



University of
Zurich^{UZH}

Dataset Generation for ML Personal Tracker Detection with a Focus on RSSI Shielding Approaches

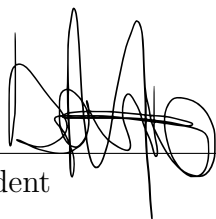
Dario Alberto Monopoli
Zürich, Switzerland
Student ID: 21-717-632

Supervisor: Katharina Müller
Date of Submission: July 01, 2024

Declaration of Independence

I hereby declare that I have composed this work independently and without the use of any aids other than those declared (including generative AI such as ChatGPT). I am aware that I take full responsibility for the scientific character of the submitted text myself, even if AI aids were used and declared (after written confirmation by the supervising professor). All passages taken verbatim or in sense from published or unpublished writings are identified as such. The work has not yet been submitted in the same or similar form or in excerpts as part of another examination.

Zürich, 01.07.2024



Signature of student

Abstract

This thesis investigates the collection of Bluetooth Low Energy (BLE) packets through passive sniffing to generate a large dataset for machine learning (ML) analysis. The primary objective is to collect an extensive dataset through sniffing to identify patterns that could pinpoint specific BLE devices, such as AirTags. This work focuses on the empirical data collection process, emphasizing the analysis of Received Signal Strength Indicator (RSSI) values at various distances to ascertain the feasibility of distinguishing individual AirTags in crowded environments. The collected data is utilized to enhance the functionality of the HomeScout application, a modular Bluetooth sensing app designed to inform users about trackers following them. By integrating RSSI shielding and ML approaches, the research aims to improve personal tracker detection capabilities, particularly for Android users who lack native AirTag tracking integration.

Acknowledgments

I want to express my deepest appreciation to Katharina Müller for her invaluable support throughout this thesis. As my supervisor, she offered not only valuable advice and constructive feedback but also dedicated her time generously to assist me whenever needed.

Additionally, I am grateful for the opportunity to conduct my bachelor's thesis within the Communication Systems Research Group (CSG) at the University of Zürich's Department of Informatics. The research field I delved into was both intellectually stimulating and challenging, and I am grateful for the chance to explore it further.

Contents

Declaration of Independence	i
Abstract	iii
Acknowledgments	v
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Goals	2
1.3 Methodology	2
1.4 Thesis Outline	2
2 Background	3
2.1 Bluetooth	3
2.1.1 Radio Spectrum	3
2.1.2 Receiver Sensitivity	4
2.1.3 Transmit Power	5
2.1.4 Antenna Gain	5
2.1.5 Path Loss	6
2.2 Bluetooth Low Energy Protocol Stack	6
2.2.1 Controller	7
2.2.2 Host	8
2.2.3 Application	11

2.3	BLE Communication	11
2.3.1	Broadcasting	11
2.3.2	Connections	12
2.4	Received Signal Strength Indicator	14
2.4.1	RSSI and Distance	14
2.4.2	RSSI and Signal Quality	15
2.4.3	Significance of RSSI Value Levels	15
2.5	Offline Finding Networks	16
2.5.1	Losing a Device	16
2.5.2	Finding a Device	16
2.5.3	Searching for a Device	17
2.6	Tracking Devices	18
2.6.1	AirTag	18
2.6.2	Tile	18
2.6.3	Chipolo ONE Spot	19
2.6.4	Samsung Galaxy SmartTag	19
2.7	Current uses of BLE in Localization	19
2.7.1	Triangulation	19
2.7.2	Fingerprinting	20
2.8	HomeScout	21
3	Related Work	23
3.1	Indoor Positioning System using BLE	23
3.2	Uses of Trilateration for localization	23
3.3	Protection from stalking attacks BLE	23

4 Design and Methodology	25
4.1 Experiments	25
4.1.1 Proximity Experiments	25
4.1.2 Experiments with Environmental Variables	26
4.2 RSSI Shield	27
5 Evaluation and Results	31
5.1 RSSI Value Comparison Across Datasets	31
5.2 Analysis of the Zürich Dataset	33
5.2.1 Data Preprocessing	33
5.2.2 Regression Analysis	34
5.2.3 Classification Analysis	37
5.3 Analysis of the Lugano Dataset	39
5.3.1 Data Preprocessing	39
5.3.2 Regression Analysis	40
5.3.3 Classification Analysis	41
5.4 Analysis and Comparisons of the two Datasets	42
5.4.1 Data Preprocessing	42
5.4.2 Regression Analysis	43
5.4.3 Classification Analysis	45
5.5 Analysis of Environmental Experiments	46
5.6 Limitations	48
6 Conclusions and Future Work	51
6.1 Conclusions	51
6.2 Future Work	52
Abbreviations	61
List of Figures	61

List of Tables	64
A Contents of the Repository	67

Chapter 1

Introduction

In 2023, 5.4 billion Bluetooth devices were shipped globally, an increase of more than 10% from 2022, as the number went from 4.9 billion to 5.4 billion, and is expected to reach 7.6 billion of annual shipments by the end of 2027 [1].

The data describes the increasing importance of Bluetooth technology in the past years, as such there is an increasing number of different Bluetooth tracking devices being manufactured, which will be the focus of this thesis. These devices support BLE, a network technology that finds its application in healthcare, fitness, security, and many other fields.

BLE devices, typically called beacons, became so known because of their reasonably compact size, low battery usage, and low price. A beacon transmits a globally unique identification token that is detected by a compatible operating system or software, hence utilizing BLE proximity sensing [2]. The approximate distance between the beacon and the client device is calculated using RSSI values. Because beacons emit radio waves, absorption and interference cause the RSSI value to fluctuate.

Some works already exist that analyze the reliability of RSSI for indoor localization [3, 4, 5] and utilize RSSI values to predict the distance between a beacon and a client device at the time of the measurement [6, 7]. However, neither approach was applicable in the detection of stalking attacks in offline finding networks. This will be the primary focus of this thesis, along with other subpoints specified in the following section.

1.1 Motivation

The proliferation of personal tracking devices, such as Apple's AirTag, has revolutionized the way individuals keep track of their personal belongings. These devices offer an unprecedented level of security by allowing users to locate lost items through their smartphones easily. However, this convenience is significantly diminished when it comes to distinguishing between multiple devices in crowded environments, especially for Android users. Unlike iOS, where AirTags owned by a user can easily be tracked with Apple's Find My network, Android users face the challenge of distinguishing their AirTags from others nearby. This issue is particularly pronounced in densely populated settings, such as

airports, where the presence of numerous AirTags can create confusion over which device belongs to whom.

The motivation behind this thesis stems from the understanding that, in the absence of native integration similar to that of iOS, Android users require alternative strategies to confidently identify their belongings. The core of this investigation revolves around leveraging AirTag’s signal strength values. Still, the methodologies and insights derived from this study are intended to be sufficiently generic, allowing for application to a broad spectrum of BLE tracking devices.

1.2 Thesis Goals

The goal of the thesis is to answer the following research questions:

1. Is it possible to predict the distance between a Non-Tracker Device (e.g., iPhone) and a Tracker Device (e.g., AirTag) using RSSI values?
2. How can RSSI values be leveraged in such a way that they allow us to distinguish between owned and unowned AirTags?
3. How do environmental conditions (e.g., indoor vs. outdoor, crowded vs. open space) impact the reliability of RSSI values for distance estimation between a Tracker Device and a Non-Tracker Device?

1.3 Methodology

Apple’s Find My App allows one to locate BLE devices both indoors and outdoors. For example, if keys or wallet are attached to an AirTag and one of these gets lost, it can be located with Apple’s Find My App, and it will tell the user the location of the AirTag [\[8\]](#). A dataset will be collected to analyze the behavior of AirTag’s strength values at different distances and in multiple environments.

1.4 Thesis Outline

The structure of the thesis is the following: background information on Bluetooth, RSSI, and current distance computation techniques is provided in Chapter 2. Indoor positioning systems with BLE, and the analysis conducted using the trilateration algorithm are presented in Chapter 3. Chapter 4 focuses on the design and methodology aspects, including a detailed explanation of the experiment settings and the design of a prototype for RSSI shielding to be integrated into the HomeScout application. Chapter 5 discusses the results and evaluation of the collected data and the implemented prototype. Finally, Chapter 6 concludes the thesis, summarizing the findings, discussing the thesis contributions, and suggesting future work.

Chapter 2

Background

This chapter introduces the foundational concepts and technologies essential to this thesis, specifically focusing on Bluetooth technology, Bluetooth Low Energy and its protocol stack, BLE communication methods, RSSI, and offline finding networks.

2.1 Bluetooth

Bluetooth technology allows devices to connect via wireless communication when these devices are close to one another. In a narrow spectrum centered on 2.4 Gigahertz (GHz), Bluetooth has 79 distinct radio frequencies. To establish a connection, two devices randomly select one of the 79 available frequencies, and then they repeatedly hop across these frequencies numerous times per second once a link is made. If the devices move too far apart, the connection will automatically be lost; once they are within range again, it will reestablish [9]. The range of Bluetooth depends on a few crucial factors:

2.1.1 Radio Spectrum

The radio spectrum spans frequencies of 30 Hertz (Hz) to 300 GHz. The range increases as the frequency decreases. Nevertheless, the maximum data rate it can handle decreases with decreasing frequency. There are therefore trade-offs between range and data rate when choosing a radio band. A fair trade-off between throughput and range is made possible by Bluetooth technology, which operates in the 2.4 GHz Industrial, Scientific and Medical (ISM) frequency band (2400 to 2483.5 Megahertz (MHz)). The 2.4 GHz band is also widely accessible, which makes it a real standard for low-power wireless communication [10]. The 2.4 GHz frequency band strikes a compromise between achieving a sufficiently long range and maintaining a high data rate. This band is designed to support minimal interference and is engineered for low power consumption.

Designation	Abbreviation	Frequencies	Wavelengths
Very Low Frequency	VLF	3 kHz - 30 kHz	100 km - 10 km
Low Frequency	LF	30 kHz - 300 kHz	10 km - 1 km
Medium Frequency	MF	300 kHz - 3 MHz	1 km - 100 m
High Frequency	HF	3 MHz - 30 MHz	100 m - 10 m
Very High Frequency	VHF	30 MHz - 300 MHz	10 m - 1 m
Ultra High Frequency	UHF	300 MHz - 3 GHz	1 m - 100 mm
Super High Frequency	SHF	3 GHz - 30 GHz	100 mm - 10 mm
Extremely High Frequency	EHF	30 GHz - 300 GHz	10 mm - 1 mm

Figure 2.1: Range of Radio Spectrum [11]

Figure 2.1 shows a list of the various Radio Frequency (RF) bands. RF bands are segmented portions of the electromagnetic spectrum that are utilized for various communication technologies. Each band's properties, such as their ground and sky wave propagation, vary according to frequency and wavelength. Lower frequency bands like Very Low Frequency (VLF), Low Frequency (LF), and Medium Frequency (MF) have longer wavelengths that enable communication over great distances and can spread around obstacles [11].

On the other hand, higher frequency bands like Very High Frequency (VHF), Ultra High Frequency (UHF), and beyond, have shorter wavelengths. They are generally used for line-of-sight communication, frequency modulation (FM) radio, television broadcasts, cellular networks, satellite communications, and Wi-Fi. These bands can carry more data due to their higher frequencies, making them suitable for modern wireless communication that requires high bandwidth [12].

2.1.2 Receiver Sensitivity

The lowest signal intensity a receiver can understand is known as receiver sensitivity. Alternatively, it represents the lowest power level at which the receiver can continue to

identify radio signals and establish a connection. Metaphorically, it can be considered as the lowest volume at which a human can perceive and comprehend sound.

The sensitivity of a receiver is typically measured in decibel-milliwatts (dBm). A lower (more negative) dBm value indicates a more sensitive receiver that can detect weaker signals. For example, a sensitivity of -90 dBm is better than -82 dBm, because the receiver can pick up even more delicate signals.

2.1.3 Transmit Power

A design trade-off between power consumption and range is made when selecting a transmit power level. The effective range and likelihood of the signal being detectable at a greater distance increase with transmit power. Nevertheless, the used gadget will use more power if the transmit power is increased. Consider transmitting power to be similar to speech loudness. Utilizing a louder voice consumes more energy but enhances the ability of the sound to travel farther distances and be heard by others.

The power output of a router, or any wireless transmitter, is quantifiable in two primary metrics: milliwatts (mW) and decibels relative to a milliwatt (dBm).

- Milliwatt (mW): this is a direct measurement of power, where one milliwatt represents a thousandth of a watt. To put this into perspective, a conventional light bulb may operate at around 40 watts. In contrast, a typical router's power output might be around 100mW, demonstrating a significantly lower power level of 400 times less than the light bulb.
- Decibel-milliwatt (dBm): dBm is a logarithmic scale used to express power levels. It offers a relative measurement starting from 0 dBm, which is equivalent to one milliwatt. As the power level increases tenfold, the dBm value increases by 10 dB. For example, 10 milliwatts equate to 10 dBm and 100 milliwatts to 20 dBm [13].

2.1.4 Antenna Gain

For the receiver, the antenna transforms electrical energy from the transmitter into electromagnetic energy, also known as radio waves, and vice versa. The efficiency of the signal's transmission and reception can be significantly influenced by the antenna's position, size, and design.

Antenna gain reflects how well an antenna can convert input power into radio waves in a specified direction. This gain is not about amplifying the power itself but rather focusing the transmitted energy more effectively. High-gain antennas can project signals further in particular directions, which is essential for long-distance communication. The placement of an antenna can greatly affect its performance: antennas mounted at higher elevations can reduce obstructions and increase the line-of-sight distance, which is particularly important in environments with many physical obstructions like buildings or trees [14].

2.1.5 Path Loss

The weakening of a radio wave's transmission during air propagation is known as path loss. The environment in which a signal is conveyed can affect path loss, also known as path attenuation, which happens naturally over distance. A signal's quality can be reduced by obstacles between the transmitter and the receiver. Urban environments, with their high density of buildings made of various materials like metal, can cause significant path loss through the reflection and dispersion of radio waves. Rural areas might experience less path loss due to fewer physical obstructions.

Moreover, atmospheric conditions such as humidity, rain, and fog can absorb and disperse radio waves, further contributing to path loss. The frequency of the signal also influences path loss; higher frequency signals tend to suffer more attenuation over the same distance compared to lower frequency signals [15].

2.2 Bluetooth Low Energy Protocol Stack

The references used for this section are [16], [17] and [18].

The protocol has the form of a stack and consists of three layers: Controller, Host, and Application. The Host Controller Interface (HCI) allows the Host and the Controller to communicate with each other. Figure 2.2 provides an overview of the BLE Protocol Stack and its layers.

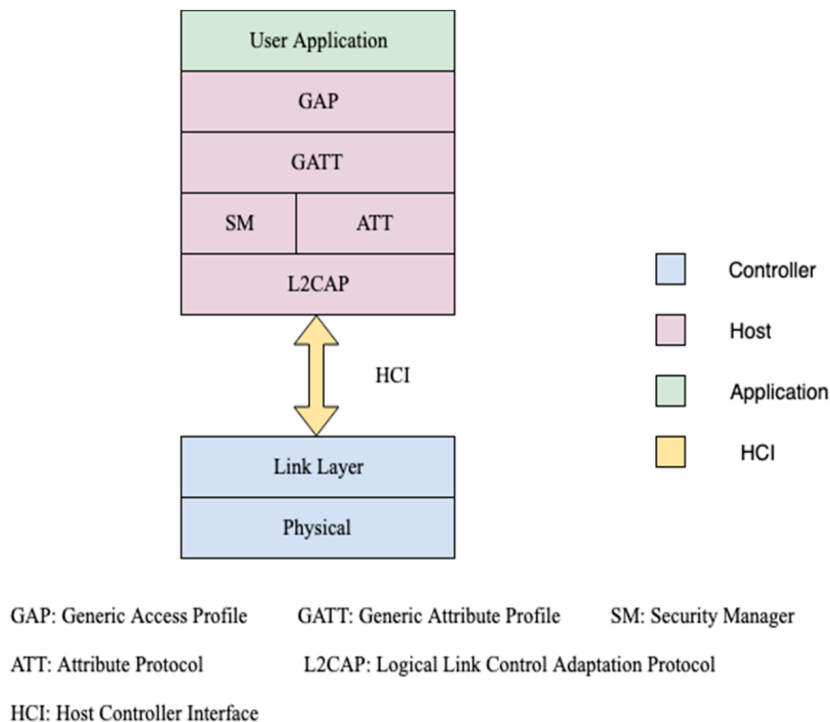


Figure 2.2: Architecture of BLE [17]

2.2.1 Controller

The controller is what most people recognize as a Bluetooth chip or radio. This definition is, however, over-simplistic, as it consists of hardware that enables packet transmission and receiving in addition to analog and digital radio frequency components. Via an antenna, the controller communicates with the external environment, and via the HCI, it communicates with the host. The controller is made of two layers: the Link Layer and the Physical Layer.

- **Link Layer:** the link layer (LL) is composed of a hardware part and a software part. This layer, by controlling the link state of the radio, establishes the kinds of communications that can be established between BLE devices. The link layer also defines the packet formats for the advertising channel and the data channel.

