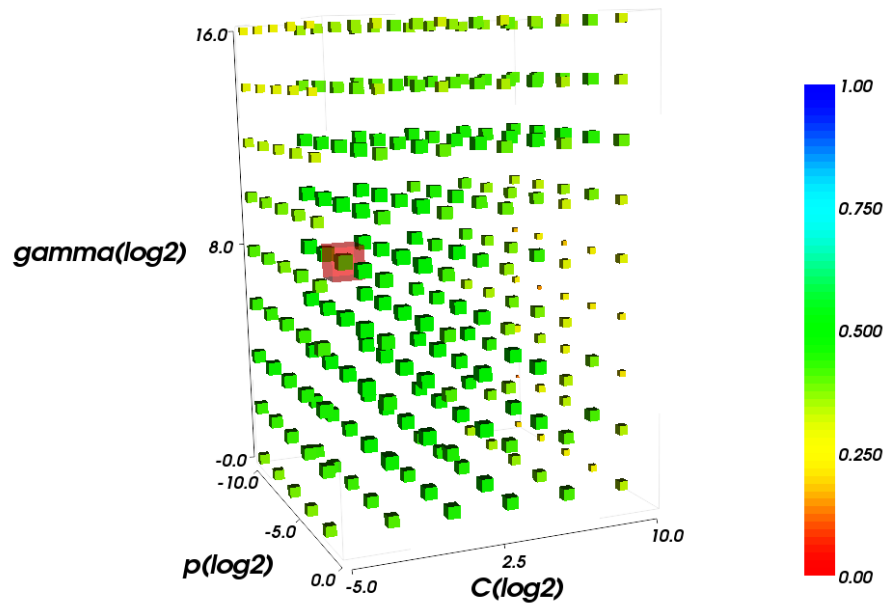


Figure 24. Plot of cross-validation rate against training parameters C , γ , and p for x-coordinate with RTT using ϵ -SVR.

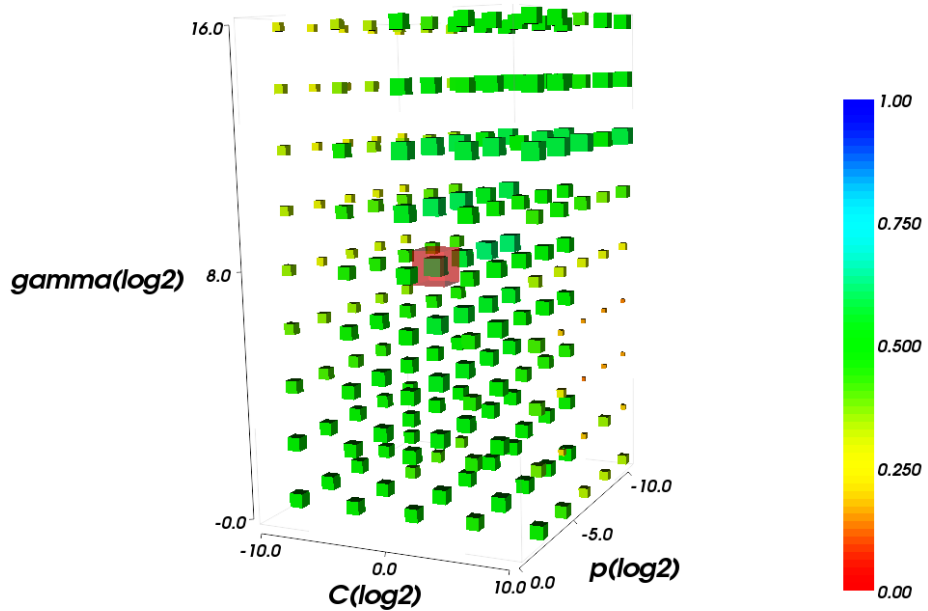


Figure 25. Plot of cross-validation rate against training parameters C , γ , and p for y-coordinate with RTT using ϵ -SVR.

CHAPTER 6

CONCLUSION

The goal was to develop a location estimation system capable of providing secure location information. To that end, this research has succeeded.

The system developed demonstrates that the secure architecture (Section 3.1) oriented around a central server is workable. Furthermore, the implementation is robust (Sections 5.3 and 5.4) and accurate, especially with the use of additional radio environment features (Section 5.7). SVRs proved to be accurate enough for the task (Section 5.7), often correctly identifying the location of the mobile node. The SVRs have the added advantage of being able to interpolate over un-surveyed regions (Section 5.4), though with limited accuracy, thus limiting the resolution of the system to that of the survey grid. However, attempts to utilize the RTT to improve the accuracy of system failed.

The shortcomings of the system include the inconvenient site survey and the long training time.

