AN INDOOR GEO-FENCING BASED ACCESS CONTROL SYSTEM FOR WIRELESS NETWORKS

by

Hossein Rahimi

Submitted in partial fulfillment of the
requirements for the degree of
Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
July 2013

Table of Contents

List of Tables . . . . **iv**

List of Figures . . . . **v**

Abstract . . . . **vii**

List of Abbreviations and Symbols Used . . . . **viii**

Acknowledgements . . . . **xi**

Chapter 1 Introduction **1**

Chapter 2 Literature Survey **5**

2.1 Indoor Positioning 5

2.2 Indoor Positioning Security 10

Chapter 3 Methodology **12**

3.1 System Overview 12
3.1.1 Network Services 12
3.1.2 Web and Mobile Applications 17

3.2 The Proposed Positioning Algorithm 19
3.2.1 Positioning Data 20
3.2.2 C4.5 Decision Tree 23
3.2.3 Random Forest 25
3.2.4 Client Status Manager 26

3.3 Generic Access Control Mechanism 28

Chapter 4 Noise, Outliers and Missing Values **29**

4.1 Smoothing Techniques 29
4.1.1 Moving Average Method 30
4.1.2 Locally Weighted Regression Scatter Plot Smoothing 31
4.1.3 Savitzky-Golay Method 31
4.1.4 Robust LOESS and LOWESS Algorithms 32

Chapter 5 Security **33**

5.1 Communication Security 33

5.2 Attacks and Countermeasures 34
5.2.1 Brute Force and Token Guessing 35
5.2.2 Denial of Service 37
5.2.3 Man in The Middle 39
5.2.4 Spoofing 40

Chapter 6 Experiments and Results **42**

6.1 Classification Results 42

6.2 Smoothing Results 44

6.3 Most Important Anchor Nodes 50

6.4 Best Anchor Node Positions 52

6.5 Brute Force and Spoofing Experiments 54

6.6 Improving The Smoothing Process 57

Chapter 7 Conclusions and Future Works **59**

Bibliography **60**

Appendix A Result Tables and Figures **66**

A.1 Tables 66

A.2 Figures 78

List of Tables

3.1 Attributes for the collected data sets. 21

6.1 Geo-fencing data set description. 42

6.2 Top 10 anchor node positions for the 5x5 geo-fenced zone while using the Bluetooth infrastructure for positioning. 53

A.1 All the best classification results achieved by runs on the original data. In these runs [0,1] normalization is also tested. 66

A.1 All the best classification results achieved by runs on the original data. In these runs [0,1] normalization is also tested. 69

A.2 Detailed classification result averages for Random Forest on all data sets using different smoothing methods, sorted based on $\mathrm{F}_{Measure}$. $Window_{Size}$ is set to 10 for all runs. 70

A.2 Detailed classification result averages for Random Forest on all data sets using different smoothing methods, sorted based on $\mathrm{F}_{Measure}$. $Window_{Size}$ is set to 10 for all runs. 72

A.3 Detailed classification results for different $Window_{Size}$ values on all data sets. 73

A.3 Detailed classification results for different $Window_{Size}$ values on all data sets. 77

List of Figures

3.1 Authenticating to a registered application using CAS. 14

3.2 Registration process of a typical customer. 15

3.3 Session initiation and positioning based on proximity based transfer of the value $Salt_{Session}$. 16

3.4 Screenshots from the monitoring user interface. 18

3.5 Sample of the Bluetooth and Wi-Fi data sets. 21

3.6 A floor plan of the experimentation area. 22

3.7 A visualization of the C4.5 tree trained on Bluetooth data. 23

3.8 Client Status Manager's structure and data buffers. 27

6.1 Random Forest result distribution on non-smoothed data sets while using the Wi-Fi infrastructure. 45

6.2 Random Forest result distribution on non-smoothed data sets using the Bluetooth infrastructure. 46

6.3 C4.5 decision tree result distribution on non-smoothed data sets while using the Wi-Fi infrastructure. 47

6.4 C4.5 decision tree result distribution on non-smoothed data sets while using the Bluetooth infrastructure. 48

6.5 Classification results for different Bluetooth data sets vs. changing the moving average $Window_{Size}$. 49

6.6 Classification results for different Wi-Fi data sets vs. changing the moving average $Window_{Size}$. 49

6.7 Random Forest result distribution on Wi-Fi 5x5 data set collected using an Asus Tablet, using different Smoothing methods. 50

6.8 Random Forest result distribution on Bluetooth 5x5 data set collected using a Samsung Smartphone, using different Smoothing methods. . . 51

6.9 The first and the second most important anchor nodes. 52

6.10 Random Forest results considering different positions of anchor nodes on a 10x10 Bluetooth zone. 54

6.11 Trend of $Growth_{rate}$ for RSSI Spoofing, Brueforcing ,and a regular users in a 2x2 zone monitored using Bluetooth RSSI values. 56

A.1 Random Forest result distribution on Wi-Fi 2x2 data set collected using a variety of devices. 78

A.2 Random Forest result distribution on Wi-Fi 2x2 data set collected using an Asus Tablet. 79

A.3 Random Forest result distribution on Wi-Fi 5x5 data set collected using a Samsung Smartphone. 80

A.4 Random Forest result distribution on Wi-Fi 10x10 data set collected using a Samsung Smartphone. 81

A.5 Random Forest result distribution on Wi-Fi 10x10 data set collected using an Asus Tablet. 82

A.6 Random Forest result distribution on Bluetooth 2x2 data set collected using an Asus Tablet. 83

A.7 Random Forest result distribution on Bluetooth 2x2 data set collected using a Samsung Smartphone. 84

A.8 Random Forest result distribution on Bluetooth 10x10 data set collected using a Samsung Smartphone and an Asus Tablet. 85

A.9 Random Forest result distribution on Bluetooth 10x10 data set collected using a Samsung Smartphone. 86

A.10 Random Forest result distribution on Bluetooth 10x10 data set collected using an Asus Tablet. 87

A.11 $F_{Measure}$ vs. smoothing $Window_{Size}$ using Moving Average for all Bluetooth data sets. 88

A.12 $F_{Measure}$ vs. smoothing $Window_{Size}$ using Moving Average for all Wi-Fi data sets. 89

Abstract

Use of wireless network information for indoor positioning has been an area of interest since wireless networks became very popular. On the other hand, the market started to grow in variety and production volumes leading to a variety of devices with many different hardware and software combinations. In the field of indoor positioning, most of the existing technologies are dependent on additional hardware and/or infrastructure, which increases the cost and requirements for both users and providers.

This thesis investigates possible methods of coupling indoor geo-fencing with access control including authentication, identification, and registration in a system. Moreover, various techniques are studied in order to improve the robustness and security of such a system. The focus of these studies is to improve the proposed system in such a way that gives it the ability to operate properly in noisy, heterogeneous, and less controlled environments where the presence of attackers is highly probable. To achieve this, a classification based geo-fencing approach using Received Signal Strength Indicator (RSSI) has been employed so that accurate geo-fencing is coupled with secure communication and computing. Experimental results show that considerable positioning accuracy has been achieved while providing high security measures for communication and transactions. Favouring diversity and generic design, the proposed implementation does not mandate users to undergo any system software modification or adding new hardware components.

List of Abbreviations and Symbols Used

RSSI Received Signal Strength Indicator

GPS Global Positioning System

RFID Radio Frequency Identification

BYOD Bring Your Own Device

RIM Research In Motion

Wi-Fi Wireless Fidelity

DoS Denial of Service

MiTM Man in The Middle

UWB Ultra Wide Band

GSM Global System for Mobile

SNR Signal to Noise Ratio

CDMA Code Division Multiplex Access

WEP Wired Equivalent Privacy

WPA/WPA2 Wi-Fi Protected Access

TCP Transport Control Protocol

IP Internet Protocol

EAP-TLS Extensible Authentication Protocol based on Transport Layer Security

JOSSO Java Open Single Sign On

CAS Central Authentication Service

LDAP Lightweight Directory Access Protocol

RADIUS Remote Authentication Dial In User Service

SPNEGO Simple and Protected GSSAPI Negotiation Mechanism

GSSAPI Generic Security Services Application Program Interface

OAuth Open Authentication

MAC Media Access Control

IMEI International Mobile Station Equipment Identity

QR-Code Quick Response Code

NFC Near Field Communication

HTTP Hypertext Transfer Protocol

SSL Secure Socket Layer

RESTful Representational state transfer

TDoA Time Difference of Arrical

AoA Angle of Arrival

CSM Client Status Manager

FIFO First In First Out

GHz Gigahertz

ISM Industrial, Scientific and Medical radio bands

SSH Secure Shell

TLS Transport Layer Security

MAC Message Authentication Code

L2CAP Logical Link Control and Adaption Protocol

ToS Terms of Service

TP True Positive Prediction Count

TN True Negative Prediction Count

FP False Positive Prediction Count

FN False Negative Prediction Count

TP_{ratio} The percentage of positive predictions that are correct.

FP_{ratio} The percentage of positive predictions that are incorrect.

TN_{ratio} The percentage of negative predictions that are correct.

FN_{ratio} The percentage of negative predictions that are incorrect.

$\hat{y}$ Average signal value

$\bar{y}_{t+1}$ Next Moving Average Estimate

σ Standard Deviation

Acknowledgements

First and foremost, I would like to thank Professor Nur Zincir-Heywood for her careful supervision, scientific guidance, and devotion to keep me on track to achieve the gift of success.

I am also proud to thank my family. Especially my parents, who brought me up to dream, believe, work hard, and to be a free man. For whatever I have had and will achieve in my life, I owe a great deal of that to them.

I also wish to thank the people at Mitacs who, together with my supervisor, created the opportunity for conducting applied research during my studies. I greatly appreciate Susan Michel's efforts, who helped the industrial partnership stay coordinated and to advance it under the Mitacs Accelerate program.

Finally, all the best to the members of the Network Information Management and Security Lab. They not only patiently tolerated the presence of flashing lights of anchor nodes around the lab and above their heads, but also greatly helped in the early data collection efforts.

> " Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains."
> —Steve Jobs

Chapter 1

Introduction

Knowledge about geographical location of a mobile device or indirectly its owner can be of enormous utility. Geo-spatial information is being used in many fields such as computer software, physical security, in addition to location aware marketing and advertisement. In this context, most of the existing technologies focus on identifying the exact location of the user via Global Positioning System (GPS) in outdoor environments. A different view of the location aware computing is to focus on the presence of a user in a virtual perimeter of a given geographical landscape. This second alternative view, which complements the first one is called Geo-fencing and has brought in many benefits and also challenges to the location based computing field.

As stated before, the term “Geo-fence” refers to a virtually fenced geographical area. This concept has been employed to implement various tasks including equipment theft control, transportation path control [65], asset management and tracking, or automatic house arrest monitoring systems. Social networks have also brought new ideas and use cases for Location Based Services, including geo-spatial networking. Targeted and location aware marketing and advertisement is also an interesting use case where promotions are sent to potential customers based on the opportunities associated with the geographical location they visit.

The GPS technology, providing an accurate positioning method, has made the outdoor location based services and geo-fencing conveniently accessible. GPS has become a popular sensor chip available on nearly all computationally powerful smartphone and tablets in the market. Navigation and location based search in outdoor areas coupled with aerial imaging and accurate mapping of urban and rural areas has brought many applications to the hand-held devices. However, the GPS technology is not completely flawless. The low bit rate of the satellite connection transmitting the timing data required for positioning makes the task of accurate positioning slow and impossible in some cases such as indoor environments. Additionally, due to required Line of Sight connection with the satellite infrastructure, the technology has not been as successful in indoor setups.

Another major difference between outdoor and indoor geo-fencing is the amount of tolerable positioning error. While in an outdoor geo-fence a dozen of meters in error might be tolerable, even half of this amount of error might result in a complete failure in an indoor setup. To bring the concept of positioning indoors, different approaches have been proposed [57]. Most of these approaches are based on wireless technologies for tracking tagged objects. It means that every user/object that is tracked needs to carry a tag device such as a tagged Smartphone or a bracelet. Many tag based approaches are based on Radio Frequency Identification (RFID) chips, which might require a proximity based sensing procedure that adds to the burden for the users. Some of the special tags carry additional and unpopular sensors such as ultrasonic or infrared to provide with more accurate positioning data.

Furthermore, the emerging phenomenon of Bring Your Own Device (BYOD) has a powerful impact on how the market is demanding newer approaches to deal with the resulting chaos [74]. With the prevalent use of mobile devices, it is difficult for industries with a controlled environment to manage employees and customers interacting with on-site services using non-corporate devices. Instead of excessively limiting users, it is preferred for industries to adapt to the situation by taking more intelligent and device/user aware service providing approaches. Considering the limitations imposed by the task of tagging every new user's device, the simple tag based world of indoor location based services is changing to a heterogeneous environment. This environment not only introduces challenges regarding location estimation, it also affects the way multiple software components have to be implemented for different mobile and tablet operating systems to support such a technology.

Implementations should be based on a set of widely available data values that is provided by devices out of the box. The most popular wireless technologies available with each handheld device are Bluetooth and Wi-Fi. As of now, Google's Android, RIM's BlackBerry, and Apple's iOS mobile operating systems are the most popular operating systems as they are dominating the mobile device market; with Android having a higher speed in growing its market share [36].

This thesis proposes a Received Signal Strength Indication (RSSI) based system for identification and access control in a geo-fenced indoor environment. Such a system is important for certain services where only the devices in the geo-fenced zone are authorized to use a service. The RSSI values are obtained from Bluetooth and/or Wi-Fi infrastructures available on-site. Using the existing wireless network infrastructure and Android devices, a proof of concept implementation of this proposed system was developed and tested. In this system, as a case study, access control to an online application is performed based on a user's presence in a restricted area, or geo-fenced zone. Altogether, the system aims to introduce an indoor geo-fencing methodology that also aims to address the concerns brought to network administrators by the BYOD phenomenon.

This system was tested using multiple types of devices and various geo-fenced zone dimensions on both Wi-Fi and Bluetooth infrastructures. Devices include different Android based tablet and smartphones. Results show that an accurate geo-fencing methodology has been built where accuracy is up to 100% in most cases, and 93.1% in average. The average can be improved even more when selected zone sizes are larger and Bluetooth infrastructure is used.

The prototype implementation of the proposed system is tested in several scenarios including in noisy environments and under network attacks. According to the behaviour of the system and possible attack and noise conditions, the system seems to meet the robustness and security requirements of such indoor environments. To eliminate the negative effect of noise and erroneous RSSI readings, several smoothing and outlier detection techniques are evaluated and benchmarked to choose the best method. Based on our evaluations, Moving Average, is chosen as the best smoothing technique and then tuned up for the best performance based on a separate set of experiments focusing on parameters and configurations of this specific technique.

Known network and system attacks such as Denial of Service (DoS), Password Guessing, Man in the Middle (MitM) attacks, and spoofing are studied on the proposed system. Additionally, system specific attacks such as RSSI value guessing, infrastructure faking, and RSSI value spoofing are also simulated and evaluated on the proposed system. Based on these experiments and evaluations, defense mechanisms are developed and integrated with the present implementation of the proposed indoors geo-fencing system to effectively tackle the aforementioned attacks and bring sufficient amount of safety to the indoor geo-fencing based access control environment.

A US provisional patent [49] application was filed by our industrial partner, describing anatomy of the proposed system. A paper describing the proof of concept implementation and experimental results was presented and published in proceedings of the IEEE Computing Intelligence in Cyber Security conference 2013 [64].

The rest of this thesis will discuss and elaborate on different aspects of the system design and implementation choices made for the proposed system. Chapter 2 presents the existing industrial and academic systems reported in the literature. Chapter 3 discusses the system design and the employed positioning technique in detail. Chapters 4 and 5 focus on the analysis of the effect of environmental noise and security threats as well as the techniques to counter each condition, respectively. Chapter 6 presents the experimental results and evaluations performed. Finally, Chapter 7 concludes the thesis and proposes areas for future research directions.

Chapter 2

Literature Survey

The work discussed in this thesis has two major phases: 1) Creating an indoor positioning framework, and 2) Improving the proposed system for better robustness and security characteristics . As a result, the related literature and industrial works will be presented in Sections 2.1 and 2.2. These sections discuss related works in indoor positioning and efforts in securing and creating robust systems, respectively.

2.1 Indoor Positioning

The third millennium is when wireless technologies started to be vastly developed and employed. Bluetooth and Wi-Fi are both the children of 2000s. As of 2002, Bluetooth v1.1 [3] was introduced as a standard. While Wi-Fi was defined in early 1990's, and later clarified in 1999, the prevalence of its usage started growing since early 2000s [1].

As the wireless technology started growing, with every available wireless technology ranging from Bluetooth, Wi-Fi to other technologies such as Global System for Mobile (GSM), Ultra Wide Band (UWB), Ultra High Frequency (UHF), and RFID tags, there have been researches on positioning of devices undertaken by academicians and industry [57]. In many cases the creators of such techniques have sought the aid of secondary positioning techniques such as infrared and ultrasonic sensors to increase the accuracy of their proposed systems [51, 58].

One of the earliest works in wireless based indoor positioning is Radar by Microsoft Research [27]. Radar uses a WaveLAN [76] network driver under the FreeBSD environment that allows collection of data sets with more information such as RSSI and Signal to Noise Ratio (SNR). Using overlapping areas of WaveLAN network access point coverages, they have provided a method based on empirical analysis, triangulation, and attenuation noise modelling to position the users in indoor environments. The accuracy of this system is between 2 and 3 meters. Horus [86] uses joint clustering techniques to convert the indoor area into tiles, and then locate each device relative to those tiles with a higher clustering performance. The idea of associating a subset of visible access points to an area/tile of the map is the technique they use as the basis of their indoor positioning approach. First a model is built based on association of access points and areas of an indoor map to respective ranges of signal strength values. Then the model is used to cluster the RSSI data received from users into access point subsets and then locations. Not giving any explanation about the noise in the experimental environment, they have achieved an accuracy of about 90% in distances above 2.1 meters. Miura et al. [60], and Chang et al. [41] have used Support Vector Machines (SVM) to classify RSSI data samples for localizing the nodes in a Wi-Fi testbed. They have used the same homogeneous wireless hardware to classify the presence of a user in a 2x2 meter square shaped zone. Clearing the area of other wireless signals, they have achieved 100% accuracy when using obstacles (e.g. walls) to separate the zone from outside areas, which yields a larger amount of signal attenuation. Their data set includes unknown number of instances sampled in 21 symmetric predefined locations of the area. Likewise, Castro et al. [40] use SVM and also triangulation to position a node inside or outside a given zone. However, not only they do not have a detailed error investigation in their report, they also have static formation of anchor nodes and a homogenous hardware environment. Furthermore, in [68] and [40], researchers have used machine learning algorithms and probabilistic models while using fine grids of Wi-Fi access points to locate wireless devices. They have achieved an accuracy of about 1.5 meters with about 50% of the samples. Their samples are collected in 270 fixed locations, they have used 8 Wi-Fi access points to extract coordinates of the users in a 16 by 40 meters office area. The environment noise is not discussed, but it is mentioned that the test area includes glass, concrete and wooden obstacles. In [30] and [69], researchers have taken the approach of employing Artificial Neural Networks (ANN) to determine the location of users, their results show an error of over 1 meter in most cases. [30] uses 3 access points with unequal transmission powers, associating the data collected with the experimentation area floor plan they can locate the users. With minimum of 5 data points sampled at the training time, they have managed to locate test samples with errors of 3 meters or above. Due to the unbalanced power of access points and their arbitrary placing, they have discussed suffering from missing values in certain blind spots of the experimentation area.

There are a handful of research works focusing on Wireless Sensor Networks since 2004.

Works like [77], [84], and [32] have used triangulation in short range wireless sensor networks like ZigBee [22] to perform location based tasks. Specifically, [32] focuses on selecting anchor nodes in between the moving objects to do relative positioning. Wightman et al. [82] and also [87], have coupled the classification and triangulation techniques with Kallman Filtering to smooth down or predict spontaneous behaviours of wireless sensor networks that are due to noise and hardware failures.

On the other hand, only a few works have tried to experiment Bluetooth for indoor positioning. Among them [23] is worth mentioning. To achieve the two goals they set, the authors have used multiple Neural Networks. First goal was dealing with noisy samples resulted by real world environments. The second goal was recovering from access point or anchor node failures. They have also tried their system with different Bluetooth hardware, but never tried to tune their system to work better with different devices.

Recently, Galvan et al. [73] and Baniukevic et al. [29] have used a combination of Bluetooth and Wi-Fi anchor nodes to implement hybrid indoor positioning systems. The work introduced in [73], uses different combinations of Bluetooth and Wi-Fi reference points to estimate the position of a user using trilateration and multilateration. They have established a simulation system based on 400 base data points and have ran simulations with different attenuation factors virtually achieving sub-meter positioning accuracy. In [29], the authors have experimented with the addition of a few Bluetooth hotspots to the present Wi-Fi infrastructure. This divides a building floor into certain number of regions. The short range hotspots increase the accuracy of a position system that approximates a device's location based on the closest access point or hotspot, similar to a proximity based approach using RFIDs. Their results based on a series of simulations has an average error of about 2 meters in the best case. Wang et al. [79] have employed bayesian filtering and simulated annealing to position users. They have run simulations that is calibrated using 5,460 data samples and is then tested using about 10,000 points. Their achievement is to reduce the positioning error from 4 meters to 2.9 meters. In many cases experiments are undertaken only using simulations, and also the scale of sampling and eventual use of additional technologies is what makes deployment of such techniques costly and less generic.

There are other works like [78] that use RSSI values, not for locating devices but for detecting obstacles and passing bodies. Ironically, they use fluctuations in installed wireless sensor infrastructure to detect the motion or position of obstacles that move in between the sensors that are also anchor nodes. One application mentioned by that work was locating persons working in a company or for advanced anti theft and motion detection systems.

Kontkanen el al. [54], introduced a Wi-Fi based indoor positioning system based on “a combination of Bayesian networks, stochastic complexity and online competitive learning”. Later, this research led to the establishment of a commercial product called “Ekahau”. Although the research paper does not investigate the error of the system deeply, the official Ekahau website [9] refers to room level accuracy for their product. It is also claimed that the accuracy can be increased to bed level accuracy in an exemplary medical facility. However, Infrared sensors are being used [9] in order to achieve such accuracy. Needless to say, the infrared beacons can only be used in combination with the company provided tags. Moreover, Ekahau also claims that their system can locate every wireless mobile device. However, there is still no official version of the product in order to support tracking wireless devices in a generic manner. Ekahau is supporting a collection of wireless devices only by installing a custom company provided wireless driver. The driver gives the device the ability of being used in the site survey process, and not in the tracking process.

GloPos [10] is a GSM/CDMA (Code Division Multiplex Access) cellular network based commercial positioning product. That only uses information from cell towers to estimate the location of mobile devices. The system accuracy is referred to as being 10 to 40 meters in suburban, urban and indoor areas. Moreover, they have claimed a 7.7 to 12.5 meters accuracy being achieved during an independent test [11]. However, the provided test report suggests that the referred accuracy is achieved in less than 75% of the test cases, and the overall average of accuracy is between 15.1 and 23.9 meters.

AeroScout [16] is a company offering enterprise indoor and outdoor positioning infrastructure. Their technology is a combination of RFID, GPS and Wi-Fi. Putting these different technologies together, their main goal is to cover the limitations of each technology with the benefits of the others. Like many other indoor positioning solutions, their system is based on tags provided by the company.

The limitations of the aforementioned technologies can be summarized as follows:

- The lack of flexibility for supporting multiple off the shelf wireless enabled devices as tracked tags.

- The use of technology that is not available in all devices. For example, GloPos uses Cell Tower information while many tablet devices do not have GSM/CDMA active modules. Ekahau is also using Infrared for increasing accuracy, which is not present in most of modern mobile devices.

- The need (in some cases) for extra devices or infrastructure for data transfer or positioning. For example, GloPos depends on cellular data networks for data transfer and AeroScout does not interact with present infrastructure and needs a set of new access points and tags to be in place in order for the customer to have an operational positioning system.
- High deployment efforts both in terms of number of site survey samples and time. This drawback will impose a fundamental change in the positioning algorithm even in case of minor changes in the geo-fenced zone.

On the other hand, the market and the regulations are moving towards the concept of BYOD [28]. BYOD is concerned with users using the devices owned and controlled by themselves. This concern has many reasons; the most important of which is the user privacy and control when using the device. Another reason is the fact that predictions show that in a few years there would be so many devices in the hands of users that retailer supplied devices are going to be far from popularly used [5]. Thus, the limitations summarized above will become even more important as the BYOD prevails. Also many security features and mechanisms have to be generalized to other platforms [74]. For further information about technologies using other radio frequency bands and technologies one can refer to [57, 51, 58].

In this research, our aim is to address some of these limitations that have not been the focus of other researches or industrial efforts. These issues are: 1) lack of flexibility for using different devices as tags, 2) using technologies such as infrared, ultrasonic, RFID, and cellular networks that are not commonly available in hand-held devices available off the shelf, 3) lack of use of available infrastructure towards positioning, which means a requirement for the installation of a secondary hardware infrastructure. For example, AeroScout needs a separate Wi-Fi and RFID reader network.

On the other hand, in this thesis, the proposed system is merely based on available facilities in the hand-held devices in the market and does not require cellular data adaptors on the devices. Potentially, the implementation can be employed with any Wi-Fi or Bluetooth infrastructure that is already in effect in indoor areas of interest. Moreover, it can work with any device enabled with a wireless or Bluetooth adaptor. Experiments show that the proposed system performs well with different devices under noisy and realistic indoors positioning context.

2.2 Indoor Positioning Security

There have not been many works in the field of securing indoor positioning algorithms. It is worth noting the generic efforts like [71, 38] by Strasser and Capkun et. al, that investigate methods to make networks that are dependent on the key exchange initialization, and resistance against low level jamming attacks. In addition to such low level threats, a handful of vulnerabilities such as encryption weaknesses, and password guessing, and denial of service attacks associated with industrial implementations of Wi-Fi and Bluetooth stacks and firmwares [81, 45] need to be considered.

As a basis for the user identification process, Authentication has always been a center of attention. Memorized string passwords have [61] ruled the world of authentication for a long time. Although this technique is limited by the ability of users for memorizing and choosing hard to guess passwords, there has not been a replacement for them up to now. Bonneau et. al [33] have studied about 35 methods that are designed to replace passwords. They have compared these methods in terms of security, usability, and deployability. Despite the fact that many of the studied methods are reportedly more secure and usable, none of them is as deployable as passwords because of the imposed extra cost and the complexity of their architectures [33].

To bring authentication and confidentiality to 802.11 wireless networks, different standards and protocols were proposed. The first popular link layer protocol was Wired Equivalent Privacy (WEP) [2], based on a pre-shared secret mechanism. However, soon after its release many experts started to discover weaknesses and vulnerabilities in the WEP mechanism [25, 34, 62, 72, 37]. Simultaneously, commercial and free tools were released and made many people able to exploit these weaknesses [17]. Wi-Fi alliance created the Wi-Fi Protected Access (WPA) [24] mechanism to address these weaknesses. In the 802.11i standard [46], WPA2 was proposed to standardize a slightly more secure way of authentication over the networks in comparison to WPA. WEP, WPA, and WPA2 operate in the Data Link Layer of the TCP/IP stack [70], mainly providing per frame confidentiality. However, the latter two support an external Authentication Server (AS) to detach the process of authentication from encryption. Despite the additional security added by this protocols, WPA and WPA2 are still vulnerable to cracking and dictionary attacks. With the availability of parallel computing and Graphical Processing Units, this has become much faster and easier for attackers to infiltrate networks that suffer a weak choice of passwords [50]. To this end, many industrial extensions were introduced to address the weaknesses discussed above. Baek et al. [26] have surveyed the most recent techniques in the wireless authentication field based on factors such as mutual authentication, identity privacy, dictionary attack resistance, session key strength, and having a tested implementation. As a conclusion, they propose some extensions of the Extensible Authentication Protocol based on Transport Layer Security (EAP-TLS) as the best method.

