Hybrid Detection Mechanism for Spoofing Attacks in Bluetooth Low Energy Networks

Hanlin Cai
20122161

Final Year Project – 2023/24
BSc Robotics & Intelligent Devices

Maynooth International Engineering College
Fuzhou University
Fujian, China

A thesis submitted in partial fulfilment of the requirements for the
BSc Robotics & Intelligent Devices

Supervisor: Prof. Zhezhuang Xu

Declaration

I hereby certify that this material, which I now submit for assessment as part of the Robotics & Intelligent Devices programme, is entirely my own work and has not been taken from the work of others - save and to the extent that such work has been cited and acknowledged within the body of my work.

I hereby acknowledge and accept that this thesis may be distributed to future final year students, as an example of the standard expected of final year projects.

Signed: Hanlin Cai **Date: 2024/04/28**

Hybrid Detection Mechanism for Spoofing Attacks in Bluetooth Low Energy Networks

Abstract

Bluetooth Low Energy (BLE) serves as a critical protocol for low-energy communication, playing a vital role in various sectors including industry, healthcare, and home automation. Despite its widespread adoption, inherent security limitations and firmware vulnerabilities expose BLE to significant risks, notably from spoofing attacks that threaten device integrity and data privacy. Addressing this challenge, this project introduces ***BLEGuard***, a hybrid detection mechanism specifically designed to identify spoofing attacks within BLE networks. BLEGuard integrates pre-detection scheme, reconstruction techniques, and classification models to effectively detect advanced spoofing threats. To refine and validate BLEGuard system, this project established a physical Bluetooth testbed to simulate attacks and generated a large-scale BLE Spoofing Attack Dataset (***BLE-SAD***). The experimental results demonstrate a high detection accuracy rate of 99.02%, with a false alarm rate of 2.04% and an un-detection rate of 0.37%. These findings highlight BLEGuard's effectiveness in enhancing the security of BLE networks, proving its potential as a robust solution to safeguard against sophisticated cyber threats in real-world applications.

Key words: Network Systems, Security and Privacy, Time-series Anomaly Detection, Machine Learning

CONTENT

Chapter 1 Introduction

1.1 Research Topic and Motivation

Named after the Viking King Harald Bluetooth, who was known for his role in unifying Danish tribes, Bluetooth technology has become a ubiquitous standard for short-range wireless communications. Since its inception, Bluetooth has revolutionized the way devices interact in close proximity[1]. The advent of the **Bluetooth Low Energy (BLE)** standard has further solidified its dominance, especially in the burgeoning era of the Internet of Things (IoT) and the emerging technologies of 6G communications[2]. BLE's low power requirements and high functionality make it an ideal choice for a multitude of IoT applications ranging from industrial automation to health monitoring, ensuring seamless connectivity between billions of devices. By 2027, the deployment of BLE devices is anticipated to burgeon to an astonishing 7.5 billion[3].

This exponential adoption, however, is overshadowed by significant security challenges within the BLE networks. BLE-enabled devices are prone to a diverse array of sophisticated attacks due to inherent I/O limitations and firmware vulnerabilities. These threats include zero-day exploits, where attackers exploit undisclosed vulnerabilities[4], DDoS (Distributed Denial of Service) attacks that cripple network services[5], and particularly spoofing attacks[6].

Spoofing attacks are alarmingly prevalent and concerning due to their low initiation costs and minimal hardware requirements, making them a preferred tactic among attackers. In these attacks, perpetrators impersonate legitimate devices, misleading network participants to intercept or manipulate sensitive data[7]. This undermines the integrity and confidentiality of BLE systems, facilitating unauthorized access and data breaches. The ease and low cost of initiating these attacks underscore the urgent need for the development of advanced detection mechanisms. These mechanisms must be capable of identifying and mitigating the sophisticated tactics used in spoofing attacks, thereby enhancing the security posture of BLE networks against these pervasive threats[8].

1.2 Problem Statement

To combat these security threats, an out-of-the-box monitoring system has been introduced, leveraging BLE's cyber-physical features to fortify defenses against spoofing attackers[9]. Additionally, various research initiatives employ machine learning techniques to detect anomalous patterns within BLE network traffic. A particularly promising learning framework that integrates reconstruction and classification models has been developed to identify network packets as either benign or malicious with remarkable precision[10].

Unfortunately, most existing methods grapple with the significant challenge of harmonizing detection accuracy, false positive rates, and resource utilization. This delicate balance severely restricts their applicability across a broader spectrum of real-world scenarios[6]. There is a pressing need for a more adaptable and efficient solution, which can uphold stringent detection standards while effectively managing resource constraints. Such an innovation would significantly broaden the utility of security frameworks, extending their deployment across a wider variety of environments and devices. This expansion is crucial for bolstering defenses against spoofing attacks in increasingly diverse and resource-constrained settings[11].

1.3 Approach and Metrics

Therefore, this project aims to introduce a novel detection mechanism that leverages cyber-physical analysis and machine learning techniques. Specifically engineered to detect sophisticated spoofing attacks, this mechanism combines extensive offline training with critical real-time online analysis. In pursuit of this goal, we will establish a tangible BLE network system for conducting attack simulations and compiling a large-scale network dataset. This broad and verifiable dataset is crucial for advancing research within the domain and ensuring the robustness of our findings. A series of experiments utilizing diverse datasets will be conducted to test the viability of the detection mechanism proposed. Subsequent to these tests, a meticulous assessment of the experimental results will be performed, and their profound implications for real-world applications will be analyzed.

1.4 Contributions of this Project

In this project, we propose *BLEGuard*, an innovative detection mechanism designed to enhance security in Bluetooth Low Energy networks. Additionally, we present *BLE-SAD*, an extensive network dataset generated from our specialized physical testbed. Our contributions are threefold:

- Development of the BLE-SAD dataset, which includes around 906,000 packets, tailored specifically for the training and evaluation of our models.
- Design and empirical validation of BLEGuard, which is proposed for effective detection of spoofing attacks.
- Integration capabilities of BLEGuard within BLE networks, designed to ensure effective detection without disrupting existing network operations or depleting network resources.

1.5 Structure of this FYP Report

The structure of this final year project report is organized for clarity and depth of understanding: **Chapter 2** provides an overview of the technical background and related works essential for understanding the context of this project. **Chapter 3** outlines the three principal research questions this project seeks to resolve, emphasizing their practical implications. **Chapter 4** explicates the creation and deployment of BLE-SAD dataset, alongside the formulation of BLEGuard mechanism. **Chapter 5** presents the experimental results, assessing the effectiveness of the BLEGuard system. Finally, **Chapter 6** concludes the project, summarizing key findings and contributions.