There are a number of ways the research can be extended. Future work can focus on the improvement of the estimation algorithms, the robustness of the system to various physical extremes, or increasing the security of the system.

As mentioned in Section 2.3.3, various algorithms can be used to provide the estimates. While [25] provides some comparison of their accuracy and algorithmic complexity, it does so without considering other practical concerns. Also, the system being implemented will not use time averaging, a common technique used to improve the accuracy of estimates[60, 36]. Time averaging can be incorporated into the algorithms as future work.

The training time for the SVRs was noted to be extremely long, most cases requiring approximately one day when training was performed on a small cluster, or several

days if performed on a single system. Each case also requires human intervention to interpret the results and determine the best set of parameters for the SVR. Furthermore, site survey and training will have to be done regularly as the radio environment changes, and each time the SVR will have to be retrained with the new data. The problem will be exacerbated as the number of sensors increase, which would increase the feature space and thus the training time for each iteration of training.

The most time-consuming portion would be the brute-force grid-search approach. There have been other methods proposed for SVR parameter tuning[51, 55], and these should be investigated further.

Another common method of addressing the issue of a large feature space is through the use of Support Vector Machine-Recursive Feature Elimination (SVM-RFE), a technique popular in bio-informatics where the feature space is in the thousands [61]. Further methods of optimizing SVMs are documented in [62], but these will have to be adapted slightly for use with SVRs.

Fortunately, the training time for SVRs decreases correspondingly with the size of the input data. Based on [37], the granularity of the site survey can be reduced significantly without much impact on estimation accuracy when SVRs are used. Reduced granularity would reduce the number of data points and thus the training time of the SVR.

From an implementation perspective, a promising alternative to LIBSVM is LIBSVMTL[63]. This library includes certain optimizations over LIBSVM but, for now, still lacks certain crucial functions.

Current systems use a map from a site survey done specifically for that purpose. This map is static and is never updated with new information. One possible research area would focus on updating this map based on new measurements of the mobile nodes done at run time. Similarly, the training time would be amortized over the extended training period, thus reducing the initial training time.

Few tests have been done on the accuracy of the map in the face of variations in wireless hardware. Commercial deployments would generally have to tolerate such variations, but there are no published results from commercial systems. More hardware configurations could be evaluated.

As mentioned in Section 4.5.2, the mobile nodes are currently identified using their MAC address. This is easily spoofed by attackers. Alternative methods of identifying the mobile nodes should be investigated.

APPENDIX A
SERVER GUI CONSOLE

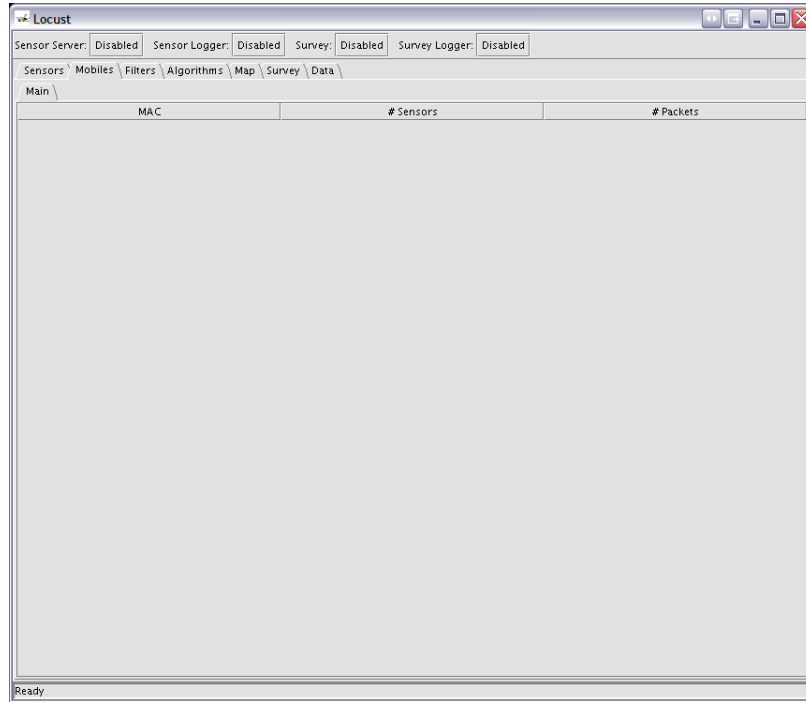


Figure 26. Sensor list.