In short, there are many reported efforts of securing Wi-Fi. However, to the best of my knowledge, there is no study for indoor positioning techniques in terms of evaluating attacks against the higher layers of the protocol stack and the system design. Therefore, in this research, I investigate some attack scenarios that are taking place in the higher layers of the network protocol stack as well as investigating how to counter them.

Chapter 3

Methodology

In this research, there are certain functionalities that must be guaranteed by the proposed system. The first functionality is authentication, that involves identifying users who interact with the system using their hand-held devices. Entangled with authentication is the identification mechanism, by device parameters. The second functionality of such a system is positioning, which consists of a mechanism to determine a user/device's indoor position when she/he requests to access resources from the protected network or infrastructure. The third functionality, which intuitively is completely dependent on existence of the previous functionalities, is access control. The access control mechanism is generally defined as an apparatus that decides, based on any arbitrary technique, a user's authorization at the time of service usage.

The aforementioned functionalities can be categorized into two groups of 1) Authentication and Identification, and 2) Indoor Positioning . The implementation consists of multiple network services in addition to mobile and web applications as its building blocks. These building blocks are described in detail in Section 3.1. However, most of the complexity of the system is obviously lying under the Indoor Positioning subset of functionalities that are also described in the aforementioned Section of this thesis.

3.1 System Overview

This section will discuss the components that build the whole system together. There are two main categories of components in the system: Services, and Applications. Sections 3.1.1 and 3.1.2 will discuss these two categories, respectively.

3.1.1 Network Services

Briefly, principal services that are required for the operation of a user under a location controlled application environment include Authentication, Authorization, and Geo-fencing (indoor positioning). While the controlled servers are of non-deniable importance, their architectural details are out of the scope of this thesis. To this end, these services are considered to be resources such as web based services or network resources. Generalizing network resources to services, leads to a design that does not focus on a low level set of requirements.

Authentication

Because all the client transactions should be centralized for ease of management and accounting purposes, Authentication must be provided as a Single Sign On (SSO) capable service. There are numerous open source and commercial products that offer this functionality out of the box. The use of OpenID [66] is opted out because it can be limiting in terms of network architecture. More precisely, use of OpenID with globally accessible providers will cause the corporation to be unable to limit the authentication steps to the local networks in geo-fenced zones.

There are many open source, free, and commercial implementations for the Single Sign On architecture [13]. The most well-known open source implementations are the Java Open Single Sign-On (JOSSO) [12] and the Central Authentication Service (CAS) [6]. CAS and JOSSO offer many similar features. Taking a closer look, CAS has a higher simplicity by use of a layered dependency injection system based on Apache's Maven [18], and is hence more customizable as well as having a better community support. CAS has also offered a layered deployment model, which allows the administrators to distribute the physical location of authentication servers, also known as authentication proxies through registered web applications. This helps to distribute the infrastructure, while keeping the ticketing operations centralized at the same time [8].

Figure 3.1 demonstrates the cycle of getting authenticated using a CAS server installation with an arbitrary authentication mechanism as the backend. CAS supports a wide variety of backends out of the box. Such mechanisms include Lightweight Directory Access Protocol (LDAP), Remote Authentication Dial In User Service (RADIUS), and database authentication in addition to a handful of other protocols such as x509 certificate based authentication, Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO), and Open Authentication (OAuth) [19]. CAS can also use an OpenID backend, regardless of version, which is helpful if a corporate needs OpenID benefits while the related limitations are removed.

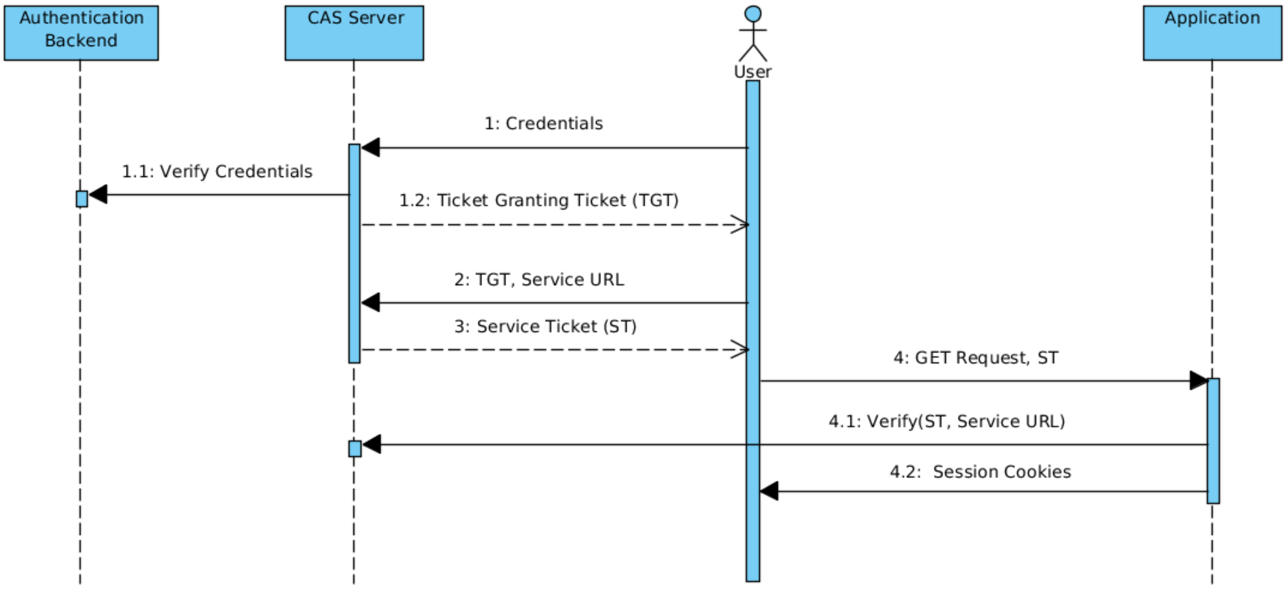


Figure 3.1: Authenticating to a registered application using CAS.

Identification

Although in many cases Authentication and Identification are referred as the same concept, in the proposed system, getting authenticated is not enough. This is because both the user and the device he/she is using need to be authorized to access system resources. As a result, the system needs a strategy to bind user credentials and device characteristics together in order to identify them as an entity eligible for authorization. To do this some hardware parameters/values are associated with every device: 1) MAC (Media Access Control) address of the Wi-Fi adapter installed on the device, 2) IMEI (International Mobile Station Equipment Identity) number of the device [1], and 3) the build model of the device. These values are chosen to represent the device permanently, as they are not subject to change as a side effect of software and firmware updates. When a user registers to the network through the operator, these values are stored and associated to his/her profile. To finish the registration a random salt value is then transferred to the device using a proximity based communication medium. This salt value, $Salt_{Init}$, is later used to hash a specific string and send to server to verify session validity [61].

To this end, the identification process uses both the device and the user information.

[1] Unless the device does not operate on cellular networks.

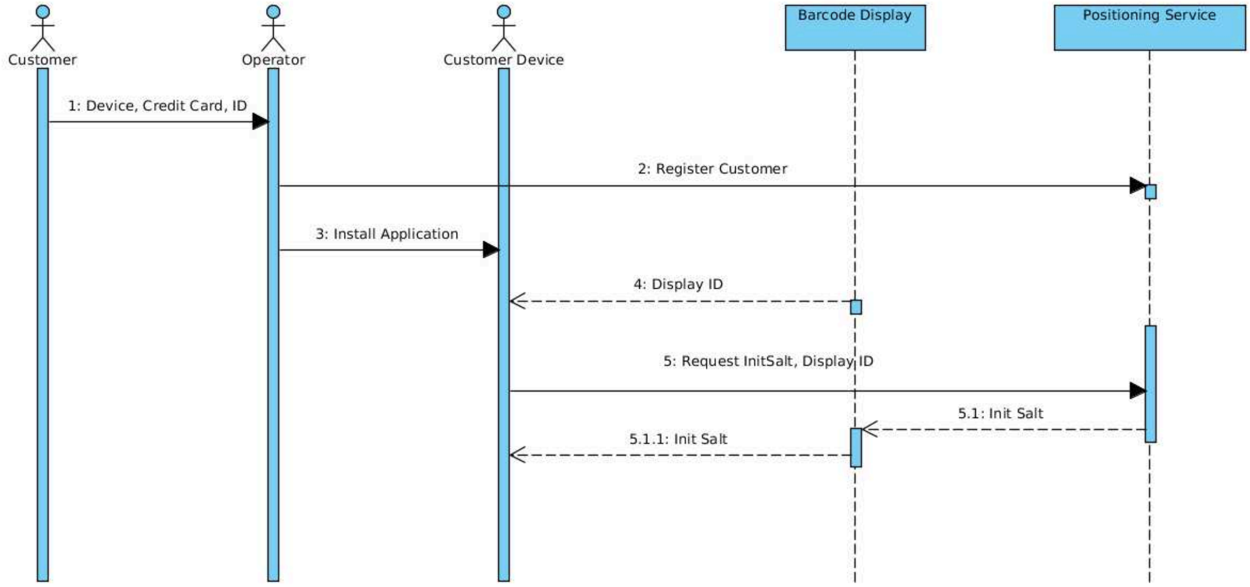


Figure 3.2: Registration process of a typical customer.

Additionally, this process involves the positioning service which adds to the levels of positioning security because a positioning service is local to the geo-fencing zone, while the authentication services are served remotely. This enforces users to be physically present at the zone, eliminating the possibility that an adversary can initiate an application session remotely.

The registration process is demonstrated in the sequence diagram given in Figure 3.2 where a user hands his/her device and identification information to the system Operator to get registered to the system. As can be seen in the diagram, the $Salt_{Init}$ value is sent to the device using a barcode display. This protects the $Salt_{Init}$ value from eavesdropping because the communications between the positioning service and barcode displays are using SSL encrypted sockets. Scanning the barcode using the device's camera is also helping the security of the process in cases where the barcode display is well protected.

Barcodes, in this case Quick Response Code (QR), are a form of Near Field Communication[2] (NFC) [80]. Although RFID based NFC chips are becoming more popular and get equipped into many devices nowadays, they are still not available in every mobile device, so I chose to use the QR barcode technology for the registration step.

[2] Not to be confused with the radio NFC chip.

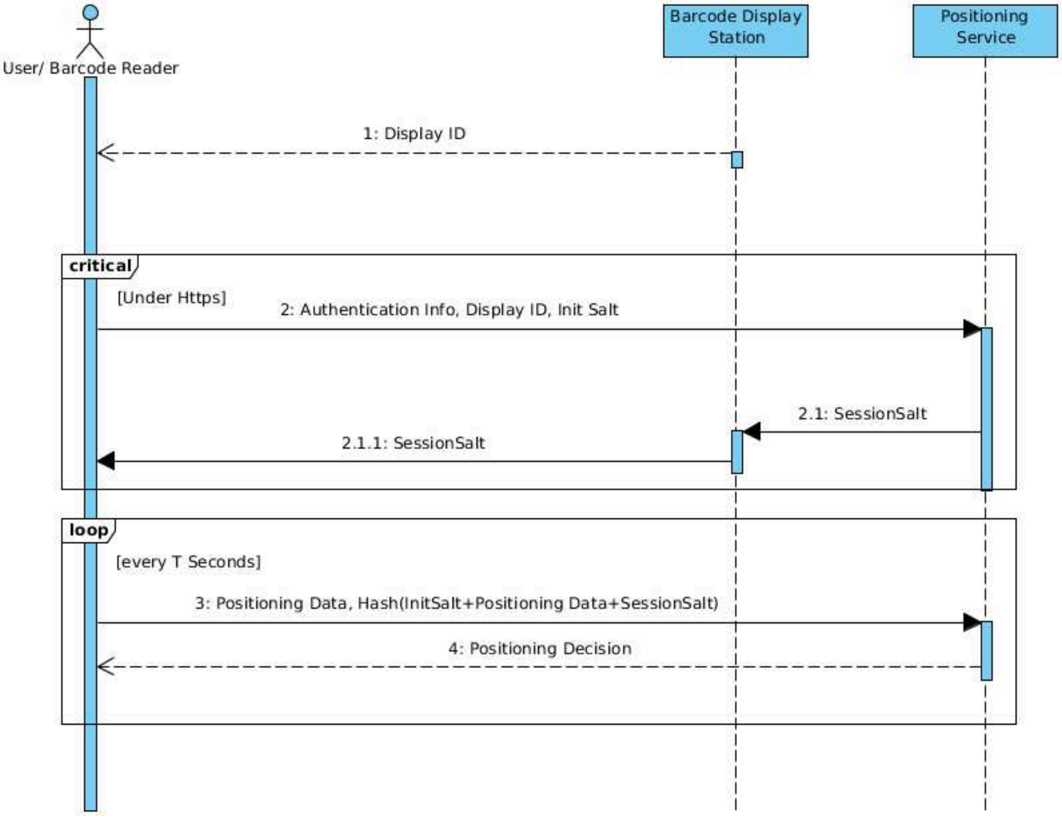


Figure 3.3: Session initiation and positioning based on proximity based transfer of the value $Salt_{Session}$.

In addition to registration, barcodes are also used at the time of an active session initiation. This complementary method is used to ensure that users use their registered devices. Moreover, coupling a strict proximity based barcode scanning results in an extra positioning step to ensure the presence of a user (in a geo-fencing zone) when initiating a working session.

Every time a user visits a geo-fenced site and intends to use services on a previously registered device, a new salt value is sent to his/her device. This salt value is called $Salt_{Session}$, which is used along with the $Salt_{Init}$ to ensure that the device is both registered and also is present at the geo-fenced zone where access to services is controlled. This process is demonstrated in Figure 3.3, assuming that the user has already authenticated to the system and intends to initialize a working session.

The positioning service is implemented using the HTTP based RESTful API standard

[47], and is served in SSL encrypted channels. As stated before, the service is deployed on-site for having shorter delays and also being only accessible from a specific subnet associated with a number of geo-fences. Each positioning request includes the RSSI data collected by the device, and depends of the infrastructure type (Wi-Fi or Bluetooth) in action which can be based on Bluetooth or Wi-Fi. This information is then associated with a hash value that is computed as shown in Equation 3.1.

$$Hash(Salt_{Init} \mid\mid Salt_{Session} \mid\mid RSSI_{string} \mid\mid Username \mid\mid BuildModel \mid\mid IMEI \mid\mid MAC_{address}) \tag{3.1}$$

The values in Equation 3.1 are concatenated then hashed. This hash value is sent to the positioning service. Then the service verifies these values by reproducing the hashes using values that are generated/entered at the time of registration, and session initiation. If the two of hashes are not equal then the service will not be provided. Otherwise, the request will be processed normally. Please note that when the initial authentication steps are being taken, before the near field salt transfer, the $Salt_{Session}$ is set to null.

3.1.2 Web and Mobile Applications

To collect the training data used for building classification models, a mobile application called “Surveyor” is developed. This application consists of three major functionalities:

1) Collecting Wi-Fi signals and RSSI information; and

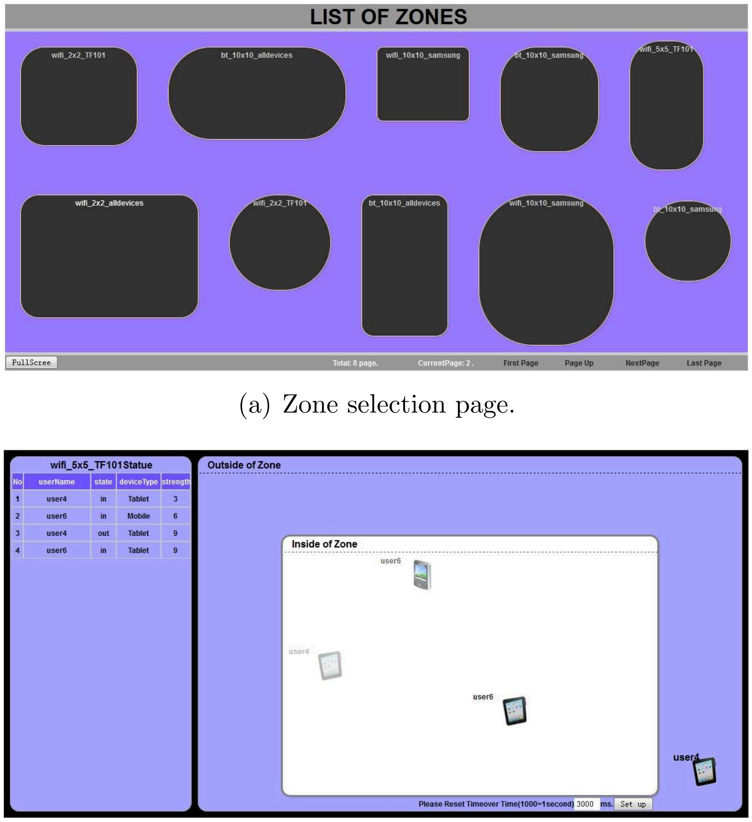
2) Collecting Bluetooth signals and RSSI information.

3) Organizing and transferring the information to the data collection server.

To gather the training data samples, Surveyor maintains a local database on the hand-held device. This mobile database allows the operator to categorize and store the samples collected during the site survey phase. The application collects and labels the samples based on the administrator’s input. There are two possible labels for each sample, which corresponds to being inside and outside the zone. Number of samples for each label is shown in a simple way to help the administrator in keeping track of the data he/she is collecting. When the administrator is done collecting samples, data can be sent over to the

SSL protected web application that stores the data in an archive and makes it ready for preprocessing and model building.

“Wi-Fi Demo” and “BT Demo” are two mobile applications that are then used to track the play on the Wi-Fi and the Bluetooth infrastructures. These two applications simply demonstrate the process of connecting and gaining access on the network and live access control while the service being accessed. In this work, the mobile platform I focus on is the Android operating system. Android versions ranging from 2.1 to 4.0.1 were tested with the proposed system without any problems.



(a) Zone selection page.

wifi_5x5_TF101Statue

No	userName	state	deviceType	strength
1	user4	in	Tablet	3
2	user6	in	Mobile	6
3	user4	out	Tablet	9
4	user6	in	Tablet	9

Please Reset Timeover Time(1000=1second) 3000 ms.

(b) Current zone status page.

Figure 3.4: Screenshots from the monitoring user interface.

A web application is also developed in order to give the network administrators a view of what is happening in the geo-fenced zones. This application connects to the positioning service using a special monitoring API and fetches statistics such as number of users, each user’s status according to the positioning system, type of the device, and geo-fences that each user is having access to. Screenshots from the monitoring user interface are demonstrated in

Figure 3.4. Figure Figure 3.4(a) allows the administrator to choose which node he/she wants to monitor. Then the window shown in Figure Figure 3.4(b) is brought to the administrator and is dynamically updated based on a customizable refresh rate.

3.2 The Proposed Positioning Algorithm

To use wireless network related data for indoor positioning, three values are popularly used. Assuming that there are more than one access point or dongles in place, these values can come in handy for triangulation, trilateration, or multilateration. These values are described as follows:

1) **Time Difference of Arrival (TDoA)**: This value has many use cases. But mainly, TDoA is used for calculating Angle of Arrival that results in position estimation. If access points are synchronized to send signals in a specified time, TDoA can be useful for directly estimating a user's position. However, it is rather difficult to synchronize access points with a small error considering the sensitivity of the equations to small errors, resulted by the high propagation speed of radio waves.

2) **Angle of Arrival (AoA)**: is the angle in which a radio frequency is propagating at the time of arrival to the receiving antenna. This value can be achieved by use of special antennas or antenna arrays. The intuitive requirement of the process is a knowledge about current direction of the user. But having this is not enough for the process. Most popular technique for measuring this angular value is to use a grid or array of antennas and estimate the angle by measuring TDoA over all the elements of the array. In this case the regularly available devices are hardly useable. Although the devices are mostly equipped with a magnetic sensor (compass) for direction information, they lack a rather sophisticated antenna array structure that is essential for obtaining the AoA information based on TDoA.

3) **Received Signal Strength Indication (RSSI)** This indicator shows the amount of power a wave is carrying at the time of arrival at the receiving antenna. This measure is built in the IEEE 802.11 standard for roaming and power optimization purposes [46]. According to the 802.11 standard [46], this should be a 8-bit integer value. Further decision about how to compute such a value is dependent on the implementation. But

the regular way of calculating the RSSI is to measure signal power in relation to the maximum assumed possible power (that is also carried in 802.11 frame headers), and representing it in decibels, as shown in Equation 3.2. A signed 8-bit value can range from -128 to +127. However, in practice this value is normally a negative integer in ranges of -30 and -100.

$$RSSI = 10 * \log_{10}(\frac{ReceivedSignalPower}{MaxPower}) \tag{3.2}$$

While AoA and TDoA are values that cannot be read by primitive antennas available on mobile devices, RSSI is available due to 802.11 methods for power management and roaming decisions. Bluetooth, also supports RSSI for connections since 802.15.1 (Version 1.1) [4]. RSSI is basically calculated using the preamble of the packets as they reach the receiving antenna.

Considering the aim of this research that is to set a generic solution for hand-held devices, AoA and TDoA based techniques are not an option for the proposed system. Because their recruitment imposes hardware and software modifications. On the other hand, RSSI is available on nearly all of the devices and software platforms that support Wi-Fi and Bluetooth connectivity with recent standards. Therefore, the proposed positioning system is based on a machine learning approach employing the RSSI values read by devices.

3.2.1 Positioning Data

As mentioned in Chapter 2, RSSI is widely used for both exact and approximate location estimation purposes. Such vast applications of this value shows its usefulness for indoor positioning practices.

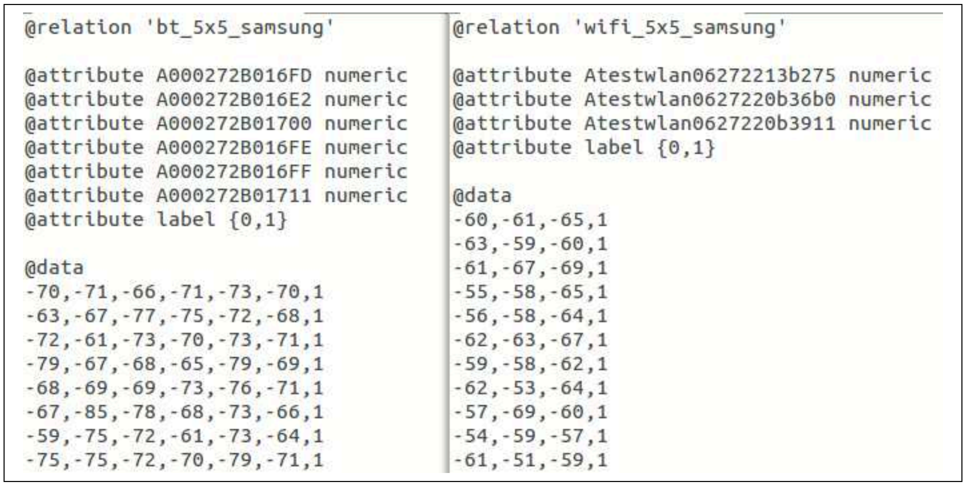
Geo-fencing is concerned with detection of the presence of a user in a specific perimeter, or zone. Although exact positioning can be used to geometrically detect the presence of the user, its complexity and vigorous need for data collection and simulation is a major drawback. However, the concept of exact position information is not necessary in the problem being studied in this thesis. To simplify the problem, this thesis work proposes classification based geo-fencing. The classification based geo-fencing engine uses RSSI data to classify devices/users as being present or absent in a specific zone.

Table 3.1: Attributes for the collected data sets.

Type	Attribute Names	No. Attributes
Wi-Fi	Access Point MAC || SSID	3 + label
Bluetooth	Dongle MAC	6 + label

Data sets are collected using mobile applications previously discussed. The already in place Wi-Fi access points and Bluetooth enabled hardware can be used as anchor nodes to collect data upon them. In this thesis, two major type of data sets are collected, Bluetooth data and Wi-Fi data. In Wi-Fi data sets the attributes are access point canonical names and the attribute values are RSSI values associated with the corresponding access point at the time of data collection. For Bluetooth data sets, the features are Bluetooth dongle/hardware canonical names and the values are RSSI values of their propagating signals at the time of data collection. Canonical names are built by concatenating MAC addresses with the display name or Service Set Identifier (SSID) depending on the type of the data set, i.e. Bluetooth or Wi-Fi. All types of data sets have a label attribute that holds the class label value. The class label is set to be false for being absent in a geo-fenced zone, or true otherwise.

Table 3.1 tabulates the attribute names and number of attributes for both Bluetooth and Wi-Fi data sets. Each data set has a label attribute, which is set to 1(== True) for being inside the geo-fenced zone, and 0(== False) otherwise.



```
@relation 'bt_5x5_samsung'

@attribute A000272B016FD numeric
@attribute A000272B016E2 numeric
@attribute A000272B01700 numeric
@attribute A000272B016FE numeric
@attribute A000272B016FF numeric
@attribute A000272B01711 numeric
@attribute label {0,1}

@data
-70,-71,-66,-71,-73,-70,1
-63,-67,-77,-75,-72,-68,1
-72,-61,-73,-70,-73,-71,1
-79,-67,-68,-65,-79,-69,1
-68,-69,-69,-73,-76,-71,1
-67,-85,-78,-68,-73,-66,1
-59,-75,-72,-61,-73,-64,1
-75,-75,-72,-70,-79,-71,1
```

```
@relation 'wifi_5x5_samsung'

@attribute Atestwlan06272213b275 numeric
@attribute Atestwlan0627220b36b0 numeric
@attribute Atestwlan0627220b3911 numeric
@attribute label {0,1}

@data
-60,-61,-65,1
-63,-59,-60,1
-61,-67,-69,1
-55,-58,-65,1
-56,-58,-64,1
-62,-63,-67,1
-59,-58,-62,1
-62,-53,-64,1
-57,-69,-60,1
-54,-59,-57,1
-61,-51,-59,1
```

Figure 3.5: Sample of the Bluetooth and Wi-Fi data sets.

Figure 3.5 is showing a sample of both Bluetooth and Wi-Fi data sets. The Bluetooth sample is shown on the left, while the Wi-Fi is shown on the right side of the figure.