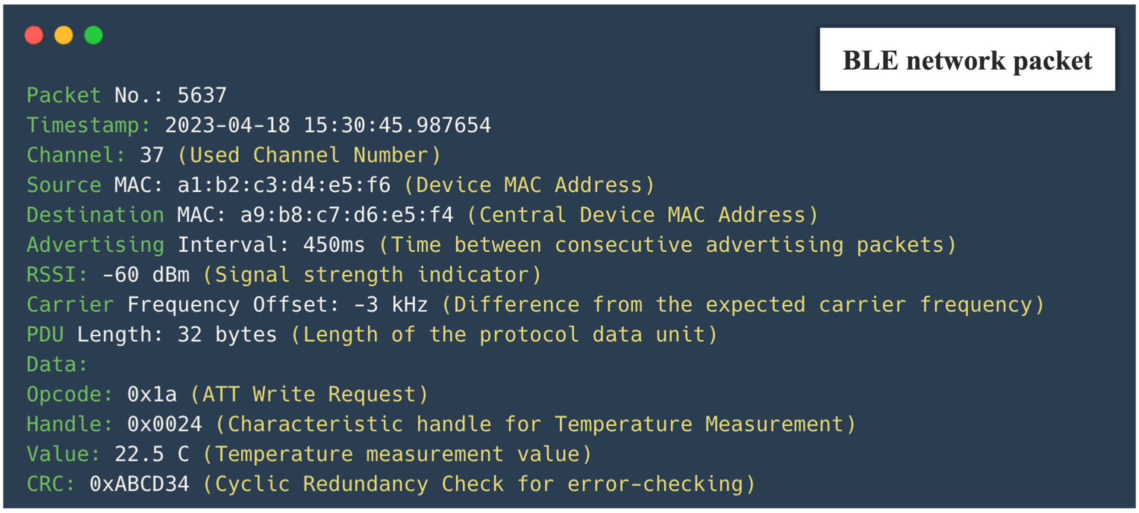
Chapter 2 Technical background

2.1 Topic material

2.1.1 Basics of Bluetooth Low Energy

Bluetooth Low Energy (BLE) is often the technology of choice for networks where energy-efficient and cost-effective communication is paramount. This is especially common with low-cost, energy-constrained devices like temperature sensors that capture specific data attributes and wirelessly transmit this information to user devices, like smartphones. BLE operates using three dedicated radio frequency (RF) channels (37, 38, and 39) for advertising, which is the process of broadcasting the presence of a BLE device to initiate a connection[1]. These are known as the advertising channels. Once a connection is established, the remaining channels, known as data channels, are used for the ongoing communication between devices.

The typical communication protocol in a BLE network encompasses four main stages: advertising, connecting, pairing, and data accessing[9]. The advertising stage is where the BLE device announces its availability to connect. In the connecting phase, a user device responds to this advertisement, establishing a bidirectional link. Pairing is the next crucial step, where security credentials are exchanged, forming the foundation for a secure communication. Finally, in the data accessing stage, the authenticated user device is able to read or write the data from or to the BLE device. **Figure 2-1** shows a typical network packet during the BLE communication, which includes data with time-series features such as packet number, timestamp, and payload data.



```
Packet No.: 5637
Timestamp: 2023-04-18 15:30:45.987654
Channel: 37 (Used Channel Number)
Source MAC: a1:b2:c3:d4:e5:f6 (Device MAC Address)
Destination MAC: a9:b8:c7:d6:e5:f4 (Central Device MAC Address)
Advertising Interval: 450ms (Time between consecutive advertising packets)
RSSI: -60 dBm (Signal strength indicator)
Carrier Frequency Offset: -3 kHz (Difference from the expected carrier frequency)
PDU Length: 32 bytes (Length of the protocol data unit)
Data:
Opcode: 0x1a (ATT Write Request)
Handle: 0x0024 (Characteristic handle for Temperature Measurement)
Value: 22.5 C (Temperature measurement value)
CRC: 0xABCD34 (Cyclic Redundancy Check for error-checking)
```

Figure 2-1 Sample data of a typical BLE network package.

2.1.2 Spoofing Attacks in BLE Networks

The spoofing attack is a type of cybersecurity attack wherein an attacker impersonates a legitimate BLE device or network entity[7]. In such attacks, the perpetrator typically masquerades as a trusted

BLE device using forged information, such as a spoofed MAC address or other identifying details, as illustrated in **Figure 2-2(a)**. In the context of a spoofing attack, the cyber-physical features of the BLE network are notably impacted, leading to significant deviations from typical benign scenarios. For instance, an anomalous shift in the RSSI (Received Signal Strength Indicator) values of the advertising packets can signal the presence of a spoofing attack, as depicted in **Figure 2-2(b)**. These deviations provide critical indicators that can be used to effectively identify potential malicious activity[12].

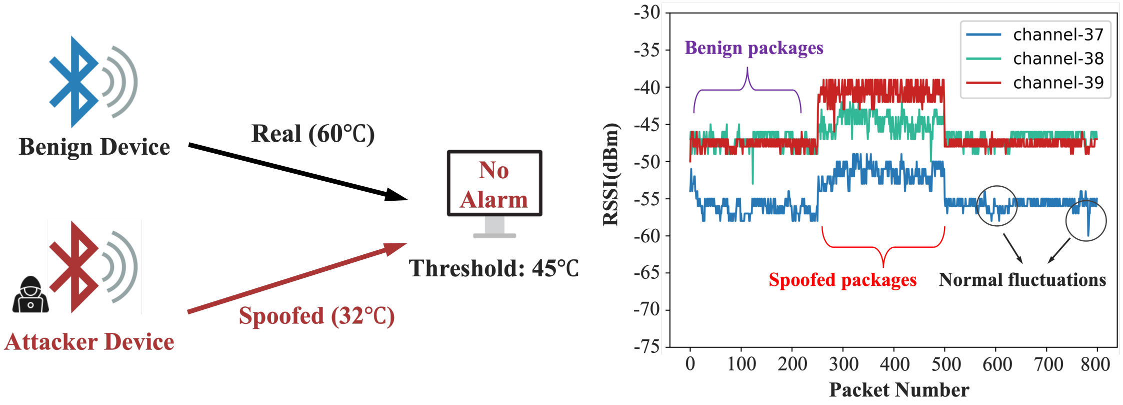


Figure 2-2 (a) Spoofing attack in BLE sensor network and (b) observed RSSI values during attack simulation.

Given the unique characteristics of BLE networks, this project has successfully identified and utilized four key cyber-physical features to enhance the detection algorithm and to facilitate the training of learning models:

- **Used Channel Numbers (UCN):** These denote the specific data channels employed during the transmission of BLE packets, crucial for analyzing communication patterns.
- **Advertising Interval (INT):** This measures the temporal interval between consecutive packets transmitted on the same advertising channel, vital for detecting timing anomalies.
- **Received Signal Strength Indicator (RSSI):** This feature represents the signal-to-noise ratio gleaned from packet exchanges, providing insights into the physical layer connectivity.
- **Carrier Frequency Offset (CFO):** Refers to the discrepancy between the expected and the actual carrier frequencies used in BLE communications, indicating potential frequency drifts or unauthorized channel usage.

2.1.3 Related Research Findings

The BLE specifications[13] provide a range of authentication mechanisms theoretically designed to prevent spoofing attacks. However, these mechanisms often fail to achieve their intended purpose in practice due to three main reasons:

- **(1) Limited Device I/O Capabilities:** A significant number of BLE devices have limited I/O capabilities, which precludes them from utilizing any robust authentication mechanisms. It is not surprising that recent research has shown that over 80% of current BLE devices communicate with user devices in plaintext without any form of authentication[14].
- **(2) Persistent Security Vulnerabilities:** For BLE devices that do implement various security measures, there are still numerous attack vectors at both the protocol level and application level that malicious actors can exploit to conduct spoofing attacks[9].
- **(3) Insufficient User Awareness:** Users of BLE devices may lack awareness or the technical knowledge required to enable and configure security features properly, leading to increased susceptibility to spoofing attacks[11].