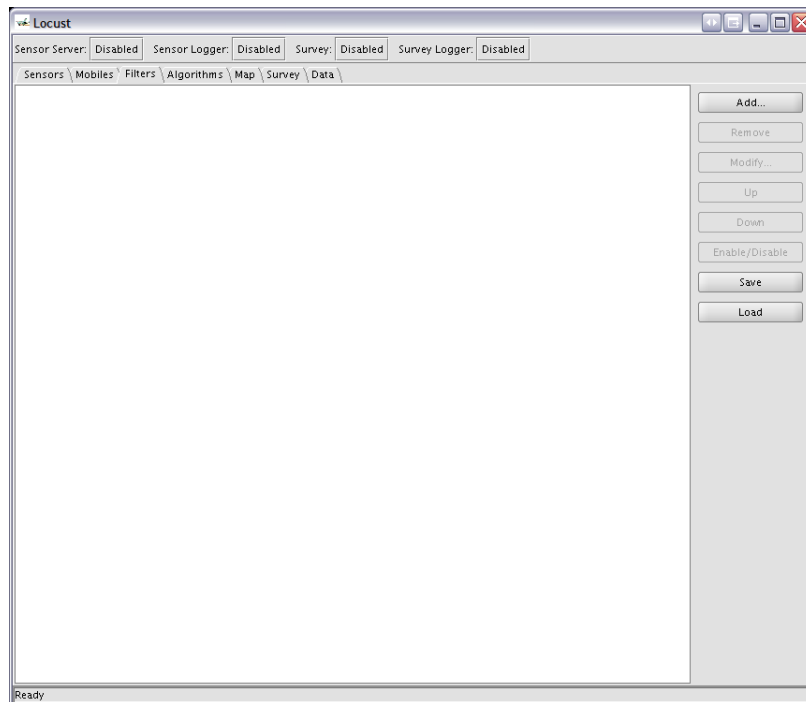


Figure 27. Sensor filters.

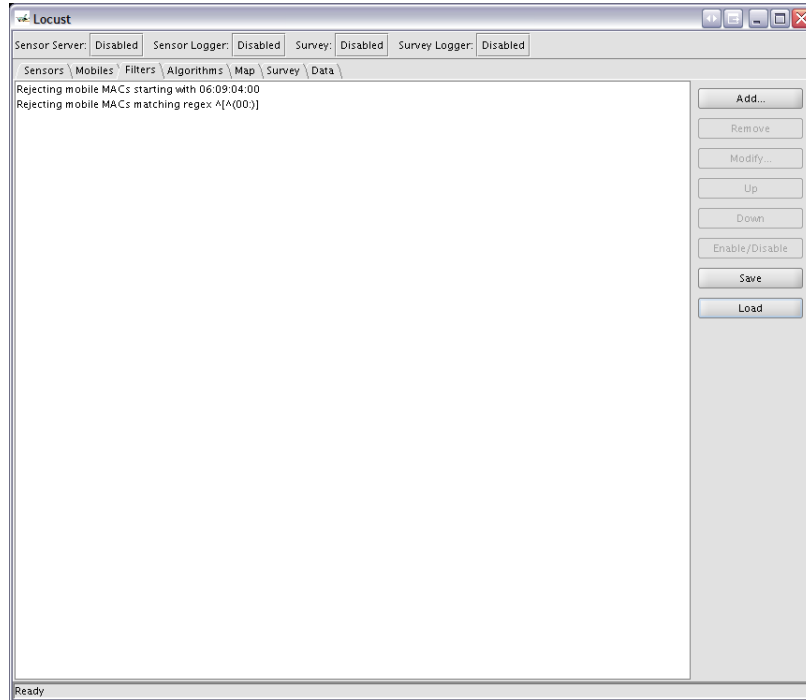


Figure 28. SVR training settings.