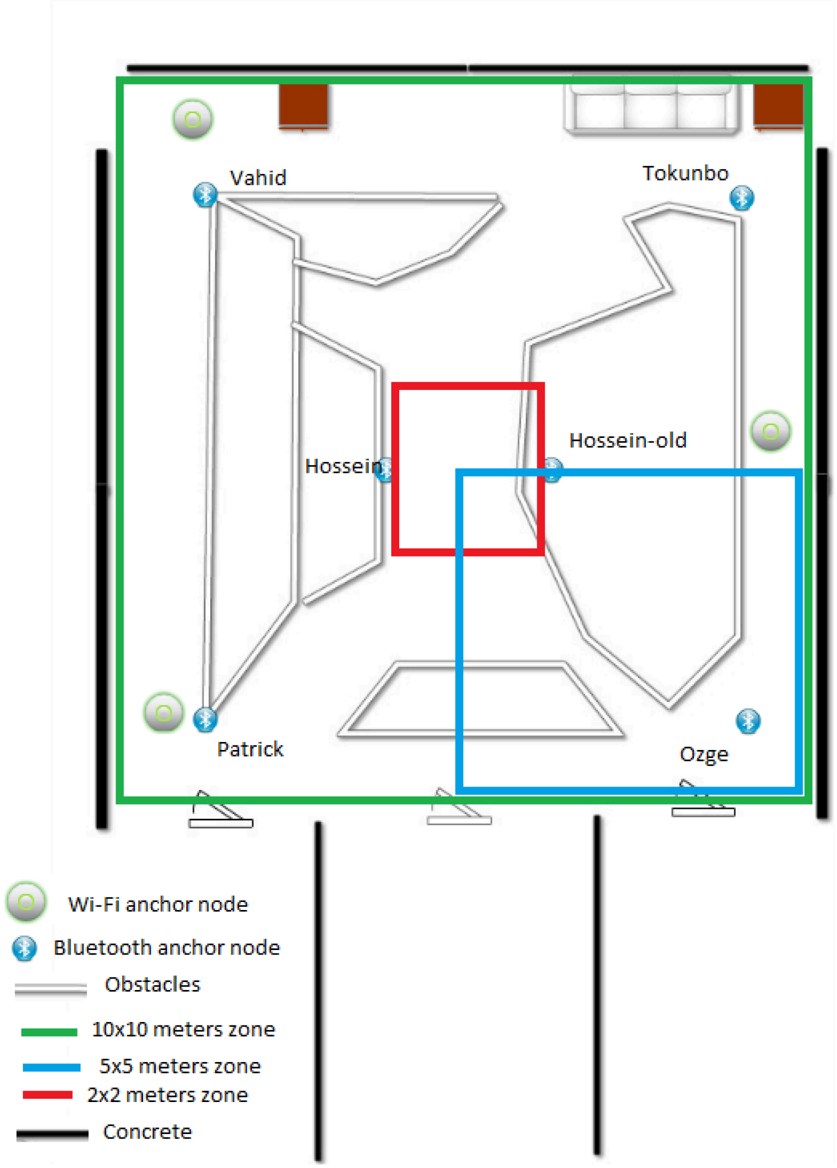


Figure 3.6: A floor plan of the experimentation area.

In this research, for evaluation purposes, the NIMS Lab area was set up for a set of experiments. Three Wi-Fi access points and 6 Bluetooth dongles are installed. Figure 3.6 illustrates a floor plan of the area and anchor node installations. The Lab is not cleaned of radio noise, where other Wi-Fi networks, Bluetooth connections, and wireless enabled devices are communicating and propagating wireless signals all the time throughout the evaluations conducted.

To build the classification engine, experiments conducted using Naive Bayes, C4.5 decision trees, Random Forest, and Support Vector Machines. Based on these experimental results, that will be discussed in more detail in Chapter 6, the top two machine learning algorithms are then chosen for further comparison purposes. Ultimately the Random Forest [35] algorithm is selected as the best performing algorithm. Two machine learning algorithms that are compared in the end, are: 1) C4.5 decision tree [63], and 2) Random Forest [35]. The C4.5 is widely used for many reasons. This algorithm has robust performance against noisy data and missing values, it also leverages a comprehensible model structure which makes it easy to analyse, modify, or embed for development and experimental purposes. However, in some cases where data is imbalanced or there are many attributes, C4.5 might build over fitted models. To address this, I have also employed the Random Forest algorithm in the proposed system. Using an ensemble of decision trees, Random Forest is known to be efficient when dealing with imbalanced data sets. This is caused by its use of a random sub-sample of the data set for training each tree. The Random Forest algorithm also performs well in terms of using as many attributes as possible by randomly using subsets of attributes for building each tree in the forest. In the following, each of these machine learning algorithms is discussed in more detail.

3.2.2 C4.5 Decision Tree

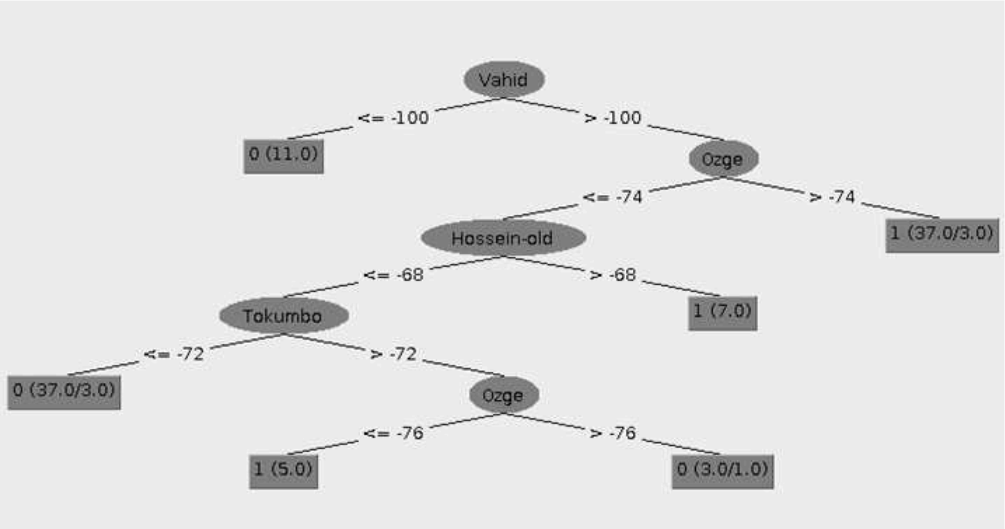


Figure 3.7: A visualization of the C4.5 tree trained on Bluetooth data.

Decision trees are built using the collected data vectors once, then the built model is used to predict the label of an unseen data record multiple times. The tree consists of three building blocks, internal nodes, edges and leaves. Each internal node represents an attribute. Edges that connect internal nodes to their children are labelled with conditional expressions based on the possible nominal or numeric value of the preceding node. Leaf nodes of the tree represent predicted labels. Any data record in a given data set will lead to traversing the tree from root to one of the leaf nodes to determine the predicted label. Traversing the tree with a sample, edge labels and the value of the corresponding attribute in the sample determine which child to choose for continuing the traversal down to a leaf node. Reaching a leaf node, the prediction is determined the label of the leaf node that is met and the end of a traversal.

Training phase involves selecting attributes and link labels in addition to leaf node associated predictions. To build a tree from root down, features need to be ranked. Features are ranked based on their *GainRatio*, where the information gain ratio for an attribute a_i in a data set S is defined in Equation 3.3.

$$GainRatio(a_i, S) = \frac{InformationGain(a_i, S)}{Entropy(a_i, S)} \tag{3.3}$$

$$InformationGain(a_i, S) = Entropy(y, S) - \sum_{\nu_{i,j} \in dom(a_i)} \frac{\mid \sigma_{a_i = \nu_{i,j}} S \mid}{\mid S \mid} \cdot Entropy(y, \sigma_{a_i = \nu_{i,j}} S) \tag{3.4}$$

$$Entropy(y, S) = \sum_{c_j \in dom(y)} - \frac{\mid \sigma_{y = c_j} S \mid}{\mid S \mid} \cdot \log_2 \frac{\mid \sigma_{y = c_j} S \mid}{\mid S \mid} \tag{3.5}$$

In equations 3.4 and 3.5, σ is the standard deviation. Variables c_j and $\nu_{i,j}$ represent labels in the class domain of the attributes y and a_i respectively. $InformationGain(a_i, S)$ is the value that shows impurity of the values in an attribute's values. Since Information Gain is biased towards nominal attributes with many different values, the gain ratio formula is designed to normalize it [67]. This will prevent the problem of zero $InformationGain$ for nominal attributes that have many different values. More detailed information on C4.5 can be found in [63] and [67].

Figure 3.7 demonstrates a visualization of a C4.5 tree trained on a Bluetooth data set. Attribute names that appear on the internal nodes, having an oval shape, of the tree are the name of dongles. The attributes are named after the lab members that have a desk close to the respective anchor node. Leaf nodes, with a rectangular shape, list the final decision by 0 (== false or outside) and 1 (== true or inside) at the leftmost. The pair of values shown on each leaf node in parenthesis represents the number of test samples that visited the specific leaf node. The first number is the number of instances that were classified correctly. The second number, if present, shows the count of instances that ended up at this leaf node with a wrong classification result, producing a False Negative or False Positive prediction.

3.2.3 Random Forest

Random Forest is a classification algorithm based on ensemble learning [35]. Random Forest grows a collection of trees on a training data set based on three rules:

1) Each tree is trained of N samples, where N is the size of the original data set. But the samples are randomly selected from the original data set **with replacement**.

2) If there are M attributes, the constant $m(\ll M)$ is specified. For splitting nodes m random attributes are selected from the original M inputs. Value of m remains constant through the forest growing process.

3) Each tree is fully grown, which means there is no pruning.

One of the interesting features of the Random Forest classifier is that it is able be unbiasedly trained on only training data. This is because the training mechanism includes 33% split of samples selected for training each tree for testing purposes. Random Forests do not suffer from over-fitting, and they are also very fast in terms of training time [14].

Additionally, Random Forest is able to compute the proximity of instances in the training data set. After each tree is built, all the instances are used to traverse the new tree. the instances that end their traverse at the same leaf node will get an incremented proximity score. This score is normalized at the end of the training process by dividing by the number of trees. Then the proximity values are used to replace missing values at the time of testing.

However, missing values are also replaced at the time of training. There are two approaches for missing value replacement at the time of training:

1) **Fast** The missing value is replaced by the median of the corresponding attribute, which yields less accurate results than the latter.

2) **Slow** First a forest is built by replacing missing values inaccurately, in order to calculate instance proximities. When the proximities are in hand, the missing numeric values are replaced calculating a weighted average of non-missing values in samples, where weights are the proximity values. For nominal values, the missing value is replaced by the most frequent non-missing value, where the frequencies are weighted based on proximity values.

Random Forest is well known for building accurate classifiers on different data sets. It also supports methods for balancing error in unbalanced data sets and is capable of handling a large number of attributes efficiently. More detailed information on Random Forest can be found in [14, 35].

3.2.4 Client Status Manager

Client Status Manager (CSM) is responsible for managing user requests and collecting data statistics. CSM holds brief and long term history of each user's activity. This component also works as a behaviour analysis system for the clients, throttling request timings and managing the number of active devices per user. It keeps track of statistical characteristics of the RSSI values sent by a device to detect anomalous activity. CSM is a component that is responsible for detecting and preventing attacks specific to the indoor geo-fencing system proposed here. Some of these attacks and their countermeasures are discussed in detail in Chapter 5. Moreover, this component is responsible for applying smoothing and outlier detection on the signal strength data provided by hand-held devices to enhance the classification and user experience.

Statistics and monitoring unit uses data collected by CSM to make an abstract view of the client activity per geo-fenced zone. A simple web interface demonstrates the status and information of clients (mobile devices) active in a specific geo-fence. The information includes parameters such as being inside or outside the zone, signal strengths, device types, etc. Figure 3.4 demonstrates some screen shots of the statistics and monitoring user interface.

As demonstrated in Figure 3.8, CSM keeps a profile for each of the clients connected to the system. Each profile keeps the following buffers:

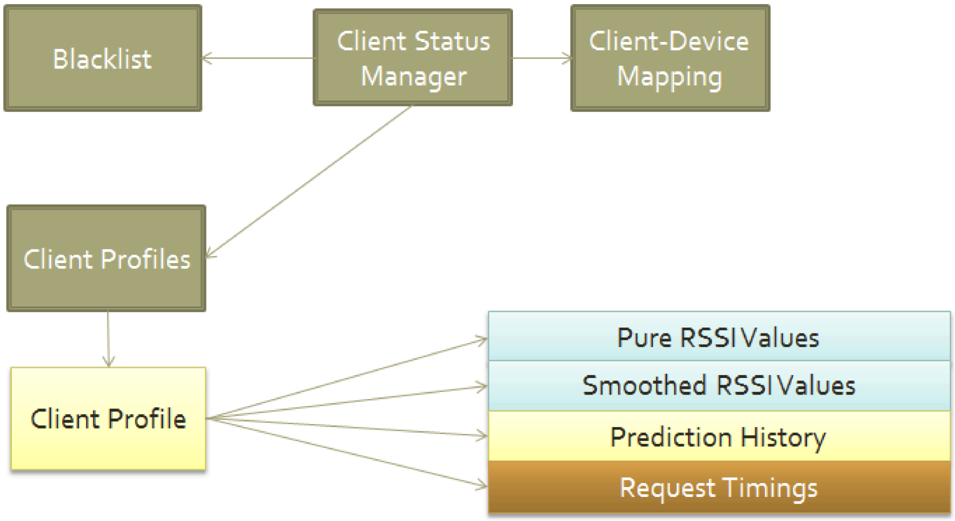


Figure 3.8: Client Status Manager's structure and data buffers.

- **Pure RSSI Values** Buffer is held in a Circular FIFO Buffer fashion. This buffer also calculates the mean RSSI value and other statistical values for smoothing and outlier detection purposes.

- **Smoothed RSSI Values** Buffer holds the smoothed values for error calculation and outlier detection.

- **Prediction History** includes the last p predictions. Each prediction consists of three boolean values : 1) Smoothed Prediction, 2) Rough Prediction, and 3) Final Prediction. Further details on the use of these values can be found in Chapter 6.

- **Request Timing** Buffer is recorded in order to keep track of frequency of requests per user-device pairs. This information is then used for request throttling and security purposes.

"Client-Device Mapping Manager" keeps track of the number of devices being active per user in each zone. This is useful for managing device identities and mapping them to their owner. Additionally, users with too many active devices are reported or blacklisted to prevent attacks.

Finally, Blacklist is a mechanism that keeps track of users that are blocked from connecting to the service due to not respecting global limits or suspicious activities. The scheduling component in the Blacklist enables the system to block users temporarily or permanently.

3.3 Generic Access Control Mechanism

The proposed system is aimed to aid the administrators for implementing location aware access control. As a result, the administrators or industry developers need the freedom to complement this solely location based access control mechanism to suit their specific business model. Another case is where the same industrial complex has different access control methods for different installations of the proposed geo-fencing system.

To this end, the system is designed in a manner to support a generic access control backend. This allows administrators and developers to integrate their custom authorization mechanisms to the system. An abstract class called "AuthorizationDecider" is provided with the installation libraries. This class can be extended to implement a tailored access control methodology. Then the implemented method can be set to be the default authorization filter for the geo-fencing system with simply modifying the configuration file.

For proof of concept purposes, a default implementation of the AuthorizationDecider is used where access control is simply performing a single rule check: if the device is authenticated and the requested service is on the same network with the device, then access is granted. Otherwise, the device is denied from accessing the requested resource/service.

Chapter 4

Noise, Outliers and Missing Values

The proposed system depends on RSSI values received from both Bluetooth and Wi-Fi infrastructures to decide upon the presence of a specific user in a geo-fenced zone. However, there are many factors that cause the Industrial, Science and Medical (ISM) band to be one of the noisiest frequency bands: 1) The use of this band is free for public and many non-regulated transmitters are propagating signals. 2) The 2.4 GHz frequency is water resonant, so it is efficiently absorbed by objects containing water (including human bodies). 3) Due to the fact of absorption of 2.4 GHz frequency by water, many appliances (e.g. microwave ovens) operate on high power adjustments in this band that adds a significant noise on the ISM band. This chapter discusses the methods that are experimented and employed in this research in order to overcome such noise for the geo-fencing service administration.

4.1 Smoothing Techniques

There are many approaches for dealing with noise in RSSI readings. Such approaches include outlier detection and value estimation which leads to smoothing the data samples. Value estimation techniques estimate the next upcoming value in a stream of data, or for replacing missing values. Estimation, in a data stream, is normally based on temporally local samples of data. The wideness of this locality can be determined by a span or $Window_{Size}$. $Window_{Size}$ determines the number of data points that are taken into account when estimating the next possible value. The larger the $Window_{Size}$, the stronger the effect of previous data samples on the next estimate. As a result, a larger $Window_{Size}$ will yield a smoother trend line in comparison to the original data. As a result, aside from the estimation methodology, parameters such as $Window_{Size}$ can greatly impact the outcome of the procedure. To choose the best, I have experimented with six different well known smoothing methods:

1. Moving Average, a low pass filter with filter coefficients equal to the reciprocal of the span.

2. Local Regression using Weighted Linear Least Squares and a 1st degree polynomial model (LOWESS).

3. Local Regression using Weighted Linear Least Squares and a 2nd degree polynomial model (LOESS).

4. Savitzky-Golay filter. A generalized moving average with filter coefficients determined by an un-weighted linear least-squares regression and a polynomial model.

5. A robust version of 2 that assigns lower weight to outliers in the regression. The method assigns zero weight to data outside six mean absolute deviations (Robust LOWESS).

6. A robust version of 3 that assigns lower weight to outliers in the regression. The method assigns zero weight to data outside six times the median absolute deviations (Robust LOESS).

4.1.1 Moving Average Method

The Moving Average method is based on the idea that RSSI readings that are close to each other in terms of time, should also be close to each other in terms of their values. Also, variations of these values in relation to the average value of the stream, should be temporally proximate.

In an indoor positioning scenario, this is more understandable considering the fact that walking speed for an average person is between 1.25 to 1.5 meters per second [39]. Based on the fact that users walk slower in indoor environments [83], the probability of causing drastic changes and shootings in the perceived signal strength decreases, and the idea behind Moving Average seems applicable. A moving average using a $Window_{Size}$ of k will result in an estimated value calculated as shown in Equation 4.1. Equation 4.2 is used to calculate the next estimate. Where $\widehat{y}_{t+1}$ is the estimate, and y_t and y_{t-1} are last two actual readings.

$$\widehat{y}_t = \frac{1}{k+1} \sum_{j=0}^{k} y_{t-j} \tag{4.1}$$

$$\bar{y}_{t+1} = \widehat{y}_t + (y_t - y_{t-1}) \tag{4.2}$$

Equation 4.1 is also called one-sided moving average. Two sided moving average techniques are useful when readings are available from both before and after the estimation/smoothing point. Please note that setting $k = 0$ leads to putting the latest sample value as the estimate which is also called “naive” Moving Average. Exponential weighting of the values in the window is another approach for managing the effect of their age on the estimate. Further information on these methods can be found in [53].

4.1.2 Locally Weighted Regression Scatter Plot Smoothing

The method LOESS (Locally Weighted Regression Scatter Plot Smoothing), was first introduced by Cleveland in 1979 [42] and was further developed in early 90’s [43]. It is a local regression technique based on a second order polynomial derived using a Least Square approximation. The polynomial is built using points from the whole data span, biased toward a range of points in vicinity of the sample which is going to be estimated by an assigned weight. The LOWESS method is the same as LOESS, unless the least square approximation is a first degree or linear polynomial. The local regression weights are calculated as demonstrated in Equation 4.3. Where $d(x)$ is the time difference between the values of x and x_i.

$$\omega_i = \left(\mid \frac{x - x_i}{d(x)} \mid ^3 \right)^3 \tag{4.3}$$

Least square approximation method is a popular approximation technique for data fitting. Least square tries to minimize the summation of squared value of errors. Error is defined in terms of the difference of an approximated value with the actual value observed at that point, this difference is also called a residual.

4.1.3 Savitzky-Golay Method

Savitzky-Golay [52] filter is a generalized form of the moving average algorithm. The filter uses an order-k polynomial regression local to the estimation point. It also assumes that all data points are evenly distributed in time, which does not hold for our collected data sets. And also might cause inconvenience when data sampling is set to be on demand for purposes such as extending device battery operation time.

4.1.4 Robust LOESS and LOWESS Algorithms

To make LOESS and LOWESS robust, the only modification to the original methods is that outliers are removed from the computation by simply assigning zero weights to them. When calculating the estimate, a 1st or 2nd order polynomial is used to approximate the trend of data in that vicinity. However, weights are assigned to the closest values to bias the least square approximation toward the most recently observed data. When zero weights are assigned to outliers, they are simply disregarded in the least square approximation process.

Outliers are detected based on comparing their residual to the median absolute deviation. Median absolute deviation is the median of the distances samples have from the mean of the data observed so far. For the robust methods, regression weights are calculated as shown in Equation 4.4. in this Equation, r_i is the i^{th} residual. This equation also shows how MAD is used to give zero weights to the samples that are considered to be outliers. MAD is calculated as shown in Equation 4.5

$$\omega_i = \begin{cases} \left(1 - (\frac{r_i}{6MAD})^2\right)^2 & \text{if } |r_i| < 6MAD \\ 0 & \text{if } |r_i| \geq 6MAD \end{cases} \tag{4.4}$$

$$MAD = median(\mid r_i \mid) \tag{4.5}$$

Chapter 5

Security

As discussed in previous chapters, there have not been many researches focusing specifically on higher level threats in an indoor positioning system. This Chapter discusses specific threats that are introduced by the architecture of such systems. Additionally, attacks and countermeasures that are related to the recruited technologies and standards employed in the proposed system are studied.

5.1 Communication Security

The greatest concern in the context of communication is to provide confidentiality for the network interactions. To do so, I experimented with two methods:

1) SSH Tunnelling.

2) SSL/TLS Sockets.

Secure Shell (SSH) is an application layer protocol, that was initially introduced to address weaknesses of telnet and remote login protocols as described in RFC 4251 [85]. This protocol provides secure communications over an insecure network infrastructure. Based on different implementations and platforms; services such as remote graphical shell access, tunnelling, and port forwarding can be provided by SSH. In this research, the OpenBSD Secure Shell (OpenSSH) open source implementation [20] was used to establish secure connections.

The working mechanism of the SSH tunnelling is described as follows:

i- A device authenticates to the SSH server, and initiates the connection using credentials.

ii- A local SOCKS proxy [56] (RFC 1928) service is then run on the device platform listening for requests issued from the device itself.

iii- The application that needs to access services and geo-fencing infrastructures, sends and receives its requests through the device-local proxy server.

Once the tunnel is established, all the communication is then passed through the local proxy. This local proxy will encrypt the whole packet using SSH connection parameters, and forward it to the destination as one or multiple packets.

Secure Socket Layer version 3.0 (SSLv3), or Transport Layer Security (TLS) [44], is a transport layer security protocol that provides with: 1) Confidentiality, 2) Message Integrity, and 3) Mutual authentication.

TLS supports multiple strong encryption techniques. Initial negotiation will be determining the confidentiality basics of a communication and random parameters that are set to prevent replay attacks. After the initial handshake, data is encrypted and is appended a Message Authentication Code (MAC), in addition to TLS specific headers for each packet. The encryption technique used can be changed in order to make breaking the ciphers more difficult. The details of TLS is out of the scope of this thesis, and available from the Internet Engineering Task Force [21].

Considering both methods provide identical techniques for providing communication security, networking advantages are in favour of TLS. Firstly, TLS is a mechanism that does not require extraneous authentication for initiating a secure connection. Although mutual authentication can be enforced on TLS connections, it is not necessarily required to provide secure communication, as client identification and authentication is already provided by the proposed system in other ways. Secondly, while TLS does only encrypt the payload and adds minimal control and integrity headers and trailers, SSH (due to port forwarding requirement) has to encrypt the whole packet, including TCP headers. This leads to a phenomenon called TCP over TCP, that is strongly discouraged due to its overhead when having small payloads [75], which applies to the proposed system's use of network resources.

Due to the aforementioned advantages of TLS over SSH, the proposed system is implemented on the TLS. Meanwhile, it should be noted here that the use of SSH is undeniable for management and remote control purposes and therefore might be useful for large scale deployment of the system.

5.2 Attacks and Countermeasures

The proposed system consists of multiple components that run under different platforms and use divergent set of system resources. With every dimension that is added to a software system through using a specific type of resource or mechanism, certain threats are arisen.

Network communication is an essential component of the whole system. Making resources available through well known network protocols brings flexibility and usability to the table. At the same time the well known contrast between usability and security limits the designer to employ techniques that are hidden from the user while at the same time provide sufficient security.

The proposed system can be subject to multiple attacks as it has multiple components and packets of sensitive information flowing over the network. Several attacks have been listed and tested against the implementation of the system. Many of these attacks are well known and commonly used against different types of Internet services. Some of the attacks are customary designed to leverage the specific design and the data flow of the system. However, even these custom attacks can be categorized in at least one of the well known categories. The generic categories that are studied in this research are as follows:

1. Brute Force and Token Guessing.
2. Denial of Service.
3. Man in The Middle.
4. Spoofing.

Sections 5.2.1 through 5.2.4 discuss these attacks and describe how the proposed system is designed to address these threats.

5.2.1 Brute Force and Token Guessing

Perhaps the least complicated and most trivial type of attacks are brute force attacks. These category of attacks are mainly designed to randomly or heuristically traverse the whole state space to guess the right parameters for acquiring access to a certain system. While user name and password guessing attacks are the most familiar ones in this category, brute forcing can be used in many other scenarios, too. Against the proposed system, an adversary can try to guess RSSI values that lead to a positive positioning outcome. Having a collection of those samples can give the adversary the ability of replaying those values to gain access. The RSSI spoofing/replay attacks are discussed in detail in Section 5.2.4. Out of the numerous parameters that can be guessed, I focus on the following:

Dictionary Attacks

As described before user names and passwords are used for authenticating the users. Brute force attacks are obviously a threat to the proposed system. Therefore the following approaches are taken to counter this type of brute force attacks:

1. Authentication is performed in a throttled manner. If a certain user name fails to authenticate successfully more than a specified number of times (threshold), the system blocks the user account. However, because the authentication server and the access control server are separate servers, the user based throttling is implemented separately from the device based throttling. CAS is responsible for throttling authentication requests based on user names. While the positioning service takes the responsibility to control the number of failed authentications given a certain device identity.
2. Enforcing security policies on password selection. This consists of forced inclusion of special characters and numbers, minimum password length, and expiration periods. These factors could increase the security and enlarge the state space which then results in a more time consuming and nearly infeasible process for an attacker. At the same time, this may cause decrease in the quality of user's experience because the passwords are going to be less and less memorable as the constraints get more strict. As a result, using a scheme which does not affect the user experience is suggested. To this end, the proposed system relies on tracking of request timings to limit the number of requests given by a user in a specific time frame.

Guessing Positive RSSI Samples

This is related to the number of RSSI values required for enabling positioning in each zone. In my experiments I have used from 3 access points to 6 dongles. Some experiments show that with some zone sizes even two dongles are enough. Each anchor node's RSSI value can approximately range from -36 to -90. Given this the size of the search space for the attacker is of size N^{64} in my experiments, when using N anchor nodes. Obviously, a range of values can be classified as inside zone. The zone size is small in comparison to the range which dongles or access points cover, but still the probability of a guess coming true is high.

In order for an adversary to guess enough samples to get away from the positioning step, it needs sufficient number of samples to evade being detected by the system's RSSI spoofing detection mechanism which is described in Section 5.2.4.

5.2.2 Denial of Service

Denial of Service attacks are defined as the category of attacks which are aimed to disable the system from providing service to its clients. Such attacks can be implemented based on using up the system resources. Such resources include network bandwidth, computational load on the server or client, and available information. Disrupting the information in a way that stops the system from serving its legitimate users is one of the most important threats that this framework can face. The important attacks that fall into this category are discussed as follows:

Fake Dongle/Access Point Installation

An adversary can bring in and install hardware in a way to pretend that the hardware is part of the positioning infrastructure. If successful, this will disrupt the quality of the data collected by devices to be sent to the server for positioning use. Duplicate cases could happen where a device discovers the same access point twice or more with different RSSI values. This attack can be classified in both Spoofing and DoS categories.