Additionally, the challenge of implementing software-based solutions (i.e., firmware updates for BLE devices or software patches on user devices) to these security vulnerabilities is compounded by four major practical challenges:

- **(1) Ineffectiveness Against Zero-Day Exploits:** The nature of software patches does not allow them to preemptively protect against zero-day vulnerabilities, which can be immediately exploited by attackers upon discovery[4].
- **(2) Fragmented Update Ecosystem:** The diversity in BLE device manufacturers leads to a fragmented ecosystem for firmware updates, which complicates the process of applying uniform security patches across devices.
- **(3) Legacy Device Constraints:** A considerable number of legacy BLE devices in use are incapable of being updated due to outdated I/O capabilities, leaving them vulnerable to new exploits.
- **(4) Resource Constraints for Update Dissemination:** Many manufacturers of BLE devices may face resource constraints that impede the timely development and distribution of necessary firmware updates, further exacerbating security challenges[15].

2.2 Technical material

Throughout the development of the BLE network testbed and the subsequent collection of the network dataset, this project employed a variety of technical tools to facilitate data acquisition and analysis. Given the extensive range of tools and methodologies utilized, a comprehensive explanation of each is beyond the scope of the main text. Detailed descriptions and justifications for these tools are therefore provided in **Appendices 1**. Below are examples of the categories of technical tools used:

- **Attacker Platforms:** Tools such as Mirage[16], Ostinato[17], and custom scripts designed to emulate spoofing attack scenarios.
- **Network Sniffers:** Tools like Wireshark, Ubertooth, BLE-Analyzer-PRO and HCI snoop log.
- **Data Acquisition Systems:** Automated systems like nRF Connect for data logging and analysis.
- **Simulation Software:** Network simulators like GNS3[18] used for virtual testing and modeling.

Chapter 3 Research Problem

This Chapter is dedicated to outlining the core objectives of this project, structured around three major questions that it seeks to address. The intent of this chapter is to present these pivotal questions, providing a clear statement of the problems this project aims to solve within the domain of BLE network security. We will also delineate our project's contributions to the field, underlining the potential impact and advancements our findings offer to the existing body of knowledge.

3.1 Problem I: Real-Time Spoofing Detection

- **Problem I: How can we achieve real-time, precise analysis to efficiently identify advanced spoofing attackers?**

The ability to identify spoofing attacks as they occur is paramount in safeguarding BLE networks. Real-time analysis ensures that security measures can react instantly to potential threats, mitigating risks before they materialize into breaches. Precision in detection is equally crucial to avoid the costs associated with false alarms. By solving this problem, the positive impact would be two-fold: the creation of a more secure network environment and the assurance of user trust through the reliable protection of sensitive data.

3.2 Problem II: Efficiency and Impact Minimization

- **Problem II: How do we ensure that our detection mechanism remains efficient while minimizing its side effects on system operations and energy consumption?**

An efficient detection mechanism must operate with minimal impact on the system it protects. Excessive energy consumption or operational delays can be as detrimental as the threats they aim to prevent. Achieving this balance is crucial for the viability of security solutions in energy-constrained environments. By addressing this challenge, the resultant detection mechanism can be widely adopted, ensuring broad-scale security that is both effective and sustainable.

3.3 Problem III: Reproducibility and Benchmarking

- **Problem III: How do we ensure the reproducibility of our proposed solution and establish a new benchmark dataset to foster progression in this field of research?**

Reproducibility is the cornerstone of scientific advancement. Ensuring that our solution can be independently verified by other researchers reinforces the validity of our findings. Furthermore, by establishing a new benchmark dataset, we contribute a valuable resource that catalyzes further research. This not only demonstrates the practicality of our solution but also propels the field forward, providing a foundation for future innovations in BLE network security.

Each of these challenges pertains to a vital facet of BLE network security. Detailed resolutions to these challenges will be elucidated in the subsequent chapter.

Chapter 4 Research Solution

4.1 Environment Setup

In this section, we will detail the construction of BLE network testbed, as well as comprehensive information regarding the deployment environment and the devices involved. To ensure the reliability and reproducibility of the experiments, all hardware devices and software platforms utilized are readily accessible and well-documented on the Internet.

4.1.1 Testbed Implementation

In a word, the testbed environment can be categorized into four parts: (i) BLE devices, (ii) user devices, (iii) attacker platforms, and (iv) network sniffers. **Table 4-1** comprehensively illustrates all the components utilized in the network testbed.

Table 4-1 Components of proposed BLE network testbed.

Component	Description	Devices Example
BLE devices	Used to build the BLE cyberspace environment	nRF51822, DA14580 chips
User devices	Used to connect and simulate usage scenarios	Apple laptop, Android phone
Attacker platforms	Used to launch advanced spoofing attacks	CSR dongle, Lenovo laptop
Network sniffers	Used to capture network advertising package	Raspberry Pi, BLE-Analyzer

4.1.2 Deployment Environment

The testbed was strategically deployed within a physical environment: a $15m \times 15m$ office space configured with 18 cubicles, as illustrated in **Figure 4-1**. The office was methodically partitioned into $1m \times 1m$ grids. This setting typifies a complex and acoustically active indoor environment, presenting significant challenges for evaluating the detection efficiency of BLEGuard. During data collection, RF signals were monitored within the range of our sniffers, uncovering considerable channel interference. This interference was primarily caused by 40 devices equipped with Bluetooth or BLE technologies, including smartphones, speakers, mice, and keyboards, along with numerous Wi-Fi access points and two microwave ovens. It was also noted that the sudden movements of individuals within the office substantially affected the channel conditions, further complicating the monitoring environment.

4.1.3 BLE Devices

For the construction of our network testbed, we utilized sixteen commonly used BLE devices featuring a range of Bluetooth chips, including nRF51822 and DA14580, as catalogued in **Table 4-2**. After evaluating their performance, we selected nine devices that demonstrated consistent stability, suitable for our data collection needs. These devices represent a wide array of typical BLE applications, providing a comprehensive overview of potential real-world uses.

Table 4-2 BLE devices used in proposed network testbed.

ID	Device Name	Manufacturers	Device Type
1	Indoor Sensor	Xiaomi	Industry
2	Smart Lock	Xiaomi	Smart Home
3	Mijia Speaker	Xiaomi	Entertainment
4	HomePod v2	Apple	Entertainment
5	Dell Speaker	Dell	Entertainment
6	Lenovo Speaker v2	Lenovo	Entertainment
7	Smart Lock	August	Smart Home
8	Key Finder	Nutale	Smart Home
9	nRF52 DK Board	Nordic	Industry
10	Mi Smart Light Bulb	Xiaomi	Smart Home
11	Mi Smart Scale	Xiaomi	Smart Home
12	Mi Band v8	Xiaomi	Health Care
13	Door & Window Sensor	Eve	Smart Home
14	Button Remote Control	Eve	Smart Home
15	Energy Socket	Eve	Industry
16	Sport Band v4	Huawei	Health Care

4.1.4 User Devices

In our testbed, user devices are employed to connect with BLE devices, simulating typical network scenarios. Monitoring software has been implemented on user laptops and PCs to facilitate interaction with network sniffers and to collect datasets. **Table 4-3** details the user devices employed within our network environment. Additionally, while approximately 40 other network devices are present in our deployment office, data collection is exclusively focused on the BLE devices we specifically deployed, with no data recorded from any unidentified devices.