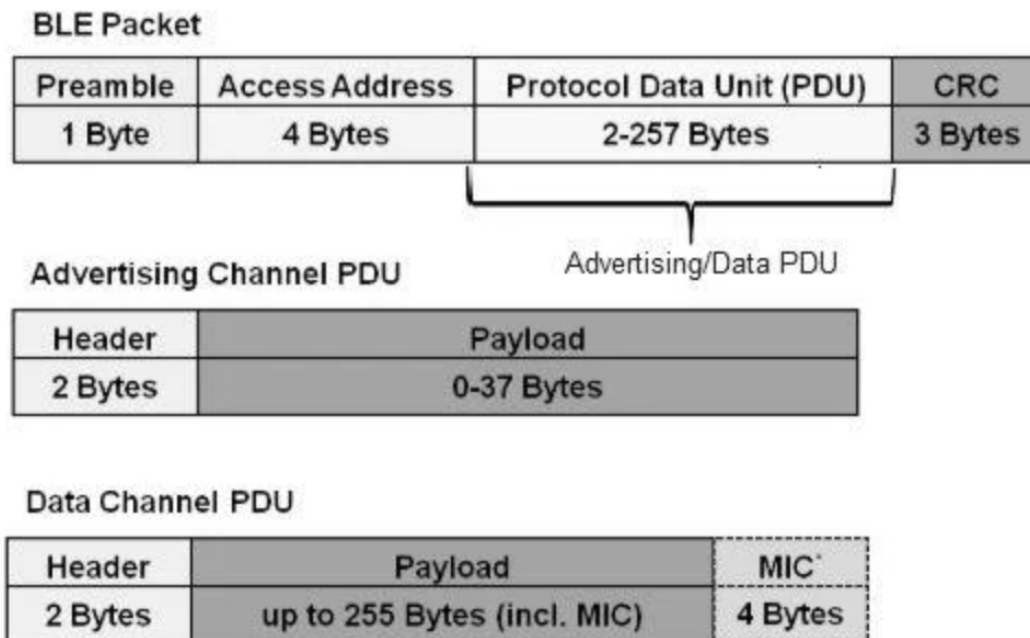


Figure 2.3: Structure of BLE packets [19]

A BLE packet, as it can be seen from Figure 2.3, needs 1 byte for the preamble, which is used by the receiver for time and frequency synchronization. The access address occupies 4 bytes and it uses different values depending on the type of packet [19]. The protocol data unit (PDU) can be either an advertising channel PDU or a data channel PDU. The Cyclic Redundancy Check (CRC) is 24 bits in size and is used for packets' error detection.

- Advertising Channel PDU: it broadcasts the data and can be categorized into different types of advertising PDUs depending on payload formats:
 - * Advertising PDUs: ADV_IND, ADV_DIRECT_IND, ADV_NONCONN_IND, ADV_SCAN_IND
 - * Scanning PDUs: SCAN_REQ, SCAN_RSP

- * Initiating PDUs: CONNECT_REQ
- Data Channel PDU: facilitates data transfer and reception between BLE devices after a link has been made between them.
- **Physical Layer:** the physical layer (PHY) in BLE technology specifies the modulation strategy and other methods it employs to transmit data over a particular RF band. This covers a wide range of factors, such as the number of channels available, how well those channels are used, the application of error correction, the safeguards against interference, and much more.

2.2.2 Host

The host manages the communication between the hardware and the user application, and it includes the following layers:

- **Generic Access Profile (GAP):** describes the procedure for device detection, connection formation, and standards-based interoperability between two BLE devices. It is collocated at the top-most level of the stack, and it communicates directly with the application layer and, through it, with the user, who can provide all the network's parameters. In addition, it acts as a conduit for user interaction with the entire stack protocol, implementing and managing all subprotocols. Figure 2.4 depicts a State Diagram of the GAP, outlining its states and components.

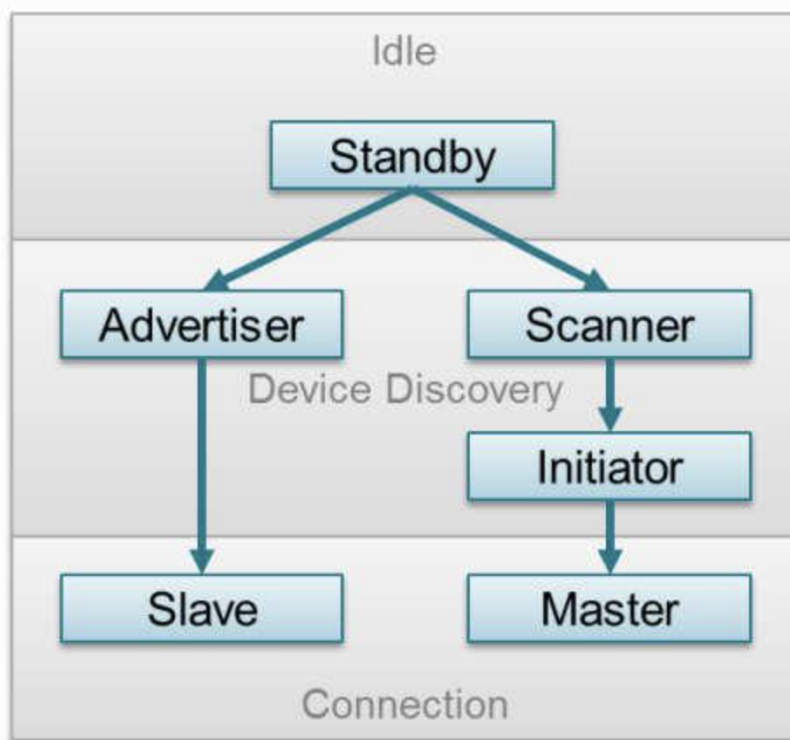


Figure 2.4: GAP State Diagram [20]

- **Generic Attribute Profile (GATT):** defines how the data is organized and exchanged in a BLE link. This information is arranged in a hierarchical framework made up of sections known as services that further subdivide information into containers known as characteristics. Figure 2.5 shows a GATT Server having two services (public and private), with a GATT Client executing several operations to read/write data in those services.

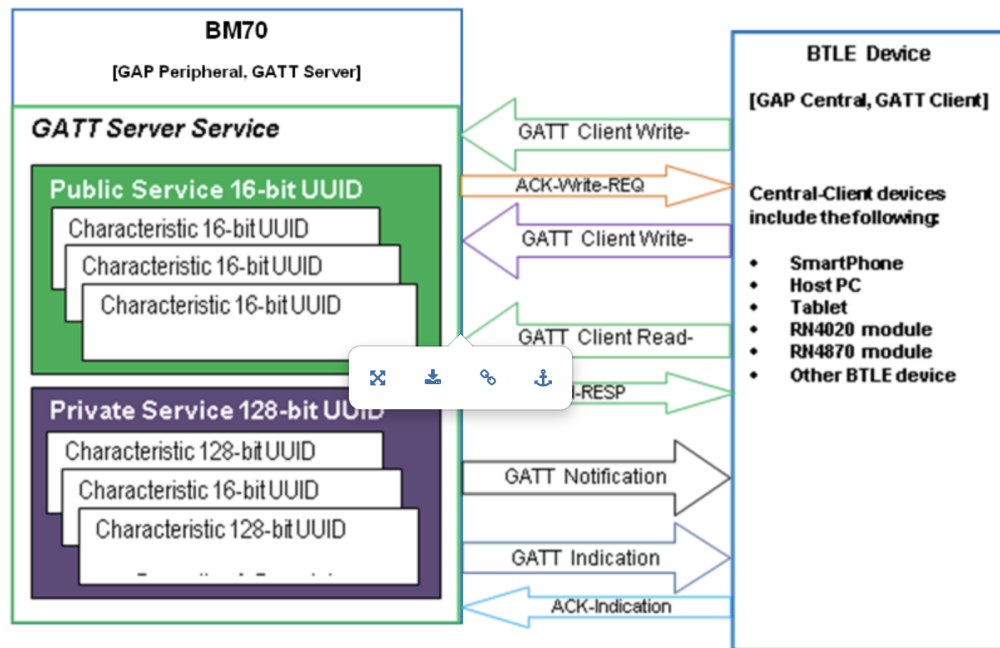


Figure 2.5: GATT Example [21]

- **Security Manager (SM):** the security manager is a protocol that enables safe communication between two BLE devices via an encrypted channel. It is therefore responsible for pairing with another device, which consists of trusting that device by authenticating it. The different pairing phases are illustrated in Figure 2.6.

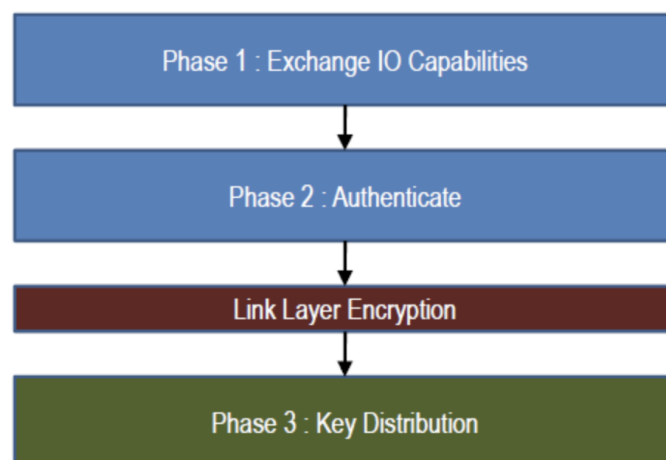


Figure 2.6: Security Manager Pairing Phases [22]

- **Attribute Protocol (ATT):** this protocol uses attributes to organize the data, where each attribute contains a 16-bit handle, a Universal Unique Identifier (UUID), a set of permissions, and a data value. This protocol can also define some attributes to have permissions, which allow a client device to read or write an attribute's value only in situations when the client has successfully authenticated or been granted permission by the server to view this value. Figure 2.7 illustrates the attribute's data structure .

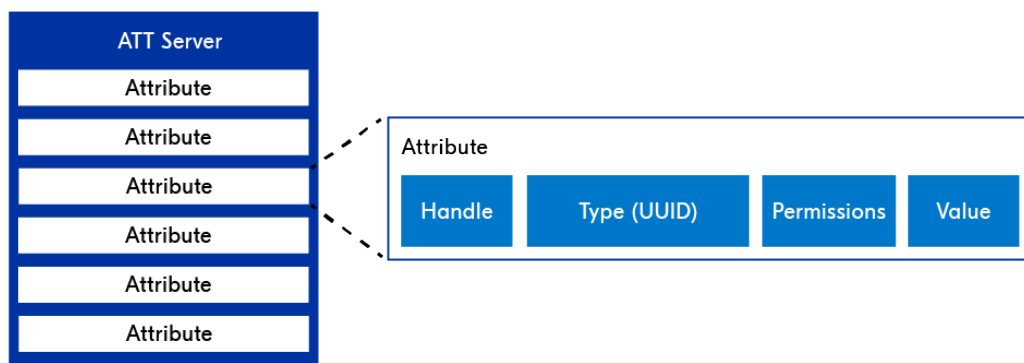


Figure 2.7: ATT attribute data structure [23]

- **Logical Link and Adaptation Protocol (L2CAP):** the L2CAP protocol serves two primary purposes:
 - (i) Fragmentation and recombination: packets received from higher layers get divided into smaller packets to fit within the maximum payload size of 27 bytes.
 - (ii) Encapsulation: combining several upper-layer protocols into a single, conventional BLE packet structure.

As indicated by its designation, it is responsible for adapting the upper and lower layers of the stack, to do this it takes the data from the lower layers and encapsulates it into the standard format of the BLE packet, according to the upper layers (fragmentation), and vice-versa (recombination). The detailed architecture is presented in Figure 2.8

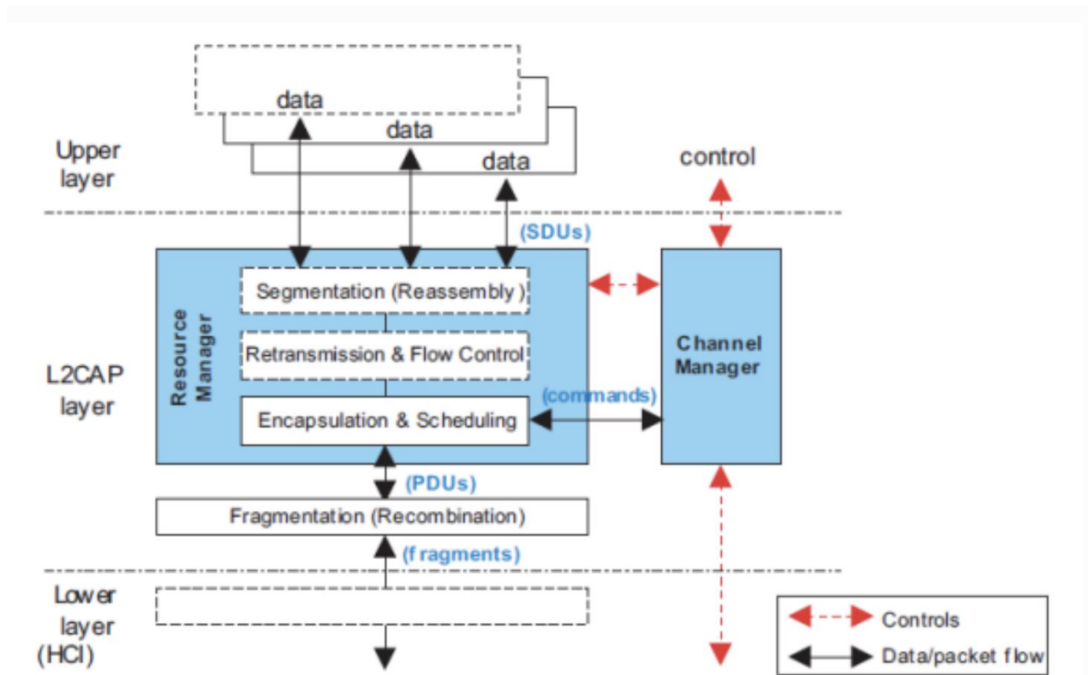


Figure 2.8: L2CAP Building Blocks [24]

2.2.3 Application

The application layer is the top layer of the BLE stack and is responsible for direct communication with users. It is responsible for the application logic and the user interface. The application layer defines three different specifications: characteristic, service, and profile. The Generic Attribute Profile serves as the foundation for each of these requirements. The Generic Attribute Profile defines grouping attributes for characteristics and services, and the application defines the specifications that use the attribute groups defined by the GATT.

2.3 BLE Communication

There are two types of communications that BLE devices use:

2.3.1 Broadcasting

Broadcasting is the quickest method to transfer data to more than one device at the same time, but its drawback is that it cannot be used for sensitive data as it lacks security and privacy controls [18]. This is due to broadcasting lacking robust security and privacy measures, as it transmits data openly. This means that any device within range can receive the broadcasted data without requiring explicit permission or verification, making it impossible to guarantee the confidentiality and integrity of the data.

Broadcasting packets serve two purposes: the first one is sending advertising packets to applications that do not require a fully active connection, and the second one is discovering slaves available for connection when a master sends advertising packets. There are two different roles in broadcasting:

1. Broadcaster (Advertiser): delivers advertising packets regularly to any device that can receive them. The broadcaster must set the Advertising Interval, which represents the rate at which the advertising packets are sent.
2. Observer (Scanner): periodically checks to see whether there are any advertising packets accessible from a broadcaster. This process is done continually. If the advertising interval is not set by the broadcaster, the observer sets the Scan Interval, which is the rate at which the radio scanner turns on. The observer also needs to set the Scan Window, which is the time the radio continues scanning for each scan interval [18]. Figure 2.9 depicts the broadcasting procedure performed by the Scanner and the Advertiser on different channels.

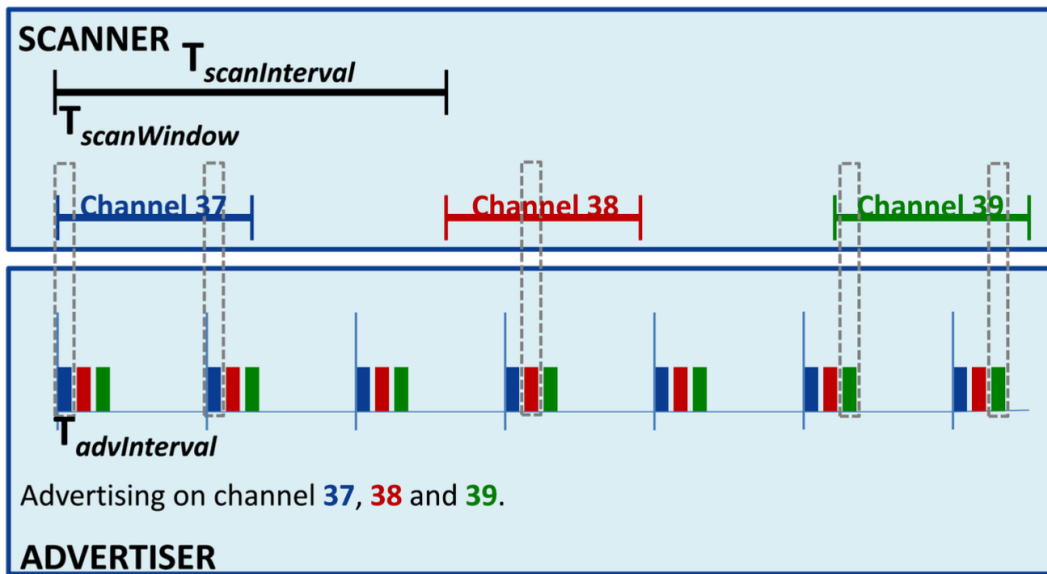


Figure 2.9: Advertising and scanning in broadcasting [18]

2.3.2 Connections

A connection is a continuous and recurring packet-by-packet data exchange between two devices. Connections are private, and in contrast to broadcasting, they promote the exchange of sensible data as they can be protected with security measures. The two different roles in a connection are Central (Master) and Peripheral (Slave). There is a distinction between the types of connections: passive and active connections [16].

Active Connections

- Central (Master): in an active connection, the master device actively scans for advertising packets from slave devices that are open for connection. The master device initiates the connection process, and as the connection becomes active, the master controls the timing and frequency of data exchanges, ensuring that communication is ongoing.
- Peripheral (Slave): the slave device periodically broadcasts advertising packets that indicate its availability to connect. When a master device initiates a connection, the slave accepts this connection. Throughout the active connection, the slave adheres to the parameters and timing set by the master device, participating in the data exchange as defined by the master.

Passive Connections

- Central (Master): the master device may listen to advertising packets without actively seeking to connect. The distinction lies in the master's role in managing a connection only after it's been established, through protocols that don't involve active scanning and initiation by the master device itself.
- Peripheral (Slave): the slave device periodically sends out advertising packets with the intent of being discovered by master devices. These advertising packets are broadcasted not only for devices actively scanning but also for those that might be in a passive scanning mode. While passive scanning devices do not actively request additional information from the advertiser (slave), they can initiate a connection based on the information passively collected from these packets. This allows slaves to communicate their presence and basic information with minimal energy consumption, relying on master devices to decide when to establish a more active connection.