It can be classified as a DoS attack because it can cause the system fail to provide with accurate positioning. This will result in a stoppage in service because access to services in the area of geo-fencing are provided based on the position and the quality of this service solely depending on positioning accuracy and robustness.

The true nature of such an attack is spoofing. To impersonate an access point or dongle, the fake nodes need to send out beacons that are at least carrying the same MAC address of the others in their beacon frame headers [46].

Overwhelming Number of Requests

Depending on security and access control requirements, frequency of location checking can be different. In some areas that users are using devices to access sensitive resources, one administrator might choose to configure the positioning service in order to check for the location of users more frequently. But in almost all cases that the system is going to be deployed, a location aware service in retail environments, the frequency of location checks does not need to be more frequent than a few times per minute. Overcoming a large number of requests sent out by a single device is achieved by properly throttling these requests based on the frequency of location checking. If location checking requests from an identified device are submitted in violation of the timing restrictions, and keeps violating the restrictions more than a certain limit, user's account will be blocked temporarily or permanently based on the administrator's configuration.

BlueSmack Attack

Under the Bluetooth stack one of the most used transport protocols is Logical link control and adaptation protocol (L2CAP). This protocol is responsible for packet multiplexing, quality of service, segmentation and reassembly of the packets. Additionally, similar to the Internet Control Message Protocol's (ICMP) echo message, L2CAP also provides the echo functionality for discovery and availability purposes.

Some versions of the BlueZ [7] implementation are vulnerable against echo packets with an extraordinarily large (greater than 600 bytes) payload [45, 31]. Because the attack aims for the availability of the Bluetooth infrastructure, it is normally classified as a DoS attack.

The Bluetooth infrastructure is essentially used for indoor positioning. No data or service is provided through Bluetooth. More specifically, Bluetooth dongles are present for sending beacon frames that make the user devices able to have RSSI readings and then submit positioning requests. In order for the Bluetooth anchor nodes to propagate signals, the system uses the BlueZ stack implementation [7]. However, BlueZ is a complete implementation of the 802.15 standard [3]. This means that there are some default features that are included but are not used by the proposed system. Because unnecessary services and equipments are considered security risks, the BlueZ installation is configured not to operate on higher level protocols of the Bluetooth standard stack. This leads to ignoring all echo packets arriving at the anchor nodes, which makes the system resistant to such attacks.

5.2.3 Man in The Middle

Replay Attacks

As mentioned before in this section, communication confidentiality and message integrity is provided using TLS on the transport layer. On the application layer, HTTP post messages for positioning are checked for integrity using session salts. Two scenarios are considered for replaying positioning request data:

1) Replaying cipher data captured on the network.

2) Replaying plain text messages.

As of the first case, TLS is made replay proof by embedding two random elements generated by the client and the server in the handshake process. These two random values are then used to generate the master key that is used for the encryption algorithm chosen in the future steps of the TLS communication. Additionally, each TLS payload is accompanied by a Message Authentication Code (MAC) that is also dependent on the master encryption key. This makes replaying TLS ciphers nearly impossible. Another way of intercepting plain text data through a TLS connection is the SSLStrip attack [59], whose countermeasures are discussed in the text that follows.

For the second case, there is the prerequisite of being able to break the TLS cipher. This can be achieved by guessing the random numbers exchanged or having access to the device and server's private keys. However, if these dependencies are met, plain text POST data can be sent to the server. As mentioned before, the POST request data is protected against alterations using a message verification hash based on the $Session_{salt}$ and $Init_{salt}$ values. $Init_{salt}$ is permanent, in contrast $Session_{salt}$ is temporary and set to expire shortly. Each user needs to renew this salt value through a NFC barcode terminal whenever access is cut for a certain amount of time, or the salt is expired. As a result, the replaying adversary not having access to both the salt values will be unable to calculate the message verification hashes and fail to replay.

SSLStrip

In 2009 the SSLStrip was introduced. This attack exploits the weaknesses in the way browsers validate certificates and warn users about invalid certificates. Low level of awareness about SSL protection by the users is another effective factor that this type of attack can leverage.

The anatomy of SSLStrip involves accessing the network gateway, or forcing the clients in a network to direct their SSL traffic through an attacker. The latter is normally achieved by launching a successful Address Resolution Protocol poisoning (ARP poisoning) attack [48]. After doing so, the attacker forwards the client browser to a plain text communication between the rogue gateway and the client, while maintaining the SSL connectivity between the rogue gateway and the intended web server. SSLStrip gives the attacker the power to manipulate content and access the client-server communication data in plain text. Meanwhile, the server will not identify an attack because of the SSL connection present between the rogue gateway and the server. Moreover, the attacker hopes that the attack is unnoticed on the client side due to incorrect browser behaviour or a user's ignorance.

To counter SSLStrip, it is recommended to use bookmarked HTTPS URLs on browsers and not trusting invalid SSL certificates. The proposed system uses hard coded HTTPS URLs with HTTP redirect handling disabled. Certificates are set to be infinitely verified down the chain. Additionally, server certificates are hard coded with the IP address and domain name of the servers that they are installed on. Applying the mentioned measures will disable the attacker from launching a SSLStrip attack.

5.2.4 Spoofing

RSSI Spoofing

This type of attack involves an adversary replaying previously recorded positive RSSI samples. This attack is experimented and discussed in Section 6.5.

Tunnelling Based Location Spoofing

In this case, some threatening scenarios are studied where a certain decision could not be made about the true class of an activity. One such scenario is studied where an activity cannot be marked as an attack due to ambiguous nature of such practice. This problem also highly depends on the terms of service (ToS) that is agreed upon by a company and its customers.

This scenario includes a device, which is physically present in a geo-fenced zone, which is facilitating the access for users not present. Such a scenario can use the third or the fourth generation data networks to give access to users remotely connecting.

A remote user can connect to the geo-fenced services over a screen sharing protocol to play remotely. Or he/she can connect through a more sophisticated set of services such as remote command lines and small applications in order to obtain values such as RSSI and hashing salt, and then can start accessing the geo-fenced services while the facilitating node acts as a relay or a network proxy.

Such an attack can be countered by two strategies. Firstly, CSM can be configured such that no more than one user at a time is able to play using a physical hand held device. Secondly, one can use the fact that in 3G networks, up-link has a smaller delay than the down-link [55]. This helps detecting such an activity when a tunnelled connection is used by the remote user to play through the facilitating device.

As mentioned before, such a scenario is highly dependent on regional legislative decisions in addition to the company policies and Terms of Service. Such situations further emphasize the need to expand legal studies in order to keep up with the evolution of the digital computing environments.

Fake Access Point/Dongle Attacks

The geo-fencing system is dependent on RSSI values for determining the position and as a result permissions of a user. One way for an attacker to disrupt this process is introducing fake Wi-Fi or Bluetooth anchor nodes. Such an attack can be achieved by running a Wi-Fi access point or Bluetooth hotspot that advertises services with the same MAC address or SSID (Service Set Identifier) as one or more of the anchor nodes employed in the geo-fencing infrastructure.

To counter such an attack, the Bluetooth hardware used in the wireless infrastructure constantly scans the environment for beacons from other anchor nodes. Each node will compare each observed MAC address and/or SSID to its own MAC and SSID. If any of the anchor nodes observe such a duplicate, an alert is sent to the administrator for further investigation.

Chapter 6

Experiments and Results

This chapter of the thesis focuses on the experiments that were carried out during the research. At a glance, multiple discussions are brought up about different aspects of the machine learning algorithms and how to extract useful and domain relevant information from these algorithms. Additionally, some statistical measures are introduced to counter specific types of custom attacks targeting RSSI based indoor positioning systems.

6.1 Classification Results

As mentioned before, based on my preliminary experiments (see Appendix A), the two best classifiers were C4.5 and Random Forest. Therefore, I compared these two classifiers against each other on my data sets for this research. These data sets collected in NIMS lab area are described in Table 6.1.

Table 6.1: Geo-fencing data set description.

Data Set Name	Positive Samples	Negative Samples	Devices
Wi-Fi 2x2 all devices	199	204	7 different Android devices
Wi-Fi 2x2 TF101	41	41	Asus TF101
BT 2x2 Samsung	50	50	Samsung Galaxy Ace
BT 2x2 TF101	32	81	Samsung Galaxy Ace and Asus TF101
Wi-Fi 5x5 TF101	50	51	Asus TF101
Wi-Fi 5x5 Samsung	50	50	Samsung Galaxy Ace
BT 5x5 Samsung	50	51	Samsung Galaxy Ace
Wi-Fi 10x10 TF101	50	50	Asus TF101
Wi-Fi 10x10 Samsung	50	50	Samsung Galaxy Ace
BT 10x10 Samsung	50	51	Samsung Galaxy Ace
BT 10x10 all devices	50	51	A variety of devices
BT 10x10 TF101	50	50	Asus TF101

Each classifier was run on all data sets using 10 fold cross validation. Multiple runs were performed using different random seeds to ensure that the results are not biased.

The main metric used for selecting the best classification algorithm is the $F_{Measure}$. This is due to the discriminative aim of classification in this research. This goal is achieved by minimizing the number of false negative and false positive predictions at the same time. To understand $F_{Measure}$, one needs to first know Precision and Recall. Precision and Recall are well known measures that are widely used in information retrieval and data mining practices. Precision and Recall are defined in Equations 6.1 and 6.2 respectively.

$$Recall = \frac{TP}{TP + FN} \tag{6.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

Needless to say, Recall decreases as the number of the false negatives (FN) increases. On the other hand, Precision decreases when the number of false positives (FP) increases. So there is a trade off between Recall and Precision. To build a discriminative classifier, both Precision and Recall must be maximized as much as possible.

$$F_{Measure} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6.3}$$

Equation 6.3 is used to calculate the $F_{Measure}$ value. Clearly, this measure is maximized to 1 when both Precision and Recall have higher values. Consequently, $F_{Measure}$ a suitable measure for evaluating a classifier to be discriminative.

As mentioned before, multiple classifiers are employed in our preliminary experiments.. These classifiers include: Naive Bayes, Support Vector Machines, C4.5 decision trees, and Random Forest. Based on the results of these initial tests, I selected C4.5 and Random Forest classifiers for further evaluations. Weka machine learning tools kit [15] was used for both initial testing and software integration purposes.

For further evaluations, both classifiers are first fine tuned for all data sets based on linear parameter search. Each classifier is run multiple times while changing parameter values in a linear manner. The best parameters for the top two classifiers being used are as follows:

- **Random Forest**: 250 trees in the forest, each split point is based on 2 random features.
- **C4.5 Decision Tree**: Confidence factor of 0.25 for final pruning.

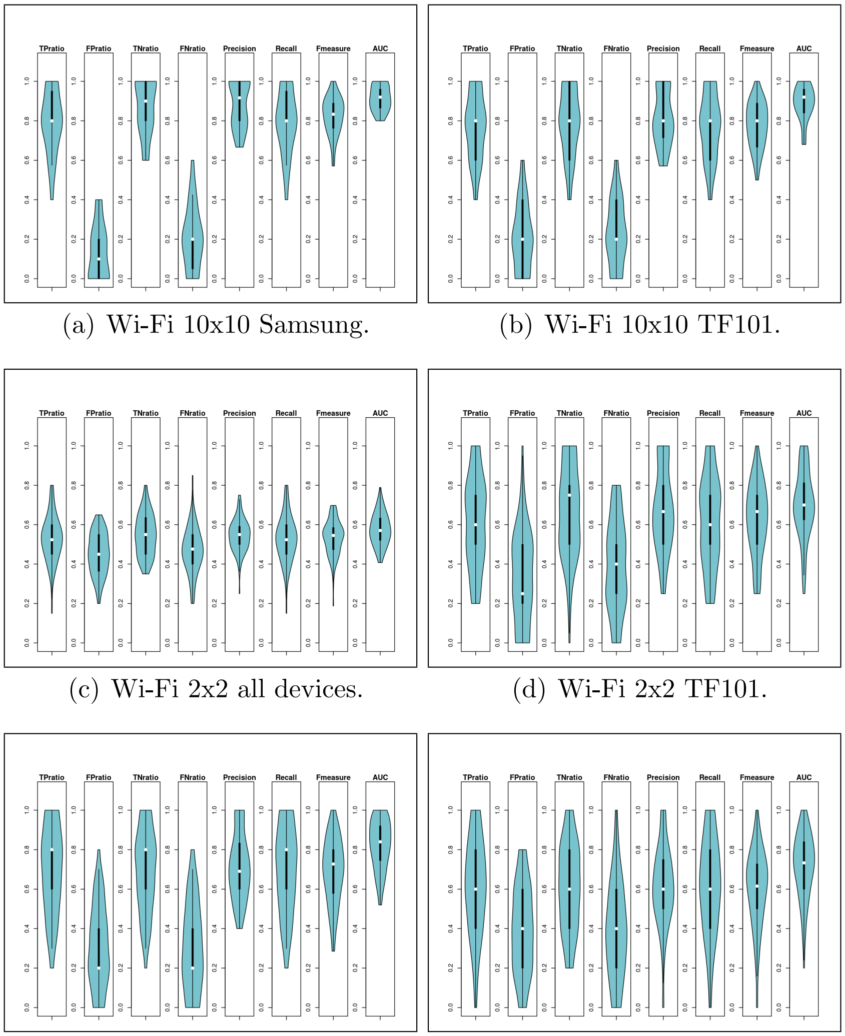
Figures 6.1 and 6.2 demonstrate the performance of the Random Forest classifier on Bluetooth and Wi-Fi infrastructure data sets respectively, whereas Figures 6.3 and 6.4 illustrate the performance of the C4.5 classifier on the same data sets. These classification results suggest that the Random Forest classifier has a better performance on all data sets. In these experiments Random Forest classifier has a final F-measure of 0.72 in average on all data sets as opposed to C4.5 having a F-measure of 0.67 (before smoothing). Therefore, Random Forest is chosen to be the main classification algorithm for the proposed system. From this point on all the experiments that are concerned with classification are run using the Random Forest algorithm.

6.2 Smoothing Results

The data collection step is basically operated by administrators, and it depends on user input for reading and recording the RSSI values. But in runtime, the testing applications that operate on user devices are configured to record Wi-Fi readings in timed periods (5 seconds for our tests). On the other hand, due to the event based design of Bluetooth scanning in Android, setting a smaller recording frequency is needed to achieve uniformly recorded samples. To experiment the effect of data collection timings on the smoothing process, a new Bluetooth data set for the 2x2 zone size was collected. After performing experiments on both the new and the old 2x2 data sets results turn out to be nearly the same. This is because most of the employed smoothing methods are assuming that values are not temporally evenly distributed. As a result, no new data set is collected for other zone sizes to conduct smoothing experiments.

For the smoothing method evaluations, two extensive sets of experiments were carried out. The first set of experiments were aimed to determine the best smoothing method for the existing data sets. Then, the second round of experiments are designed to discover the best parameter values for the method chosen in the previous round of experimentation.

In the first round of experiments each of the 12 data sets are smoothed using all the methods described in Chapter 4. Based on average $F_{Measure}$ of all runs on each data set,



(a) Wi-Fi 10x10 Samsung.

(b) Wi-Fi 10x10 TF101.

(c) Wi-Fi 2x2 all devices.

(d) Wi-Fi 2x2 TF101.

(e) Wi-Fi 5x5 Samsung.

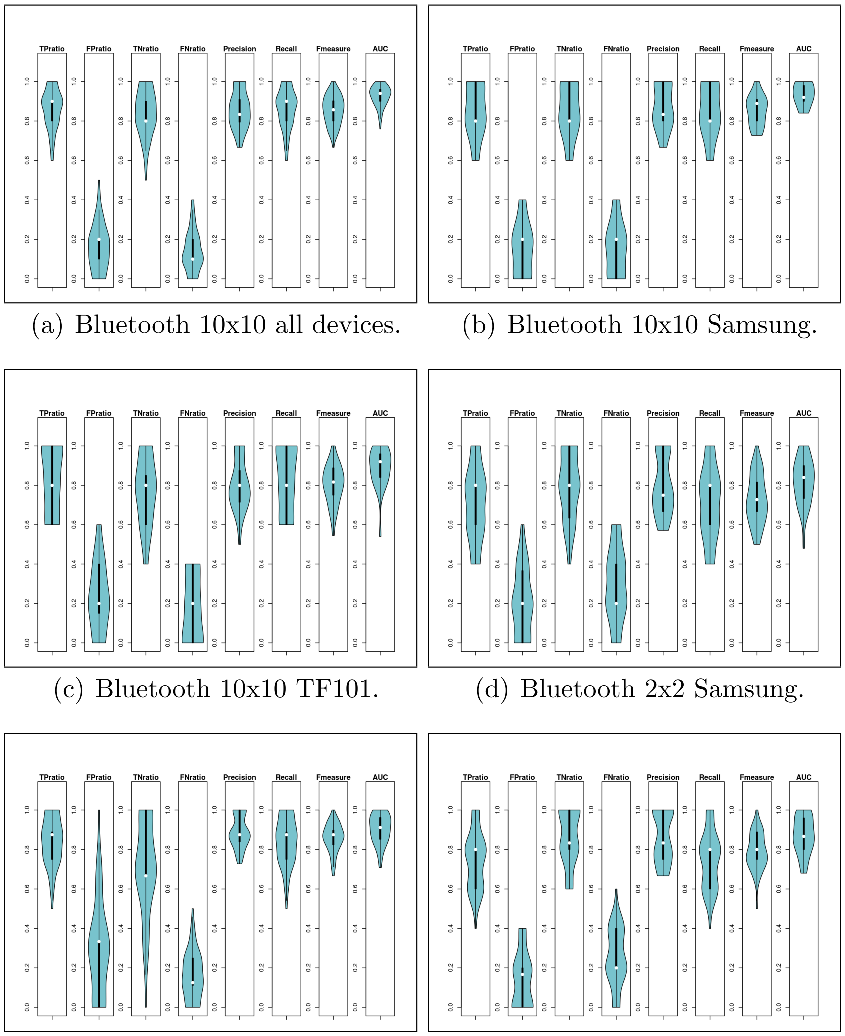
(f) Wi-Fi 5x5 TF101.

Figure 6.1: Random Forest result distribution on non-smoothed data sets while using the Wi-Fi infrastructure.

the best smoothing algorithm is selected. To do so, smoothed datasets are sorted based on average $F_{Measure}$ then the method with the best ranks is chosen as the main algorithm.

These results show that the Moving Average has shown more steadiness in performance across different data sets. Figures 6.7 and 6.8 show the results of classification on 5x5 zone using the Wi-Fi infrastructure. Appendix A includes detailed experimental results on all data sets. Table A.2 tabulates the average classification measures for all the methods on all data sets for different smoothing methods.

The smoothing process is performed in a few steps. First, data sets are loaded into



(a) Bluetooth 10x10 all devices.

(b) Bluetooth 10x10 Samsung.

(c) Bluetooth 10x10 TF101.

(d) Bluetooth 2x2 Samsung.

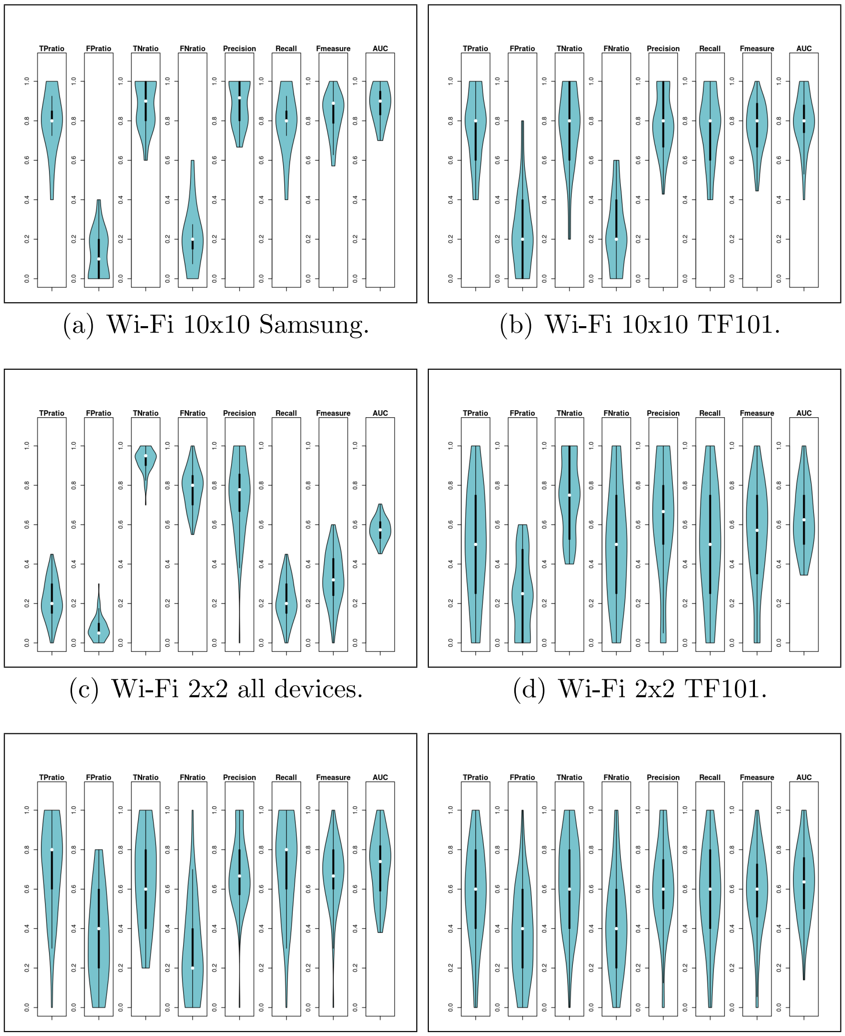
(e) Bluetooth 2x2 TF101.

(f) Bluetooth 5x5 Samsung.

Figure 6.2: Random Forest result distribution on non-smoothed data sets using the Bluetooth infrastructure.

Matlab environment from CSV file formats. Sorting the samples based on their reading time, each data set represents a recorded time series of RSSI readings. Then, starting from the beginning of the sorted data set, the algorithms aim to adjust the value of next sample based on the samples observed up to that point. Finally, after smoothing filter is applied, the data sets are saved into separate CSV files for classification purposes.

Applied smoothing techniques (filters) have a delay for starting to remove the noise. This delay is in direct relation to the $Window_{Size}$ selected. As mentioned before, there is a trade-off between the delay for a filter to start its impact on the values and the amount of



(a) Wi-Fi 10x10 Samsung.

(b) Wi-Fi 10x10 TF101.

(c) Wi-Fi 2x2 all devices.

(d) Wi-Fi 2x2 TF101.

(e) Wi-Fi 5x5 Samsung.

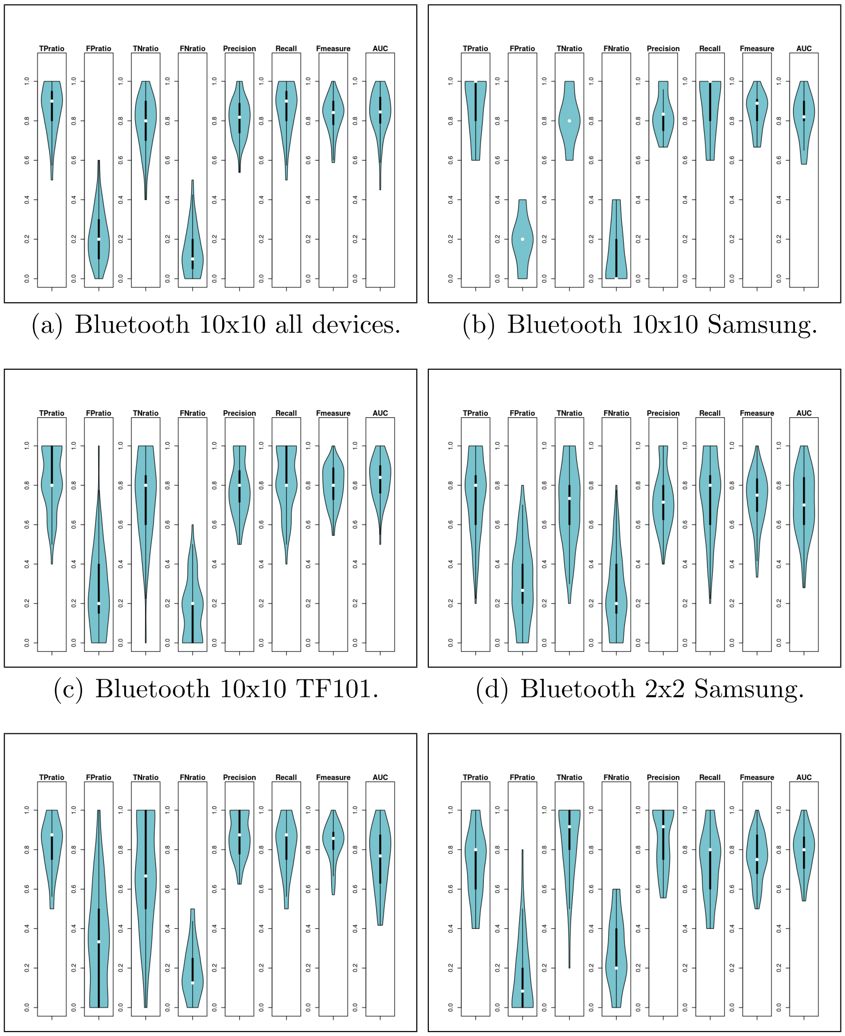
(f) Wi-Fi 5x5 TF101.

Figure 6.3: C4.5 decision tree result distribution on non-smoothed data sets while using the Wi-Fi infrastructure.

the noise that is removed by the filter.

In each data set smoothing is performed separately for each attribute and also separately for each label, because each attribute value is technically independent from the other values. A user's location and orientation in relation to a specific anchor node is highly effective on the noise. Additionally, the noise on each anchor node is also independent because they are operating on different frequency channels.

To further examine the moving average method, experiments have been undertaken with different values of $Window_{Size}$. The $Window_{Size}$ has been changed from 5 to 50 readings.



(a) Bluetooth 10x10 all devices.

(b) Bluetooth 10x10 Samsung.

(c) Bluetooth 10x10 TF101.

(d) Bluetooth 2x2 Samsung.

(e) Bluetooth 2x2 TF101.

(f) Bluetooth 5x5 Samsung.

Figure 6.4: C4.5 decision tree result distribution on non-smoothed data sets while using the Bluetooth infrastructure.

Then the Random Forest classifier is run and results are compared based on average $F_{Measure}$. Figures 6.5 and 6.6 illustrate the effect of the changes in $Window_{Size}$ on the results of classification experiments. Based on these experiments, a $Window_{Size}$ of between 20 and 30 is shown to be the best for both Bluetooth and Wi-Fi data sets. Additionally, Table A.3 lists detailed classification results achieved on all data sets when changing the smoothing $Window_{Size}$ from 5 to 50.