4.1.5 Attacker Platforms

To generate multiple spoofing attacks, we deployed four distinct types of attacker platforms, each comprising three identical samples situated at different locations. The specifics of the attacker platforms used in our testbed are outlined in **Table 4-4**. We opted for these platforms due to their accessibility, programmability, and utilization of various transmit power values[19]. In addition, we provided the MAC address for each device to distinguish between identical devices performing different functions, since we have multiple duplicate copies.

4.1.6 Network Sniffers

Within the office environment, three network sniffers were precisely positioned at predetermined grid coordinates, each powered by a Raspberry Pi running BLE-Analyzer-PRO software. This adaptable platform, suitable for further development, enabled the detailed capture of network packets and their cyber-physical features. The Raspberry Pi also managed the transmission of the collated datasets to the monitoring systems. The total cost of this sniffer configuration was approximately 80 dollars, demonstrating a budget-friendly approach to comprehensive network monitoring. For a more in-depth overview of the sniffer equipment used, refer to **Appendices 1**.

Table 4-3 User devices used in BLEGuard testbed.

Device Name	Operating System	MAC Address
Lenovo V15-IIL	Windows 10 Pro	0d:76:9a:3f:e7:0b
MacBook Pro M1	macOS 13.1	0f:2e:4d:1a:8c:5b
Google Pixel 7	Android 13	08:5b:3c:2f:a1:6d
iPhone 13	iOS 16	0a:9f:7e:2d:6b:8f
Surface Laptop 5	Windows 11	06:3d:1f:7e:a8:4c
Dell 7050 PC	Windows 10 Pro	0b:4a:5e:2c:9f:7d

Table 4-4 Attacker platform used in BLEGuard testbed.

Device Name	Operating Platform	MAC Address
Lenovo 15IIL laptop	Mirage tool	04:6c:59:05:9c:8a
CSR 4.0 BT dongle	Mirage tool	02:42:07:cd:65:a4
HM-10 development board	Mirage tool	02:42:13:02:c7:f0
CYW920735 development board	Ostinato tool	00:16:3e:0d:95:65

4.2 Data Preparation

This section delineates the procedures for implementing the network testbed, simulating spoofing attacks, and constructing network datasets. To ensure the reliability and reproducibility of our experiments, we will make both our data and the associated code publicly accessible[20].

4.2.1 Attack Simulation

As mentioned above, the BLEGuard system was operationalized within the testbed using user monitoring devices and network sniffers. These devices collaborated to manage the collection of network datasets, utilizing the Wireshark tool to analyze traffic. The coordination of network activities, from communication between monitors and sniffers to the exchange of advertising packets

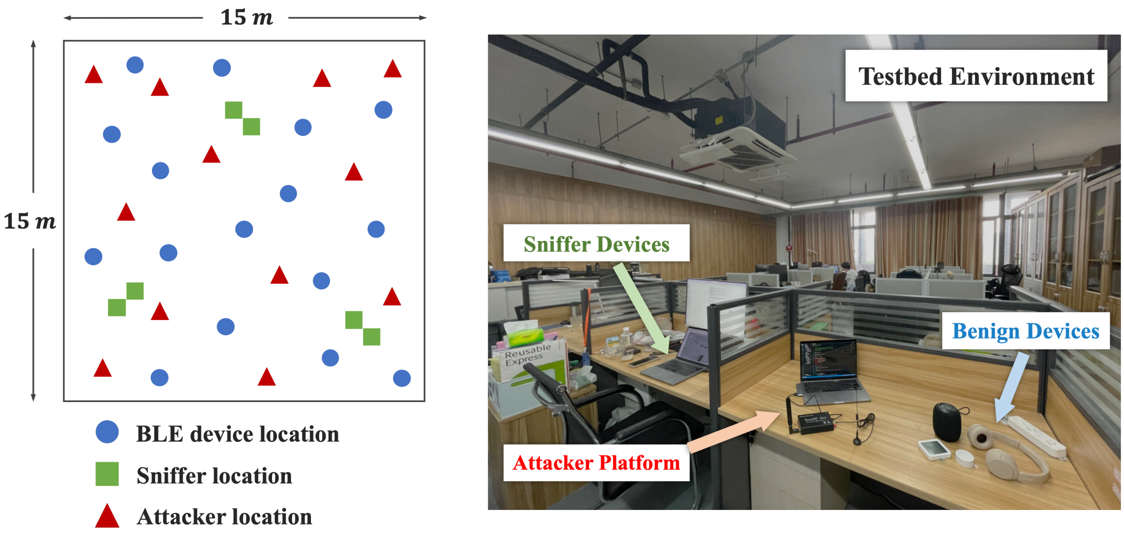


Figure 4-1 Locations of three types of devices in BLEGuard testbed.

and gathering of cyber-physical features, was handled by a Python script comprising approximately 2000 lines of code. This principal script is included in the supplementary materials.

4.2.2 Data Pre-processing

As outlined in **Table 4-4**, four types of attacker platforms were strategically deployed, with three units of each type, across twelve distinct locations. This setup enriched the cyber-physical feature datasets, particularly for RSSI and CFO. **Figure 4-1** visually represents this arrangement within the testbed environment, where sixteen BLE devices are marked by blue circles, six sniffers by green squares, and twelve attacker platforms by red triangles. In the simulated attack scenarios, identity information was cloned using USB dongles, the Mirage tool[16], and the Ostinato tool[17] to disrupt normal network connections between BLE devices and user devices. This experiment underscores the ease with which advanced attackers could exploit vulnerabilities in BLE to manipulate device settings, posing significant security risks.

4.2.3 Datasets Building

Regarding the building of our dataset, we meticulously collected normal advertising packets from each BLE device over a period of approximately eight hours—five hours during daytime and three hours at nighttime. Additionally, for each attacker platform situated in various positions, malicious packets were collected for about 20 minutes. Currently, our BLE Spoofing Attack Dataset (*BLE-SAD*) contains 906,000 advertising packets, of which 81.6% are benign and 18.4% are malicious. The open-source dataset can be accessed at: https://github.com/BLEGuard/supplement.

4.3 Detection Mechanism

In this section, we will discuss our proposed BLEGuard system, a hybrid detection mechanism combined cyber-physical analysis with machine learning techniques.

4.3.1 Pre-detection Scheme

In BLEGuard, suspicious activities are identified through the detection of atypical fluctuations in cyber-physical features such as Used Channel Numbers (UCN), Advertising Interval (INT), Carrier Frequency Offset (CFO), and Received Signal Strength Indicator (RSSI). Abrupt changes in UCN and INT may indicate potential security threats, while RSSI and CFO are crucial for a continuous pre-detection mechanism that anticipates advanced spoofing attacks.