A connection between a master and a slave follows a specific timeline: the time during which the master and the slave exchange packets with each other is called a Connection Event, while the rest of the time when the communication is disabled, is called Radio Idle. The anchor point is the start of a connection event, and exactly at this time, the master starts transmitting data to the slave.

The Connection Interval is the sum of the connection event and the radio idle. Within the interval of 7.5 ms to 4.0 s, the connection interval must be a multiple of 1.25 ms.

The Connection Supervision Timeout is the maximum amount of time that can pass without receiving two valid packets, and if it gets exceeded, the connection is lost. The Connection Slave Latency is the amount of connection events that can be disregarded without risking the connection to turn off. Figure [2.10](#) illustrates the timeline of a BLE connection with its parameters.

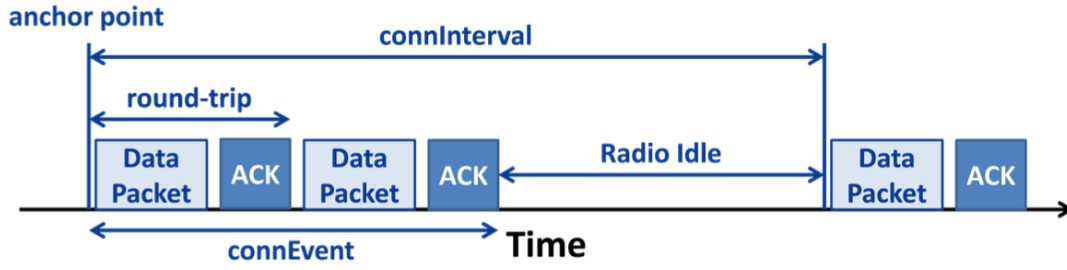


Figure 2.10: BLE connection with its parameters [18]

Two modes of communication can be chosen to exchange data between the master and the slave:

- One-way communication: the slave, in response to a poll, sends a notification to the master.
- Round-trip communication: the first step is done by the master that asks for data from the slave, afterwards the slave sends a response to the master.

2.4 Received Signal Strength Indicator

RSSI measures how well a device can hear a signal from an access point or router. RF systems, such as Bluetooth devices, Wi-Fi networks, and cellular networks, are the main applications of RSSI. Devices can evaluate the quality of the received signal thanks to its indication of the signal's power level. The received signal intensity is usually expressed in dBm. Higher RSSI values indicate stronger signals, while lower values suggest weaker signals. Several variables, including ambient conditions, interference, barriers, and distance, impact RSSI [25].

2.4.1 RSSI and Distance

Due to signal attenuation and propagation losses, the RSSI value generally decreases as the distance between the transmitter and receiver increases. Therefore, while not an exact distance measurement, RSSI can offer an approximate sense of how close devices are to one another [25].

The paper [26] proposes a formula to calculate RSSI values given the distance:

$$RSSI[dBm] = -(10n \log_{10} d - Tx) \quad (2.1)$$

where d is the distance between the transmitter and the receiver, n is a signal attenuation constant, and Tx is the strength of the signal being transmitted, measured at a distance of one meter from the transmitter.

From Equation [2.1](#) the variable d can be isolated; the equation reformulated looks as follows:

$$d[m] = 10^{\frac{RSSI - Tx}{-10n}} \quad (2.2)$$

2.4.2 RSSI and Signal Quality

RSSI is frequently used to assess the quality of a received signal. A stronger and more stable connection is typically indicated by a higher RSSI score, which also generally signals better signal quality. Lower RSSI levels, on the other hand, might indicate a weak signal that is prone to attenuation or interference [\[25\]](#).

2.4.3 Significance of RSSI Value Levels

Signal Strength	Signal Quality
-50 dBm	Excellent
-60 dBm	Very Good
-70 dBm	Good
-80 dBm	Poor
-90 dBm	Very Poor
-100 dBm	No Signal

Table 2.1: Table of RSSI value levels [\[27\]](#)

The relevance of various RSSI values for signal strength and quality is shown in Table [2.1](#). A great signal strength of -50 dBm denotes a strong and functioning connection between the devices. The signal quality declines in proportion to the RSSI value. For example, a very good signal is represented by a value of -60 dBm, while a value of -70 dBm indicates a good signal. However, the signal quality deteriorates and results in poor or very bad connections (-80 dBm for poor, -90 dBm for very poor) when the RSSI value falls further to -80 dBm and beyond. Lastly, a signal strength of -100 dBm indicates that there is no signal and that the signal received is insufficient to create a working connection.

2.5 Offline Finding Networks

Offline Finding Networks (OFN) utilize Bluetooth technology through online finder devices to locate misplaced gadgets or devices, transmitting their approximate location to the owner via the Internet. Using a set of known rolling public keys of the lost device, another owner device contacts the central server for location data when looking for a lost device. The location can be obtained by the owner by using the matching private key to decode the reports. Offline finding (OF) seeks to protect the confidentiality of location reports, and prevent owner device tracking and finder anonymity [28].

2.5.1 Losing a Device

When an OF device is disconnected from the Internet, it begins to broadcast BLE adverts. The public part of the advertisement key is 224 bits (28 bytes) long [29]. As shown in Figure 2.11, a BLE advertisement packet consists of a total of 37 bytes, with 6 bytes allocated for the advertising MAC address and up to 31 bytes available for the payload [30]. The 4-byte header is used for manufacturer-specific data, which leaves 27 bytes. The BLE standard requires that the first two bits of an address are $0b11$.

Bytes	Content (details cf. [6, § 5.1])
0–5	BLE address $((p_i[0] \mid (0b11 \ll 6)) \parallel p_i[1..5])$
6	Payload length in bytes (30)
7	Advertisement type (0xFF for manufacturer-specific data)
8–9	Company ID (0x004C)
10	OF type (0x12)
11	OF data length in bytes (25)
12	Status (e.g., battery level)
13–34	Public key bytes $p_i[6..27]$
35	Public key bits $p_i[0] \gg 6$
36	Hint (0x00 on iOS reports)

Figure 2.11: Offline Finding Advertising Format [28]

During a 15-minute interval, the same key is broadcasted, and upon the conclusion of this period, the subsequent key, designated as p_{i+1} , is used. OF-enabled iOS and macOS devices send one BLE advertising every two seconds when they lose connectivity.

2.5.2 Finding a Device

Every finder device periodically checks for OF adverts. An encrypted location report is created by the finder and uploaded to Apple’s servers whenever it gets a packet in the

OF advertising format. The public key is extracted from the advertisement by the finder. Next, it establishes its present geolocation and generates a message including its position and accuracy. Then, the finder generates a location report that includes timestamps, the public key, and the encrypted message. The binary representation of a location report is depicted in Figure 2.12.

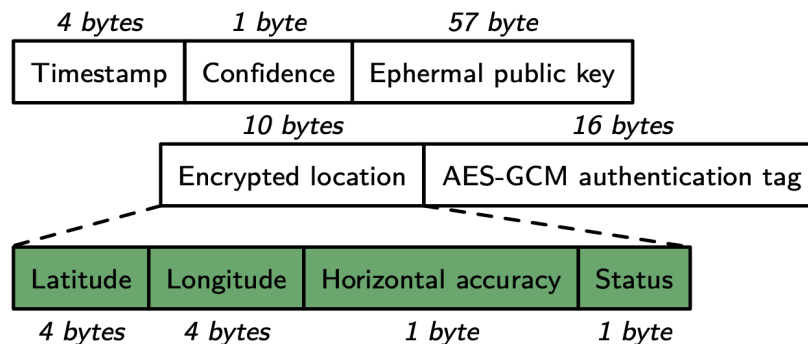


Figure 2.12: Binary representation of a location report [28]

2.5.3 Searching for a Device

When looking for a missing device, the owner of a device can ask Apple’s servers for the device’s reported position. The owner can obtain and decrypt the location reports using Apple’s Find My application on any other device because the advertising keys are synchronized across all of their devices. The detailed procedure adopted by Apple’s Find My Network is shown in Figure 2.13.

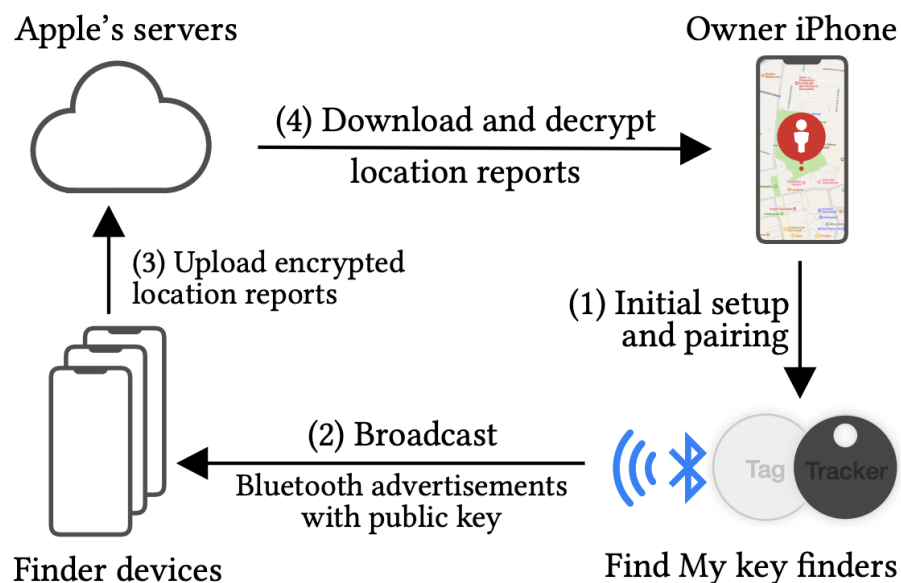


Figure 2.13: Apple’s Find My Network [31]

OS	Prevalent Type	Prevalent Content
Windows 10	ADV_NONCONN_IND	Company Identifier 0x0006
macOS, iOS	ADV_IND	Company Identifier 0x004c
Android	SCAN_REQ	Target Address
Fitbit	ADV_IND	Fitbit Service Class UUID
Surface Pen	ADV_DIRECT_IND	Target Address

Figure 2.14: Advertising Features of Different OSs [32]

Figure 2.14 illustrates that Apple’s operating system has 0x004c as the company identifier. The packet header of AirTags is 0x2560, and the length of the manufacturer-specific advertising data packet is 30 bytes. This information is crucial to be able to isolate the data related to AirTags from the rest of the data before conducting the analysis.

2.6 Tracking Devices

This section includes the main BLE Tracking Devices currently available on the market.

2.6.1 AirTag

AirTags are small coin-shaped tracking devices that weigh 11 grams and have a diameter of 31.9mm [33]. As the name says, they are used to track or tag objects that people are afraid of losing, such as keys and wallets. Thanks to Apple’s Find My App, the object to which an AirTag is attached can be located from an iPhone at any time, and its exact position will be provided [34]. With iOS 17, AirTags can be shared with up to 5 people, so that shared items, such as bikes or car keys, can be tracked simultaneously by different persons. When an item equipped with an AirTag is lost, the corresponding AirTag can be activated into Lost Mode. This way, when it gets detected by a device in the network, the user will automatically be notified. As security is an important aspect to consider with tracking devices, Apple only allows the owner of an AirTag to locate it, and it does not store any location or history data on the AirTag itself [34].

2.6.2 Tile

As with Apple AirTags, the purpose of Tile devices is to locate lost items. However, one additional feature that Tile provides is that it enables one to locate a nearby phone by clicking two times on a Tile device, which will make the phone ring [35]. There are different types of Tile trackers: the Tile Pro is the most powerful one, the Tile Mate is the most versatile, the Tile Slim is a slim Tile tracker that perfectly fits inside wallets,

has the shape of a credit card; and lastly, the Tile Sticker, which is a small Tile device that can be attached to any object and works like a sticker [36].

2.6.3 Chipolo ONE Spot

The Chipolo ONE Spot can be used with an Apple device inside the Apple's Find My App. However, they cannot be used for Android devices, but Chipolo ONE Point can [37]. There is also an app, called Chipolo App, which can be used with the original Chipolo trackers, but it is not compatible with Chipolo ONE Spot trackers or Chipolo ONE Point. Chipolo One Spot trackers are the most relevant for this thesis and they are water resistant, with a 1-year replaceable battery, have a range of up to 60 meters, and are very easy to hear as their sound can be of up to 120 dB. Also, they come in handy when searching for devices to attach to keys, as they have a keyring hole in them [38].

2.6.4 Samsung Galaxy SmartTag

This tracker serves the same purpose as the previously mentioned tracking devices, and it can be located with the SmartThings mobile app. The SmartTag+ uses ultra-wideband technology to locate the device and weighs 13 grams [39].

2.7 Current uses of BLE in Localization

In this section, two main algorithms that can be used to calculate the location of a mobile device or a tracker device based on the distance between the two are explained. These algorithms are Triangulation and Fingerprinting. It is important to acknowledge the limitation of Triangulation in the setting of this thesis due to the presence of only two data points: the AirTag and the device collecting BLE packets.

2.7.1 Triangulation

Triangulation methods work in a way that enables the location of a mobile device to be determined by triangulating the distances between a set of reference points, which results in an intersection point [5]. This method creates circles centered at the access points, with each circle's radius determined by either 1) the mobile device's observed RSSI value or 2) the amount of time it takes for the signal to be transmitted between the access point and the mobile device. When three or more access points are present within a specific range, an intersection point occurs, and the intersection point provides an approximate location for the mobile terminal. In Figure 2.15, x is the intersection point resulting from the triangulation algorithm performed with access points x_1 , x_2 and x_3 .

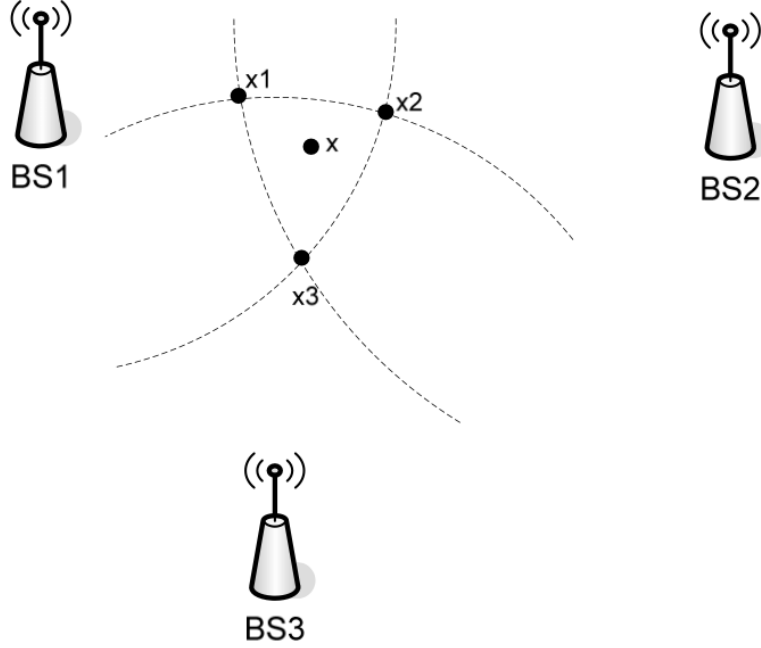


Figure 2.15: Estimation of location using triangulation. [5]

2.7.2 Fingerprinting

Fingerprinting is an algorithm for location estimation that obtains the user's location by comparing the derived RSSI values with a radio map. A radio map is a database that contains RSSI readings from different transmitters, such as Wi-Fi access points, Bluetooth beacons, or cellular towers, captured at many known locations within the region.

The radio map is created in an offline phase and includes the measured RSSI values at specific places. This avoids modeling complicated signal propagation and captures the characteristics of signal transmission in indoor contexts [40].

Using a floor plan as a guide, the area of interest is divided into cells to begin building the radio map. The radio map contains the RSSI values of the radio signals that access points broadcast, which are gathered for a certain amount of time at calibration locations inside the cells. The radio map's i th element has the following form:

$$\mathcal{M}_i = (B_i, \underbrace{\{\vec{a}_{ij} \mid j \in N_i\}}_{R_i \in R}, \theta_i), \quad i = 1, \dots, M \quad (2.3)$$

where B_i is the i th cell, \vec{a}_{ij} denotes the vector that contains the RSSI values measured from the access point AP_j and θ_i contains any other information used for the location estimation phase. R_i represents the i th fingerprint where R is the set of all fingerprints $R = \{R_1, \dots, R_M\}$. The i th element of the radio map is $\mathcal{M}_i = (B_i, R_i)$.

Afterward, either the deterministic or the probabilistic framework can be applied to infer the location from the received measurements. The accuracy of the location can be adjusted by applying either the Bayesian filter or the Kalman filter.

2.8 HomeScout

HomeScout is an Android application developed by a Master's student at the University of Zürich that allows users to customize an algorithm that informs users about trackers that are following them [41], according to the values of the parameters defined in the settings page, which is shown in Figure 2.16. This tracking algorithm can be used not only for BLE trackers but also for BLE devices in general, as they can also be the target of stalking attacks. As of now, HomeScout can scan BLE devices and identify different personal trackers with high precision using passively collected packets, but it does not contain an RSSI shielding feature that ensures only relevant devices with strong and clear signals are considered, while irrelevant and non-vulnerable devices are ignored.