To integrate the Moving Average method into a system that operates on live data streams rather than pre-recorded data sets, the CSM (Section 3.2.4) maintains a Circular FIFO Buffer

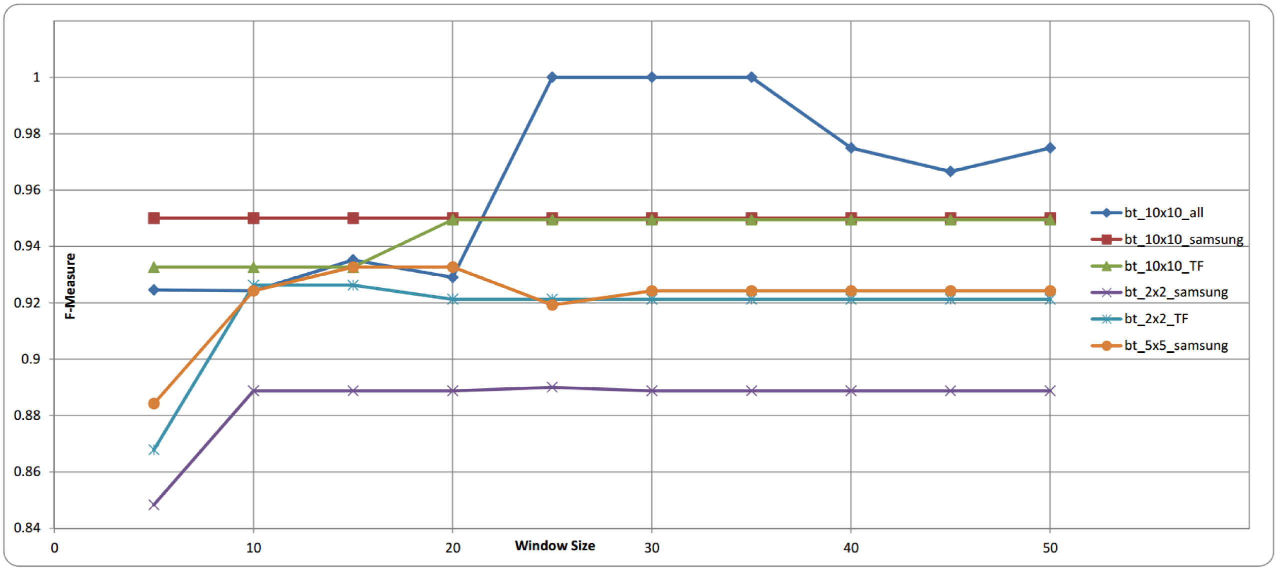


Figure 6.5: Classification results for different Bluetooth data sets vs. changing the moving average $Window_{Size}$.

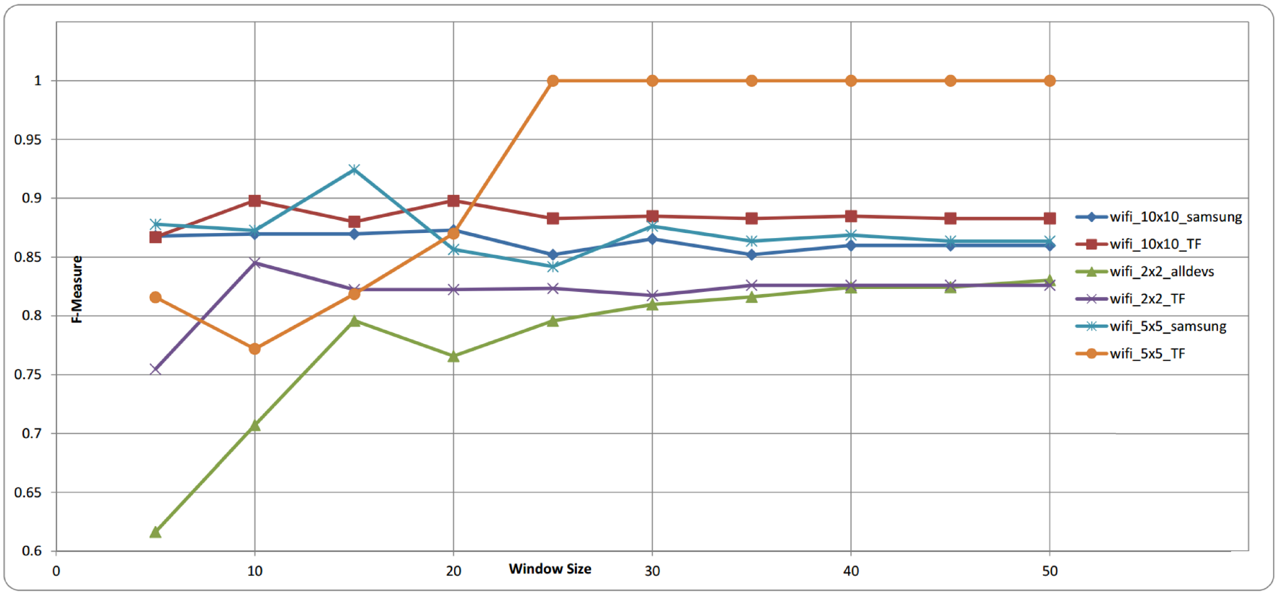
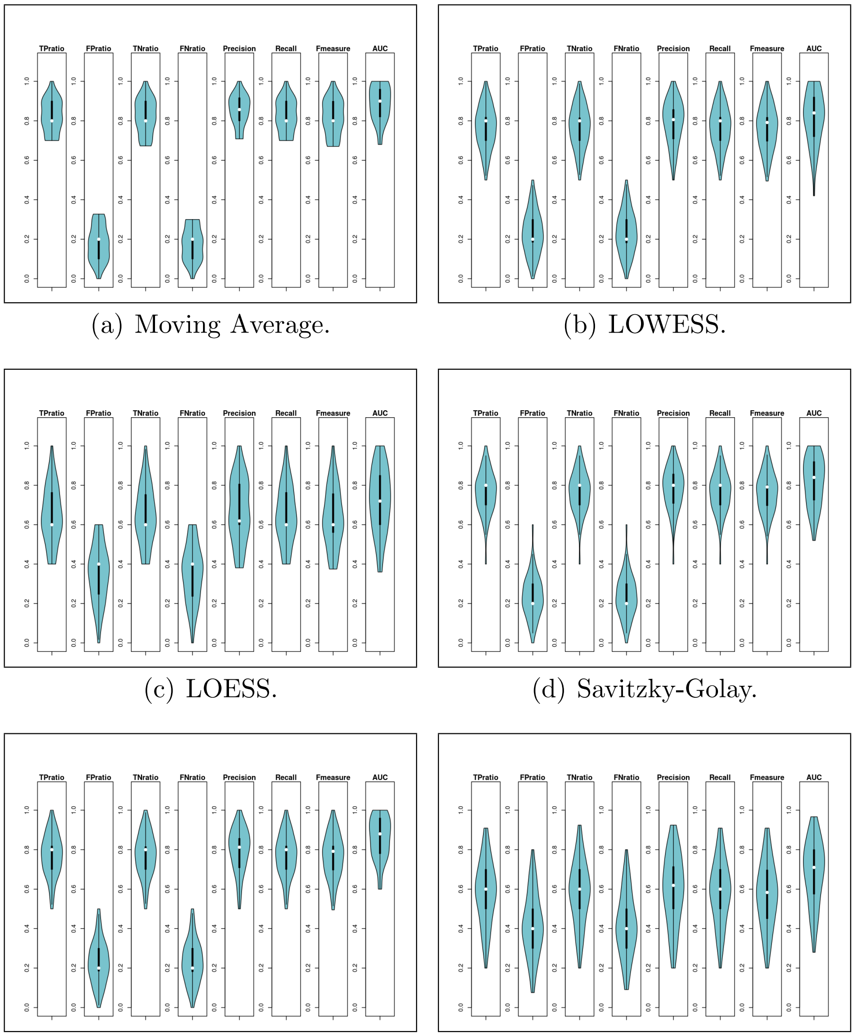


Figure 6.6: Classification results for different Wi-Fi data sets vs. changing the moving average $Window_{Size}$.

with a size equal to the $Windos_{Size}$ parameter. When each sample arrives the average RSSI over the smoothing window is updated on the fly, and then an estimate is made based on the Moving Average method.



(a) Moving Average.

(b) LOWESS.

(c) LOESS.

(d) Savitzky-Golay.

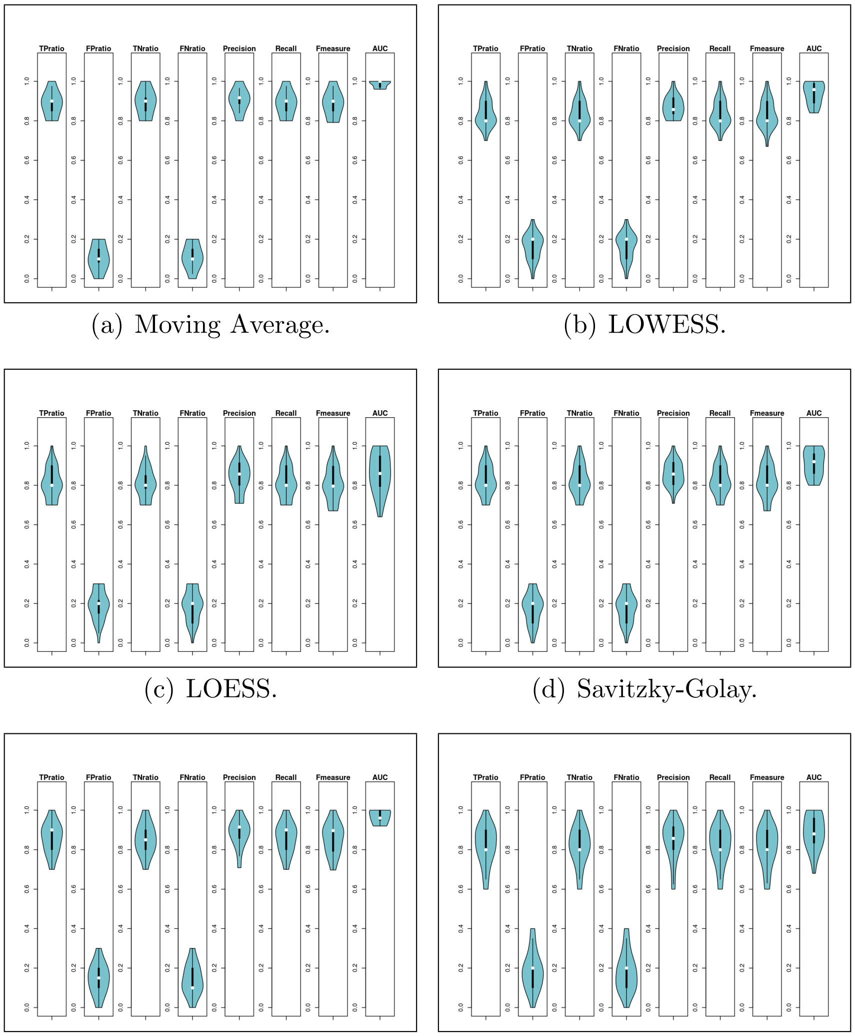
(e) Robust LOWESS.

(f) Robust LOESS.

Figure 6.7: Random Forest result distribution on Wi-Fi 5x5 data set collected using an Asus Tablet, using different Smoothing methods.

6.3 Most Important Anchor Nodes

Although the Random Forest classifier yields better classification results, the structure of a C4.5 decision tree can be helpful, too. To find the most important anchor node in the positioning decision process, a C4.5 tree is trained on the data set. Then the most important anchor node that is the anchor node that is placed in the root node of the tree is analysed. This is because features are ranked based on information gain when the decision tree is choosing the next attribute to recursively build the tree. After finding the most important



(a) Moving Average.

(b) LOWESS.

(c) LOESS.

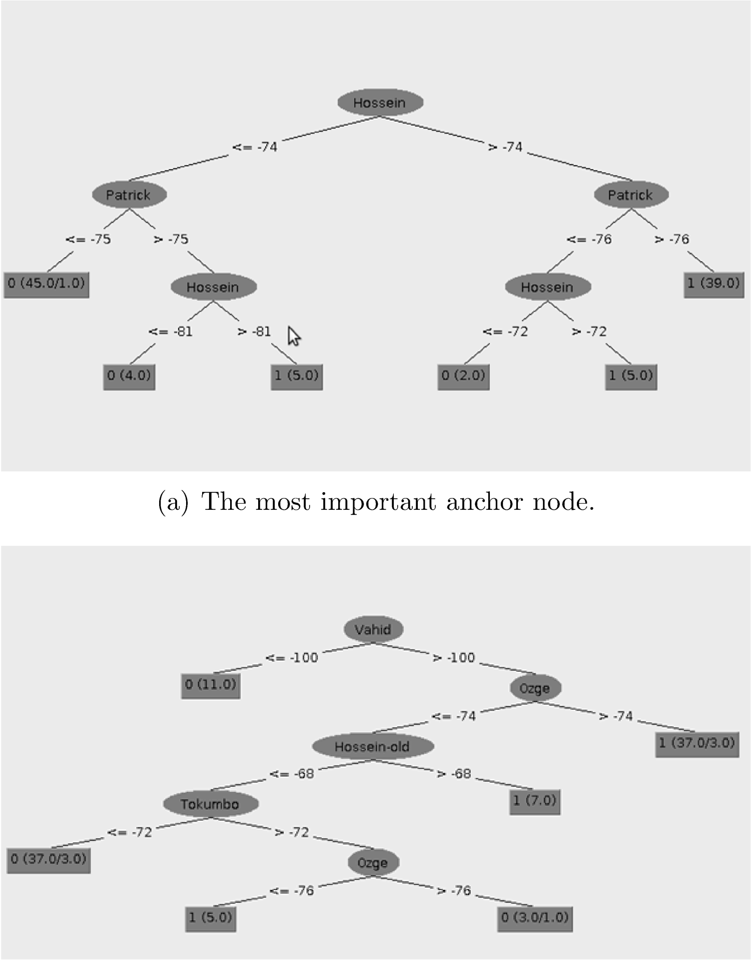
(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure 6.8: Random Forest result distribution on Bluetooth 5x5 data set collected using a Samsung Smartphone, using different Smoothing methods.

anchor node, the corresponding attribute is removed from the data set and a new decision tree is built on the remaining attributes, leading to selection of the next most important anchor node and so on. This process is repeated until no less than two attributes remain. Figure 6.9 demonstrates the decision tree visualizations for the first and the second most important anchor nodes on a 2x2 geo-fenced zone using Bluetooth infrastructure. This shows that which anchor node is the most influential in the geo-fencing decision process. Studying characteristics of their placement in relation to the geo-fenced zone being studied can give us clues in order to perform more accurate geo-fencing installations. Information from this



(a) The most important anchor node.

(b) The second most important anchor node

Figure 6.9: The first and the second most important anchor nodes.

phase together with another method described in the next section, can be of essence for saving on infrastructure hardware and optimize the infrastructure for a better performance.

6.4 Best Anchor Node Positions

In general, one might think that in indoor positioning, having more anchor nodes leads to a more accurate positioning result. Although this might seem true, because of having less blind spots and more reference points, it is not necessarily true.

To experiment with the position and the number of access points required, I employed the following technique in this thesis. To find the best positions for installation of anchor nodes in relation to the geo-fencing zone, the site survey process is started with an arbitrary

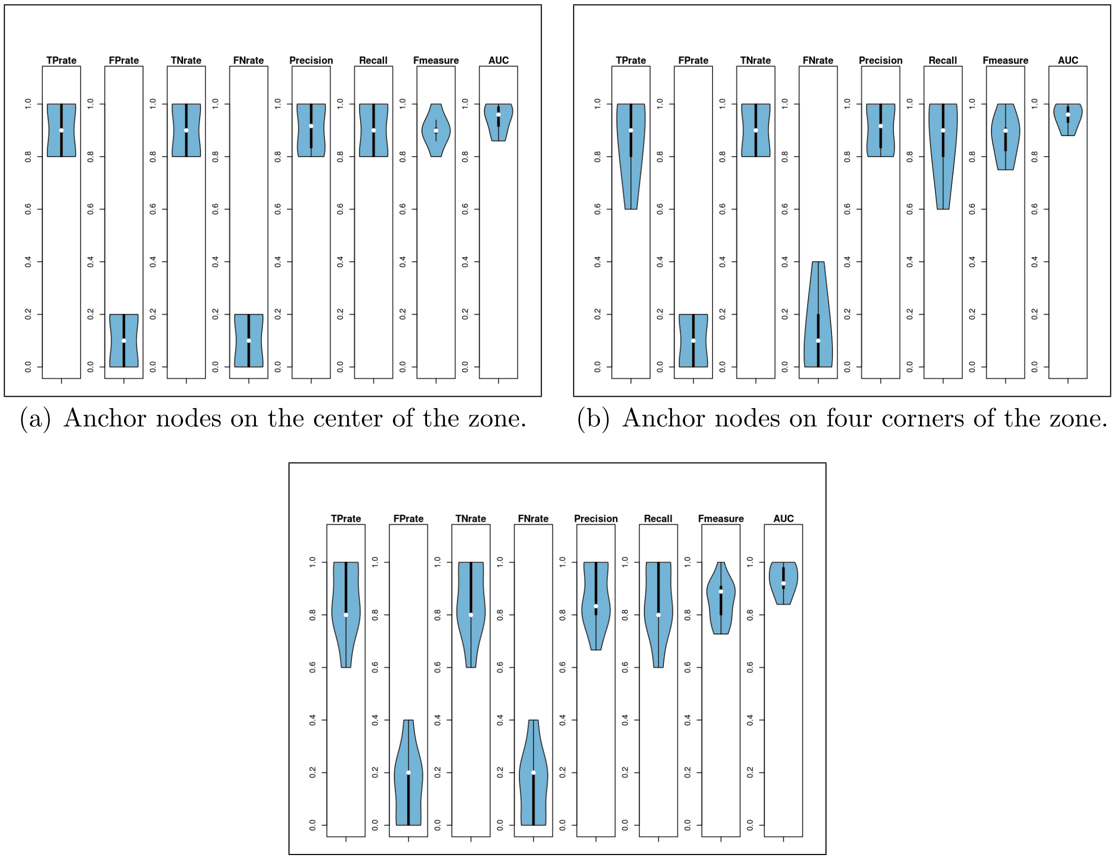
Table 6.2: Top 10 anchor node positions for the 5x5 geo-fenced zone while using the Bluetooth infrastructure for positioning.

Count	**Anchor Nodes**	$\mathbf{F}_{Measure}$ **(average)**
3	Tokunbo, Hossein, Hossein-Old	0.78
6	Patrick, Vahid, Tokunbo, Hossein, Ozge, Hossein-Old	0.77
4	Tokunbo, Hossein, Ozge, Hossein-Old	0.76
4	Vahid, Tokunbo, Hossein, Hossein-Old	0.76
1	Hossein	0.75
5	Patrick, Tokunbo, Hossein, Ozge, Hossein-Old	0.75
4	Patrick, Tokunbo, Hossein, Hossein-Old	0.75
5	Vahid, Tokunbo, Hossein, Ozge, Hossein-Old	0.747
3	Hossein, Ozge, Hossein-Old	0.746
5	Patrick, Vahid, Hossein, Ozge, Hossein-Old	0.74

number of anchor nodes. When a data set is collected, a tool called "Subseteer" is used to find the best formation of access points. Subseteer first creates subsets of the data set and trains the Random Forest classifier on each of the subsets. Then, each subset is tested for performance using 10 fold cross validation. These test results are then sorted based on $F_{Measure}$ in order to choose the most discriminative model.

Table 6.2 tabulates the top 10 results from a run of Subseteer on a Bluetooth data set collected on a 5x5 meter zone. Apparently, having only three of the anchor nodes (Tokunbo, Hossein, and Hossein-Old) is enough for performing a slightly better positioning than the whole set of anchor nodes. Considering Figure 3.6, anchor nodes "Hossein" and "Hossein" are close to the border of the 5x5 zone. And anchor node "Tokunbo" is far from the zone in a place where users have the freedom to move when they are outside the zone perimeter.

Another run of Subseteer on a 10x10 Bluetooth data set denotes that the top 2 formations are (Hossein, Hossein-Old) and (Vahid, Tokunbo, Patrick, Ozge). The (Hossein, Hossein-old) subset includes two anchor nodes from the center of the zone. On the other hand, the (Vahid, Tokunbo, Patrick, Ozge) subset, includes the nodes located at four corners of the zone being experimented. Figure 6.10, demonstrates the classification results for both the top subsets relative to the original data set. In this case, the $(Hossein, Hossein - Old)$ subset performs best by only having two anchor nodes. This denotes the importance of the infrastructure optimization for having an accurate geo-fencing engine, which is provided by the proposed system.



(a) Anchor nodes on the center of the zone.

(b) Anchor nodes on four corners of the zone.

(c) All anchor nodes considered.

Figure 6.10: Random Forest results considering different positions of anchor nodes on a 10x10 Bluetooth zone.

6.5 Brute Force and Spoofing Experiments

In this case, an adversary present in a geo-fence controlled environment can access a service outside the specific geo-fenced zone. In order to do so, he/she needs to collect a set of positive samples to be used for replaying when a spoofing attack is launched. By nature, guessing positive RSSI values is more successful than password brute forcing, because guessing a positive value is more probable. In contrast, for a password guessing attack, there is only one correct answer for the whole process [1].

In this case, there are different methods to detect or disable a brute force attempt. The main method for delaying brute force attacks is to throttle the number of requests a user can submit to the server. As mentioned before, CSM keeps track of request timings from

[1] Not considering hashing algorithm conflicts if the password hash is attacked.

each user. CSM is designed to be configurable with the following parameters:

i) **Minimum Request Delay** This determines the minimum time that a user needs to wait to submit a new positioning request.

ii) **Maximum Number of Violations** This determines the maximum number of times a user can violate a constraint before the account is blacklisted, including the violation of Minimum Request Delay.

iii) **Time in Blacklist** This is the duration that the user will be blacklisted. This could be set to infinity by setting a negative value or exponential blacklisting times, if it is set to a positive value.

Generating signal data randomly, an attacker attempts to keep the positive samples for future replay. Comparing the randomly generated signals to regular users' behaviour according to the data buffers that CSM records can distinguish users from attackers. For each client, a long term history of RSSI values per anchor node is stored. This history is held as a set of unique RSSI values each client has sent for positioning requests.

Because of the nature of a retail environment and human movement speed, a regular user sending legitimate values is limited to a number of possible values in a specific time frame. However, because the attacker is generating random values aiming to cover the state space, his/her chosen RSSI values are not as limited as a benign user. To differentiate these two behaviours, CSM keeps track of the growth rate of the each anchor node's value set per each user. The growth rate is calculated as shown in Equation 6.4.

$$Growth_{rate} = \frac{Size(ValueSet)}{TotalNumberofRequests} \tag{6.4}$$

At the beginning of a session the set is empty. Consequently, the growth rate is high and close to 1. After a number of values are submitted to the server, a user's growth rate starts to fall, because new unique values are less likely to be added to the set. This is because in normal conditions, a user's behaviour most probably will consist of movements which are limited in speed and distance [39, 83]. Experiments show that normal users' growth rate falls much earlier and faster than that of a random brute forcing attacker.

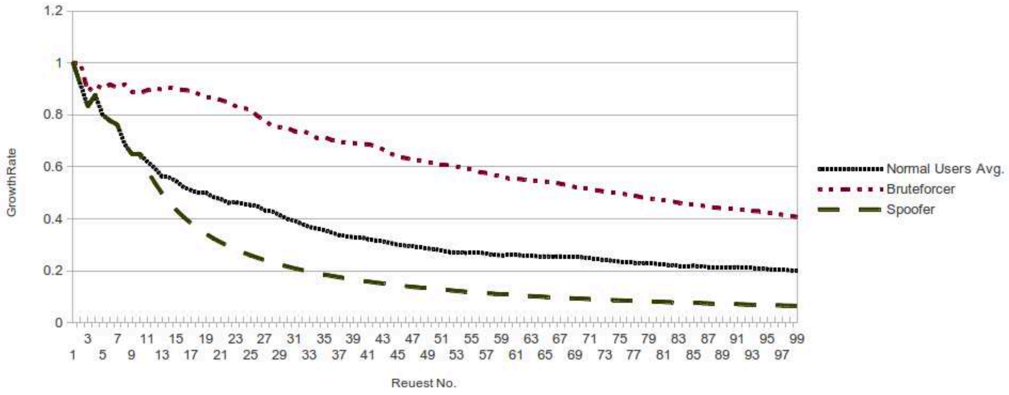


Figure 6.11: Trend of $Growth_{rate}$ for RSSI Spoofing, Brueforcing ,and a regular users in a 2x2 zone monitored using Bluetooth RSSI values.

Assuming that attacks are anomalous activities, detection of such behaviour is based on the following principles:

1 Value sets are reset at every $Flush_{Point}$ requests.

2 A user must have submitted at least $Min_{Requests}$ requests to become eligible for the detection process.

3 A user is reported as suspicious when its average $Growth_{rate}$ is higher than the average of $Growthrate$ over all the eligible users with a distance of at least 3 Median Absolute Deviation (3MAD) of average growth rates.

Intuitively, the variations observed in the signal values is also dependent on the geo-fencing zone sizes, as the user has more freedom to move and send a more diverse set of values in larger zones. To this end, $Flush_{Point}$ and $Min_{Requests}$ can be tuned to suit different zone sizes.

Equation 6.5 demonstrates the trend of average $Growth_{rate}$ for three different users over 100 positioning requests. A brute force attacker is sending random RSSI values between -30 and -90. A regular user is normally moving or standing inside the geo-fenced zone. A spoofer is replaying 10 positive samples while staying outside the geo-fenced zone. In summary, this shows that both the RSSI spoofing and the RSSI brute forcing attacks can be distinguished using the unique signal values sent by each user. Brute forcing attacks tend to have a more diverse set of values, while spoofers will have less deviation because of their use of a limited number of previously recorded samples.

6.6 Improving The Smoothing Process

Although smoothing can improve the classification performance by eliminating noise and outliers, it has specific drawbacks in the geo-fencing system. The most important drawback is the transition of a user from being inside to going outside or vice versa. Even a relatively small window size (between 5 to 15 data samples) will cause the estimates to strongly reflect the previously observed conditions. As a result, this will improve user experience by removing sudden decision changes, but will introduce the risk of giving access to resources while the user is outside a zone. As mentioned before, a weighted moving average method can help result in making predictions in favour of the latest data points. However, due to the significant amount of noise, this may also make the smoothing method prone to failing when outliers or severe shootings are introduced to the sensors.

To address this issue, I use the information stored by CSM. In Figure 3.8 two circular FIFO buffers with sizes equal to the smoothing window size are illustrated. The "Pure RSSI Values" buffer stores the RSSI data points as received from a client, opposite is the "Smoothed RSSI Values" buffer, which stores the RSSI values that are the output of the smoothing algorithm, Moving Average. A third buffer, "Predictions History", stores the three latest predictions. As one can see in Figure 3.8, this information is stored for each client separately. Based on these data points, two predictions are made per each geo-fencing request:

1) A prediction made by the classifier trained on the raw data based on the latest raw data point.

2) A prediction made by the classifier trained on the smoothed data based on the latest smoothed data point.

In cases where these two decisions disagree the final decision is made in favour of the smoothed decision maker. However, whenever the two classifiers have disagreed more than a k number of times, the decision will be put in favour of the raw classifier and all smoothing windows will be flushed. By flushing the smoothing windows, the effect of the smoothing history on the latest changes will be removed. This can also balance the training and testing phases for the classifiers. As mentioned before, positive and negative samples are smoothed independently. So, flushing the buffers at the time a user crosses a border can bring the same smoothing strategy to runtime.