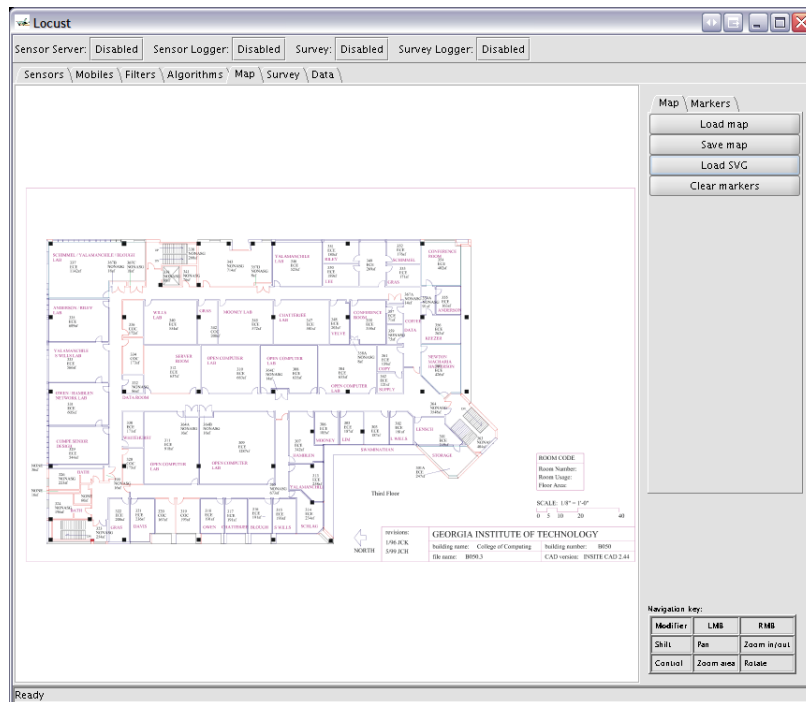


Figure 29. Map visualization for survey, training, and testing data.

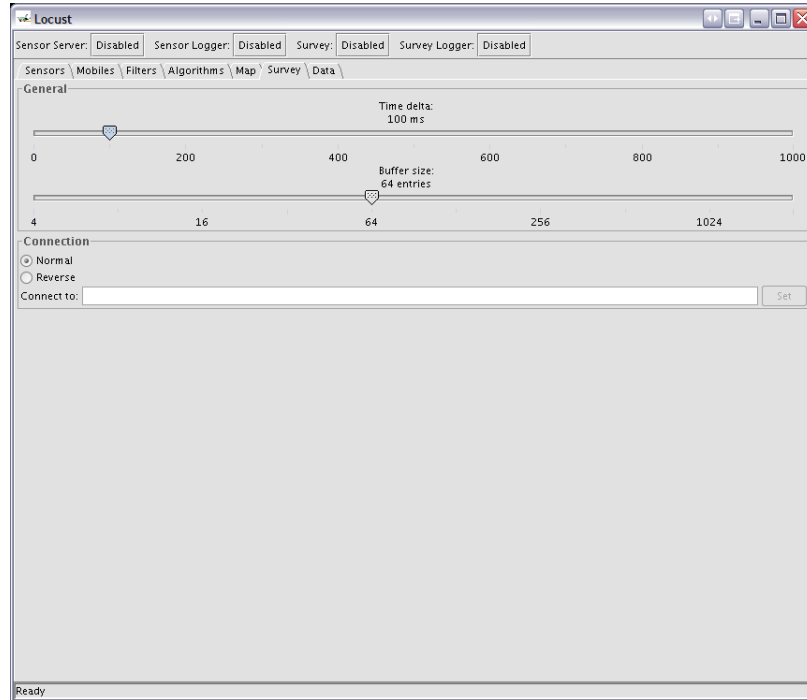


Figure 30. Sensor monitoring settings.

The screenshot shows the Locust application window displaying a table of raw sensor data. The table has the following columns: sample, name, ip, mobile, mactime, timestamp, signal, and noise. The data is as follows:

sample	name	ip	mobile	mactime	timestamp	signal	noise
1	linux6.eecom.ga...	/130.207.231.171	00.11.5ca1.c0.10	1818194974		183	163
2	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.bf.10	2535347909		180	172
3	wolf.eecom.gate...	/130.207.231.126	00.11.5ca1.b4.10	1767747869		179	169
4	wolf.eecom.gate...	/130.207.231.126	00.11.5ca1.b4.10	1767758972		182	169
5	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.bf.10	2535360036		183	171
6	ATMNT2.eecom....	/130.207.231.166	00.02.2d8a.15.d4	2535360859		177	171
7	ATMNT2.eecom....	/130.207.231.166	00.02.2d8a.15.d4	2535361418		177	171
8	ATMNT2.eecom....	/130.207.231.166	00.02.2d8a.15.d4	2535363385		177	172
9	ATMNT2.eecom....	/130.207.231.166	00.02.2d8a.15.d4	2535364108		177	172
10	ATMNT2.eecom....	/130.207.231.166	00.02.2d8a.15.d4	2535366252		177	172
11	ATMNT2.eecom....	/130.207.231.166	00.02.2d8a.15.d4	2535367156		177	172
12	linux6.eecom.ga...	/130.207.231.171	00.11.5ca1.be.90	1818221648		167	164
13	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.be.90	2535374221		174	171
14	wolf.eecom.gate...	/130.207.231.126	00.11.5ca1.be.90	1767773154		175	170
15	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.bf.10	2535377426		184	172
16	wolf.eecom.gate...	/130.207.231.126	00.11.5ca1.b5.80	1767776355		180	169
17	wolf.eecom.gate...	/130.207.231.126	00.11.5ca1.b4.10	1767776800		182	169
18	linux6.eecom.ga...	/130.207.231.171	00.11.5ca1.b4.70	1818238674		178	163
19	linux6.eecom.ga...	/130.207.231.171	00.11.5ca1.c0.10	1818253151		182	163
20	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.bf.10	2535409614		180	172
21	linux6.eecom.ga...	/130.207.231.171	00.11.5ca1.b5.80	1818260044		170	163
22	wolf.eecom.gate...	/130.207.231.126	00.11.5ca1.b5.80	1767811550		174	169
23	wolf.eecom.gate...	/130.207.231.126	00.11.5ca1.bf.30	1767812606		176	169
24	linux6.eecom.ga...	/130.207.231.171	00.11.5ca1.b4.70	1818274415		177	167
25	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.b6.10	2535426987		172	169
26	wolf.eecom.gate...	/130.207.231.126	00.11.5ca1.b4.10	1767827247		182	169
27	linux6.eecom.ga...	/130.207.231.171	00.11.5ca1.b4.70	1818276042		175	165
28	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.bf.10	2535429722		180	169
29	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.b6.10	2535431070		175	170
30	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.b6.10	2535432229		174	169
31	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.bf.10	2535434984		183	170
32	linux6.eecom.ga...	/130.207.231.171	00.11.5ca1.bf.10	1818282412		173	163
33	ATMNT2.eecom....	/130.207.231.166	00.02.2d8a.15.d4	2535464896		177	170
34	ATMNT2.eecom....	/130.207.231.166	00.02.2d8a.15.d4	2535466219		177	170
35	wolf.eecom.gate...	/130.207.231.126	00.02.2d8a.15.71	1767866391		173	178
36	ATMNT2.eecom....	/130.207.231.166	00.11.5ca1.be.90	2535476620		174	170