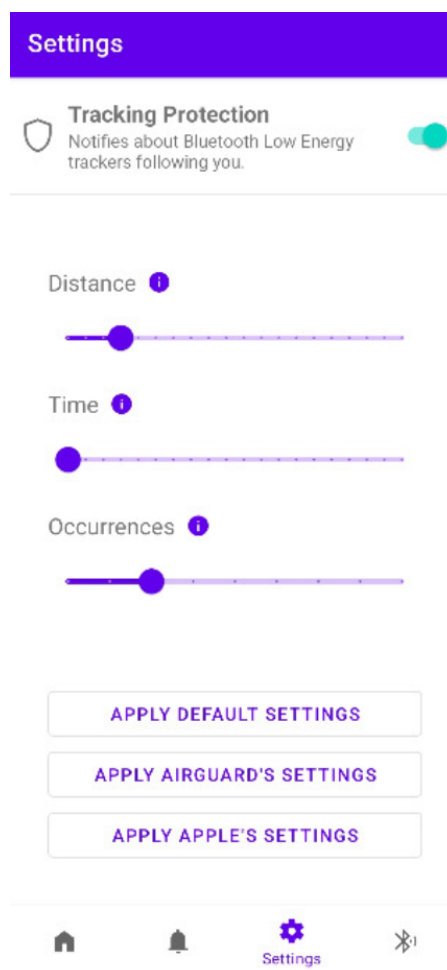


Figure 2.16: Settings of the tracking algorithm in the HomeScout application. [41]

Chapter 3

Related Work

This chapter reviews existing research related to this thesis. The works in this chapter focus primarily on advancements in indoor positioning systems using BLE and techniques for localization and protection against misuse of BLE tracking devices.

3.1 Indoor Positioning System using BLE

The works [42] and [43] use RSSI values to estimate the distance between different devices inside a building. The papers [44], [45], [46], and [47] add in addition to the use of RSSI values the integration of machine learning algorithms, such as Artificial Neural Networks, for distance estimation.

3.2 Uses of Trilateration for localization

The works [48], [49], [50], and [51] propose an indoor localization method that uses the trilateration algorithm to locate tags. The method through which trilateration operates is to identify a sequence of circles that intersect. Using triangles formed from three points and the observed lengths and angles, triangulation can calculate the distances between them. They concluded that RSSI signals are of crucial importance when it comes to determining the proximity of a certain tag to a reader antenna or device.

3.3 Protection from stalking attacks BLE

BLE location trackers allow for misuse and may be used to follow individuals. In an effort to create a vertically integrated solution, location tracker manufacturers like Apple have published suggestions for other manufacturers and included notifications within their ecosystem. However, this solution cannot be extended for detecting potential AirTags

used for stalking from Android devices, this is what the works [31], [41], [52] try to solve. Additionally, the OpenHaystack project [53] enables users to add their own accessories to Apple's Find My network. However, its application is limited to Apple devices only.

Chapter 4

Design and Methodology

This chapter delineates the experimental design and methodology adopted for the empirical collection of data from AirTags. The goal of the experiments is to accurately capture and analyze the RSSI values at varying distances to determine the feasibility of distinguishing an individual's AirTag in crowded areas.

The following sections provide an in-depth examination of the experimental setup, data collection procedures, and an explanation of the prototype created involving RSSI shielding for the HomeScout application.

4.1 Experiments

Two types of experiments will be conducted during the data collection phase: proximity experiments and environmental experiments, to analyze how the behavior of RSSI values changes over distance and in different settings.

4.1.1 Proximity Experiments

The experimental setup is systematically arranged to collect data at incremental distances from the AirTag to the detecting device. The progression is as follows:

1. Immediate Proximity (0m): The experiment begins with the detecting device in direct contact with the AirTag, establishing a baseline for RSSI values at zero distance.
2. Short Range Proximity (0.3m, 0.5m): Subsequent measurements are taken at short-range intervals of 0.3 meters and 0.5 meters, typical of personal space distances (e.g., pockets, backpacks).

3. Medium Range (2m, 4m): The setup then extends to medium-range distances, first at 2 meters, which could resemble an arm's length distance in daily interactions, followed by 4 meters, which is more indicative of a room's width or the distance within small group settings.
4. Long Range (10m): Finally, the detection is tested at a long-range distance of 10 meters, simulating scenarios where the AirTag might be across a large room or hall.

Each distance is methodically tested to capture and record the RSSI values, providing a comprehensive dataset that reflects the varying signal strengths at different distances. This sequence of distances was chosen to mirror common scenarios in which users might find themselves relative to their AirTag, from holding or wearing the item to being close to it but not in immediate proximity.

A laptop was placed on one side of the living room and any reachable Bluetooth packet was captured by the nRF board, including the strength values emitted by AirTags placed at the pre-defined distance. This procedure was conducted both in Zürich, in an environment (room in a palace) that tried to simulate a real-world crowded scenario, and in Lugano, where the same experiment was done in a controlled and isolated setting, and its goal was to be able to read clean RSSI values and understand what is the behavior of RSSI values without interfering signals.

4.1.2 Experiments with Environmental Variables

In addition to the distance measurements, it is crucial to account for environmental variables that can affect the detection and accuracy of RSSI values. This subsection outlines the experiments designed to simulate real-world scenarios, where variables such as physical obstructions and the dynamic nature of human movement are introduced. The different experiments try to simulate the following real-world scenarios:

1. Human Interaction: To mimic conditions where human bodies interfere with signal propagation, the experiment includes scenarios where a person walks between the AirTag and the detector. This can help in understanding how RSSI fluctuates with moving obstacles like a crowd in a public space. The data collected at a distance of 2 meters without and then with a person walking in between will be compared.
2. Multiple AirTag Interference: Scenarios where multiple AirTags are present are tested to observe how the presence of additional BLE devices influences the RSSI values of the target AirTag. The data collected at a distance of 2 meters with 1 AirTag and then with 10 AirTags will be compared.
3. Environmental Obstacles: it involves assessing RSSI fluctuations when AirTags are placed in various indoor scenarios with potential physical obstructions. This could include behind walls and inside closets, to simulate the signal behavior in typical everyday indoor spaces. The data collected at a distance of 2 meters inside a closet and then outside the same closet will be compared.

As in the previous experiments, a laptop was placed on one side of the living room and the AirTags at a specific distance (i.e., 2m) from it. Then, RSSI values were measured when the person was not walking in between (similar to proximity experiments). Afterward, RSSI values were measured when a person walked in between (for the human interaction experiment). After collecting these two datasets their RSSI values were compared. For the multiple AirTag interference experiment a similar approach was used to first measure the data at a pre-defined distance for just 1 AirTag, and then data was collected for 10 AirTags at that distance.

The experiment that tested how the behavior of RSSI values changes according to environmental obstacles was conducted as follows: first, a dataset was collected with all the AirTags placed at 2 meters from the nRF board. Then, all the AirTags were put inside a closet at the same distance and after the data collection, the datasets were compared with each other.

4.2 RSSI Shield

As [41] mentions, HomeScout currently lacks an implementation that uses RSSI as a shield to filter and ignore signals from BLE devices that are owned by the user or that are considered too far away to act as stalking ware. By setting a threshold RSSI value (e.g., -90 dBm), the `BLUETOOTHSCANNINGSERVICE` can ignore (or shield against) devices that are beyond a certain distance, as it is known that a weaker signal strength correlates with greater distance.

The implementation of this filter involves ignoring devices that are too far away and focusing the attention only on closer devices that could realistically be attempting to track the user.

The first step involved adding a variable to the `BLUETOOTHSCANNINGSERVICE` that holds the value of the RSSI threshold, which is stored in a companion object.

```
1 class BluetoothScanningService : LifecycleService() {  
2  
3     companion object {  
4         const val RSSI_THRESHOLD = -90  
5     }  
6  
7     // rest of code  
8  
9 }
```

Listing 4.1: Code Snippet of the companion object for the RSSI Threshold

The `onScanResult` function needs to be updated to only display BLE packets that are greater than the RSSI threshold being set.

The following code snippet displays the updated *onScanResult* function.

```
1  override fun onScanResult(callbackType: Int, result: ScanResult) {  
2  
3      lastKnownLocation?.let {  
4          if (result.rssi > RSSI_THRESHOLD) {  
5  
6              val mac = result.device.address  
7  
8              val rssi = result.rssi  
9  
10             if (!scanResults.containsKey(mac)) {  
11  
12                 val timestampInMilliseconds = Calendar.getInstance().timeInMillis  
13                 val lat = it.latitude  
14                 val lng = it.longitude  
15                 val deviceType = DeviceTypeManager.identifyDeviceType(result).  
16                     type  
17  
18                 val bleDevice = BLEDevice(  
19                     mac,  
20                     timestampInMilliseconds,  
21                     lat,  
22                     lng,  
23                     deviceType,  
24                     rssi)  
25  
26                 scanResults[mac] = bleDevice  
27             }  
28         }  
29     }  
30 }  
31
```

Listing 4.2: Code snippet of the updated *onScanResult* function.

Originally, the `SETTINGSFRAGMENT` featured filters such as "Default," "Apple," and "Air-Guard." However, to better cover diverse user needs and improve detection capabilities, these were replaced with two new filters: "NearField" and "Persistent."

- **NearField:** it optimizes the app to detect devices with strong signals over a short time. It is best suited for crowded areas where a user expects devices to be very close.
- **Persistent:** it focuses on devices detected multiple times over a longer period. It uses a less restrictive RSSI threshold than the NearField filter, allowing it to capture most of the signals. This setting is best for identifying trackers that are consistently following a user.

Listing 4.3 displays the values used to accommodate the different filters. The feasibility and reliability of these values in real-world scenarios are discussed in the next chapter.

```
1 class SettingsFragment : Fragment() {
2     private fun setupButtonsForDifferentTrackingPreferences() {
3
4         // NearField filter
5         binding.buttonNearField.setOnClickListener {
6             val distance = 50.0F
7             val timeInMin = 1.0F
8             val occurrences = 2.0F
9             val rssiThreshold = -70.0F
10
11             binding.sliderDistance.value = distance
12             binding.sliderTimeInMin.value = timeInMin
13             binding.sliderOccurrences.value = occurrences
14             binding.sliderRssiThreshold.value = rssiThreshold
15
16             settingsViewModel.updateDistance(distance)
17             settingsViewModel.updateTimeInMin(timeInMin)
18             settingsViewModel.updateOccurrences(occurrences)
19             settingsViewModel.updateRssiThreshold(rssiThreshold)
20         }
21
22         // Persistent filter
23         binding.buttonPersistent.setOnClickListener {
24             val distance = 1000.0F
25             val timeInMin = 20.0F
26             val occurrences = 2.0F
27             val rssiThreshold = -90.0F
28
29             binding.sliderDistance.value = distance
30             binding.sliderTimeInMin.value = timeInMin
31             binding.sliderOccurrences.value = occurrences
32             binding.sliderRssiThreshold.value = rssiThreshold
33
34             settingsViewModel.updateDistance(distance)
35             settingsViewModel.updateTimeInMin(timeInMin)
36             settingsViewModel.updateOccurrences(occurrences)
37             settingsViewModel.updateRssiThreshold(rssiThreshold)
38         }
39     }
40 }
41 }
42 }
```

Listing 4.3: Code snippet of the updated SettingsFragment

Chapter 5

Evaluation and Results

This chapter aims to showcase the behavior of RSSI values depending on the environment (specified in [Experiments](#)) where the different data sets were collected.

Linear and exponential regression models have been built for both the datasets from Zürich and Lugano that predict the distance between the two devices given the RSSI value, which is the single predictor variable of the model. Subsequently, a classification task was performed by categorizing the continuous distance predictions into discrete bins to assess whether this approach would increase the practical utility of the predictions. The regression and classification analyses were performed on the Zürich and Lugano datasets.

5.1 RSSI Value Comparison Across Datasets

The following tables showcase the different RSSI value levels related to the measurement distance between the six different complete datasets collected in Zürich with the aim to simulate a crowded scenario and the two collected in Lugano whose goal is to be able to read clean RSSI values, as mentioned in [Experiments](#).

Distance (m)	RSSI value (dBm)
0.0	-28.63
0.3	-44.83
0.5	-51.21
1.0	-64.61
2.0	-67.35
4.0	-77.09
10.0	-79.62

Table 5.1: RSSI Value Levels from the Zürich Dataset

Distance (m)	RSSI value (dBm)
0.0	-19.99
0.3	-58.72
0.5	-61.28
1.0	-64.85
2.0	-73.26
4.0	-76.36
10.0	-82.88

Table 5.2: RSSI Value Levels from the Lugano Dataset

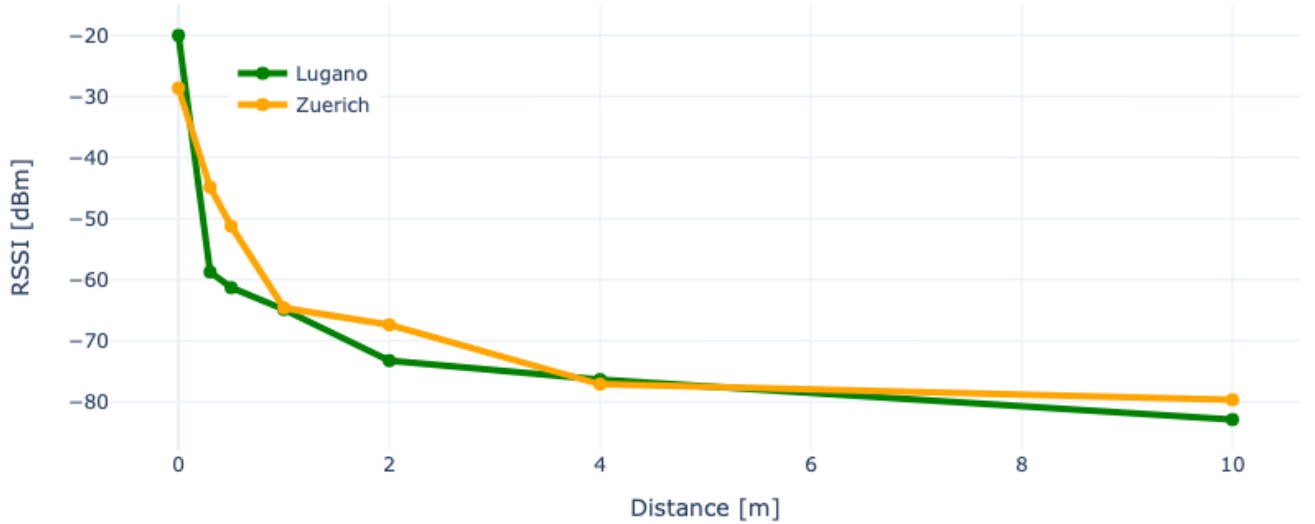


Figure 5.1: Comparison of RSSI values between datasets

Figure 5.1 shows that the RSSI values collected in Lugano seem to be generally lower than the ones collected in Zürich, except for the measurements at distances of 0 meters and 4 meters. This can be caused by having a high number of devices nearby for the Zürich dataset, causing the RSSI values to be inflated due to overlapping signals. The smallest difference between the two datasets is 0.24 dBm (at a distance of 1 meter), whereas the largest difference is 13.89 dBm (at a distance of 0.3 meters).

When comparing Table 5.1 and 5.2 with Table 2.1 from [27], it can be seen for all the values measured between -20dBm and -50dBm there is no category assigned indicating the signal quality. Additionally, it is important to note that the distance at which the measurements were taken in [27] is unknown, which further limits the direct applicability of their categorization to the collected datasets. Consequently, any categorizations or conclusions drawn from [27] should not be used as is but should be adapted to the

specific environment and conditions of the conducted experiment. This adaptation is essential because significant changes have been observed between different locations and environments in the collected data.

5.2 Analysis of the Zürich Dataset

This section presents a comprehensive analysis of the dataset collected in Zürich. The analysis is divided into several key steps, including data preprocessing, regression analysis, and classification analysis.

Data preprocessing is vital to cleaning and preparing the data, addressing errors, missing values, and inconsistencies. It ensures that the dataset is suitable for analysis and that algorithms perform optimally. Following data preprocessing, regression analysis is conducted to understand the relationships between variables. This type of analysis is crucial for predicting continuous outcomes based on the input variables. By applying regression techniques, trends can be identified, predictions can be made, and the strength of the relationships between different variables in the dataset can be quantified.

The final step is classification analysis, which aims to develop a predictive model that accurately categorizes the distance between two devices based on their RSSI values. It is essential for understanding group differences within the data.

5.2.1 Data Preprocessing

During the data preprocessing step, the dataset has been filtered only to contain rows whose *AdvPduType* is *ADV_IND*, the *PacketHeader* is equal to *0x2560* and the length of the packet is 30 bytes. This filtering process was done to remove data not related to AirTags. As the next step, the column *DTAP* (Distance According to Paper) was added and calculated with Equation 2.2 by using $n = 2$ (path loss exponent in free space) and the mean RSSI value of -64.61 dBm (according to Table 5.1). This calculation was performed to verify if the formula from [26] accurately describes the distribution of the collected data and to assess any deviation from the actual distances. The values inside the RSSI, Distance, and DATP columns have been all transformed into *Float* data type for consistency in data representation and to ensure compatibility for analysis and modeling purposes.

To avoid biases in the analysis, as the dataset had a different number of observations depending on the distance category (unbalances among the classes), a subset of the data has been taken so that for each class there are exactly 13244 observations (according to the class with the least number of observations at the beginning). This approach ensures that distances with more observations do not disproportionately influence the results compared to distances with fewer observations, thereby promoting balanced contributions from all distance categories to the outcome.