Taking this approach results in a small delay when switching from the ”being present” condition (in the zone) to the ”being absent” condition (not in the zone) in the geo-fenced zone. In return, two negative factors are removed at the same time. Firstly, the spontaneous behaviour of preempting a user's access caused by noise and outliers is eliminated. Secondly, the smoothing drawback of making predictions relative to the past is also addressed. Experiments show that setting $k = 3$ will lead to a good balance between switching time and a smooth behaviour.

Chapter 7

Conclusions and Future Works

In this thesis an indoor geo-fencing and access control system is proposed and then studied in noisy and insecure environments. Measures and modifications are applied to make the system more robust and secure.

Robustness is achieved by applying smoothing algorithms to RSSI data read by the wireless adapters equipped in wireless devices in the geo-fencing environment. Smoothing removes outliers and reduces the spontaneous changes in decisions made by the positioning system. Negative effects of smoothing is addressed using different classifiers on different data sets at the same time. Results show that smoothing not only improves the behaviour of the software, it also improves the average accuracy up to 100%.

To the best of my knowledge, this is the first work that studies threats faced by a real world deployment of indoor location aware access control. Security of the system is assured by adding throttling and per user statistical analysis. Many of the commonly known attacks are countered by using mechanisms such as static ARP entries, and request throttling. However, to address system specific attacks including RSSI value brute forcing and spoofing, new measures and detection mechanisms such as outlier detection upon RSSI value $Growth_{rate}$ and infrastructure monitoring are introduced. Detection of fake Wi-Fi access points and Bluetooth hotspots is also built-in to the system.

The proposed system architecture has been filed for a US provisional patent by the industrial partner and a paper describing early stages of the design and implementation is published in IEEE CICS'2013.

To further improve the system, new factors and features for positioning can be taken as the next future direction. Such factors can include: hybrid positioning based on both Wi-Fi and Bluetooth, a user's direction (magnetometer sensor), as well as integration of NFC for two way proximity based communication.

Bibliography

[1] A brief history of wi-fi. *The Economist*, 2004(Q2).

[2] Ieee standard for information technology- telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-1997*, pages i–445, 1997.

[3] Ieee standard for telecommunications and information exchange between systems - lan/man - specific requirements - part 15: Wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (wpans). IEEE Standards C/LM-LAN/MAN Standards Committee, 2002.

[4] Ieee standard for telecommunications and information exchange between systems - lan/man - specific requirements - part 15: Wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (wpans). *IEEE Std 802.15.1-2002*, pages 1–473, 2002.

[5] Byod trend pressures corporate networks. `http://www.eweek.com/c/a/Mobile-and-Wireless/BYOD-Trend-Puts-Pressure-on-Corporate-Networks-186705/`, May 2011.

[6] About cas — jasig communitys. `http://www.jasig.org/cas/about`, September 2012.

[7] BlueZ. `www.bluez.org`, December 2012.

[8] Cas proxy authentication documentation. `www.jasig.org/cas/proxy-authentication`, September 2012.

[9] Ekahau - wi-fi tracking systems, rtls and wlan site survey. `http://www.ekahau.com`, September 2012.

[10] Glopos - a revolution in indoor positioning technology. `http://www.glopos.com/site/`, September 2012.

[11] Glopos indoor positioning accuracy tests conducted by vtt. `http://www.glopos.com/site/technology.html`, September 2012.

[12] Josso official project website. `www.josso.org/confluence/`, September 2012.

[13] List of single sign on implementations. `en.wikipedia.org/wiki/List_of_single_sign-on_implementations`, September 2012.

[14] Random forests (website). `http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm`, Oct 2012.

[15] Weka 3 - data mining with open source machine learning software in java. http://www.cs.waikato.ac.nz/ml/weka/, May 2012.

[16] Wifi based rtls solutions and wireless sensor technologies by aeroscout. http://www.aeroscout.com/, September 2012.

[17] Aircrack-ng. http://www.aircrack-ng.org/, Jun 2013.

[18] Maven. http://maven.apache.org/, August 2013.

[19] Oauth community site. http://oauth.net/, Jun 2013.

[20] Openssh. http://www.openssh.org/, Jun 2013.

[21] Transport layer security (tls) charter. http://datatracker.ietf.org/wg/tls/charter/, Jun 2013.

[22] ZigBee Alliance. Zigbee specification. *ZigBee Document 053474r13*, pages 344–346, 2006.

[23] Marco Altini, Davide Brunelli, Elisabetta Farella, and Luca Benini. Bluetooth indoor localization with multiple neural networks. In *Proceedings of the 5th IEEE international conference on Wireless pervasive computing*, ISWPC'10, pages 295–300, Piscataway, NJ, USA, 2010. IEEE Press.

[24] William A Arbaugh et al. *Real 802.11 security: Wi-Fi protected access and 802.11 i*. Addison-Wesley Longman Publishing Co., Inc., 2003.

[25] William A Arbaugh, Narendar Shankar, and YC Justin Wan. Your 802.11 wireless network has no clothes, 2001.

[26] Kwang-Hyun Baek, Sean W Smith, and David Kotz. A survey of wpa and 802.11 i rsn authentication protocols. *Dartmouth Computer Science Technical Report2004*, 2004.

[27] Paramvir Bahl, Venkata N. Padmanabhan, and Anand Balachandran. Enhancements to the radar user location and tracking system. Technical report, 2000.

[28] Rafael Ballagas, Michael Rohs, Jennifer G. Sheridan, and Jan Borchers. Byod: Bring your own device. Technical report, Media Computing Group, RWTH Aachen University, 2004.

[29] A. Baniukevic, D. Sabonis, C.S. Jensen, and Hua Lu. Improving wi-fi based indoor positioning using bluetooth add-ons. In *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, volume 1, pages 246 –255, june 2011.

[30] Roberto Battiti, Thang Le Nhat, and Alessandro Villani. Location-aware computing: A neural network model for determining location in wireless lans. Technical report, 2002.

[31] Andreas Becker and Ing Christof Paar. Bluetooth security & hacks. *Ruhr-Universität Bochum*, 2007.

[32] Chen Bingjie, Huang Xiaoping, and Wang Yan. A localization algorithm in wireless sensor networks based on mds with rssi classified. In *Computer Science and Education (ICCSE), 2010 5th International Conference on*, pages 1465 –1469, aug. 2010.

[33] Joseph Bonneau, Cormac Herley, Paul C van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 553–567. IEEE, 2012.

[34] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting mobile communications: the insecurity of 802.11. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 180–189. ACM, 2001.

[35] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[36] M. Butler. Android: Changing the mobile landscape. *Pervasive Computing, IEEE*, 10(1):4–7, 2011.

[37] Nancy Cam-Winget, Russ Housley, David Wagner, and Jesse Walker. Security flaws in 802.11 data link protocols. *Communications of the ACM*, 46(5):35–39, 2003.

[38] Srdjan Capkun. Jamming resistance. In *Encyclopedia of Cryptography and Security (2nd Ed.)*, pages 661–662. 2011.

[39] N Carey. Establishing pedestrian walking speeds. *Karen Aspelin, Portland State University*, 2005.

[40] Paul Castro, Patrick Chiu, Ted Kremenek, and Richard Muntz. A probabilistic room location service for wireless networked environments. pages 18–34, 2001.

[41] Jhih-Chung Chang, Chih-Chang Shen, Ann-Chen Chang, and Yi-Cheng Chung. Indoor lbs based on svm and rssi method. In *Proceedings of BAI Conference*, 2011.

[42] William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836, 1979.

[43] William S. Cleveland and E. Grosse. Computational methods for local regression. *Statistics and Computing*, 1:47–62, 1991. 10.1007/BF01890836.

[44] Tim Dierks. The transport layer security (tls) protocol version 1.2. 2008.

[45] John Paul Dunning. Taming the blue beast: A survey of bluetooth based threats. *Security & Privacy, IEEE*, 8(2):20–27, 2010.

[46] Bruce Kraemer et. al. Wireless lan medium access control (mac) and physical layer (phy) specifications. July 2012.

[47] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures.* PhD thesis, UNIVERSITY OF CALIFORNIA, IRVINE, 2000.

[48] Bob Fleck and Jordan Dimov. Wireless access points and arp poisoning: Wireless vulnerabilities that expose the wired network. *Cigital, Inc.[En ligne]. Accessible à ladresse suivante: http://www. cigitallabs. com/resources/papers/download/arppoison. pdf*, 2001.

[49] Bharat Gadher, Hossein Rahimi, and A. Nur Zincir-Heywood. Computer system and method for indoor geo-fencing and access control (US 61757488).

[50] Steve Gold. Cracking wireless networks. *Network Security*, 2011(11):14 – 18, 2011.

[51] Yanying Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *Communications Surveys Tutorials, IEEE*, 11(1):13–32, 2009.

[52] Xing He. *Signal Processing, Perceptual Coding and Watermarking of Digital Audio: Advanced Technologies and Models*. BrainMedia LLC, 2011.

[53] Rob J Hyndman. Moving averages. `http://www.robjhyndman.com/papers/movingaverage.pdf`, 2009.

[54] Petri Kontkanen, Petri Myllymki, Teemu Roos, Henry Tirri, Kimmo Valtonen, and Hannes Wettig. Topics in probabilistic location estimation in wireless networks, 2004.

[55] Markus Laner, Philipp Svoboda, and Markus Rupp. Latency analysis of 3g network components. In *European Wireless, 2012. EW. 18th European Wireless Conference*, pages 1–8, 2012.

[56] Marcus Leech. Socks protocol version 5. 1996.

[57] Hui Liu, H. Darabi, P. Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067 –1080, nov. 2007.

[58] Hui Liu, H. Darabi, P. Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067 –1080, nov. 2007.

[59] Moxie Marlinspike. Sslstrip. *Thoughtcrime Labs.[Online]*, 2009.

[60] Hirokazu Miura, Junichi Sakamoto, Noriyuki Matsuda, Hirokazu Taki, Noriyuki Abe, and Satoshi Hori. Adequate rssi determination method by making use of svm for indoor localization. In *Proceedings of the 10th international conference on Knowledge-Based Intelligent Information and Engineering Systems - Volume Part II*, KES'06, pages 628–636, Berlin, Heidelberg, 2006. Springer-Verlag.

[61] Robert Morris and Ken Thompson. Password security: a case history. *Commun. ACM*, 22(11):594–597, November 1979.

[62] Nick L Petroni Jr and William A Arbaugh. The dangers of mitigating security design flaws: a wireless case study. *Security & Privacy, IEEE*, 1(1):28–36, 2003.

[63] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[64] Hossein Rahimi, A. Nur Zincir-Heywood, and Bharat Gadher. Indoor geo-fencing and access control for wireless networks. In *Proceedings of the IEEE Conference on Computational Intelligence and Cyber Security 2013*, pages 1–8. IEEE CIS, April 2013.

[65] F. Reclus and K. Drouard. Geofencing for fleet and freight management. In *Intelligent Transport Systems Telecommunications,(ITST),2009 9th International Conference on*, pages 353–356, Oct 2009.

[66] David Recordon and Drummond Reed. Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, DIM '06, pages 11–16, New York, NY, USA, 2006. ACM.

[67] Lior Rokach and Oded Z. Maimon. *Data Mining with Decision Trees: Theroy and Applications*, volume 69 of *Series in Machine Perception and Artificial Intelligence*. World Scientific Publishing Co. Pte. Ltd., 2007.

[68] Teemu Roos, Petri Myllymäki, Henry Tirri, Pauli Misikangas, and Juha Sievänen. A Probabilistic Approach to WLAN User Location Estimation. *International Journal of Wireless Information Networks*, 9(3):155–164, July 2002.

[69] S. Saha, K. Chaudhuri, D. Sanghi, and P. Bhagwat. Location determination of a mobile device using ieee 802.11b access point signals. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 3, pages 1987 –1992 vol.3, march 2003.

[70] W Richard Stevens. *TCP/IP Illustrated Vol. I: The Protocols*. Pearson Education India, 1994.

[71] Mario Strasser, Christina Pöpper, Srdjan Capkun, and Mario Cagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *IEEE Symposium on Security and Privacy*, pages 64–78, 2008.

[72] Adam Stubblefield, John Ioannidis, and Aviel D Rubin. Using the fluhrer, mantin, and shamir attack to break wep. In *Proceedings of the 2002 Network and Distributed Systems Security Symposium*, volume 1722, 2002.

[73] Carlos E. Galvan T, Issac Galvan-Tejada, Ernesto Ivan Sandoval, and Ramon Brena. Wifi bluetooth based combined positioning algorithm. *Procedia Engineering*, 35(0):101 – 108, 2012. International Meeting of Electrical Engineering Research 2012.

[74] Gordon Thomson. Byod: enabling the chaos. *Network Security*, 2012(2):5 – 8, 2012.

[75] Olaf Titz. Why tcp over tcp is a bad idea. *http://sites.inka.de/bigred/devel/tcp-tcp.html*, 58, 2001.

[76] Bruce Tuch. Development of wavelan, an ism band wireless lan. *AT & T TECH J.*, 72(4):27–37, 1993.

[77] F. Viani, L. Lizzi, P. Rocca, M. Benedetti, M. Donelli, and A. Massa. Object tracking through rssi measurements in wireless sensor networks. *Electronics Letters*, 44(10):653 –654, 8 2008.

[78] E.A. Wan, A.S. Paul, and P.G. Jacobs. Tag-free rssi based indoor localization. In *Proceedings of the 2012 International Technical Meeting of The Institute of Navigation.*

[79] Rui Wang, Fang Zhao, Haiyong Luo, Bo Lu, and Tao Lu. Fusion of wi-fi and bluetooth for indoor localization. In *Proceedings of the 1st international workshop on Mobile location-based service*, MLBS '11, pages 63–66, New York, NY, USA, 2011. ACM.

[80] R. Want. Near field communication. *Pervasive Computing, IEEE*, 10(3):4–7, 2011.

[81] Donald J Welch and Scott Lathrop. A survey of 802.11 a wireless security threats and security mechanisms. *United States Military Academy West Point*, 2003.

[82] Pedro Wightman, Daladier Jabba, and Miguel A. Labrador. An rssi-based filter for mobility control of mobile wireless ad hoc-based unmanned ground vehicles. pages 694308–694308–10, 2008.

[83] Carin Willen, Kirsten Lehmann, and Katharina Sunnerhagen. Walking speed indoors and outdoors in healthy persons and in persons with late effects of polio. *Journal of Neurology Research*, 3(2):62–67, 2013.

[84] A. Willig, M. Kuhm, and A. Wolisz. Cooperative distance classification using an ieee 802.15.4-compliant transceiver. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1 –6, april 2009.

[85] Tatu Ylonen and Chris Lonvick. The secure shell (ssh) protocol architecture. 2006.

[86] Moustafa Youssef, Ashok Agrawala, and A. Udaya Shankar. Wlan location determination via clustering and probability distributions. In *In IEEE PerCom 2003*, 2003.

[87] Liping Zhang and Yiping Chen. A new indoor mobile node tracking scheme based on rssi and kalman filter. In *Wireless Mobile and Computing (CCWMC 2011), IET International Communication Conference on*, pages 216 –220, nov. 2011.

Appendix A

Result Tables and Figures

A.1 Tables

This section includes tables that present the reader with detailed information about the classification runs. Table A.1 tabulates the classification results on the original data sets using 4 classifiers as mentioned in section 6.1. This table also includes classification results on a normalized copy of data sets, in order to investigate usefulness of such a normalization on the classification. Tables A.2 and A.3, show the results achieved from the smoothing algorithm runs. Table A.2 tabulates the sorted results for the first set of experiments aimed to choose the best smoothing algorithm. Afterwards, Table A.3 is presenting the classification results obtained when changing the smoothing $Window_{Size}$ from 5 to 50.

Table A.1: All the best classification results achieved by runs on the original data. In these runs [0,1] normalization is also tested.

Data	Run	Normalize	TP_{ratio}	FP_{ratio}	TN_{ratio}	FN_{ratio}	Precision	Recall	$F_{measure}$	AUC
BT-10x10-alldevs	NaiveBayes	none	0.8	0	1	0.2	1	0.8	0.888889	0.94
BT-10x10-alldevs	LibSVM	none	1	0	1	0	1	1	1	1
BT-10x10-alldevs	LibSVM	[0,1]	0.9	0.1	0.9	0.1	0.9	0.9	0.9	0.9
BT-10x10-alldevs	J48	none	1	0	1	0	1	1	1	1
BT-10x10-alldevs	RandomForest	none	1	0	1	0	1	1	1	1
BT-10x10-alldevs	RandomForest	[0,1]	1	0	1	0	1	1	1	1
BT-10x10-Samsung	NaiveBayes	none	1	0	1	0	1	1	1	1
BT-10x10-Samsung	LibSVM	none	1	0	1	0	1	1	1	1
BT-10x10-Samsung	LibSVM	[0,1]	1	0	1	0	1	1	1	1

Data	Run	Normalize	TP_{ratio}	FP_{ratio}	TN_{ratio}	FN_{ratio}	Precision	Recall	$F_{measure}$	AUC
BT-10x10-Samsung	J48	none	1	0	1	0	1	1	1	1
BT-10x10-Samsung	RandomForest	none	1	0	1	0	1	1	1	1
BT-10x10-Samsung	RandomForest	[0,1]	1	0	1	0	1	1	1	1
BT-10x10-Asus	NaiveBayes	none	1	0	1	0	1	1	1	1
BT-10x10-Asus	LibSVM	none	1	0	1	0	1	1	1	1
BT-10x10-Asus	LibSVM	[0,1]	1	0.2	0.8	0	0.833333	1	0.909091	0.9
BT-10x10-Asus	J48	none	1	0	1	0	1	1	1	1
BT-10x10-Asus	RandomForest	none	1	0	1	0	1	1	1	1
BT-10x10-Asus	RandomForest	[0,1]	1	0	1	0	1	1	1	1
BT-2x2-Samsung	NaiveBayes	none	1	0	1	0	1	1	1	1
BT-2x2-Samsung	LibSVM	none	1	0	1	0	1	1	1	1
BT-2x2-Samsung	LibSVM	[0,1]	0.6	0	1	0.4	1	0.6	0.75	0.8
BT-2x2-Samsung	J48	none	1	0	1	0	1	1	1	1
BT-2x2-Samsung	RandomForest	none	1	0	1	0	1	1	1	1
BT-2x2-Samsung	RandomForest	[0,1]	1	0	1	0	1	1	1	1
BT-2x2-Asus	NaiveBayes	none	1	0	1	0	1	1	1	1
BT-2x2-Asus	LibSVM	none	1	0	1	0	1	1	1	1
BT-2x2-Asus	LibSVM	[0,1]	1	1	0	0	0.75	1	0.857143	0.5
BT-2x2-Asus	J48	none	1	0	1	0	1	1	1	1
BT-2x2-Asus	RandomForest	none	1	0	1	0	1	1	1	1
BT-2x2-Asus	RandomForest	[0,1]	1	0	1	0	1	1	1	1
BT-5x5-Samsung	NaiveBayes	none	1	0	1	0	1	1	1	1
BT-5x5-Samsung	LibSVM	none	1	0	1	0	1	1	1	1
BT-5x5-Samsung	LibSVM	[0,1]	0.8	0	1	0.2	1	0.8	0.888889	0.9
BT-5x5-Samsung	J48	none	1	0	1	0	1	1	1	1
BT-5x5-Samsung	RandomForest	none	1	0	1	0	1	1	1	1
BT-5x5-Samsung	RandomForest	[0,1]	1	0	1	0	1	1	1	1
WF-10x10-Samsung	NaiveBayes	none	1	0	1	0	1	1	1	1
WF-10x10-Samsung	LibSVM	none	1	0	1	0	1	1	1	1

Data	Run	Normalize	TP_{ratio}	FP_{ratio}	TN_{ratio}	FN_{ratio}	Precision	Recall	$F_{measure}$	AUC
WF-10x10-Samsung	LibSVM	[0,1]	1	0	1	0	1	1	1	1
WF-10x10-Samsung	J48	none	1	0	1	0	1	1	1	1
WF-10x10-Samsung	RandomForest	none	1	0	1	0	1	1	1	1
WF-10x10-Samsung	RandomForest	[0,1]	1	0	1	0	1	1	1	1
WF-10x10-Asus	NaiveBayes	none	1	0	1	0	1	1	1	1
WF-10x10-Asus	LibSVM	none	1	0	1	0	1	1	1	1
WF-10x10-Asus	LibSVM	[0,1]	1	0	1	0	1	1	1	1
WF-10x10-Asus	J48	none	1	0	1	0	1	1	1	1
WF-10x10-Asus	RandomForest	none	1	0	1	0	1	1	1	1
WF-10x10-Asus	RandomForest	[0,1]	1	0	1	0	1	1	1	1
WF-2x2-alldevs	NaiveBayes	none	0.7	0.2	0.8	0.3	0.777778	0.7	0.736842	0.76875
WF-2x2-alldevs	LibSVM	none	0.85	0.4	0.6	0.15	0.68	0.85	0.755556	0.725
WF-2x2-alldevs	LibSVM	[0,1]	1	1	0	0	0.525	1	0.688525	0.5
WF-2x2-alldevs	J48	none	0.45	0.05	0.95	0.55	0.9	0.45	0.6	0.705
WF-2x2-alldevs	RandomForest	none	0.75	0.4	0.6	0.25	0.652174	0.75	0.697674	0.745
WF-2x2-alldevs	RandomForest	[0,1]	0.75	0.4	0.6	0.25	0.652174	0.75	0.697674	0.745
WF-2x2-Asus	NaiveBayes	none	1	0	1	0	1	1	1	1
WF-2x2-Asus	LibSVM	none	1	0	1	0	1	1	1	1
WF-2x2-Asus	LibSVM	[0,1]	1	0	1	0	1	1	1	1
WF-2x2-Asus	J48	none	1	0	1	0	1	1	1	1
WF-2x2-Asus	RandomForest	none	1	0	1	0	1	1	1	1
WF-2x2-Asus	RandomForest	[0,1]	1	0	1	0	1	1	1	1
WF-5x5-Samsung	NaiveBayes	none	1	0	1	0	1	1	1	1
WF-5x5-Samsung	LibSVM	none	1	0	1	0	1	1	1	1
WF-5x5-Samsung	LibSVM	[0,1]	1	0.2	0.8	0	0.833333	1	0.909091	0.9
WF-5x5-Samsung	J48	none	1	0	1	0	1	1	1	1
WF-5x5-Samsung	RandomForest	none	1	0	1	0	1	1	1	1
WF-5x5-Samsung	RandomForest	[0,1]	1	0	1	0	1	1	1	1
WF-5x5-Asus	NaiveBayes	none	1	0.2	0.8	0	0.833333	1	0.909091	0.88

Data	Run	Normalize	TP_{ratio}	FP_{ratio}	TN_{ratio}	FN_{ratio}	Precision	Recall	$F_{measure}$	AUC
WF-5x5-Asus	LibSVM	none	1	0	1	0	1	1	1	1
WF-5x5-Asus	LibSVM	[0,1]	1	0.2	0.8	0	0.857143	1	0.923077	0.9
WF-5x5-Asus	J48	none	1	0	1	0	1	1	1	1
WF-5x5-Asus	RandomForest	none	1	0	1	0	1	1	1	1
WF-5x5-Asus	RandomForest	[0,1]	1	0	1	0	1	1	1	1

Table A.1: All the best classification results achieved by runs on the original data. In these runs [0,1] normalization is also tested.

Table A.2: Detailed classification result averages for Random Forest on all data sets using different smoothing methods, sorted based on $\mathrm{F}_{Measure}$. $Window_{Size}$ is set to 10 for all runs.