Figure 31. Raw sensor data table.

APPENDIX B

MAP OF SENSOR LOCATIONS

Three sensors have been deployed on the third floor of the College of Computing Building of Georgia Tech. The locations are indicated in the map in Figure 33.

215 feet (65.5 meters)

149 feet (45.3 meters)

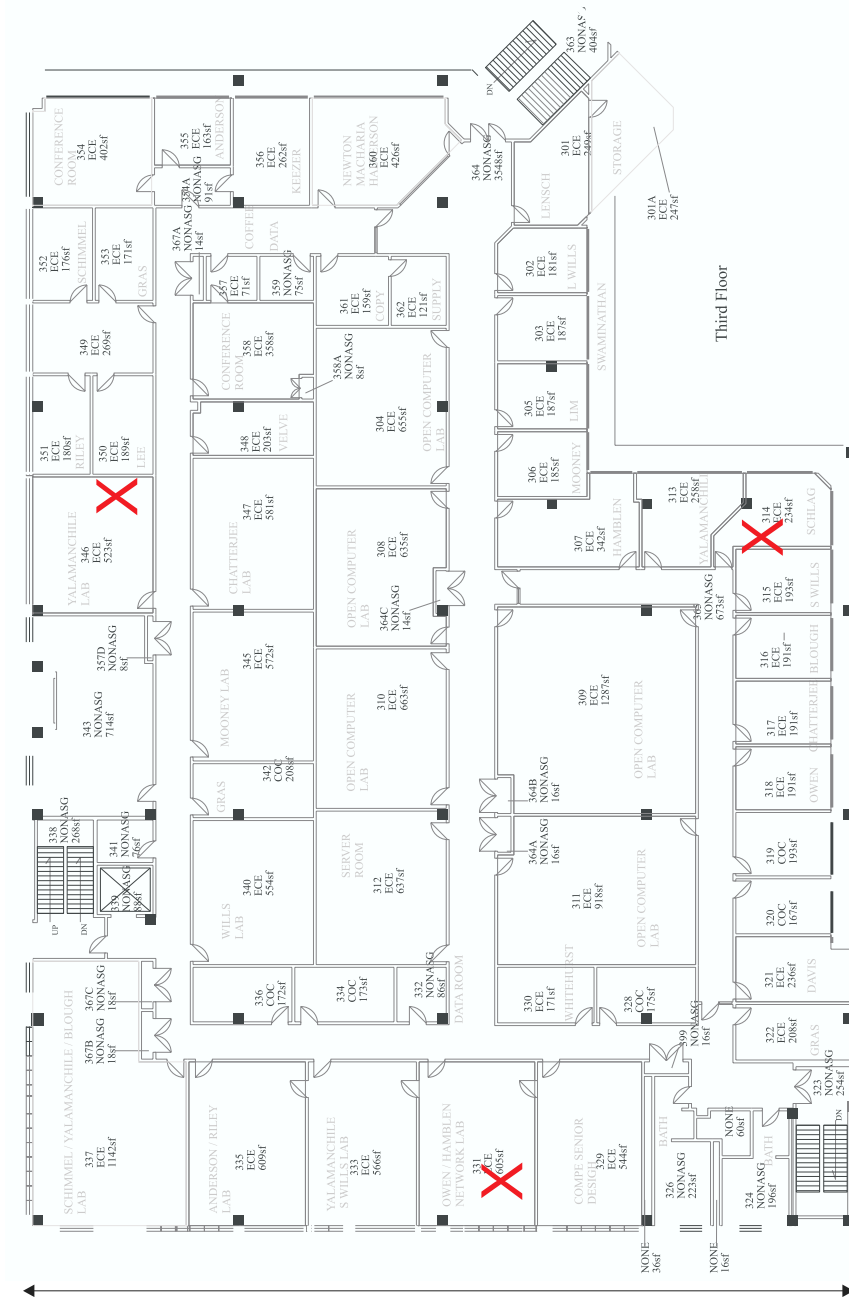


Figure 33. Location of sensors on third floor of the College of Computing building.

APPENDIX C

MAP OF LOCATIONS FOR SITE SURVEY

Survey readings were taken at six-foot intervals in areas that were accessible. While attempts were made to take readings at the same locations over each sampling interval, certain locations were inaccessible on occasion. Thus not all points were sampled equally.

LIST OF TERMS

Angle of Arrival (AOA)

4

Basic Service Set Identifier (BSSID)

33

Commercial off-the-Shelf (COTS)

15, 27

Clear-To-Send (CTS)

19, 36, 43

Denial of Service (DoS)

18

Federal Communications Commission (FCC)

3

Geographic Information System (GIS)

3

Global Positioning System (GPS)

See [10]. 3, 4

Hidden Markov Model (HMM)

7, 20

***k*-Nearest Neighbor (KNN)**

7, 20

LIBSVM

Programming library for Support Vector Machine (SVM)/Support Vector Regression (SVR) functions[53]. 27, 38, 59, 71

LIBSVMTL

Programming library for SVM/SVR functions, supposedly faster than LIBSVM[63].
59

Line of Sight (LOS)

4

Media Access Control (MAC)

29, 33, 36, 37, 60

Multiple In/Multiple Out (MIMO)

The draft IEEE 802.11n standard that uses multiple antennae to “shape” the radio signal to boost the signal strength and thus the transmission speeds. 6

Mean Squared Error (MSE)

A measure of the variance. $MSE = \sqrt{RMS}$. 39–42

Quadratic Programming (QP)

26

Radial Basis Function (RBF)

Also known as the Gaussian function. 23–25

Radio-Frequency IDentification (RFID)

13

Root-Mean-Square (RMS)

9, 39

Received Signal Strength (RSS)

Signal strength at the receiver. 4–7, 16, 17, 28, 40

Request-To-Send (RTS)

19, 36, 43

Round Trip Time (RTT)

6, 20, 21, 28, 29, 36, 37, 41, 43, 44

Sequential Minimal Optimization (SMO)

A faster alternative training method for SVMs[56, 57]. 26, 27

Secure SHell (SSH)

34

Support Vector Machine (SVM)

Statistical method used for pattern recognition and classification[64, 50]. 20, 23, 24, 38, 59, 71, 72