To effectively monitor these indicators, BLEGuard employs three network sniffers that capture the values of these features within a *lookback window*. The lookback window refers to a predefined period prior to the current analysis point, during which data is collected to establish a baseline for normal behavior. This historical data is essential for understanding typical network conditions and variations. Subsequently, the system evaluates the current network activity by examining the values from an *observation window*, which is the period immediately following the lookback window. This approach allows BLEGuard to compare present data against the baseline to spot any irregularities or deviations.

An alarm is triggered if there are deviations from the established norms in any of the monitored features, indicating a potential security breach. This method can be seamlessly integrated into existing BLE networks without causing disruption or significant resource consumption. Detailed detection schemes for each feature are outlined as follows:

• Metric 1: Used Channel Numbers

In BLE networks, Used Channel Numbers (UCN) designates the sequence of radio channels that BLE devices utilize for transmission, adhering to a preconfigured pattern to enhance connectivity and reduce noise interference. The stability of UCN patterns can be compromised during spoofing attacks, as attackers may instigate an irregular shift in the communication channels, thus disrupting the network's harmonious channel utilization. To quantify such fluctuations, we introduce the metric UCN_{change}, which represents the cumulative measure of channel switching activity:

$$UCN_{\text{change}} = \sum_{i=1}^{N_{\text{obs}}} |UCN_i - UCN_{i-1}|, \tag{4-1}$$

where N_{obs} is the count of observed transmission packets and UCN_i corresponds to the utilized channel for the i^{th} packet transmission. An elevated UCN_{change} value is indicative of more frequent channel alternations, potentially signaling an ongoing spoofing attack. For operational integrity in BLE networks, an acceptable threshold for UCN_{change}, denoted by $\Delta UCN_{\text{normal}}$, is set at 2.8. This threshold indicates the maximum allowable frequency of channel changes within a defined observation period. A breach of this threshold is symptomatic of anomalous behavior:

$$\text{If } UCN_{\text{change}} > \Delta UCN_{\text{normal}}, \text{ then activate further detection.} \tag{4-2}$$

Employing UCN_{change} as a heuristic enables a robust security framework capable of detecting and responding to potential spoofing threats, thereby fortifying the BLE network's defenses.

- **Metric 2: Advertising Interval**

The Advertising Interval (INT) is also a key parameter in BLE communications, defining the time gap between consecutive advertising packets. This interval is crucial for maintaining the orderly transmission of broadcast information in BLE networks. By definition, the INT between any two consecutive advertising packets should never fall below a predefined lower bound, which is set based on the specifications of the BLE device and the operational requirements of the network. This lower bound is denoted as L_{int}. The formula used to compute the runtime INT value, INT, for the interval between two packets is given by:

$$INT = T_{\text{current}} - T_{\text{previous}} \tag{4-3}$$

where T_{current} is the timestamp of the current advertising packet, and T_{previous} is the timestamp of the immediately preceding advertising packet. The Advertising Interval (INT) is calculated as the difference between these two timestamps. If INT is found to be less than the predefined lower limit L_{int}, the monitor identifies this condition as anomalous. Such a scenario indicates a potential operational fault or a security breach, such as a spoofing attack that attempts to flood the network with frequent, unauthorized advertising packets. Upon detecting such an anomaly, the monitor triggers an alarm, alerting the system to the potential threat. Typically, L_{int} is set to a threshold value of 10 milliseconds to detect rapid, unscheduled transmissions[9]. The corresponding condition can be mathematically expressed as:

$$\text{If } INT < L_{\text{int}}, \text{ then activate further detection.} \tag{4-4}$$

This monitoring mechanism ensures the integrity and correct functioning of the BLE network by verifying that the advertising packets are transmitted within the expected intervals, adhering to the designed operational parameters.

- **Metric 3: CFO level**

BLEGuard continuously monitors the CFO (Carrier Frequency Offset) and RSSI (Received Signal Strength Indicator) values from advertising packets. Upon activation of the CFO and RSSI inspection, BLEGuard analyzes these values through the following procedure. For a BLE device exhibiting intermittent advertising patterns, we define the lookback window as the time period T_l (with N_l packets) before the transition from advertising to connection state, and the observation window as the time period T_o (with N_o packets) after the transition from connection back to advertising state. In BLEGuard, following the reception of a connection request packet, the monitoring system initiates the CFO and RSSI inspections for advertising packets collected from each device across the three advertising channels (37, 38 and 39). The system first calculates the acceptable ranges for CFO and RSSI values using data from the lookback window. It then evaluates these metrics in the advertising packets during the observation window. If an anomaly is detected in either the CFO or RSSI readings, an alarm is triggered.

The CFO values observed from BLE networks are expected to conform to a Gaussian distribution[9]. Consequently, when μ_0 and σ_0 represent the mean and standard deviation of these CFO values, the probability distribution function for the CFO can be articulated as:

$$F_{cfo}(x_i) = \frac{1}{\sigma_0\sqrt{2\pi}} \cdot e^{-\frac{(x_i-\mu_0)^2}{2\sigma_0^2}} \tag{4-5}$$

where x_i denotes a sample CFO value. In BLEGuard, the monitor employs the CFO values from advertising packets within a lookback window, comprising N_l packets, to calculate μ_0 and σ_0. These parameters are then integrated into the probability function previously mentioned. If the advertising packets from both the lookback and subsequent observation windows originate from the same BLE device, the CFO values from the observation window's advertising packets should statistically align with the given distribution. This is verified by the monitoring system calculating the negative log-likelihood of the CFO values from the observation window packets, defined as:

$$L_{cfo} = \frac{1}{N_o}\sum_{i=1}^{N_o} -\log F_{cfo}(x_i) \tag{4-6}$$

If the log-likelihood value is less than a predetermined CFO inspection threshold, denoted by β_{cfo} (i.e., $L_{cfo} < \beta_{cfo}$), the CFO values are considered to be within the normal range for the BLE device. This threshold β_{cfo} is a tunable parameter within BLEGuard that dictates the permissible range of CFO values during the observation window. In contrast, if the log-likelihood value exceeds β_{cfo} (i.e., $L_{cfo} > \beta_{cfo}$), an anomaly is recognized, and an alarm is activated, signaling a possible spoofing attack. In most real-world scenarios, β_{cfo} is generally set to 3.

- **Metric 4: RSSI level**

To detect anomalies in RSSI values amid strong signal reflections in BLE networks, we utilize a two-component Gaussian mixture model. This approach is chosen because RSSI values in environments with high noise can be effectively modeled using two normal distributions[21]. The probability distribution function for RSSI values is given by:

$$F_{rssi}\left(y_i\right) = w \cdot \frac{1}{\sigma_1\sqrt{2\pi}} \cdot e^{-\frac{(y_i-\mu_1)^2}{2\sigma_1^2}} + (1-w) \cdot \frac{1}{\sigma_2\sqrt{2\pi}} \cdot e^{-\frac{(y_i-\mu_2)^2}{2\sigma_2^2}} \tag{4-7}$$

In this equation, μ_1 and μ_2 are the means of the two components, σ_1 and σ_2 are their standard deviations, w is a weight parameter that balances the two components, and y_i is an RSSI sample. Using the BLEGuard system, N_l of RSSI values from advertising packets within a lookback window are analyzed to estimate the parameters $\mu_1, \mu_2, \sigma_1, \sigma_2$, and w through a conventional expectation-maximization (EM) algorithm[22]. Following this, the monitor calculates the negative log-likelihood that the RSSI values $(y_i, \forall i \in [1, N_o])$ from the observation window conform to the model specified by Equation (4-7):

$$L_{rssi} = \frac{1}{N_o} \sum_{i=1}^{N_o} - \log F_{rssi}\left(y_i\right) \tag{4-8}$$

An anomaly is detected when the negative log-likelihood surpasses a predefined RSSI inspection threshold, denoted as δ_{rssi} (i.e., $L_{\text{rssi}} > \delta_{\text{rssi}}$). The threshold δ_{rssi} is a crucial parameter in BLEGuard, calibrated to optimize detection sensitivity and specificity. Typically, δ_{rssi} is set to 5.