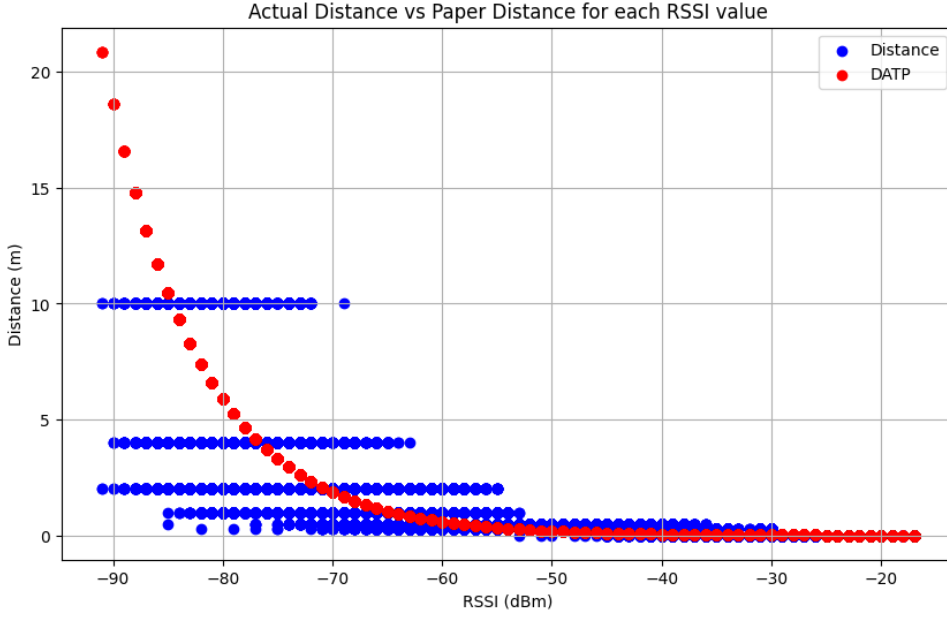


Figure 5.2: Relationship between Distance and DATP Variables

Figure 5.2 shows the difference between the data collected at the different distances versus the distance calculated with Equation 2.2 according to the RSSI values in the dataset. The plot shows that the two variables are correlated and they both follow the exponential distribution. The mean difference between the labeled distance and the distance predicted by Equation 2.2 is 0.04 meters. The median difference is 0.07 meters, whereas the minimum difference is -18.87 meters, and the maximum difference is 8.34 meters. This result indicates that, with few exceptions, there is almost no difference between the distance given by the formula presented in [27] and the actual distance when using free space (air) as the signal attenuation constant. This small difference demonstrates a similar distribution between the collected data and the equation from [26], which confirms the formula's accuracy in estimating distances based on RSSI values.

5.2.2 Regression Analysis

The goal of this analysis is to infer from the data the type of relationship that exists between RSSI values and the distance between two devices. Understanding this relationship is important as it enables the prediction of the physical distance between devices based on their RSSI values, which is crucial for the detection and identification of malicious trackers. For example, if a user suspects that a device is nearby, he/she can use RSSI values to determine its proximity. If the RSSI values indicate that a device is unexpectedly close, it could be a sign of a potential malicious tracker, assuming the user does not own any AirTags.

Regression analysis is essential for achieving accurate predictions and reliable detection by using different models to identify the best fit for the data. The process begins with selecting the appropriate regression approach, such as linear or exponential regression, to

determine which best captures the relationship between RSSI values and distance. Afterward, each model is fitted to the data, and its performance is evaluated based on metrics such as R-squared or Mean Squared Error. The model with the highest accuracy and lowest error rates is considered the most suitable. Additionally, it is essential to check the assumptions of the regression models, such as linearity, homoscedasticity, independence, and normality of residuals. Ensuring these assumptions hold true is essential for the validity of the results provided by the regression analysis. This step not only provides a deeper understanding of how RSSI values correlate with distance but also ensures that the predictive models are reliable for practical applications.

Linear Regression

Before making predictions with the linear model, the assumptions need to be checked. This can be done straightforwardly with the four diagnostic plots shown in Figure 5.3.

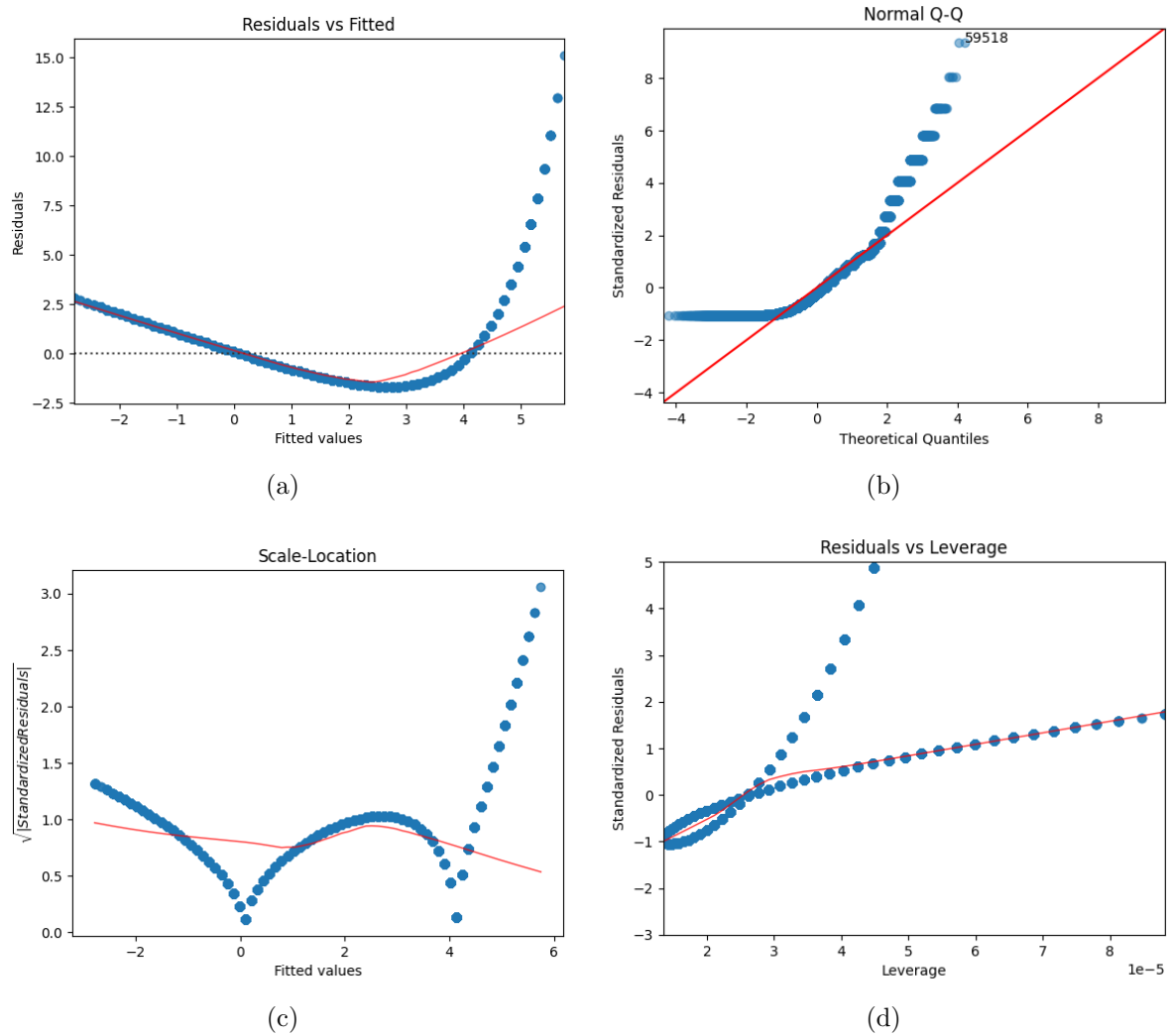


Figure 5.3: Diagnostic Plots of Linear Model for the Zürich Dataset

- Residuals vs Fitted plot: checks the assumption of linearity between the residuals. The plot in Figure 5.2 shows that the residuals are not randomly scattered around the horizontal line, which suggests that the linearity assumption is violated. This is important because it indicates that the model may not be capturing all the underlying patterns in the data. When the linearity assumption is violated, it means that there might be a non-linear relationship between the predictor variables and the response variable that the current model fails to address. This can lead to biased estimates and poor predictive performance. Checking this plot helps in diagnosing model misspecification and guides the need for considering more complex models or transformations to fit the data better.
- Normal Q-Q plot: checks the assumption of normally distributed residuals. The plot shows that the residuals deviate significantly from the diagonal line, indicating non-normality. Additionally, there is a small vertical jump with no values in between, which could be caused by the difficulty of replicating the exact setting used in the data collection process for the proximity experiments or other device signals interfering with the signal emitted by AirTags. The plot suggests that this assumption is violated. This is important because normally distributed residuals ensure the validity of hypothesis tests and confidence intervals. When this assumption is violated, it implies that the linear regression model may not be appropriate for the collected data, potentially leading to unreliable estimates and inferences.
- Scale-Location plot: checks the assumption of homoscedasticity (constant variance between the residuals). The plot shows a pattern where the spread of the standardized residuals is not constant across the range of fitted values. This indicates heteroscedasticity, implying that the assumption of constant variance between the residuals is violated. This is important because homoscedasticity is a key assumption in linear regression, ensuring that the model's errors are evenly distributed across all levels of the independent variables. Heteroscedasticity can result in biased standard errors, which affect the reliability of hypothesis tests and confidence intervals.
- Residuals vs Leverage plot: checks for influential points that might affect the regression model. The plot shows high standardized residuals which are potential outliers. This is important because influential points can disproportionately affect the fit of the regression model, leading to misleading estimates and conclusions. However, given that the other diagnostic plots clearly show that the assumptions of normality and linearity are violated, it is evident that a linear model cannot be used for inference purposes on this data.

Since the data does not follow a normal distribution, an attempt was made to model it using an exponential regression approach, as illustrated by Equation 2.2, which describes the exponential relationship between RSSI values and the distance between two devices. The purpose of employing linear regression was to assess whether the collected data conformed to this theoretical exponential relationship or exhibited indications of linearity. Figure 5.3 confirmed the non-normal distribution of the data, and Figure 5.4 demonstrates that the relationship between RSSI values and distance for the collected data is exponential.

Exponential Regression

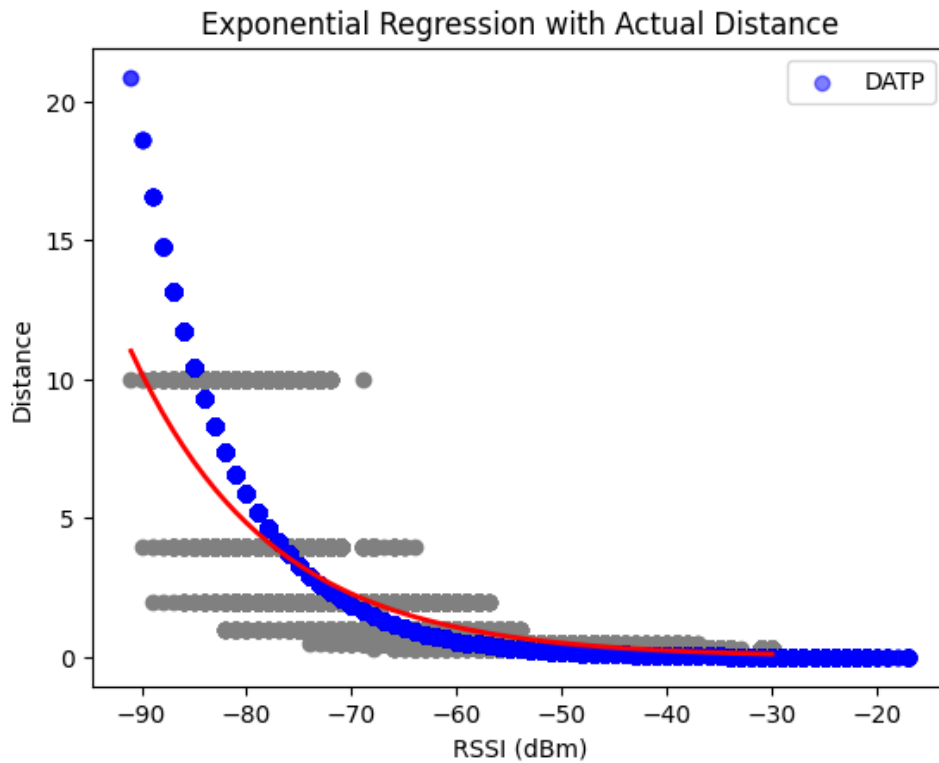


Figure 5.4: Line of Exponential Regression for Zürich Dataset

Figure 5.4 shows the exponential regression model fit on the data from Zürich with a blue line indicating the distance predicted by equation 2.2 for the respective RSSI value. As the equation and the plot show, the relationship between RSSI values and the distance between two devices is exponential. This is evident from the plot, where the red curve represents the exponential regression line fitting the data points. The curve decreases rapidly at first and then flattens out, indicating an exponential decay pattern. The R-squared of the model is 0.5021 (defined between 0 and 1; the higher, the better), and the Mean Squared Error is 5.6370 (the closer to 0, the better).

5.2.3 Classification Analysis

Classification analysis aims to predict the distance category into which a given distance falls. The chosen categories are:

- 0 - 1 meters
- 1 - 2 meters
- 2 - 4 meters

- 4 - 10 meters
- More than 10 meters

These specific distance categories were selected based on insights derived from the exponential regression model. The model revealed a pattern of drastic changes in measured distances at the lower ranges, followed by a more gradual plateau as distances increased. By using exponential regression, the study aimed to identify categories that capture these variations effectively, ensuring the classification model is appropriately tuned to distinguish between different stalking scenarios based on distance. To identify the best classifier for this task, several algorithms were evaluated, such as Random Forest, Decision Tree, Naive Bayes Classifier, and Multilayer Perceptron (MLP). By testing a diverse set of algorithms, it becomes possible to evaluate their performance with different datasets and compare them effectively. This approach facilitates the selection of the most effective classifier, thereby ensuring optimal accuracy and reliability in prediction tasks.

A cross-validation approach was conducted to test the over/under-fitting of the different models, as well as their accuracy on unseen data. Overfitting occurs when a model learns not only the underlying patterns in the training data but also the noise and random fluctuations. As a result, while the model may perform exceptionally well on the training data, its performance significantly deteriorates on new, unseen data. Overfitting is problematic because it leads to poor generalization, where the model fails to predict outcomes for data outside the training set accurately.

The cross-validation approach involves partitioning the dataset into multiple subsets and training the model on some subsets while testing it on others. This process is repeated several times to ensure that the model's performance is consistent and not overly dependent on a particular subset of data. By using cross-validation, models' performance can be assessed more rigorously and ensure that the selected model has good predictive power and robustness.

The models performed as follows:

Table 5.3: Model Performance Comparison for Zürich Dataset

Model	Training R-squared	Cross-validated R-squared	Diff	Test Accuracy
Random Forest	0.7329	0.7302	0.0027	0.6593
Decision Tree	0.7330	0.7302	0.0028	0.6593
Naive Bayes	0.5575	0.5573	0.0002	0.6850
MLP	0.7238	0.7164	0.0074	0.6466

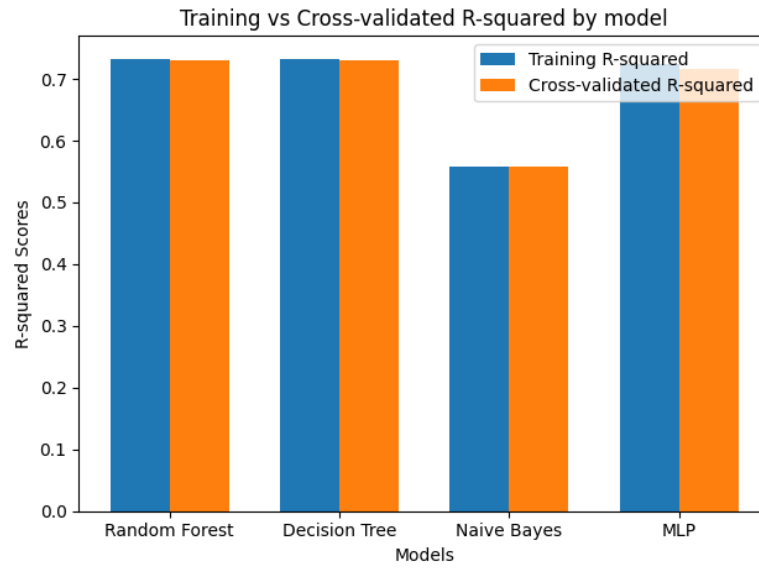


Figure 5.5: Comparison of Training R-squared and Cross-validated R-squared by model

The best model is the Naive Bayes Classifier, it has the lowest difference between training R-squared score and cross-validated R-squared score and the highest accuracy among all the models (0.6850). Figure 5.5 and Table 5.3 indicate that none of the models exhibit overfitting, as evidenced by the minimal difference between the Training R-squared and Cross-validated R-squared values across all models.

5.3 Analysis of the Lugano Dataset

This section provides an in-depth analysis of the dataset collected in Lugano. The structure of the analysis is similar to the one applied to the Zürich dataset, focusing on data preprocessing, regression analysis, and classification analysis.

5.3.1 Data Preprocessing

The data preprocessing step has been done in a similar way to the Zürich dataset. The subset of the data contained 3134 observations for each class (according to the class with the least number of observations at the beginning). This is lower than the number of observations per class for the Zürich dataset (as defined in Data Preprocessing).

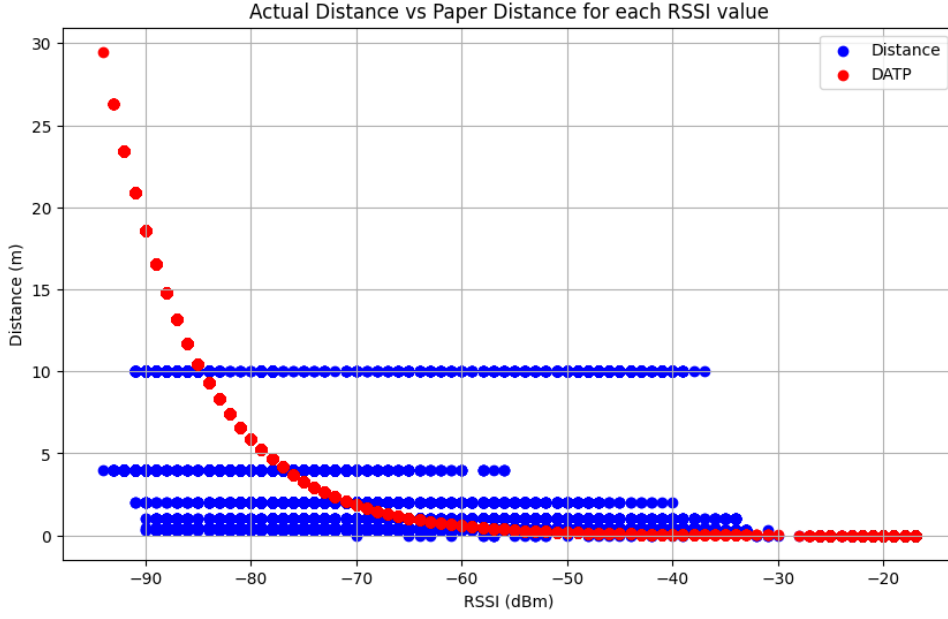


Figure 5.6: Relationship between Distance and DATP Variables

Figure 5.6 shows a similar pattern to Figure 5.2. The mean difference between the labeled distance and the distance predicted by Equation 2.2 is -1.4 meters. The median difference is 0.82 meters, whereas the minimum difference is -25.48 meters, and the maximum difference is 9.96 meters. This result indicates that, compared to the Zürich dataset, the distance given by the formula presented in [26] differs remarkably from the actual distance when using free space (air) as the signal attenuation constant for the Lugano dataset. The significance of this result should be investigated further to assess whether this is due to having a controlled environment with zero interfering signals or if this is caused by having a small data set.