Data	Method	TP_{ratio}	FP_{ratio}	TN_{ratio}	FN_{ratio}	Precision	Recall	$\mathrm{F}_{Measure}$	AUC
BT-10x10-Samsung	Moving Average	0.95	0.05	0.95	0.05	0.96	0.95	0.95	1.00
BT-10x10-Samsung	RLOWESS	0.95	0.05	0.95	0.05	0.96	0.95	0.95	1.00
BT-10x10-Asus	Moving Average	0.93	0.07	0.93	0.07	0.94	0.93	0.93	0.99
BT-10x10-alldevs	Moving Average	0.93	0.08	0.93	0.08	0.93	0.93	0.92	0.98
BT-10x10-alldevs	RLOWESS	0.91	0.09	0.91	0.09	0.92	0.91	0.91	0.98
BT-10x10-Asus	LOWESS	0.90	0.10	0.90	0.10	0.92	0.90	0.90	0.95
BT-10x10-Samsung	LOWESS	0.90	0.10	0.90	0.10	0.92	0.90	0.90	1.00
BT-10x10-alldevs	LOWESS	0.89	0.11	0.89	0.11	0.90	0.89	0.89	0.98
BT-10x10-alldevs	RLOESS	0.89	0.11	0.89	0.11	0.90	0.89	0.89	0.94
WF-10x10-Samsung	RLOWESS	0.89	0.11	0.89	0.11	0.91	0.89	0.89	0.95
WF-10x10-Samsung	LOWESS	0.89	0.11	0.89	0.11	0.91	0.89	0.89	0.90
BT-5x5-Samsung	Moving Average	0.89	0.11	0.89	0.11	0.90	0.89	0.88	0.99
BT-10x10-alldevs	Savitzky-Golay	0.88	0.12	0.88	0.12	0.90	0.88	0.88	0.96
WF-5x5-Samsung	Moving Average	0.88	0.12	0.88	0.12	0.90	0.88	0.88	0.98
WF-10x10-Asus	RLOWESS	0.88	0.13	0.88	0.13	0.89	0.88	0.87	0.99
WF-10x10-Samsung	Moving Average	0.87	0.13	0.87	0.13	0.90	0.87	0.87	0.93
BT-2x2-Asus	Moving Average	0.87	0.19	0.81	0.13	0.89	0.87	0.87	0.97
WF-10x10-Samsung	LOESS	0.87	0.13	0.87	0.13	0.89	0.87	0.87	0.93
WF-10x10-Asus	Moving Average	0.87	0.13	0.87	0.13	0.90	0.87	0.87	0.92
WF-10x10-Samsung	Savitzky-Golay	0.87	0.13	0.87	0.13	0.89	0.87	0.87	0.91
BT-10x10-Asus	RLOWESS	0.87	0.13	0.87	0.13	0.88	0.87	0.86	0.97
WF-10x10-Samsung	RLOESS	0.86	0.14	0.86	0.14	0.88	0.86	0.86	0.90
WF-5x5-Samsung	LOWESS	0.86	0.14	0.86	0.14	0.88	0.86	0.86	0.95
BT-5x5-Samsung	RLOWESS	0.86	0.15	0.85	0.14	0.88	0.86	0.86	0.97
BT-2x2-Asus	RLOWESS	0.86	0.22	0.78	0.14	0.88	0.86	0.85	0.95

Data	Method	TP_{ratio}	FP_{ratio}	TN_{ratio}	FN_{ratio}	Precision	Recall	$F_{Measure}$	AUC
BT-10x10-alldevs	LOESS	0.85	0.15	0.85	0.15	0.86	0.85	0.85	0.93
BT-2x2-Samsung	Moving Average	0.85	0.14	0.86	0.15	0.88	0.85	0.85	0.98
BT-10x10-Samsung	Savitzky-Golay	0.85	0.15	0.85	0.15	0.87	0.85	0.85	0.97
BT-10x10-Asus	LOESS	0.84	0.16	0.84	0.16	0.86	0.84	0.84	0.91
BT-2x2-Asus	LOWESS	0.84	0.22	0.78	0.16	0.86	0.84	0.83	0.93
BT-5x5-Samsung	LOWESS	0.84	0.16	0.84	0.16	0.87	0.84	0.83	0.94
WF-5x5-Samsung	RLOWESS	0.83	0.17	0.83	0.17	0.85	0.83	0.83	0.96
WF-10x10-Asus	LOWESS	0.83	0.17	0.83	0.17	0.86	0.83	0.83	0.92
BT-10x10-Samsung	RLOESS	0.83	0.17	0.83	0.17	0.85	0.83	0.83	0.93
BT-2x2-Samsung	RLOWESS	0.82	0.17	0.83	0.18	0.84	0.82	0.82	0.92
BT-10x10-Asus	RLOESS	0.82	0.18	0.82	0.18	0.85	0.82	0.82	0.93
BT-5x5-Samsung	Savitzky-Golay	0.82	0.18	0.82	0.18	0.86	0.82	0.82	0.91
BT-2x2-Samsung	LOWESS	0.82	0.18	0.82	0.18	0.83	0.82	0.82	0.93
BT-5x5-Samsung	RLOESS	0.82	0.19	0.81	0.18	0.84	0.82	0.82	0.89
WF-5x5-Asus	Moving Average	0.82	0.18	0.82	0.18	0.85	0.82	0.82	0.89
WF-10x10-Asus	RLOESS	0.82	0.18	0.82	0.18	0.85	0.82	0.81	0.86
BT-2x2-Asus	Savitzky-Golay	0.82	0.26	0.74	0.18	0.83	0.82	0.81	0.91
BT-2x2-Samsung	RLOESS	0.82	0.19	0.81	0.18	0.84	0.82	0.81	0.92
BT-10x10-Samsung	LOESS	0.81	0.19	0.81	0.19	0.84	0.81	0.81	0.92
BT-2x2-Asus	RLOESS	0.81	0.28	0.72	0.19	0.83	0.81	0.80	0.92
BT-5x5-Samsung	LOESS	0.81	0.19	0.81	0.19	0.84	0.81	0.80	0.86
BT-10x10-Asus	Savitzky-Golay	0.81	0.19	0.81	0.19	0.84	0.81	0.80	0.95
WF-10x10-Asus	LOESS	0.80	0.20	0.80	0.20	0.84	0.80	0.79	0.89
BT-2x2-Samsung	Savitzky-Golay	0.80	0.20	0.80	0.20	0.82	0.80	0.79	0.93
WF-5x5-Samsung	Savitzky-Golay	0.79	0.21	0.79	0.21	0.82	0.79	0.78	0.89
BT-2x2-Asus	LOESS	0.79	0.30	0.70	0.21	0.80	0.79	0.78	0.89
WF-2x2-Asus	RLOWESS	0.78	0.22	0.78	0.22	0.81	0.78	0.78	0.83
WF-10x10-Asus	Savitzky-Golay	0.78	0.22	0.78	0.22	0.81	0.78	0.77	0.87
BT-2x2-Samsung	LOESS	0.77	0.23	0.77	0.23	0.80	0.77	0.77	0.83
WF-5x5-Asus	RLOWESS	0.77	0.23	0.77	0.23	0.80	0.77	0.77	0.86

Data	Method	TP_{ratio}	FP_{ratio}	TN_{ratio}	FN_{ratio}	Precision	Recall	$\mathrm{F}_{Measure}$	AUC
WF-5x5-Asus	LOWESS	0.77	0.23	0.77	0.23	0.79	0.77	0.76	0.81
WF-5x5-Asus	Savitzky-Golay	0.76	0.24	0.76	0.24	0.78	0.76	0.76	0.82
WF-2x2-Asus	Moving Average	0.76	0.24	0.76	0.24	0.80	0.76	0.75	0.81
WF-5x5-Samsung	RLOESS	0.75	0.25	0.75	0.25	0.79	0.75	0.74	0.87
WF-2x2-Asus	LOWESS	0.74	0.26	0.74	0.26	0.78	0.74	0.74	0.78
WF-5x5-Samsung	LOESS	0.74	0.26	0.74	0.26	0.77	0.74	0.73	0.85
WF-2x2-Asus	Savitzky-Golay	0.73	0.28	0.72	0.27	0.77	0.73	0.72	0.75
WF-2x2-Asus	RLOESS	0.69	0.30	0.70	0.31	0.72	0.69	0.68	0.76
WF-2x2-Asus	LOESS	0.68	0.32	0.68	0.32	0.71	0.68	0.67	0.74
WF-5x5-Asus	LOESS	0.65	0.35	0.65	0.35	0.67	0.65	0.64	0.72
WF-2x2-alldevs	Moving Average	0.62	0.38	0.62	0.38	0.62	0.62	0.62	0.68
WF-2x2-alldevs	RLOWESS	0.60	0.40	0.60	0.40	0.60	0.60	0.59	0.63
WF-2x2-alldevs	Savitzky-Golay	0.58	0.42	0.58	0.42	0.59	0.58	0.58	0.64
WF-2x2-alldevs	LOWESS	0.58	0.42	0.58	0.42	0.58	0.58	0.58	0.64
WF-2x2-alldevs	RLOESS	0.57	0.43	0.57	0.43	0.58	0.57	0.57	0.62
WF-5x5-Asus	RLOESS	0.58	0.42	0.58	0.42	0.60	0.58	0.57	0.68
WF-2x2-alldevs	LOESS	0.55	0.45	0.55	0.45	0.55	0.55	0.55	0.58

Table A.2: Detailed classification result averages for Random Forest on all data sets using different smoothing methods, sorted based on $\mathrm{F}_{Measure}$. $Window_{Size}$ is set to 10 for all runs.

Table A.3: Detailed classification results for different $Window_{Size}$ values on all data sets.

Data	$Window_{Size}$	TP$_{ratio}$	FP$_{ratio}$	TN$_{ratio}$	FN$_{ratio}$	F$_{Measure}$	AUC
BT-10x10-alldevs	5	0.93	0.08	0.93	0.08	0.92	0.98
BT-10x10-alldevs	10	0.93	0.08	0.93	0.08	0.92	1.00
BT-10x10-alldevs	15	0.94	0.06	0.94	0.06	0.94	0.99
BT-10x10-alldevs	20	0.93	0.07	0.93	0.07	0.93	1.00
BT-10x10-alldevs	25	1.00	0.00	1.00	0.00	1.00	1.00
BT-10x10-alldevs	30	1.00	0.00	1.00	0.00	1.00	1.00
BT-10x10-alldevs	35	1.00	0.00	1.00	0.00	1.00	1.00
BT-10x10-alldevs	40	0.98	0.03	0.98	0.03	0.97	1.00
BT-10x10-alldevs	45	0.97	0.03	0.97	0.03	0.97	1.00
BT-10x10-alldevs	50	0.98	0.03	0.98	0.03	0.97	1.00
BT-10x10-Samsung	5	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Samsung	10	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Samsung	15	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Samsung	20	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Samsung	25	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Samsung	30	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Samsung	35	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Samsung	40	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Samsung	45	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Samsung	50	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Asus	5	0.93	0.07	0.93	0.07	0.93	0.99
BT-10x10-Asus	10	0.93	0.07	0.93	0.07	0.93	0.99
BT-10x10-Asus	15	0.93	0.07	0.93	0.07	0.93	0.99
BT-10x10-Asus	20	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Asus	25	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Asus	30	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Asus	35	0.95	0.05	0.95	0.05	0.95	1.00

Data	$Window_{Size}$	TP$_{ratio}$	FP$_{ratio}$	TN$_{ratio}$	FN$_{ratio}$	F$_{Measure}$	AUC
BT-10x10-Asus	40	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Asus	45	0.95	0.05	0.95	0.05	0.95	1.00
BT-10x10-Asus	50	0.95	0.05	0.95	0.05	0.95	1.00
BT-2x2-Samsung	5	0.85	0.14	0.86	0.15	0.85	0.98
BT-2x2-Samsung	10	0.89	0.10	0.90	0.11	0.89	0.98
BT-2x2-Samsung	15	0.89	0.10	0.90	0.11	0.89	0.98
BT-2x2-Samsung	20	0.89	0.10	0.90	0.11	0.89	0.98
BT-2x2-Samsung	25	0.89	0.10	0.90	0.11	0.89	0.96
BT-2x2-Samsung	30	0.89	0.10	0.90	0.11	0.89	0.99
BT-2x2-Samsung	35	0.89	0.10	0.90	0.11	0.89	0.97
BT-2x2-Samsung	40	0.89	0.10	0.90	0.11	0.89	0.97
BT-2x2-Samsung	45	0.89	0.10	0.90	0.11	0.89	0.97
BT-2x2-Samsung	50	0.89	0.10	0.90	0.11	0.89	0.97
BT-2x2-Asus	5	0.87	0.19	0.81	0.13	0.87	0.97
BT-2x2-Asus	10	0.93	0.12	0.88	0.07	0.93	1.00
BT-2x2-Asus	15	0.93	0.12	0.88	0.07	0.93	1.00
BT-2x2-Asus	20	0.92	0.12	0.88	0.08	0.92	0.99
BT-2x2-Asus	25	0.92	0.12	0.88	0.08	0.92	0.99
BT-2x2-Asus	30	0.92	0.12	0.88	0.08	0.92	0.99
BT-2x2-Asus	35	0.92	0.12	0.88	0.08	0.92	0.99
BT-2x2-Asus	40	0.92	0.12	0.88	0.08	0.92	0.99
BT-2x2-Asus	45	0.92	0.12	0.88	0.08	0.92	0.99
BT-2x2-Asus	50	0.92	0.12	0.88	0.08	0.92	0.99
BT-5x5-Samsung	5	0.89	0.11	0.89	0.11	0.88	0.99
BT-5x5-Samsung	10	0.93	0.08	0.93	0.08	0.92	0.99
BT-5x5-Samsung	15	0.93	0.07	0.93	0.07	0.93	0.99
BT-5x5-Samsung	20	0.93	0.07	0.93	0.07	0.93	0.99
BT-5x5-Samsung	25	0.92	0.08	0.92	0.08	0.92	0.96
BT-5x5-Samsung	30	0.93	0.08	0.93	0.08	0.92	0.97
BT-5x5-Samsung	35	0.93	0.08	0.93	0.08	0.92	0.97

Data	$Window_{Size}$	TP_{ratio}	FP_{ratio}	TN_{ratio}	FN_{ratio}	$\mathrm{F}_{Measure}$	AUC
BT-5x5-Samsung	40	0.93	0.08	0.93	0.08	0.92	0.97
BT-5x5-Samsung	45	0.93	0.08	0.93	0.08	0.92	0.97
BT-5x5-Samsung	50	0.93	0.08	0.93	0.08	0.92	0.97
WF-10x10-Samsung	5	0.87	0.13	0.87	0.13	0.87	0.93
WF-10x10-Samsung	10	0.88	0.13	0.88	0.13	0.87	0.94
WF-10x10-Samsung	15	0.88	0.13	0.88	0.13	0.87	0.95
WF-10x10-Samsung	20	0.88	0.12	0.88	0.12	0.87	0.94
WF-10x10-Samsung	25	0.86	0.14	0.86	0.14	0.85	0.98
WF-10x10-Samsung	30	0.87	0.13	0.87	0.13	0.87	0.95
WF-10x10-Samsung	35	0.86	0.14	0.86	0.14	0.85	0.98
WF-10x10-Samsung	40	0.87	0.13	0.87	0.13	0.86	0.96
WF-10x10-Samsung	45	0.87	0.13	0.87	0.13	0.86	0.96
WF-10x10-Samsung	50	0.87	0.13	0.87	0.13	0.86	0.96
WF-10x10-Asus	5	0.87	0.13	0.87	0.13	0.87	0.92
WF-10x10-Asus	10	0.90	0.10	0.90	0.10	0.90	0.92
WF-10x10-Asus	15	0.88	0.12	0.88	0.12	0.88	0.95
WF-10x10-Asus	20	0.90	0.10	0.90	0.10	0.90	0.92
WF-10x10-Asus	25	0.89	0.11	0.89	0.11	0.88	0.94
WF-10x10-Asus	30	0.89	0.11	0.89	0.11	0.88	0.93
WF-10x10-Asus	35	0.89	0.11	0.89	0.11	0.88	0.95
WF-10x10-Asus	40	0.89	0.11	0.89	0.11	0.88	0.94
WF-10x10-Asus	45	0.89	0.11	0.89	0.11	0.88	0.95
WF-10x10-Asus	50	0.89	0.11	0.89	0.11	0.88	0.95
WF-2x2-alldevs	5	0.62	0.38	0.62	0.38	0.62	0.68
WF-2x2-alldevs	10	0.71	0.29	0.71	0.29	0.71	0.79
WF-2x2-alldevs	15	0.80	0.20	0.80	0.20	0.80	0.87
WF-2x2-alldevs	20	0.77	0.23	0.77	0.23	0.77	0.86
WF-2x2-alldevs	25	0.80	0.20	0.80	0.20	0.80	0.87
WF-2x2-alldevs	30	0.81	0.19	0.81	0.19	0.81	0.90
WF-2x2-alldevs	35	0.82	0.18	0.82	0.18	0.82	0.90

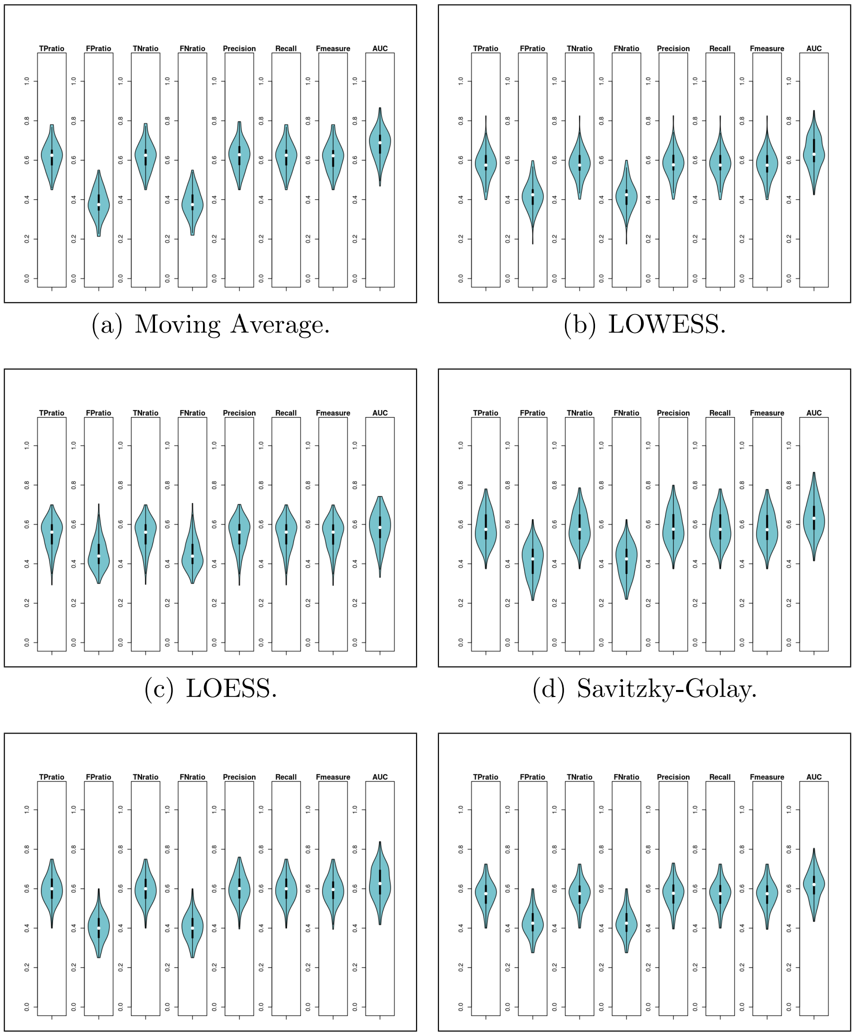
Data	$Window_{Size}$	TP_{ratio}	FP_{ratio}	TN_{ratio}	FN_{ratio}	$\mathrm{F}_{Measure}$	AUC
WF-2x2-alldevs	40	0.82	0.18	0.82	0.18	0.82	0.91
WF-2x2-alldevs	45	0.82	0.17	0.83	0.18	0.82	0.92
WF-2x2-alldevs	50	0.83	0.17	0.83	0.17	0.83	0.91
WF-2x2-Asus	5	0.76	0.24	0.76	0.24	0.75	0.81
WF-2x2-Asus	10	0.85	0.16	0.84	0.15	0.85	0.90
WF-2x2-Asus	15	0.83	0.17	0.83	0.17	0.82	0.88
WF-2x2-Asus	20	0.83	0.17	0.83	0.17	0.82	0.90
WF-2x2-Asus	25	0.83	0.17	0.83	0.17	0.82	0.91
WF-2x2-Asus	30	0.82	0.17	0.83	0.18	0.82	0.91
WF-2x2-Asus	35	0.83	0.17	0.83	0.17	0.83	0.90
WF-2x2-Asus	40	0.83	0.17	0.83	0.17	0.83	0.90
WF-2x2-Asus	45	0.83	0.17	0.83	0.17	0.83	0.90
WF-2x2-Asus	50	0.83	0.17	0.83	0.17	0.83	0.90
WF-5x5-Samsung	5	0.88	0.12	0.88	0.12	0.88	0.98
WF-5x5-Samsung	10	0.88	0.13	0.88	0.13	0.87	0.98
WF-5x5-Samsung	15	0.93	0.08	0.93	0.08	0.92	0.99
WF-5x5-Samsung	20	0.86	0.14	0.86	0.14	0.86	0.99
WF-5x5-Samsung	25	0.84	0.16	0.84	0.16	0.84	0.96
WF-5x5-Samsung	30	0.88	0.12	0.88	0.12	0.88	0.99
WF-5x5-Samsung	35	0.87	0.13	0.87	0.13	0.86	0.98
WF-5x5-Samsung	40	0.87	0.13	0.87	0.13	0.87	0.98
WF-5x5-Samsung	45	0.87	0.13	0.87	0.13	0.86	0.98
WF-5x5-Samsung	50	0.87	0.13	0.87	0.13	0.86	0.98
WF-5x5-Asus	5	0.82	0.18	0.82	0.18	0.82	0.89
WF-5x5-Asus	10	0.78	0.22	0.78	0.22	0.77	0.88
WF-5x5-Asus	15	0.82	0.18	0.82	0.18	0.82	0.94
WF-5x5-Asus	20	0.87	0.13	0.87	0.13	0.87	0.97
WF-5x5-Asus	25	1.00	0.00	1.00	0.00	1.00	1.00
WF-5x5-Asus	30	1.00	0.00	1.00	0.00	1.00	1.00
WF-5x5-Asus	35	1.00	0.00	1.00	0.00	1.00	1.00

Data	$Window_{Size}$	TP$_{ratio}$	FP$_{ratio}$	TN$_{ratio}$	FN$_{ratio}$	F$_{Measure}$	AUC
WF-5x5-Asus	40	1.00	0.00	1.00	0.00	1.00	1.00
WF-5x5-Asus	45	1.00	0.00	1.00	0.00	1.00	1.00
WF-5x5-Asus	50	1.00	0.00	1.00	0.00	1.00	1.00

Table A.3: Detailed classification results for different $Window_{Size}$ values on all data sets.

A.2 Figures

Violin plot was a useful visual representation when I needed to understand and compare the classification results distribution. Figures A.1 through A.10 visualize the results tabulated in Table A.2 for each data set. For all the runs shown in these figures the $Window_{Szie}$ parameter is equal and set to 10. At the end of this appendix, Figure A.12 and Figure A.11 are representing how window sizes affect the classification $F_{Measure}$ for each data set individually. Intuitively, information given in these figures is also represented in Table A.3, and figures 6.5 and 6.6.



(a) Moving Average.

(b) LOWESS.

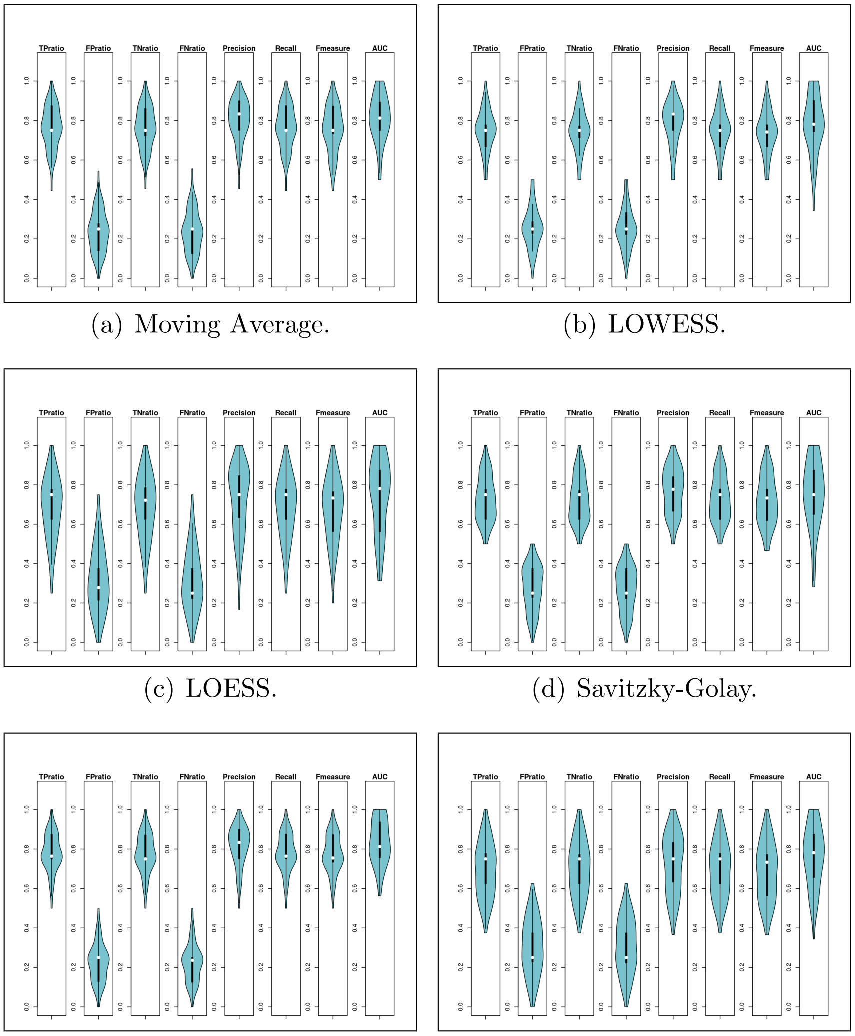
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.1: Random Forest result distribution on Wi-Fi 2x2 data set collected using a variety of devices.



(a) Moving Average.

(b) LOWESS.

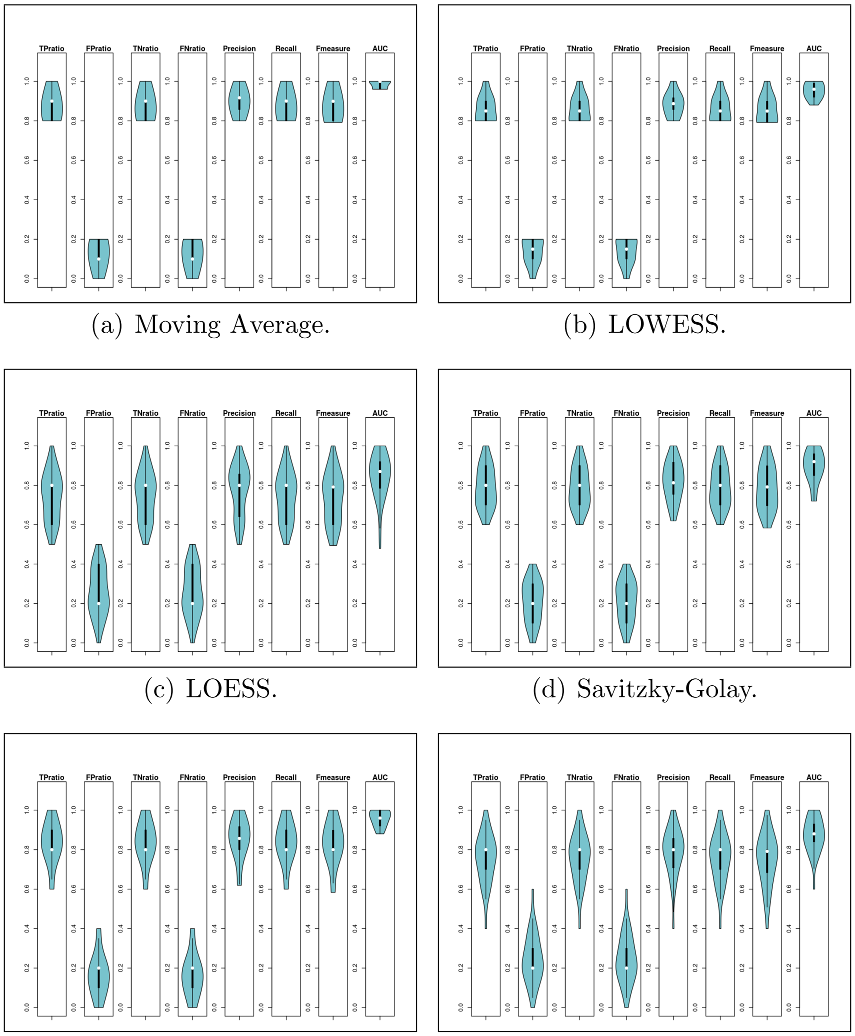
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.2: Random Forest result distribution on Wi-Fi 2x2 data set collected using an Asus Tablet.



(a) Moving Average.

(b) LOWESS.

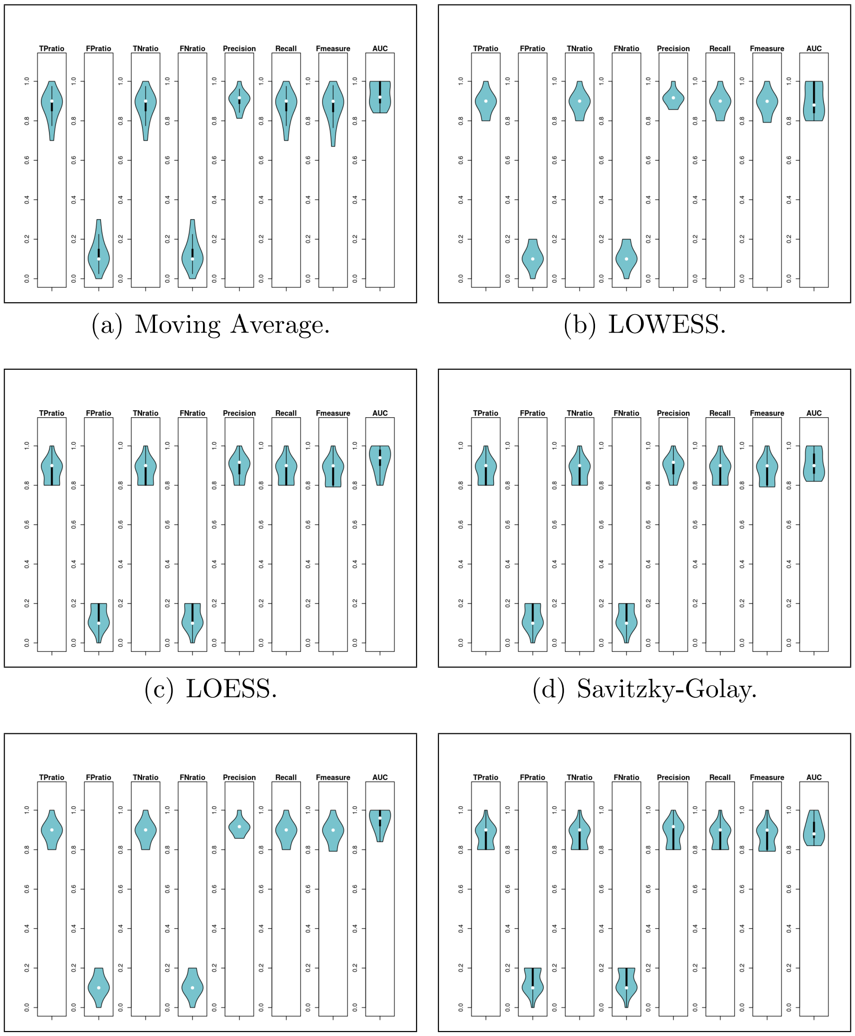
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.3: Random Forest result distribution on Wi-Fi 5x5 data set collected using a Samsung Smartphone.