4.3.2 Reconstruction Model

Upon identifying suspicious activities, a thorough analysis is initiated on anomalous data batches. To facilitate this, a Temporal Convolutional Network (TCN)[23], as illustrated in **Figure 4-2**, is employed to reconstruct traffic patterns. This approach helps isolate aberrant data through comparative analysis. A TCN is a type of neural network specifically designed for sequence modeling that combines convolutional layers with causal connections to ensure that predictions for a specific time step can only depend on past data. This structure makes TCNs particularly effective for time series predictions where past context is crucial.

During the offline training phase, the objective is to minimize the discrepancy between the learned data D_L and the original dataset D_T. In the online testing phase, the presence of malicious packets in the input data triggers an increase in the reconstruction error, indicative of potential spoofing threats. The residual, defined as $R(D_T, D_L) = |D_T - D_L|$ with $D_L = f(D_T)$, where f represents the transformation function employed by the TCN auto-encoder, serves as a critical metric. This residual is assessed to calculate the anomaly score α for each data batch, as depicted in Equation (4-9). Here, R_α denotes the calculated residual, μ is the mean value of the residual, and σ is its standard deviation. **Figure 4-2** demonstrates how benign inputs lead to benign outputs, whereas malicious inputs result in malicious outputs, showing the operational mechanism of TCN.

$$\alpha = \begin{cases} 0, \, when \ |R_\alpha - \mu R_\alpha| \leq 3 * \sigma R_\alpha \ \rightarrow \ Normal \ Data \ Batch \\ 1, \, when \ |R_\alpha - \mu R_\alpha| > 3 * \sigma R_\alpha \ \rightarrow \ Suspicious \ Data \ Batch \end{cases} \tag{4-9}$$

4.3.3 Classification Models

Following the identification of suspicious data batches, the next step involves classifying these packets into two categories: benign or malicious. In this research, a text-convolutional neural network (text-CNN)[24] is employed for the extraction of traffic features. Text-CNNs are specialized types of convolutional neural networks designed to handle text data. They apply convolutional layers to extract higher-level features from text data structured as input vectors, making them highly effective for tasks involving natural language processing and text analysis.

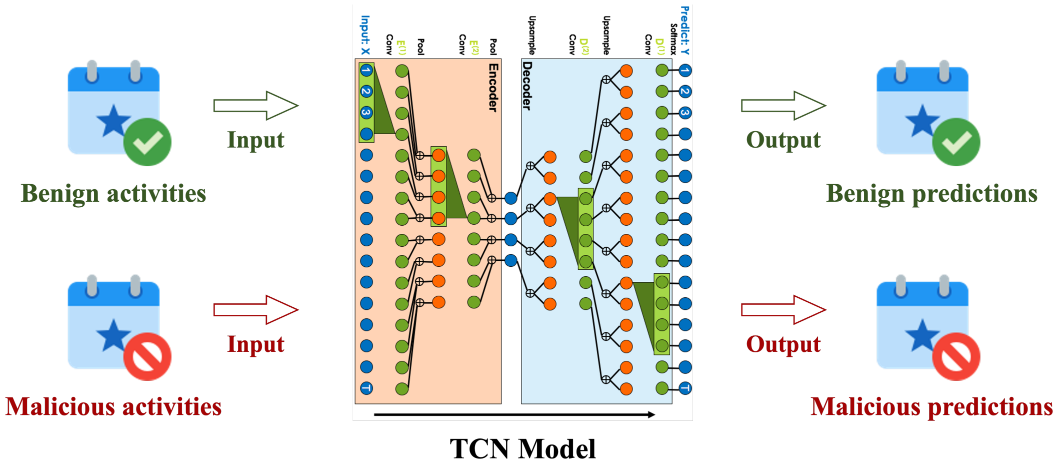


Figure 4-2 Operational mechanism of the temporal convolutional network.

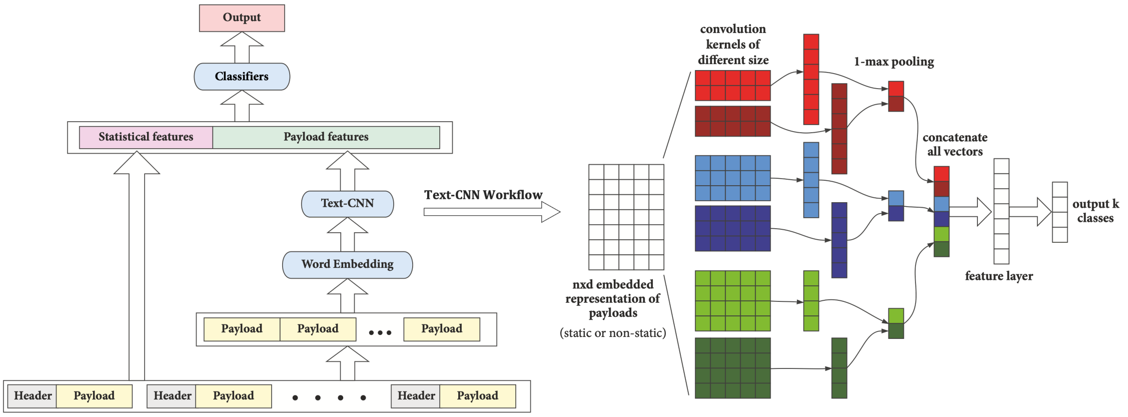


Figure 4-3 The general architecture of the classification models.

For packet classification, this project employs four cost-efficient classifiers: Support Vector Machine (SVM)[25], K-Nearest Neighbors (KNN)[10], Random Forest (RF)[24], and Naïve Bayes[26]. This multi-classifier approach helps to mitigate potential biases in text analysis by diversifying the analytical perspectives. Network payload-based features are generated by converting the payload bytes into low-dimensional vectors using *Word2Vec* techniques, which effectively capture the semantic relationships within the data. These vectors form the input for the text-CNN, where key traffic features are extracted. The features extracted by the text-CNN are then concatenated with statistical features to create a comprehensive feature set for the final classification models.

The workflow for packet classification using text-CNN is illustrated in **Figure 4-3**, where it can be seen how raw data is transformed through various stages of processing to classify packets accurately as either benign or malicious.