5.3.2 Regression Analysis

This section aims to analyze the regression analysis for the Lugano dataset and compare the exponential regression model with the one from the Zürich dataset.

Linear Regression

Since the data does not conform to a normal distribution but instead follows an exponential distribution (similar in diagnostic plots and distribution to the Zürich data), it is directly modeled using exponential regression.

Exponential Regression

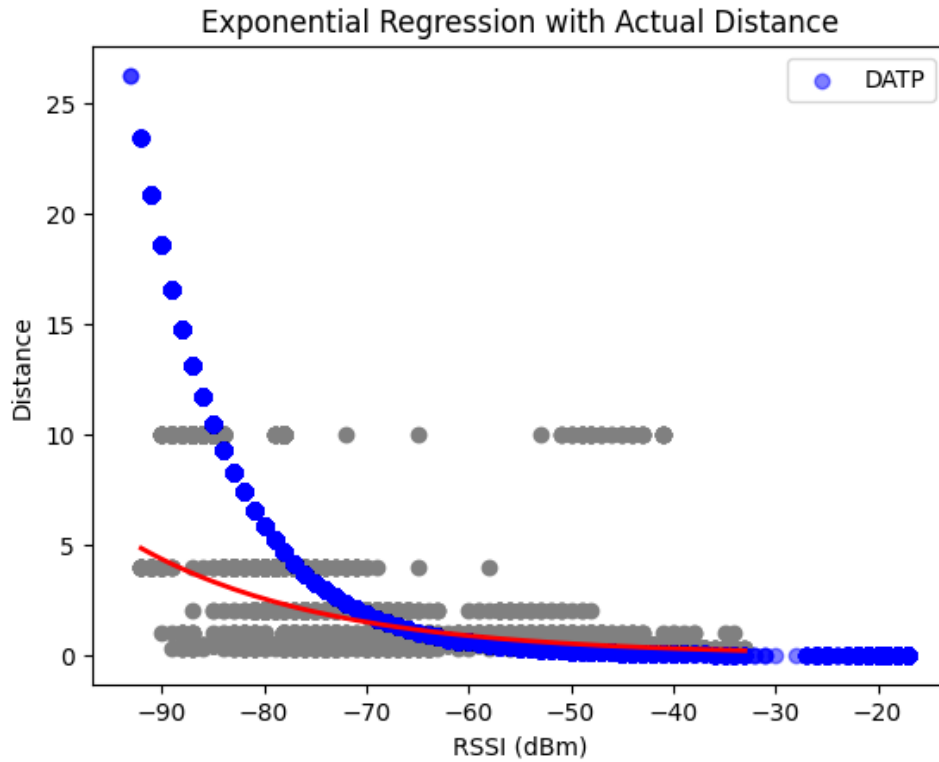


Figure 5.7: Line of Exponential Regression for Lugano Dataset

As the equation and the plot in Figure 5.7 show, the relationship between RSSI values and the distance between two devices is exponential. The R-squared of the model is 0.2429, and the Mean Squared Error is 8.5067. The model performs worse on this dataset than on the dataset from Zürich.

5.3.3 Classification Analysis

The models performed as follows:

Table 5.4: Model Performance Comparison for Lugano Dataset

Model	Training R-squared	Cross-validated R-squared	Diff	Test Accuracy
Random Forest	0.5715	0.5733	-0.0018	0.4289
Decision Tree	0.5717	0.5734	-0.0017	0.4218
Naive Bayes	-0.0752	-0.0709	-0.0043	0.5444
MLP	0.4672	0.4890	-0.0218	0.4861

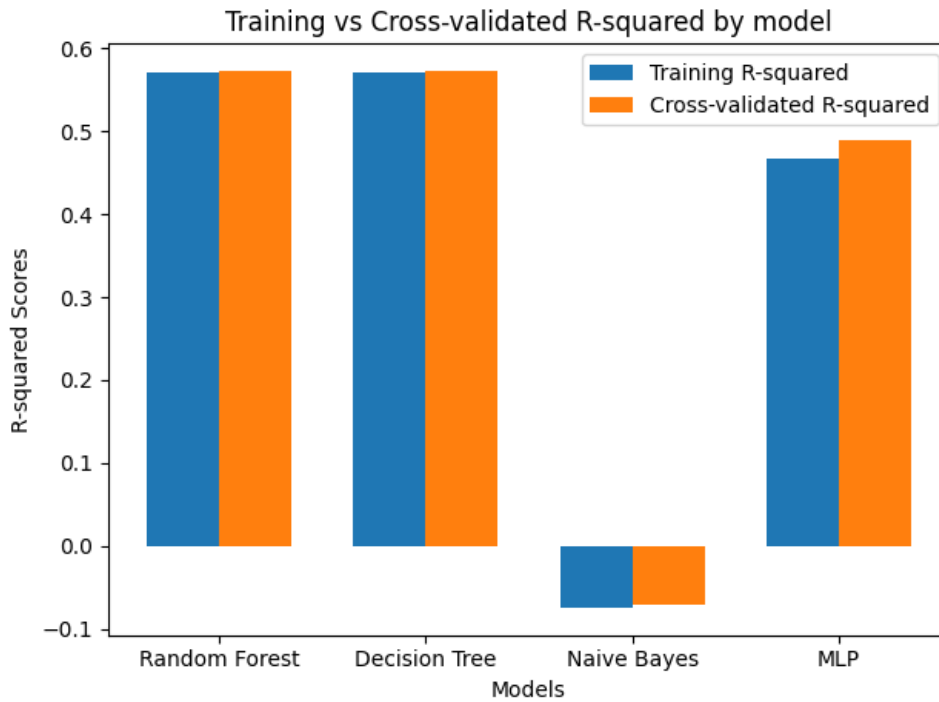


Figure 5.8: Comparison of Training R-squared and Cross-validated R-squared by model

The best model in terms of accuracy is the Naive Bayes Classifier model. However, Figure 5.8 and Table 5.8 illustrate that this model has negative R-squared values, which could be a sign of overfitting. Therefore, the preferred model is the Random Forest, as it has a medium-high accuracy and does not overfit (the difference between the Training R-squared and the Cross-validated square is close to 0). Although the MLP achieves higher accuracy compared to the Random Forest, the significantly larger difference in R-squared values suggests potential overfitting in the MLP model.

5.4 Analysis and Comparisons of the two Datasets

For this analysis, the two datasets have been combined. This section offers a detailed examination of the combined dataset, following a structure akin to the individual analyses conducted for the Lugano and Zürich datasets.

5.4.1 Data Preprocessing

The data preprocessing step has been done in a similar way to the other two datasets. The subset of the data contained 51294 observations for each class (according to the class with the least number of observations at the beginning).

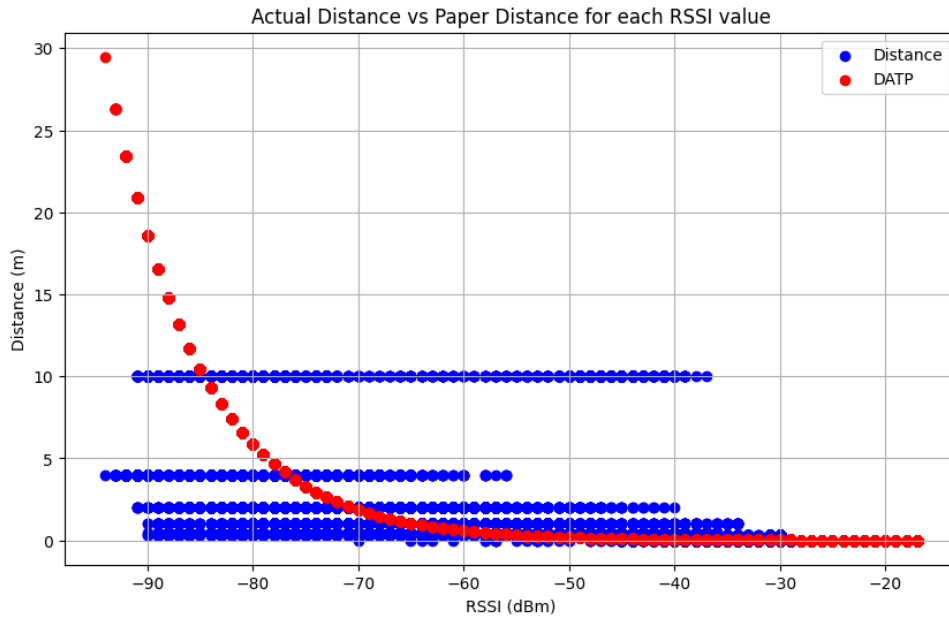


Figure 5.9: Relationship between Distance and DATP Variables

Figure 5.9 shows a similar pattern to Figure 5.2 and Figure 5.6.

5.4.2 Regression Analysis

Similar to the analyses conducted for the individual datasets, this section explores the combined dataset using linear and exponential regression models to assess the accuracy of predicting distance from RSSI values.

Linear Regression

As Figures 5.3, 5.4 and 5.7 show, the data follows an exponential distribution. Therefore, exponential regression is employed directly.

Exponential Regression

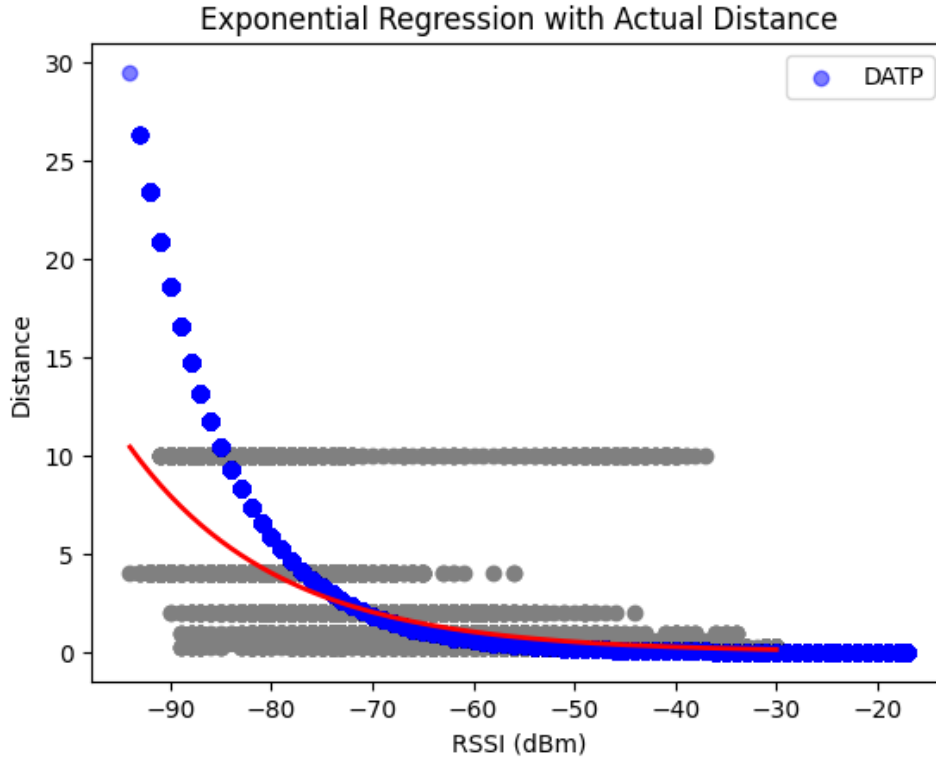


Figure 5.10: Line of Exponential Regression for the Combined Dataset

As the equation and Figure 5.10 show, the relationship between RSSI values and the distance between two devices is exponential. The R-squared of the model is 0.5394, and the Mean Squared Error is 5.2458. The model performs better on this dataset than on the other two datasets because the information from data collected in two different environments (specified in Experiments) enhances the model's ability to generalize and make accurate predictions.

5.4.3 Classification Analysis

The models performed as follows:

Table 5.5: Model Performance Comparison for Combined Dataset

Model	Training R-squared	Cross-validated R-squared	Diff	Test Accuracy
Random Forest	0.7219	0.7210	0.0009	0.6432
Decision Tree	0.7219	0.7211	0.0008	0.6432
Naive Bayes	0.5644	0.5657	-0.0013	0.7030
MLP	0.7077	0.7086	-0.0009	0.6322

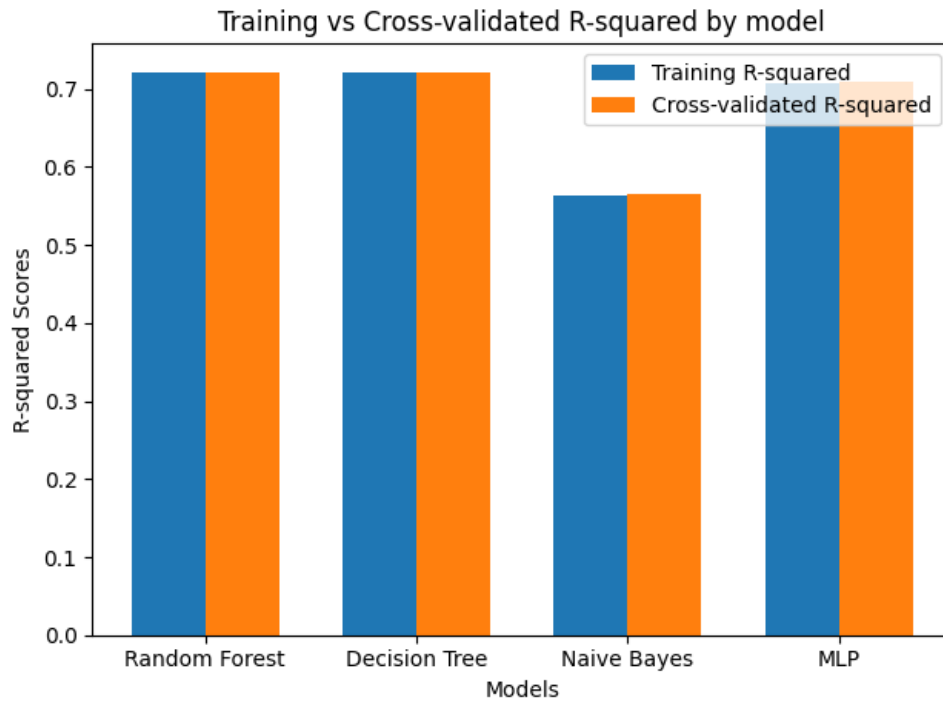


Figure 5.11: Comparison of Training R-squared and Cross-validated R-squared by model

The best model is the Naive Bayes Classifier, it has the lowest difference between training R-squared score and cross-validated R-squared score and the highest accuracy among all the models (0.7030). It has the highest accuracy out of three Naive Bayes Classifiers built for the three analyses. Figure 5.11 and Table 5.5 illustrate that none of the models exhibit overfitting, as evidenced by the close similarity between the Training R-squared and Cross-validated R-squared values across all models

5.5 Analysis of Environmental Experiments

This section has the goal of studying the data collected for the various environment experiments (specified in [Experiments](#)) to better understand the behavior of RSSI values.