(a) Moving Average.

(b) LOWESS.

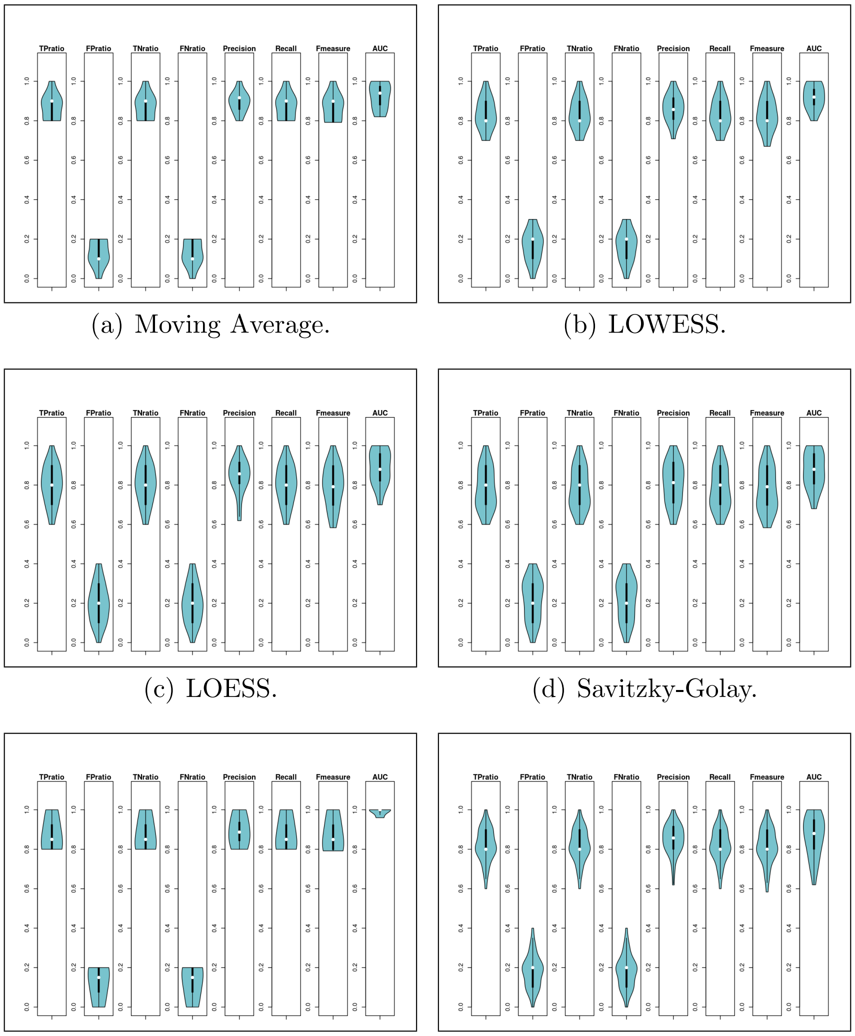
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.4: Random Forest result distribution on Wi-Fi 10x10 data set collected using a Samsung Smartphone.



(a) Moving Average.

(b) LOWESS.

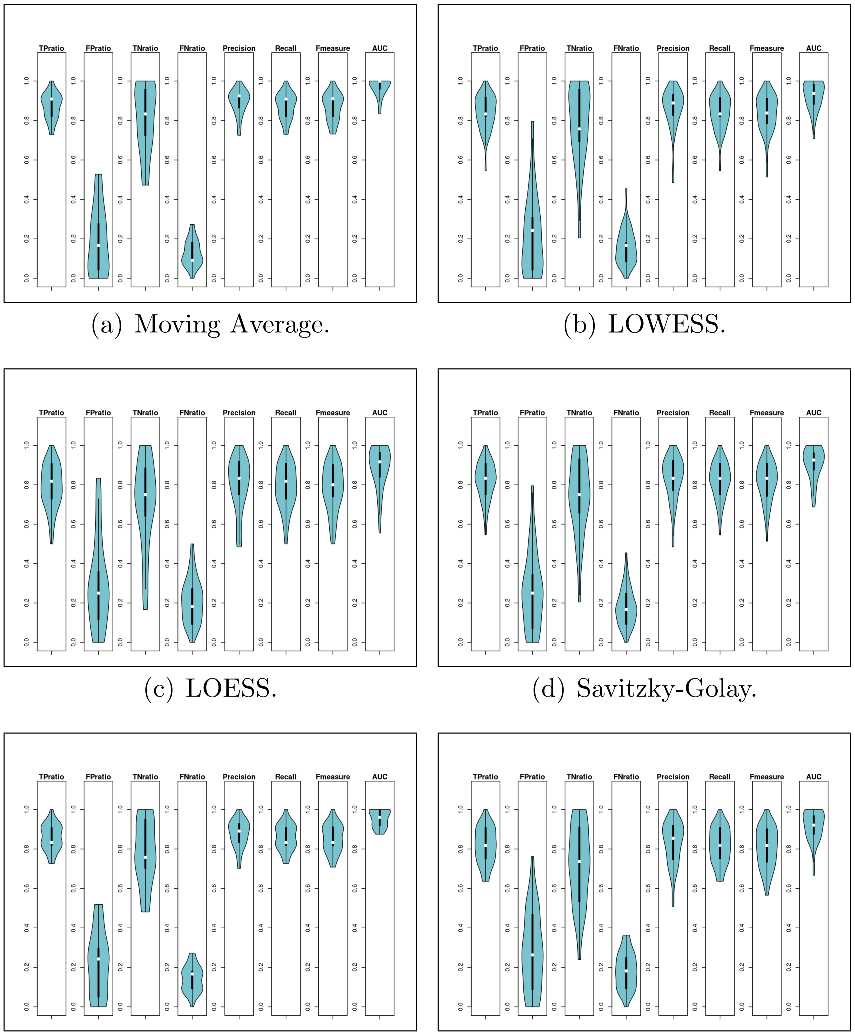
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.5: Random Forest result distribution on Wi-Fi 10x10 data set collected using an Asus Tablet.



(a) Moving Average.

(b) LOWESS.

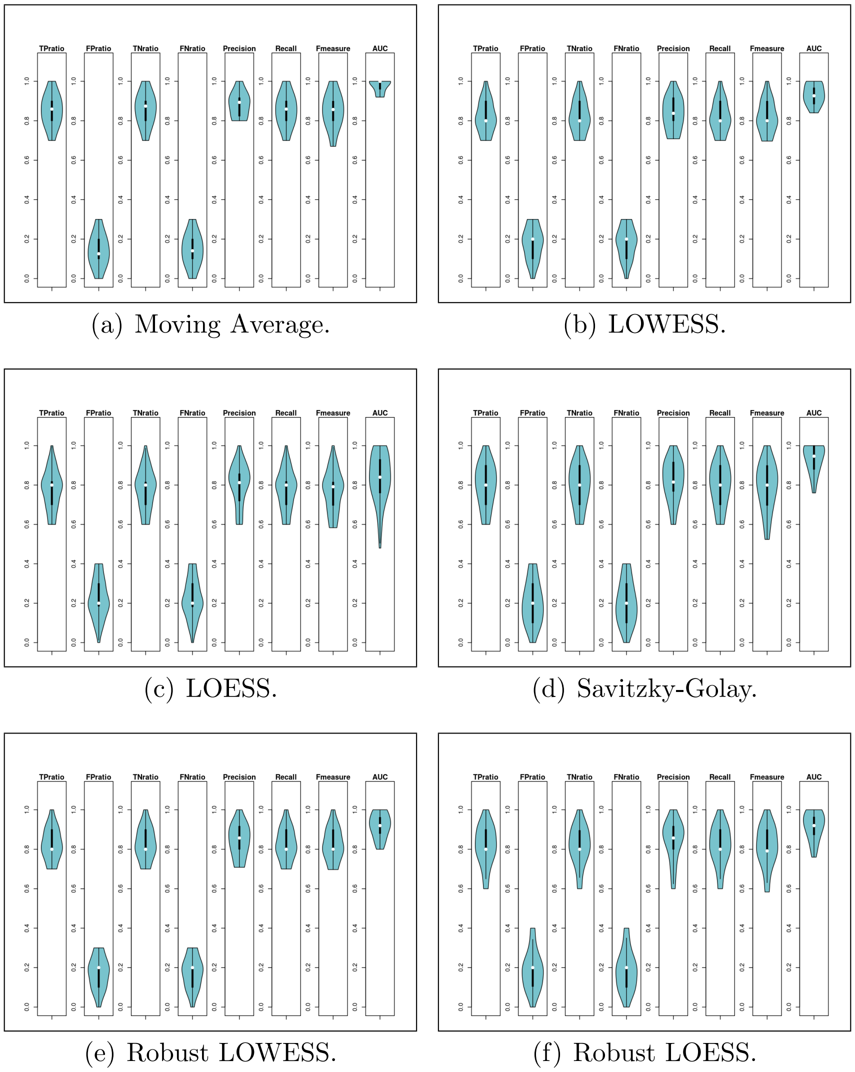
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.6: Random Forest result distribution on Bluetooth 2x2 data set collected using an Asus Tablet.



(a) Moving Average.

(b) LOWESS.

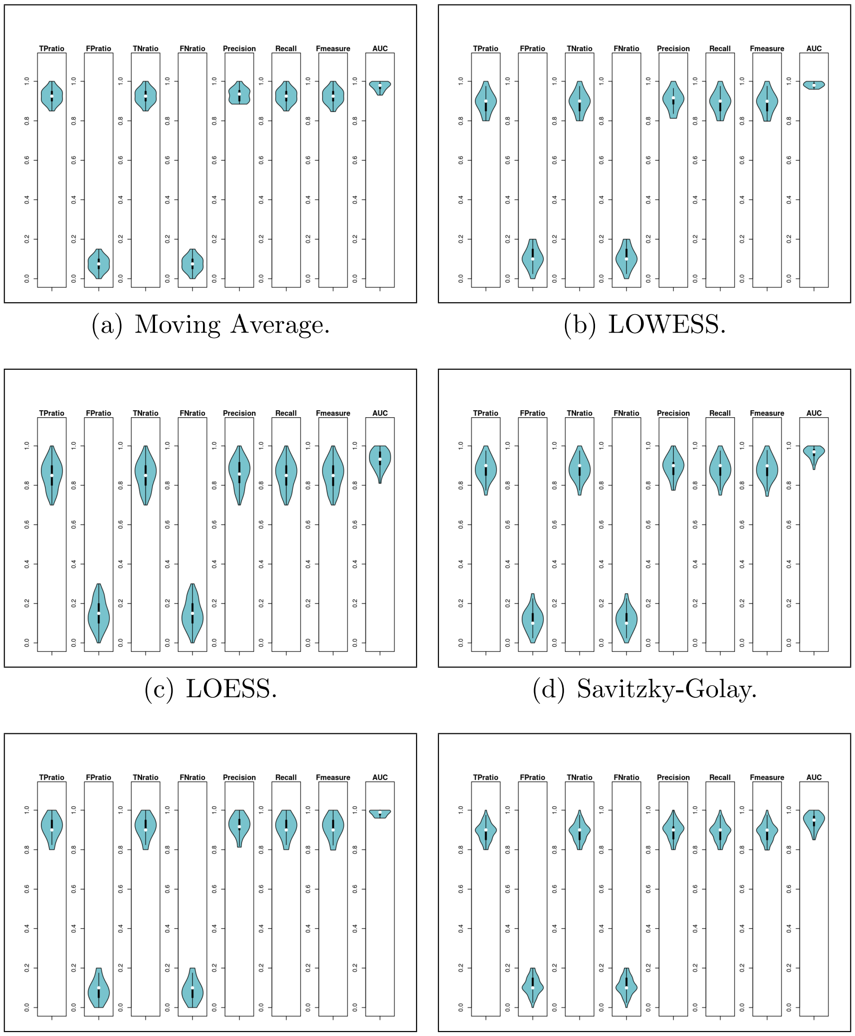
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.7: Random Forest result distribution on Bluetooth 2x2 data set collected using a Samsung Smartphone.



(a) Moving Average.

(b) LOWESS.

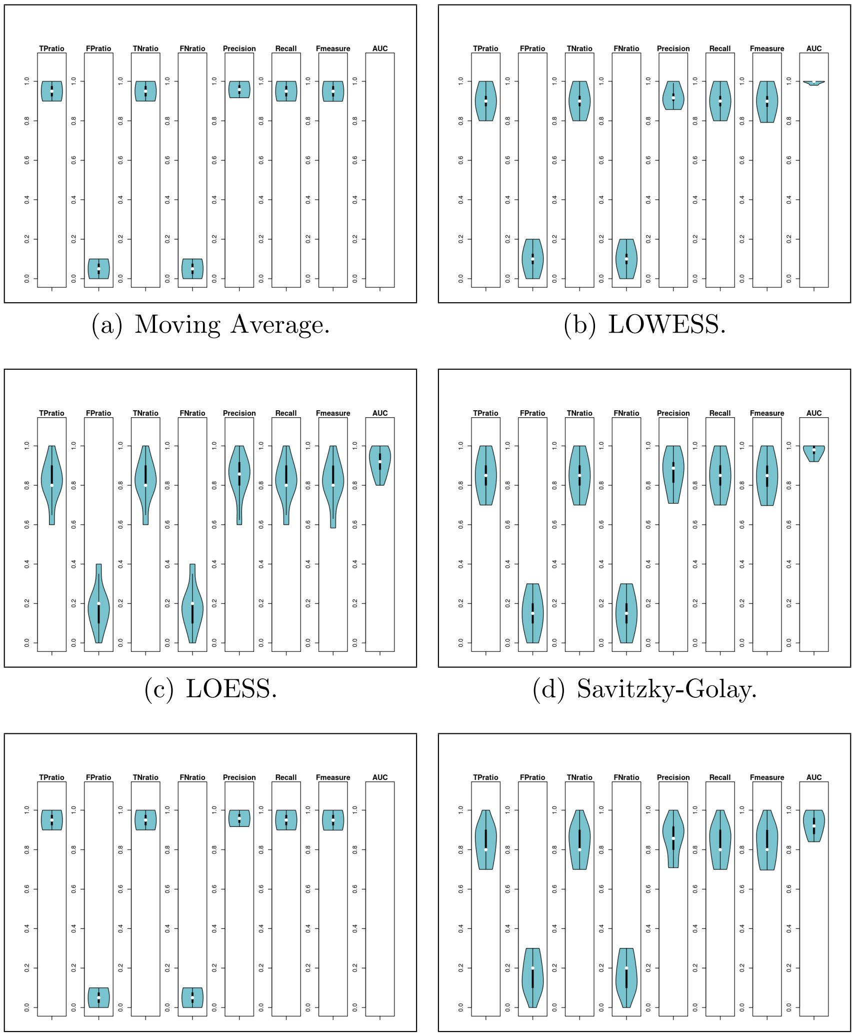
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.8: Random Forest result distribution on Bluetooth 10x10 data set collected using a Samsung Smartphone and an Asus Tablet.



(a) Moving Average.

(b) LOWESS.

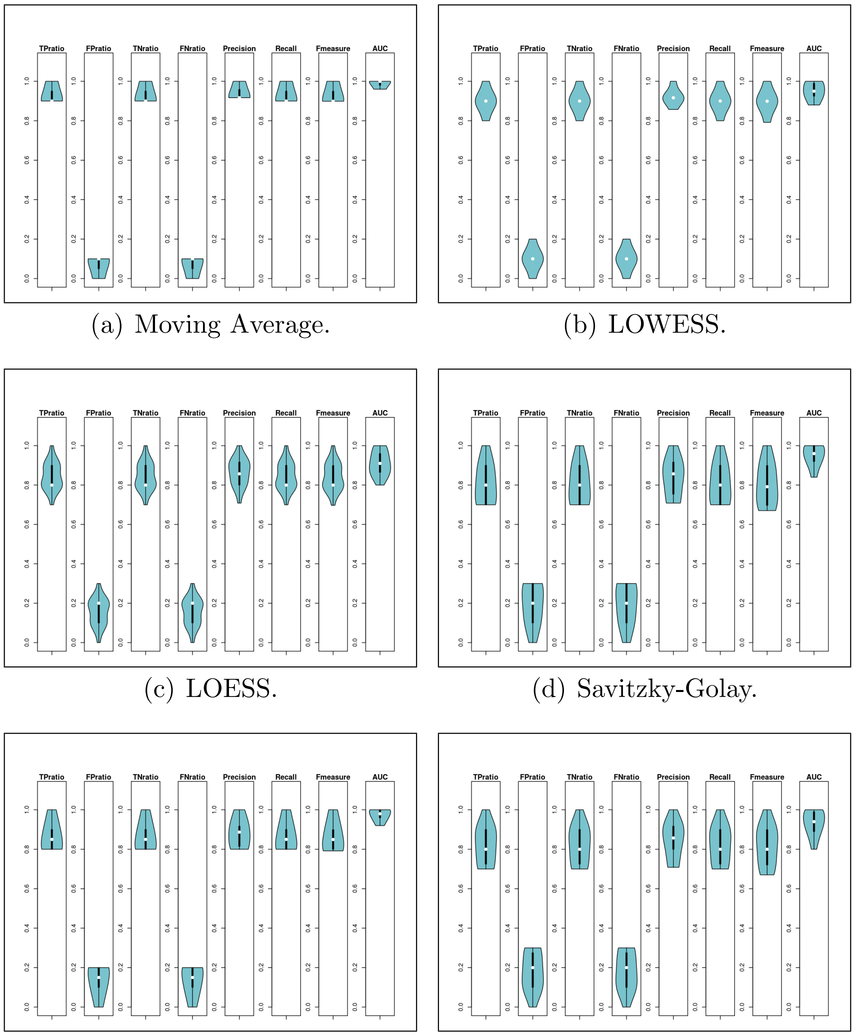
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.9: Random Forest result distribution on Bluetooth 10x10 data set collected using a Samsung Smartphone.



(a) Moving Average.

(b) LOWESS.

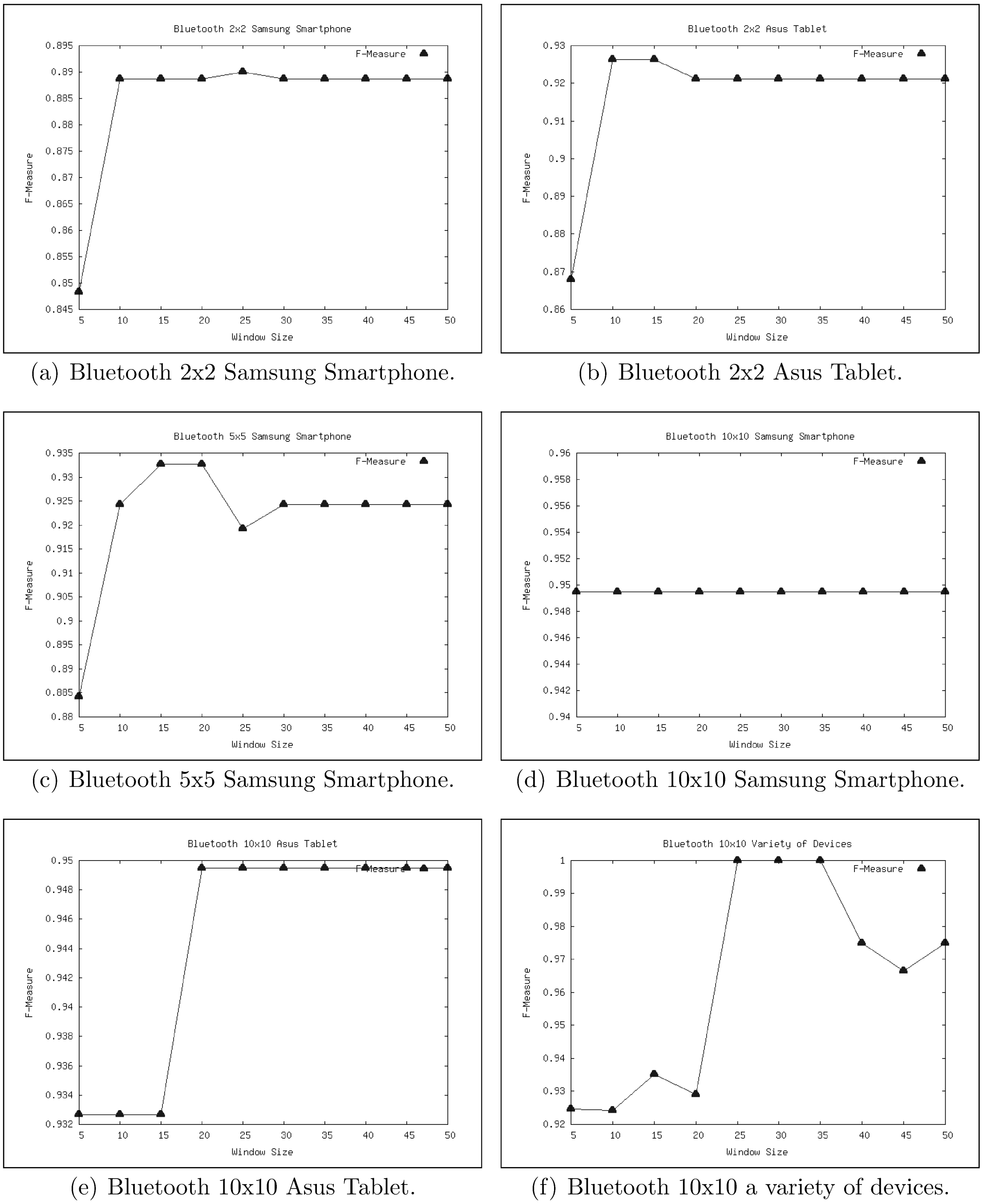
(c) LOESS.

(d) Savitzky-Golay.

(e) Robust LOWESS.

(f) Robust LOESS.

Figure A.10: Random Forest result distribution on Bluetooth 10x10 data set collected using an Asus Tablet.



(a) Bluetooth 2x2 Samsung Smartphone.

(b) Bluetooth 2x2 Asus Tablet.

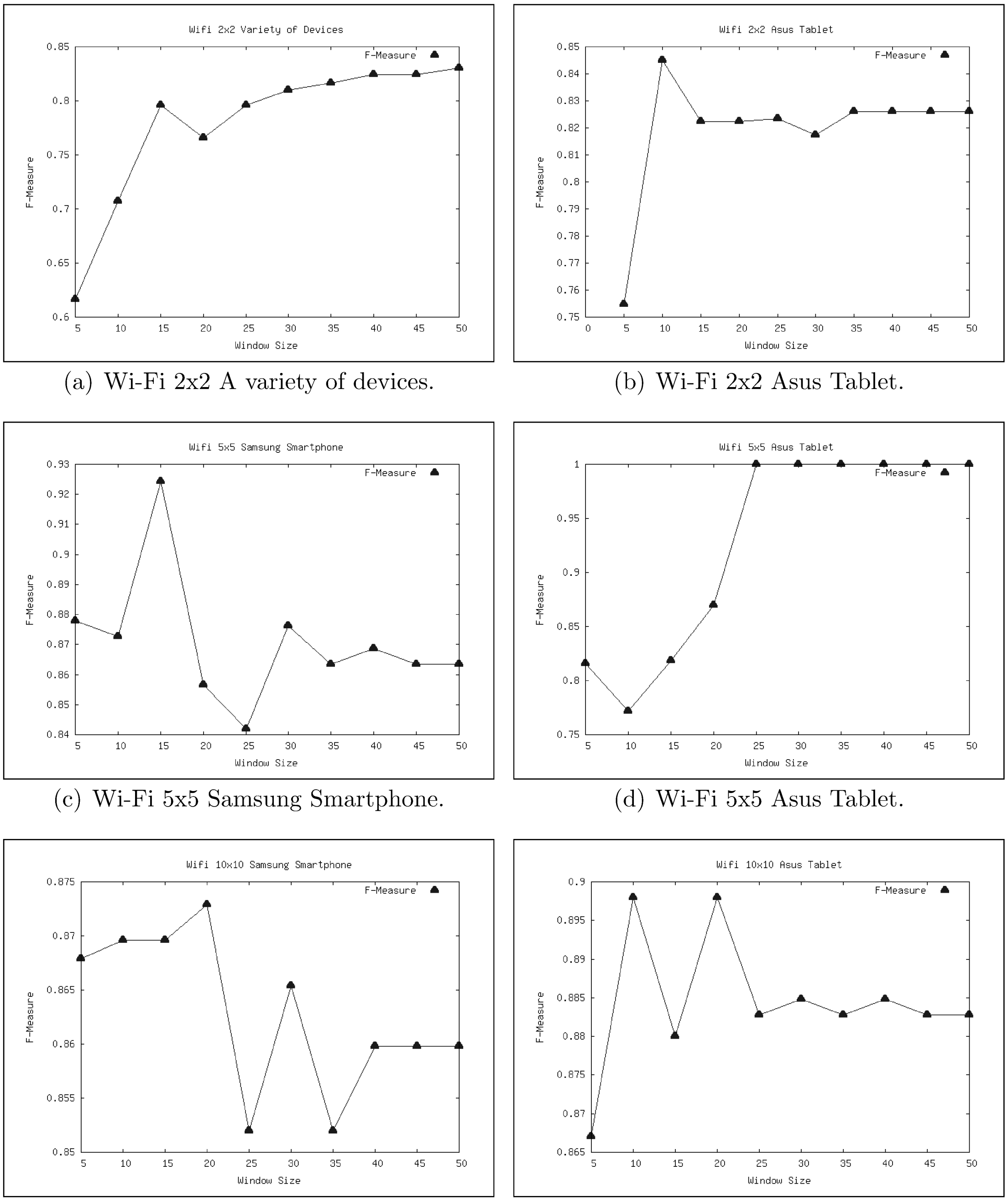
(c) Bluetooth 5x5 Samsung Smartphone.

(d) Bluetooth 10x10 Samsung Smartphone.

(e) Bluetooth 10x10 Asus Tablet.

(f) Bluetooth 10x10 a variety of devices.

Figure A.11: $F_{Measure}$ vs. smoothing $Window_{Size}$ using Moving Average for all Bluetooth data sets.



(a) Wi-Fi 2x2 A variety of devices.

(b) Wi-Fi 2x2 Asus Tablet.

(c) Wi-Fi 5x5 Samsung Smartphone.

(d) Wi-Fi 5x5 Asus Tablet.

(e) Wi-Fi 10x10 Samsung Smartphone.

(f) Wi-Fi 10x10 Asus Tablet.

Figure A.12: $F_{Measure}$ vs. smoothing $Window_{Size}$ using Moving Average for all Wi-Fi data sets.