Support Vector Machine-Recursive Feature Elimination (SVM-RFE)

59

Support Vector Regression (SVR)

Statistical method used for regression analysis of multi-dimensional data sets[64].

1, 9, 23, 25, 27, 29, 37–44, 58, 59, 71

Time of Arrival (TOA)

4

Voice-over-IP (VoIP)

3

Wireless Local Area Network (WLAN)

2

REFERENCES

- [1] “ANSI/IEEE standard IEEE 802.11.” Online document, 1999. Available HTTP: <http://standards.ieee.org/getieee802/download/802.11-1999.pdf>.
- [2] M. S. Gast, *802.11 Wireless Networks: The Definitive Guide*. O’Reilly & Associates, April 2002.
- [3] D. E. Denning and P. F. MacDoran, “Location-based authentication: Grounding cyberspace for better security,” in *Computer Fraud & Security*, Elsevier Science Ltd., 1996.
- [4] K. Pahlavan, X. Li, and J.-P. Mäkelä, “Indoor geolocation science and technology,” *IEEE Communications Magazine*, vol. 40, pp. 112–118, February 2002.
- [5] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, “Cyberguide: A mobile context-aware tour guide,” *Wireless Networks*, pp. 421–433, 1997.
- [6] P. J. Brown, “The stick-e document: A framework for creating context-aware applications,” in *Electronic Publishing*, pp. 259–272, 1996.
- [7] “Herecast.” Website, August 2005. Available HTTP: <http://www.herecast.com/>.
- [8] “Skyhook wireless.” Website, August 2004. Available HTTP: <http://www.skyhookwireless.com/>.
- [9] J. Hightower, B. Brumitt, and G. Borriello, “The location stack: A layered model for location in ubiquitous computing,” in *WMCSA 2002*, pp. 22–28, IEEE, June 2002.
- [10] E. D. Kaplan, *Understanding GPS: Principles and Applications*. Artech House, 1996.
- [11] “FCC adopts rules to implement Enhanced 911 for wireless services.” Online document. Available HTTP: http://www.fcc.gov/Bureaus/Wireless/News_Releases/1996/nrwl6026.txt.
- [12] S. Li, G. Zhao, and L. Liao, “User location service over an 802.11 ad-hoc network.” Available HTTP: <http://www.cs.washington.edu/homes/liaolin/Courses/networks02.pdf>, December 2002.
- [13] S. Čapkun, M. Hamdi, and J.-P. Hubaux, “GPS-free positioning in mobile ad-hoc networks,” in *34th Annual Hawaii International Conference on System Sciences*, IEEE, January 2001.