- **1 AirTag vs 10 AirTags experiment:**

- RSSI value for 1 AirTag: -61.03 dBm
- Mean RSSI value for 10 AirTags at the same distance: -56.25 dBm

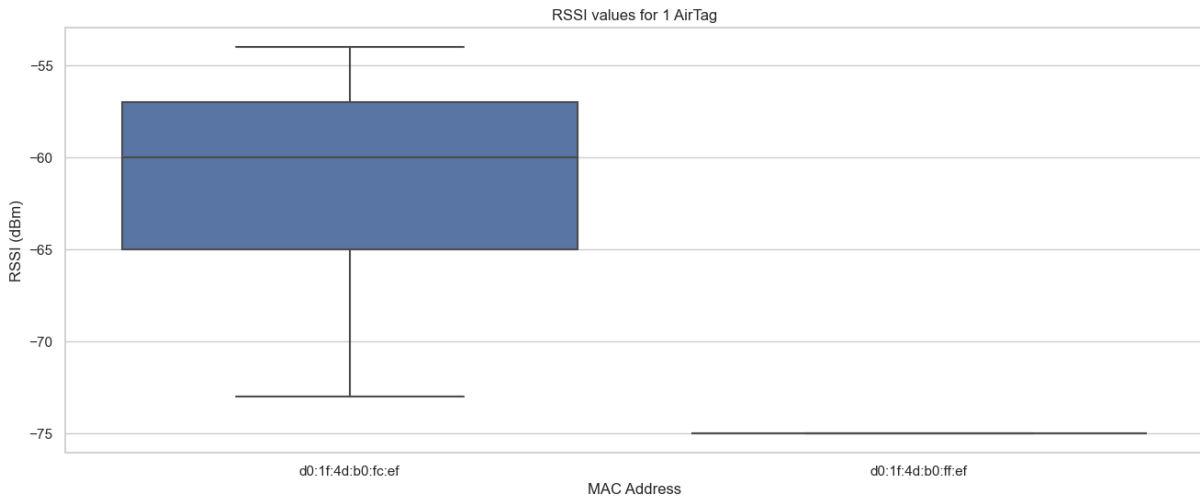


Figure 5.12: RSSI values measured with 1 AirTag

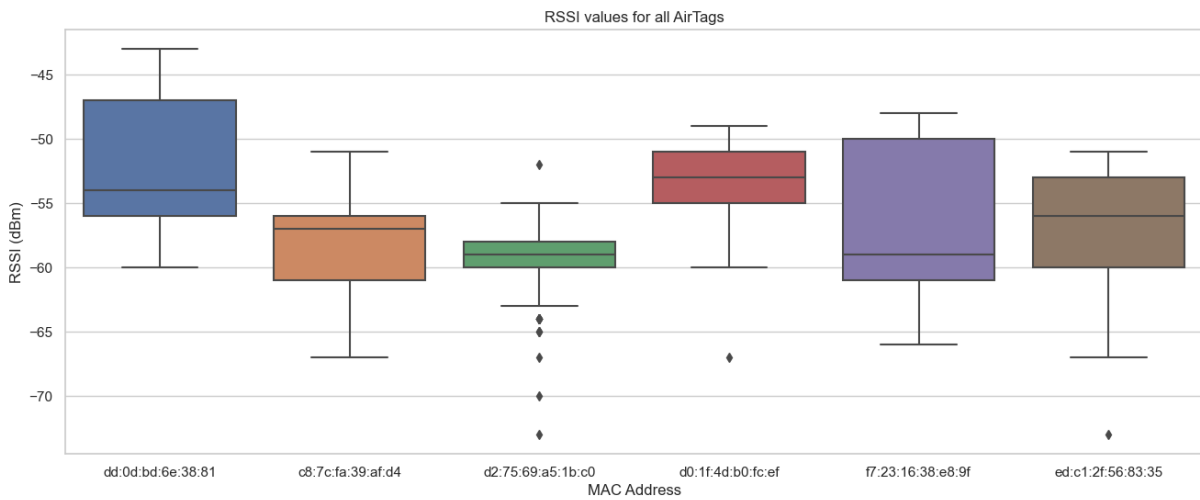


Figure 5.13: Comparison of RSSI values measured with 10 AirTags

From the data and Figures 5.12 and 5.13, it can be deduced that an increase in the number of AirTags results in a higher average RSSI. This is due to the overlapping signals reinforcing each other.

- **No person vs person walking between transmitter and receiver:**
 - Mean RSSI value when no person is walking in between: -56.25 dBm
 - Mean RSSI value when a person is walking in between: -59.63 dBm

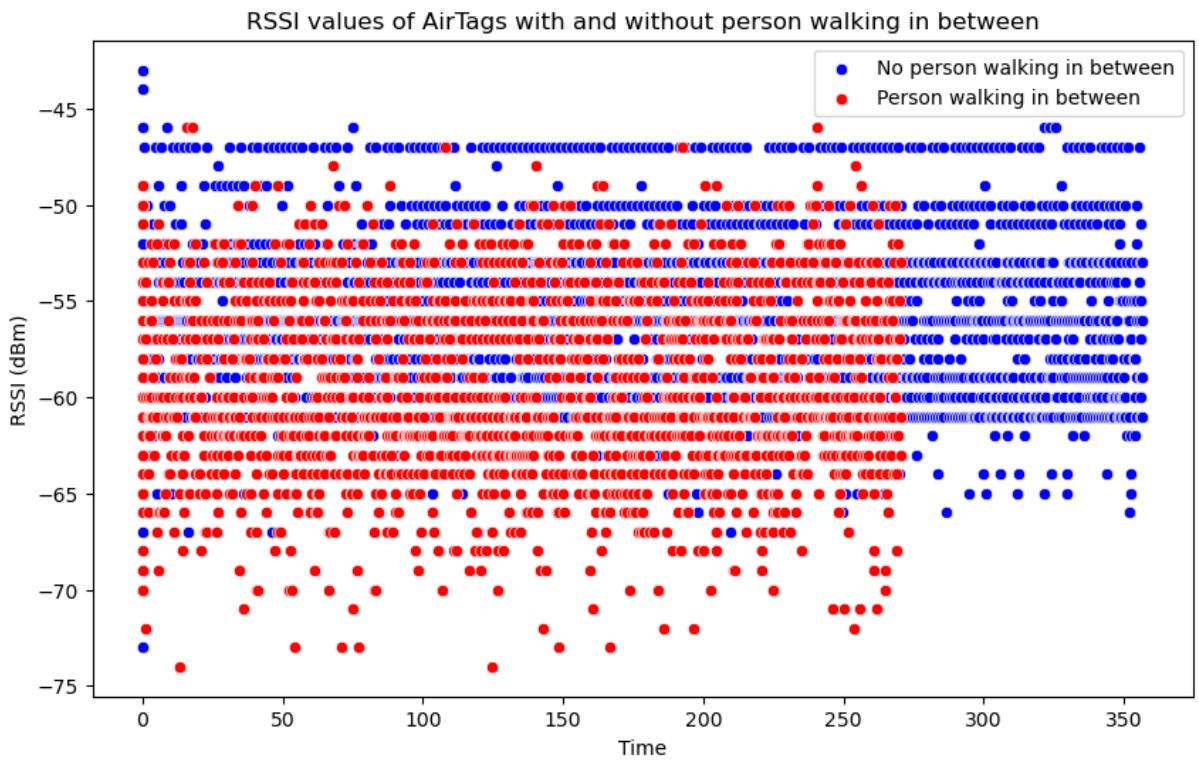


Figure 5.14: Comparison of RSSI values measured with and without a person walking in between

Based on the data and Figure 5.14, it is evident that the average RSSI is lower when a person is walking between the transmitter and receiver. This reduction in signal strength occurs because human bodies, which are largely composed of water, effectively absorb RF signals.

- **AirTags inside closet vs outside closet (same distance):**
 - Mean RSSI value for AirTags outside closet: -50.8 dBm
 - Mean RSSI value for AirTags inside closet: -57.09 dBm

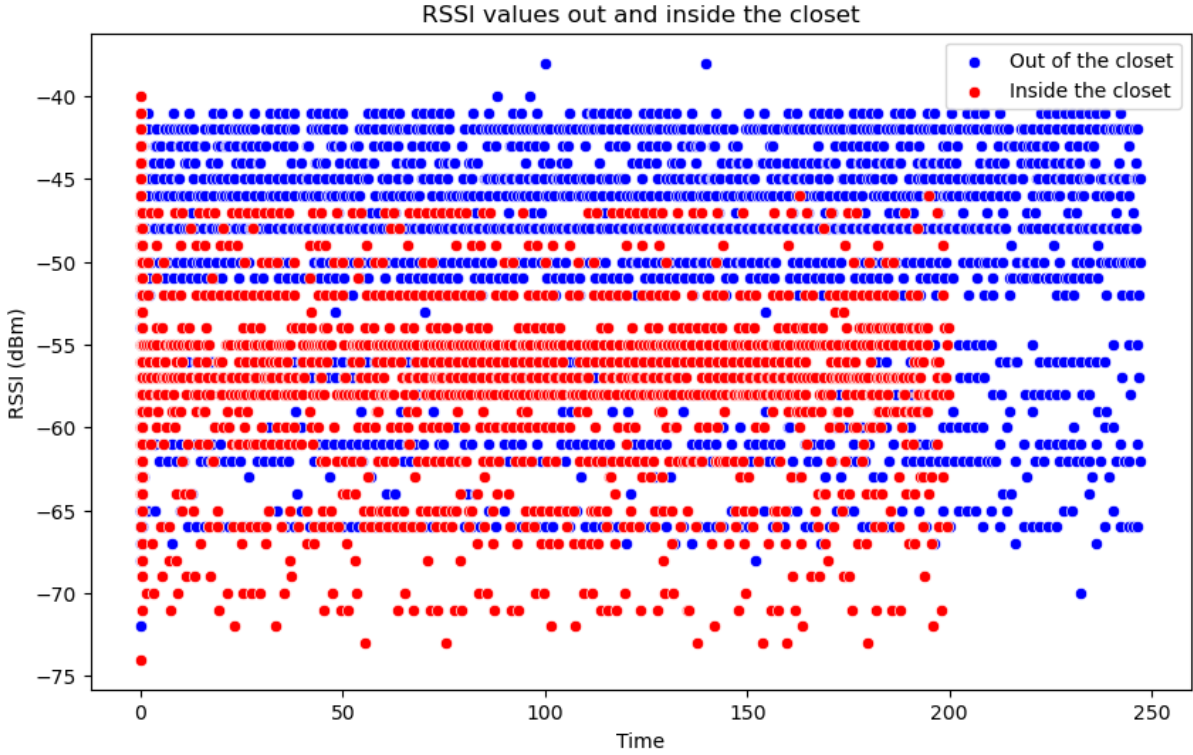


Figure 5.15: Comparison of RSSI values measured inside and outside of a closet

Figure 5.15 demonstrates that the average RSSI value is lower when AirTags are placed inside a closet. This is because materials and physical barriers attenuate RF signals, resulting in a weaker signal.

5.6 Limitations

While [54] determined that the optimal distance parameter for the HomeScout tracking algorithm is 200 meters, the measurements obtained in this study differ significantly. The experiments from [54] aimed to identify parameter values that would most reliably detect signals and thereby optimize the algorithm's performance. The definition of distance varies significantly between the two contexts: HomeScout's distance parameter is determined by a person walking along a route and detecting signals from nearby devices, where the distance from the tracking devices to the scanning device is unknown. This distance parameter is selected based on maximizing signal capture and optimizing algorithmic performance. In contrast, this thesis directly measures the distance between AirTags and the scanning device up to 10 meters. To address this difference in distance applicability, an option would be to conduct HomeScout experiments again, placing AirTags at known distances. This approach would provide a clear understanding of how distance, mapped with occurrences and time parameters, relates to detection reliability. Conducting controlled experiments with AirTags labeled according to their distance would provide comprehensive insights into these relationships. Moreover, labeling AirTags by their state can provide valuable insights. An AirTag in a connected state suggests that the device it

is paired with is within 30 meters of the AirTag [41]. This information is crucial for users who may want to filter out AirTags in a connected state if they suspect potential tracking from a greater distance. Conversely, if a user wants to focus on detecting nearby threats, they may include AirTags in a connected state while excluding those in a lost state. However, in crowded environments, this approach presents challenges as many AirTags may be in a connected state, making it difficult to distinguish between benign and potentially malicious tracking scenarios.

Additionally, a limitation of the current study is that it does not account for vertical distance. For example, if a stalker places an AirTag at the top of a backpack of a person while his/her phone is in a pocket, the vertical distance (approximately 0.5 meters) needs to be measured to understand if and how the behavior of RSSI values changes. The orientation of antennas should be taken into account as it plays a critical role in RSSI measurements, as highlighted by [55]. This study found that RSSI values vary depending on the angle of the receiver antenna relative to the transmitter antenna. Specifically, the strongest RSSI values were measured when the receiver antenna was vertically oriented towards the transmitter antenna, whereas the weakest values occurred when the antenna was vertically oriented away from the transmitter. The current experiments only tested horizontal distances, leaving a gap in understanding the full three-dimensional spatial relationships in tracking scenarios.

It is essential to validate Equation 2.2 in environmental experiments using different attenuation constants, depending on the material through which the signal passes. This is helpful to assess whether the formula from [26] accurately matches the actual distance between two devices when using a signal attenuation constant that is not free space.

Moreover, it's notable that the difference between the actual distance and the distance predicted by [26] is higher for the Lugano dataset than that observed for the Zürich dataset. Therefore, further data collection in controlled settings is essential to determine whether this discrepancy stems from the differences in nature from the formula derived by [26] or simply due to the few collected data.

While RSSI values were measured between AirTags and the receiving device in both controlled (Lugano) and semi-controlled (Zürich) environments, the latter setting doesn't entirely reflect highly crowded environments such as airports or train stations, where numerous Bluetooth devices and signals are present simultaneously. Although the data collected in Zürich provides insights into real-world scenarios to some extent, it may not fully encapsulate the complexities of such crowded environments. Future data collection efforts should prioritize methods that better replicate these conditions to enhance the model's real-world accuracy.

The values in the SETTINGSFRAGMENT for the NearField and Persistent filter in the HomeScout application are currently hardcoded and not chosen based on the collected data. This is due to the difference in contexts for the distance parameter, which would require data labeled according to the distance to be collected in a setting similar to HomeScout's. This data would provide insights into how the occurrences and time parameters correlate with labeled distances, allowing consequently for the integration of measured RSSI values for better shielding. Therefore, the current values may not be very effective when used for the prototype under real-world conditions.

The categories chosen for the classification task are not highly flexible and are tailored to the specific type of data collected. For instance, the models might classify a distance of 20 meters and a distance of 50 meters into the same category, such as "10+ meters." In practical scenarios, however, more precise distance categories would be desirable. Future research should explore more adaptable classification methodologies to enhance the model's accuracy and applicability in real-world settings.

Based on the analysis conducted in this chapter, it is evident that RSSI values exhibit significant variability based on environmental conditions and experimental setups. The datasets from Zürich and Lugano provided valuable insights into how RSSI values correlate with distance, showcasing the challenges posed by factors such as signal interference and environmental obstacles. Regression and classification analyses underscored the utility of predictive models in estimating distances and categorizing them into meaningful bins. Moreover, experiments examining different scenarios, including multiple AirTags and human interference, highlighted the dynamic nature of RSSI measurements. These findings underscore the importance of context-specific calibration and careful consideration of environmental variables when utilizing RSSI for proximity detection and distance estimation applications.

Chapter 6

Conclusions and Future Work

This chapter concludes this work by providing a summary of the key discoveries and contributions. Furthermore, it offers guidance for researchers interested in delving deeper into this field.

6.1 Conclusions

The primary objective of this thesis was to collect a comprehensive dataset to analyze the behavior of RSSI values emitted by tracking devices such as AirTags. Subsequently, the study aimed to develop an RSSI shielding prototype for the existing HomeScout application to prevent notifications about potential stalking attacks from owned or irrelevant trackers.

The first research question explored the feasibility of predicting the distance between a Non-Tracker Device (e.g., iPhone) and a Tracker Device (e.g., AirTag) using RSSI values. Equations 2.1 and 2.2 confirm that this prediction is possible, although the precision varies based on the dataset. Overall, the Zürich dataset produced more promising results compared to the Lugano dataset, indicating that the accuracy of distance estimation is influenced by the quality and characteristics of the collected data.

The second research question aimed to understand how RSSI values could be utilized to distinguish between owned and unowned devices. Based on the research in this thesis, it became apparent that a labeled dataset is required for more precise predictions. This dataset should include the state of the AirTag, such as whether it is connected, unpaired, nearby, or lost. Further data collection and analysis of RSSI values in combination with AirTag states are necessary to expand on this research question.

The third research question investigated how environmental conditions affect the accuracy and reliability of RSSI values for distance estimation. The data showed that controlled environments with minimal interference (i.e., the Lugano dataset) tend to produce higher average RSSI values per distance measurement. Additionally, RSSI values decrease when obstacles, such as people or objects, obstruct the signal. This is because human bodies,

composed largely of water, effectively absorb RF signals, weakening the RSSI values. To quantify the influence of these environmental factors on RSSI accuracy, further data collection is required. This data should be labeled according to distance, and the predicted distances should be compared to actual distances, accounting for the varying signal attenuation constants of different obstacles. Additionally, the collected data with different attenuation constants can be compared to the data with free space (air) as the attenuation constant to assess the extent of the difference in RSSI values when the signal travels through different materials.

In conclusion, while the study demonstrated the potential for using RSSI values in distance estimation and highlighted the impact of environmental conditions, further research with more comprehensive and labeled datasets is essential to enhance the reliability of the current shielding prototype.

Additionally, it should be noted that the decision was made not to delve deeply into time and frequency dispersion, as it was determined that knowing the specific type of distortion was beyond the scope of this work. Nevertheless, this thesis identified whether any distortion or interference on the signals occurred in the first place.

6.2 Future Work

The research conducted has provided valuable insights, particularly in understanding the feasibility of using RSSI values for distance estimation with tracking devices like AirTags. Key takeaways include the variability of RSSI values based on environmental conditions, including situations where human bodies or objects can introduce signal distortion. Another crucial takeaway is the need for a labeled dataset to effectively distinguish between owned and unowned devices in different environmental contexts. This refinement is essential for accurately implementing the RSSI shielding prototype within the HomeScout application.

To advance this research, the validation of Equation 2.2 for distance estimation demands further investigation through additional experiments. It is essential to verify the formula's accuracy across diverse environmental conditions and materials, considering different attenuation constants. It should also be explored whether the differences between the formula's predicted distance and the actual distances decrease in controlled settings.

Secondly, future experiments should incorporate the consideration of vertical distance in conjunction with horizontal distance. It should be studied whether vertical distance can significantly influence RSSI values, and if this is the case, in which way RSSI values are influenced by it. By accounting for both horizontal and vertical dimensions, the tracking accuracy can be improved, leading to more precise detection of malicious trackers.

Additionally, it is essential to accommodate more flexible distance categorization. Current classification approaches are specific to the collected data, leading to less precise estimation, especially with increasing distances.

Furthermore, conducting experiments with AirTags labeled according to their distance and state in a setting similar to that of HomeScout is essential. This allows the es-

establishment of a mapping between labeled distances and HomeScout's distance, thereby allowing the incorporation of RSSI as a shield into the tracking algorithm. By integrating RSSI shielding while also considering the occurrences and time parameters, the HomeScout functionality is maintained and the algorithm's reliability in real-world scenarios is enhanced.

Lastly, collecting a larger dataset is crucial to enhance the ability to make the analysis more robust. It is ideal to include measurements across a wider range of distances, extending up to the maximum distance where AirTags can effectively be used for stalking purposes. This approach ensures comprehensive coverage of all feasible distances within the operational range of AirTags.

By addressing these aspects, future research can build on the findings of this thesis to develop more effective solutions for tracking device detection and shielding, laying the groundwork for enhanced privacy and security measures.