4.3.4 System Overview

BLEGuard is designed to optimize the balance between detection accuracy and power consumption. As depicted in **Figure 4-4**, the system employs a flexible approach where the pre-detection algorithm is utilized to maintain efficiency under GPU resource constraints, minimizing power and computational overhead. In scenarios where high detection accuracy is paramount, the reconstruction model is activated to enhance analytical precision. Moreover, the classification models within BLEGuard are adept at precisely identifying malicious advertising packets, providing targeted feedback that significantly augments the efficacy of the detection modules. This versatile framework ensures that BLEGuard can adapt to varying operational demands, thereby maintaining robust security measures without compromising on network performance.

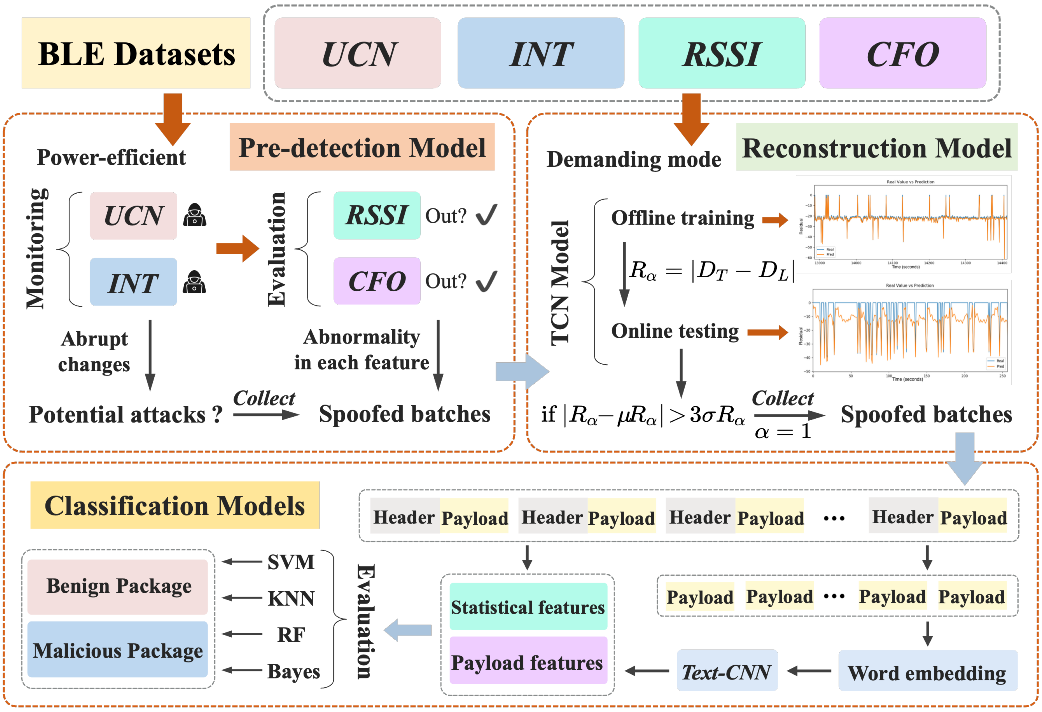


Figure 4-4 The overall workflow of *BLEGuard* detection mechanism.

In this chapter, we have provided a comprehensive description of the construction process for the BLE-SAD dataset, as well as a detailed exposition of the mathematical models and operational mechanisms underpinning the BLEGuard System. The subsequent chapter will delve into the specific parameter settings of the models, as well as present the experimental data and results, demonstrating the effectiveness of BLEGuard in various testing scenarios.

Chapter 5 Evaluation

5.1 Experiment Preparation

This section describes the experimental framework used to evaluate the efficacy of the BLEGuard system. The settings are meticulously designed to mimic realistic scenarios in which BLE networks operate, ensuring that the results are both robust and applicable to real-world applications.

5.1.1 Experimental Details

The BLE-SAD dataset was compiled from nine distinct BLE devices, with their information detailed in **Appendices 1-1**. The dataset was divided into training and testing sets at a ratio of 8.5 to 1.5, comprising 762,850 effective BLE network packets for training and 134,088 for testing, respectively. The model training was conducted using an Intel Core i5-13600 CPU processor (3.50 GHz) with 32GB of RAM and an NVIDIA GeForce RTX 4060 Ti GPU equipped with 24GB of memory. The algorithms were implemented in Python 3.8, utilizing the PyTorch 1.8.1 framework.

5.1.2 Parameter Settings

In BLEGuard's Pre-detection Scheme, four crucial parameters ($\Delta UCN_{\text{normal}}$, L_{int}, β_{cfo}, and δ_{rssi}) are meticulously configured within specific ranges to maximize detection accuracy, as summarized in **Table 5-1**. These settings are the result of comprehensive testing and fine-tuning, ensuring that BLEGuard efficiently and reliably identifies spoofing attacks within BLE networks.

Table 5-1 Optimal Parameter Settings for the Pre-detection Scheme.

Network Features	Parameter	Setting Range	Optimal Setting
Used Channel Numbers (UCN)	$\Delta UCN_{\text{normal}}$	(2.0, 5.0)	**2.8**
Advertising Interval (INT)	L_{int}	(5.0, 20.0) ms	**10.0 ms**
Carrier Frequency Offset (CFO)	β_{cfo}	(1.0, 5.0)	**3.0**
Received Signal Strength Indicator (RSSI)	δ_{rssi}	(3.0, 10.0)	**5.0**

Additionally, **Table 5-2** provides a detailed account of the Temporal Convolutional Network's configuration, designed to process dynamic time-series data for immediate anomaly detection. **Table 5-3** expounds on the hyperparameters of the text-convolutional neural network model, employed to categorize packet data effectively into benign or malicious classes.

5.2 Performance Evaluation Metrics

In an effort to mirror a realistic network environment, the BLE-SAD dataset was intentionally constructed with an imbalanced proportion of benign to malicious packets. This imbalance, characterized by a predominance of benign samples, can induce skewed results. Consequently, the conventional metric of accuracy loses its representativeness and reliability as a sole measure of

Table 5-2 The hyperparameters of the temporal convolutional network network.

Hyperparameters	Value
Optimizer	RMSprop
Learning rate	5e-4
Kernel size	8
Number of filters	9
Loss function	MSE
Hidden units	10
Dropout rate	0.05
Gradient clipping	1

Table 5-3 The hyperparameters of the text-convolutional neural network.

Hyperparameters	Value
Optimizer	Adam
Learning rate	1e-4
Batch size	50
Epoch number	50
Loss function	Binary cross-entropy
Validation metric	Accuracy
Validation split	0.2
Deep learning framework	PyTorch 1.8.1, Gensim (WordVec) 3.7.1

performance evaluation in such contexts. To mitigate the risk of biased analysis, more robust metrics like False Alarm Rate (FAR) and Un-detection Rate (UND) have been advocated for[10]. Therefore, in our performance evaluation, we extend beyond mere accuracy and incorporate FAR and UND. **Table 5-4** presents these evaluation metrics along with their respective formulas, defining TP, TN, FP, FN as follows:

- TP (True Positive): Represents the number of malicious packets correctly classified as malicious.
- TN (True Negative): Represents the number of benign packets correctly identified as benign.
- FP (False Positive): Represents the number of benign packets incorrectly identified as malicious.
- FN (False Negative): Represents the number of malicious packets incorrectly identified as benign.