- [14] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O’Dea, “Relative location estimation in wireless sensor networks,” *Transactions on Signal Processing*, vol. 51, August 2003.
- [15] D. Niculescu and B. Nath, “Ad hoc positioning system (APS) using AOA,” in *INFOCOM 2003*, vol. 3, IEEE, March 2003.
- [16] N. Sundaram and P. Ramanathan, “Connectivity based location estimation scheme for wireless ad hoc networks,” in *GLOBECOM 2002*, vol. 1, pp. 143–147, IEEE, 2002.
- [17] “Airdefense.” Website, August 2004. Available HTTP: <http://www.airdefense.net/>.
- [18] “Aeroscout.” Website, August 2004. Available HTTP: <http://www.aeroscout.com/>.
- [19] “Ekahau.” Website, August 2004. Available HTTP: <http://www.ekahau.com/>.
- [20] A. Hatami and K. Pahlavan, “In-building intruder detection for WLAN access,” in *Position Location and Navigation Symposium, 2004. PLANS 2004*, pp. 592–597, 2004.
- [21] J. Small, A. Smailagic, and D. P. Siewiorek, “Determining user location for context aware computing through the use of a wireless LAN infrastructure.” Available HTTP: <http://www-2.cs.cmu.edu/~aura/docdir/small100.pdf>, December 2000.
- [22] A. M. Ladd, K. E. Bekris, G. Marceau, A. Rudys, L. E. Kaviraki, and D. S. Wallach, “Robotics-based location sensing using wireless ethernet,” in *MOBICOM*, pp. 227–238, ACM, September 2002.
- [23] M. Berna, B. Sellner, B. Lisien, S. Thrun, G. Gordon, and F. Pfenning, “A learning algorithm for localizing people based on wireless signal strength that uses labeled and unlabeled data.” Available HTTP: <http://www-2.cs.cmu.edu/~fp/papers/wireless03.pdf>, 2003.
- [24] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz, “A probabilistic room location service for wireless networked environments,” in *UBICOMP*, pp. 18–34, ACM, 2001.
- [25] R. Battiti, M. Brunato, and A. Villani, “Statistical learning theory for location fingerprinting in wireless LANs,” tech. rep., University of Trento, October 2002. Available HTTP: <http://eprints.biblio.unitn.it/archive/00000238/01/86.pdf>.
- [26] S. Ganu, A. S. Krishnakumar, and P. Krishnan, “Infrastructure-based location estimation in WLAN networks,” in *WCNC 2004*, IEEE, 2004.