Bibliography

- [1] Bluetooth SIG, “2023 bluetooth market update,” https://img.anfulai.cn/bbs/118741/2023%20Market%20Update%20_%20Bluetooth%20Technology%20Website.pdf, 2023, visited: 20.02.2024.
- [2] Wikipedia, “Bluetooth low energy beacon,” https://en.wikipedia.org/wiki/Bluetooth_Low_Energy_beacon, visited: 21.02.2024.
- [3] Q. Dong and W. Dargie, “Evaluation of the reliability of rssi for indoor localization,” *2012 International Conference on Wireless Communications in Underground and Confined Areas*, pp. 1–6, 2012.
- [4] A. Mussina and S. Aubakirov, “Rssi based bluetooth low energy indoor positioning,” *2018 International Conference on Advances in ICT for Emerging Regions (ICTer)*, October 2018.
- [5] Y. Wang, X. Yang, Y. Zhao, and Y. Liu, “Bluetooth positioning using rssi and triangulation methods,” *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*. Las Vegas, NV, USA, IEEE, Jan. 2013.
- [6] J. Du, C. Yuan, M. Yue, and T. Ma, “A novel localization algorithm based on rssi and multilateration for indoor environments,” *Sensors*, Vol. 17, No. 9, 2017.
- [7] M. N. Amr, H. M. ELAttar, M. H. A. E. Azeem, and H. E. Badawy, “An enhanced indoor positioning technique based on a novel received signal strength indicator distance prediction and correction model,” *Sensors*, Vol. 21, No. 3, p. 719, 2021.
- [8] Apple, “Find your lost apple device or airtag with find my,” <https://support.apple.com/en-us/104978>, visited: 21.02.2024.
- [9] Britannica, “How does bluetooth work,” <https://www.britannica.com/story/how-does-bluetooth-work>, visited: 21.02.2024.
- [10] Bluetooth, “What determines bluetooth range?” <https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/#:~:text=Bluetooth%C2%AE%20technology%20uses%20the,balance%20between%20range%20and%20throughput>, visited: 21.02.2024.
- [11] Rajiv, “What are radio frequency bands and its uses?” <https://www.rfpage.com/what-are-radio-frequency-bands-and-its-uses/>, Nov 2023, visited: 07.04.2024.

- [12] Wikipedia, “Radio frequency — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/wiki/Radio_frequency, 2024, visited: 07.04.2024.
- [13] Saylor Academy, “More wireless basics: Power and receiver sensitivity,” <https://learn.saylor.org/mod/book/view.php?id=29826&chapterid=5500>, 2024, visited: 07.04.2024.
- [14] “Gain (antenna),” [https://en.wikipedia.org/wiki/Gain_\(antenna\)](https://en.wikipedia.org/wiki/Gain_(antenna)), visited: 07.04.2024.
- [15] P. T. Z. Tun, “Path loss prediction by using rssi values,” 2018.
- [16] R. Heydon, *Bluetooth Low Energy: The Developer 's Handbook*. Pearson Always Learning. Prentice Hall, 2012.
- [17] S. Zeadally, F. Siddiqui, and Z. Baig, “25 years of bluetooth technology,” *Future Internet*, Vol. 11, p. 194, 2019.
- [18] J. Tosi, F. Taffoni, M. Santacatterina, R. Sannino, and D. Formica, “Performance evaluation of bluetooth low energy: A systematic review,” *Sensors*, Vol. 17, No. 12, 2017.
- [19] RF Wireless World, “Ble advertising packet format | ble data packet format,” <https://www.rfwireless-world.com/Terminology/BLE-Advertising-and-Data-Packet-Format.html>, 2024, visited: 27.02.2024.
- [20] Texas Instruments, *Generic Access Profile (GAP)*, https://software-dl.ti.com/lprf/simplelink_cc2640r2_sdk/1.35.00.33/exports/docs/ble5stack/ble_user_guide/html/ble-stack/gap.html, visited: 16.03.2024.
- [21] Microchip Technology Inc. (2023) Generic attribute profile (gatt) overview. <https://developerhelp.microchip.com/xwiki/bin/view/applications/ble/introduction/bluetooth-architecture/bluetooth-host-layer/bluetooth-generic-attribute-profile-gatt/Overview/>. Visited: 16.03.2024.
- [22] J. Wong. (2019, June) Security manager (sm) in bluetooth low energy. <https://jimmywongiot.com/2019/06/12/security-manager-sm-in-bluetooth-low-energy/>. Visited: 05.04.2024.
- [23] Nordic Semiconductor, “Services and characteristics,” <https://academy.nordicsemi.com/courses/bluetooth-low-energy-fundamentals/lessons/lesson-4-bluetooth-le-data-exchange/topic/services-and-characteristics/>, 2023, visited: 05.04.2024.
- [24] Texas Instruments, “Logical link control and adaptation layer protocol (l2cap) - ble-stack user’s guide for bluetooth 4.2,” <https://software-dl.ti.com/lprf/sdg-latest/html/ble-stack-3.x/l2cap.html>, 2016, visited: 05.04.2024.
- [25] Telecom Trainer, “RSSI (Receive Signal Strength Indicator),” <https://www.telecomtrainer.com/rssi-receive-signal-strength-indicator/>, visited: 21.02.2024.

- [26] J.-H. Huh and K. Seo, “An indoor location-based control system using bluetooth beacons for iot systems,” *Sensors*, Vol. 17, No. 12, p. 2917, 2017.
- [27] NetSpot, “What is rssi and its relation to a wi-fi network,” <https://www.netspotapp.com/wifi-signal-strength/what-is-rssi-level.html>, visited: 21.02.2024.
- [28] A. Heinrich, M. Stute, T. Kornhuber, and M. Hollick, “Who can find my devices? security and privacy of apple’s crowd-sourced bluetooth location tracking system,” *Proceedings on Privacy Enhancing Technologies*, Vol. 2021, pp. 227–245, 2021.
- [29] M. Woolley, “Bluetooth core specification version 5.2 feature overview,” https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf, Bluetooth SIG, Tech. Rep., 2020, visited: 05.04.2024.
- [30] Apple Inc., *Accessory Design Guidelines for Apple Devices*, <https://developer.apple.com/accessories/Accessory-Design-Guidelines.pdf>, 2023, visited: 05.04.2024.
- [31] A. Heinrich, N. Bittner, and M. Hollick, “AirGuard - Protecting Android Users From Stalking Attacks By Apple Find My Devices,” 2022.
- [32] J. K. Becker, D. Li, and D. Starobinski, “Tracking anonymized bluetooth devices,” *Proceedings on Privacy Enhancing Technologies*, Vol. 2019, No. 3, pp. 50–65, 2019.
- [33] Apple Inc., “Airtag - technical specifications,” https://support.apple.com/kb/SP840?locale=en_US, 2022, visited: 26.02.2024.
- [34] —, “Airtag,” <https://www.apple.com/airtag/>, 2024, visited: 26.02.2024.
- [35] Tile, Inc., “How it works,” <https://ch.tile.com/en/how-it-works>, 2024, visited: 26.02.2024.
- [36] —, “Tile - find lost keys phone,” <https://ch.tile.com/en>, 2024, visited: 26.02.2024.
- [37] Chipolo d.o.o., “Finde deine schlüssel, portemonnaie und telefon - chipolo,” <https://chipolo.net/de/>, 2024, visited: 26.02.2024.
- [38] —, “Chipolo one spot - the key finder that works with the apple find my app,” <https://chipolo.net/en/products/chipolo-one-spot>, 2024, visited: 26.02.2024.
- [39] Samsung Electronics Co., Ltd., “Galaxy smarttag black | bluetooth tracker,” <https://www.samsung.com/ch/mobile-accessories/galaxy-smarttag-black-ei-t5300bbegeu/>, 2024, visited: 26.02.2024.
- [40] V. Honkavirta, T. Perälä, S. Ali-Löytty, and R. Piché, “A comparative survey of wlan location fingerprinting methods,” *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*. Tampere, Finland, IEEE, Apr. 2009, tampere University of Technology, Finland.
- [41] L. Bienz, “Homescout: A modular bluetooth low energy sensing android app.” Zurich, Communication Systems Group (University of Zurich), 2023.
- [42] E. Essa, B. Abdullah, and A. Wahba, “Improve performance of indoor positioning system using ble,” 12 2019, pp. 234–237.

- [43] A. A. Kalbandhe and S. C. Patil, "Indoor positioning system using bluetooth low energy," *2016 International Conference on Computing, Analytics and Security Trends (CAST)*. Pune, Maharashtra, India, Savitribai Phule Pune University, Dec 2016, pp. 19–21, rajarshi Shahu College of Engineering, Tathawade, College of Engineering Pune, India.
- [44] I. Alexander and G. P. Kusuma, "Predicting indoor position using bluetooth low energy and machine learning," *International Journal of Scientific Technology Research*, Vol. 8, No. 09, p. 1661, 2019.
- [45] K. Konstantinos and T. Orphanoudakis, "Bluetooth beacon based accurate indoor positioning using machine learning," School of Science and Technology, Hellenic Open University, Patras, Greece, 2019.
- [46] A. A. S. AlQahtani and N. Choudhury, "Machine learning for location prediction using rssi on wi-fi 2.4 ghz frequency band," *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada, 2021, pp. 336–342.
- [47] S. Üstebay, Z. Turgut, Ş. D. Odabaşı, M. A. Aydın, and A. Sertbaş, "A machine learning approach based on indoor target positioning by using sensor data fusion and improved cosine similarity," *Electrica*, 2023, early View Article, published November 27, 2023.
- [48] N. D. R. Rose, L. T. Jung, and M. Ahmad, "3d trilateration localization using rssi in indoor environment," *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 11, No. 2, 2020.
- [49] A. De Blas and D. López-de Ipiña, "Improving trilateration for indoors localization using ble beacons," *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, 2017, pp. 1–6.
- [50] R. Bembenik and K. Falcman, "Ble indoor positioning system using rssi-based trilateration," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, Vol. 11, pp. 50–69, 2020.
- [51] C. S. Mouhammad, A. Allam, M. Abdel-Raouf, E. Shenouda, and M. Elsabrouty, "Ble indoor localization based on improved rssi and trilateration," *2019 7th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC)*, Alexandria, Egypt, 2019, pp. 17–21.
- [52] J. Briggs and C. Geeng, "Ble-doubt: Smartphone-based detection of malicious bluetooth trackers," *2022 IEEE Security and Privacy Workshops (SPW)*, 2022, pp. 208–214.
- [53] A. Heinrich, M. Stute, and M. Hollick, "Openhaystack: a framework for tracking personal bluetooth devices via apple's massive find my network," *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '21. New York, NY, USA, Association for Computing Machinery, 2021, pp. 374–376.

- [54] K. O. E. Müller, L. Bienz, B. Rodrigues, C. Feng, and B. Stiller, “HomeScout: Anti-Stalking Mobile App for Bluetooth Low Energy Devices,” Zürich, Switzerland, 2023.
- [55] M. Lin, B. Chen, W. Zhang, and J. Yang, “Characteristic analysis of wireless local area network’s received signal strength indication in indoor positioning,” *IET Communications*, Vol. 14, No. 3, pp. 497–504, 2020, received on 2nd July 2019, Revised 14th October 2019, Accepted on 5th November 2019, E-First on 22nd January 2020.

Abbreviations

AP	Access Point
ATT	Attribute Protocol
BLE	Bluetooth Low Energy
CRC	Cyclic Redundancy Check
dBi	Decibel Isotropic
dBm	Decibel-milliwatts
FM	Frequency Modulation
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GHz	Gigahertz
GPS	Global Positioning System
HCI	Host Controller Interface
Hz	Hertz
ISM	Industrial, Scientific, and Medical
LF	Low Frequency
LL	Link Layer
L2CAP	Logical Link and Adaptation Protocol
MF	Medium Frequency
MHz	Megahertz
ML	Machine Learning
Mw	Milliwatt
OF	Offline Finding
OFN	Offline Finding Network
PDU	Protocol Data Unit
PHY	Physical Layer
RF	Radio Frequency
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
SM	Security Manager
TxPower	Transmitted Power
UHF	Ultra High Frequency
UUID	Universal Unique Identifier
UWB	Ultra-wideband
VHF	Very High Frequency
VLF	Very Low Frequency

List of Figures

2.1	Range of Radio Spectrum [11]	4
2.2	Architecture of BLE [17]	6
2.3	Structure of BLE packets [19]	7
2.4	GAP State Diagram [20]	8
2.5	GATT Example [21]	9
2.6	Security Manager Pairing Phases [22]	9
2.7	ATT attribute data structure [23]	10
2.8	L2CAP Building Blocks [24]	11
2.9	Advertising and scanning in broadcasting [18]	12
2.10	BLE connection with its parameters [18]	14
2.11	Offline Finding Advertising Format [28]	16
2.12	Binary representation of a location report [28]	17
2.13	Apple's Find My Network [31]	17
2.14	Advertising Features of Different OSs [32]	18
2.15	Estimation of location using triangulation. [5]	20
2.16	Settings of the tracking algorithm in the HomeScout application. [41]	21
5.1	Comparison of RSSI values between datasets	32
5.2	Relationship between Distance and DATP Variables	34
5.3	Diagnostic Plots of Linear Model for the Zürich Dataset	35
5.4	Line of Exponential Regression for Zürich Dataset	37
5.5	Comparison of Training R-squared and Cross-validated R-squared by model	39

5.6 Relationship between Distance and DATP Variables	40
5.7 Line of Exponential Regression for Lugano Dataset	41
5.8 Comparison of Training R-squared and Cross-validated R-squared by model	42
5.9 Relationship between Distance and DATP Variables	43
5.10 Line of Exponential Regression for the Combined Dataset	44
5.11 Comparison of Training R-squared and Cross-validated R-squared by model	45
5.12 RSSI values measured with 1 AirTag	46
5.13 Comparison of RSSI values measured with 10 AirTags	46
5.14 Comparison of RSSI values measured with and without a person walking in between	47
5.15 Comparison of RSSI values measured inside and outside of a closet	48

List of Tables

2.1	Table of RSSI value levels [27]	15
5.1	RSSI Value Levels from the Zürich Dataset	31
5.2	RSSI Value Levels from the Lugano Dataset	32
5.3	Model Performance Comparison for Zürich Dataset	38
5.4	Model Performance Comparison for Lugano Dataset	41
5.5	Model Performance Comparison for Combined Dataset	45

Appendix A

Contents of the Repository

The code repository is divided into two main parts: the HomeScout application and the Data Analysis folder.

HomeScout Application

The HomeScout project is organized as follows:

- **app/src/main/kt/android/example/homescout/**
 - **database**
 - * **BLEDevice.kt** - Represents a Bluetooth Low Energy (BLE) device.
 - * **BLEDeviceDao.kt** - Data Access Object for BLE devices, providing methods to interact with the database.
 - * **HomeScoutDatabase.kt** - Database class for the HomeScout application.
 - * **MaliciousTracker.kt** - Represents a malicious BLE tracker.
 - * **MaliciousTrackerDao.kt** - Data Access Object for malicious trackers, providing methods to interact with the database.
 - **di**
 - * **AppModule.kt** - module for application-level dependencies.
 - * **ServiceModule.kt** - module for service-level dependencies.
 - **models**
 - * **AirTag.kt** - Model class representing an AirTag device.Additional model classes for other devices (Chipolo, Tile, ...).
 - **repositories**
 - * **MainRepository.kt** - Handles main data operations, including BLE data interactions and local database management.

- * **TrackingPreferencesRepository.kt** - Manages user preferences related to BLE tracking and stores configuration settings.
- **services**
 - * **BluetoothScanningService.kt** - Service responsible for scanning Bluetooth devices.
 - * **LocationTrackingService.kt** - Service responsible for tracking the location of the user.
 - * **TrackerClassificationService.kt** - Service responsible for classifying BLE trackers as malicious or benign.
- **ui**
 - * Contains activities and fragments for the user interface, managing user interactions and UI updates.
- **utils**
 - * **BluetoothAPILogger.kt** - Utility class for logging Bluetooth-related activities.
 - * **Constants.kt** - Class containing constant values used throughout the application.
 - * **RingBuffer.kt** - Implementation of a ring buffer data structure.
- **app/src/main/res/**
 - **layout** - XML layout files that define the visual structure of the screens in the app.
 - **values** - Resource files containing strings, colors, and other static values used in the app.
 - **drawable** - Resource files for images and graphics used in the app.

Data Analysis Folder

The Data Analysis folder contains scripts and tools used for analyzing the collected BLE data. The organization is as follows:

- **Jupyter Notebooks**

- **Analysis_0m.ipynb** - Analysis at 0 meters distance.
- **Analysis_0comma3m.ipynb** - Analysis at 0.3 meters distance.
- **Analysis_0comma5m.ipynb** - Analysis at 0.5 meters distance.
- **Analysis_1m.ipynb** - Analysis at a 1-meter distance.
- **Analysis_2m.ipynb** - Analysis at 2 meters distance.
- **Analysis_4m.ipynb** - Analysis at 4 meters distance.
- **Analysis_10m.ipynb** - Analysis at 10 meters distance.
- **Analysis_1vsAll.ipynb** - Analysis comparing RSSI values emitted by 1 AirTag and 10 AirTags.
- **Analysis_in_out_closet.ipynb** - Analysis comparing RSSI values of 10 AirTags inside and outside a closet.
- **Analysis_noperson_vs_person.ipynb** - Analysis comparing RSSI values with and without a person walking between the two devices.
- **Analysis_average_rssi_Zuerich.ipynb** - Analysis of average RSSI values for each distance measurement in Zürich.
- **Analysis_average_rssi_Lugano.ipynb** - Analysis of average RSSI values for each distance measurement in Lugano.
- **rssi_comparison.ipynb** - RSSI comparison analysis between the two datasets.
- **ml_analysis_Zuerich.ipynb** - Machine learning analysis for data collected in Zürich.
- **ml_analysis_Lugano.ipynb** - Machine learning analysis for data collected in Lugano.
- **ml_analysis_Combined.ipynb** - Machine learning analysis for combined Lugano and Zürich datasets.

- **Plots**

- **rssi_comparison_plot.png** - Plot comparing RSSI values across different distances for the two datasets.