- [27] M. Youssef and A. Agrawala, "Small-scale compensation for WLAN location determination systems," in *WCNC 2003*, IEEE, March 2003.
- [28] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *INFOCOM 2004*, IEEE, 2004.
- [29] Y. Gwon, R. Jain, and T. Kawahara, "Robust indoor location estimation of stationary and mobile users," in *INFOCOM 2004*, IEEE, 2004.
- [30] J. Yin, Q. Yang, and L. Ni, "Adaptive temporal radio maps for indoor location estimation," in *PERCOM 2005*, IEEE, 2005.
- [31] M. Brunato and C. K. Kalló, "Transparent location fingerprinting for wireless services," in *Med-Hoc-Net 2002*, 2002.
- [32] T. Tonteri, "A statistical modeling approach to location estimation," Master's thesis, University of Helsinki, May 2001.
- [33] A. Neskovic, N. Nescovic, and G. Paunovic, "Modern approaches in modeling of mobile radio systems propagation environments," *Communications Surveys*, 2000.
- [34] G. V. Záruba, M. Huber, and F. A. Kamangar, "Monte carlo sampling based in-home location tracking with minimal RF infrastructure requirements," in *GLOBECOM 2004*, vol. 6, pp. 3624–3629, IEEE, 2004.
- [35] A. Günther and C. Hoene, "Measuring round trip times to determine the distance between WLAN nodes," tech. rep., TKN, December 2004.
- [36] M. Youssef and A. K. Agrawala, "Continuous space estimation for WLAN location determination systems," in *ICCCN*, no. 13, IEEE, October 2004.
- [37] J. Krumm and J. Platt, "Minimizing calibration effort for an indoor 802.11 device location measurement system," in *NIPS 2003*, 2003.
- [38] N. Michalakis, "PAC: Location aware access control for pervasive computing environments." Available HTTP: <http://www.org.lcs.mit.edu/pubs/michalakis.pdf>, September 2002.
- [39] J. E. Bardram, R. E. Kjær, and M. Ø. Pedersen, "Context-aware user authentication: Supporting proximity-based login in pervasive computing," in *UBICOMP 2003*, 2003.
- [40] M. Wallbaum and P. Dornbusch, "Design considerations for a platform supporting location-aware services," in *MIV 2001*, 2001.
- [41] M. A. Youssef and A. Agrawala, "On the optimality of WLAN location determination systems," in *CNDS 2004*, SCS, January 2004.

- [42] H. Srivasta, "Location, location-based services." Website, December 2004. Available HTTP: <http://www-106.ibm.com/developerworks/wireless/library/wi-loc/>.
- [43] M. Wallbaum, "Wheremops: An indoor geolocation system," in *International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, September 2002.
- [44] P. Bahl, V. N. Padmanabhan, and A. Balachandran, "A software system for locating mobile users: Design, evaluation, and lessons," online document, Microsoft Research, February 2000. Available HTTP: <http://research.microsoft.com/~padmanab/papers/radar.pdf>.
- [45] M. Youssef, A. Agrawala, and U. Shankar, "WLAN location determination via clustering and probability distributions," in *PerCom 2003*, IEEE, March 2003.
- [46] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *INFOCOM (2)*, vol. 2, pp. 775–784, IEEE, March 2000.
- [47] Y.-X. Lim, T. Schmoyer, J. Levine, and H. Owen, "Wireless intrusion detection and response," in *Proceedings of the 4th Annual Information Assurance Workshop*, June 2003.
- [48] T. Schmoyer, Y.-X. Lim, and H. Owen, "Wireless intrusion detection and response: A case study using the classic man-in-the-middle attack," in *Proceedings of Wireless Communications and Networking Conference*, 2004.
- [49] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *Proceedings of the 12th USENIX Security Symposium*, USENIX, 2003.
- [50] M. A. Hearst, "Trends & controversies: Support vector machines," *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 18–28, 1998.
- [51] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification." Online document, July 2003. Available HTTP: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>.
- [52] V. Vapnik, *Statistical Learning Theory*. Wiley, 1998.
- [53] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [54] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for svm regression," *Neural Networks*, vol. 17, pp. 113–126, January 2004.

- [55] F. Friedrichs and C. Igel, “Evolutionary tuning of multiple svm parameters,” in *Proceedings of the 12th European Symposium on Artificial Neural Networks*, 2004.
- [56] J. C. Platt, “Sequential minimal optimization: A fast algorithm for training support vector machines,” online document, Microsoft Research, April 1998. Available HTTP: <http://www.research.microsoft.com/~jplatt/smoTR.pdf>.
- [57] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization.” Online document, 2000. Available HTTP: <http://research.microsoft.com/~jplatt/smo-book.pdf>.
- [58] J. Malinen, “Host AP driver for Intersil Prism2/2.5/3 and WPA supplicant.” Website, January 2005. Available HTTP: <http://hostap.epitest.fi/>.
- [59] L. D. Harvel, L. Liu, G. D. Abowd, Y.-X. Lim, C. Scheibe, and C. Chatham, “Context cube: Flexible and effective manipulation of sensed context data,” in *Proceedings of the 2nd International Conference on Pervasive Computing*, pp. 51–68, 2004.
- [60] M. Youssef and A. Agrawala, “Handling samples correlation in the Horus system,” in *INFOCOM 2004*, IEEE, October 2004.
- [61] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.
- [62] C. J. C. Burges and B. Schölkopf, “Improving the accuracy and speed of support vector machines,” in *Advances in Neural Information Processing Systems*, vol. 9, p. 375, The MIT Press, 1997.
- [63] O. Ronneberger, *LIBSVM TL — a Support Vector Machine Template Library*, 2004. Software available at <http://lmb.informatik.uni-freiburg.de/lmbsoft/libsvm/tml/index.en.html>.
- [64] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.