Table 5-4 Formula for three evaluation metrics

Evaluation metrics	Corresponding formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN} \times 100\%$
False alarm rate (FAR)	$\frac{FP}{FP+TN} \times 100\%$
Un-detection rate (UND)	$\frac{FN}{FN+TP} \times 100\%$

5.3 Overall Performance Evaluation

During the evaluation phase, three key metrics are employed to assess the effectiveness of our proposed methods. Accuracy, defined as the overall proportion of correctly classified instances, serves as a fundamental measure of the model's capability to accurately differentiate between benign and malicious packets. This metric is critical in evaluating the overall efficacy of the detection system. The False Alarm Rate (FAR) quantifies how often BLEGuard erroneously activates an alert when processing benign advertising packets from legitimate BLE devices, reflecting the model's precision. Conversely, the Un-detection Rate (UND) measures the frequency with which BLEGuard fails to identify a spoofing attack, highlighting potential vulnerabilities in detecting sophisticated threats.

BLEGuard's performance evaluation is conducted on a robust and imbalanced dataset collected from nine different BLE devices, such as Xiaomi sensors, Apple HomePod, and Dell speakers. The devices and the corresponding evaluation results are comprehensively detailed in **Table 5-5**. The table presents the performance metrics for each device, including the accuracy, FAR, and UND, thus providing a granular view of the system's effectiveness across varied hardware configurations. The empirical data from **Table 5-5** reveals BLEGuard's formidable detection capabilities, achieving an exemplary average accuracy of 99.02%, complemented by a low false alarm rate of 2.04% and an un-detection rate of 0.37%. These statistics not only validate the robustness of BLEGuard but also illustrate its adaptability and reliability in diverse operational environments.

Table 5-5 Detection performance of *BLEGuard* mechanism.

ID	Device (Number)	Accuracy	FAR	UND
1	Xiaomi Sensor (*3)	98.92%	2.23%	0.43%
2	Xiaomi Locker (*2)	99.11%	2.04%	0.32%
3	Xiaomi Speaker (*2)	98.93%	1.84%	0.36%
4	Apple HomePod (*1)	99.04%	2.11%	0.34%
5	Dell Speaker (*1)	99.21%	2.51%	0.17%
6	Lenovo Speaker (*1)	98.71%	1.81%	0.76%
7	August Smart Lock (*2)	99.00%	2.43%	0.19%
8	Nutale Key Finder (*2)	99.05%	1.45%	0.52%
9	Nordic nRF52 DK (*2)	99.20%	1.96%	0.22%
Overall		**99.02%**	**2.04%**	**0.37%**

BLEGuard's performance, with a 2.04% FAR and a 0.37% UND, indicates high reliability for real-world BLE network monitoring. In a weeklong, intensive-use scenario, it would minimally misidentify benign activity or miss spoofing incidents, maintaining network integrity with little disruption. Such reliability translates into reduced maintenance demands and allows network administrators to focus their efforts on proactive improvements rather than reactive troubleshooting.

Chapter 6 Conclusion

In this project, we developed the ***BLEGuard*** system, a novel hybrid detection mechanism designed to safeguard Bluetooth Low Energy (BLE) networks against sophisticated spoofing attacks. BLEGuard's unique integration of a pre-detection scheme, reconstruction techniques, and classification models enables it to effectively identify and neutralize threats, thereby enhancing network security. The system's high detection accuracy, combined with a low false alarm rate and un-detection rate, underscores its potential not only as a specialized tool for BLE security but also for broader applications in industry, healthcare, and smart home sectors. The practical application of BLEGuard in these sectors can significantly mitigate risks associated with the inherent security vulnerabilities of BLE technologies, providing a reliable security solution that aligns with the needs of modern connected environments.

The construction of the ***BLE-SAD*** dataset was a cornerstone of this project, providing a critical resource for testing and refining the BLEGuard system. This large-scale dataset, generated through comprehensive simulations of spoofing attacks within a controlled testbed environment, offers an invaluable asset for related cybersecurity research. Its detailed representation of varied attack scenarios enables researchers and security professionals to rigorously evaluate Bluetooth security solutions and contributes to the ongoing development of advanced defensive methods against an array of cyber threats.

Looking to the future, there are several avenues for further enhancing the BLEGuard system and the BLE-SAD dataset. Enriching the dataset with a broader spectrum of attack scenarios, including Man in the Middle (MITM) and Distributed Denial of Service (DDoS) attacks, will extend our research's reach and enrich the training resources available for developing robust BLE security measures. Additionally, adapting BLEGuard to function as a general framework for Bluetooth security could transform it into a versatile tool capable of defending against a diverse range of cyber threats. Such advancements will not only fortify the security of BLE networks but also pave the way for next-generation protection mechanisms in the evolving landscape of digital communication technologies.

Reference

[1] Gomez C, Oller J, Paradells J. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology[J]. sensors, 2012, 12(9): 11734-11753.

[2] Pattnaik S K, Samal S R, Bandopadhaya S, et al. Future wireless communication technology towards 6G IoT: An application-based analysis of IoT in real-time location monitoring of employees inside underground mines by using BLE[J]. Sensors, 2022, 22(9): 3438.

[3] Bluetooth-SIG. Bluetooth Market Update[Z]. https://bluetooth.com/2024-market-update/. Accessed on: 20 April 2024.

[4] Stellios I, Kotzanikolaou P, Psarakis M. Advanced persistent threats and zero-day exploits in industrial Internet of Things[J]. Security and Privacy Trends in the Industrial Internet of Things, 2019: 47-68.

[5] Ditton S, Tekeoglu A, Bekiroglu K, et al. A proof of concept denial of service attack against bluetooth iot devices[C]. in: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). 2020: 1-6.

[6] Wu J, Nan Y, Kumar V, et al. {BLESA}: Spoofing attacks against reconnections in bluetooth low energy[C]. in: 14th USENIX Workshop on Offensive Technologies (WOOT 20). 2020.

[7] Zhang P, Nagarajan S G, Nevat I. Secure location of things (SLOT): Mitigating localization spoofing attacks in the Internet of Things[J]. IEEE Internet of Things Journal, 2017, 4(6): 2199-2206.

[8] Wu J, Wu R, Xu D, et al. SoK: The Long Journey of Exploiting and Defending the Legacy of King Harald Bluetooth[C]. in: 2024 IEEE Symposium on Security and Privacy (S&P). 2023.

[9] Wu J, Nan Y, Kumar V, et al. {BlueShield}: Detecting spoofing attacks in bluetooth low energy networks[C]. in: 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020). 2020.

[10] Lahmadi A, Duque A, Heraief N, et al. MitM attack detection in BLE networks using reconstruction and classification machine learning techniques[C]. in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. 2020: 149-164.

[11] Cäsar M, Pawelke T, Steffan J, et al. A survey on Bluetooth Low Energy security and privacy [J]. Computer Networks, 2022, 205: 108712.

[12] Cai H. Securing Billion Bluetooth Devices Leveraging Learning-Based Techniques[C]. in: Proceedings of the AAAI Conference on Artificial Intelligence: vol. 38: 21. 2024: 23731-23732.

[13] Bluetooth-SIG. Bluetooth Core Specification 5.4[Z]. https://www.bluetooth.com/specifications/. Accessed on: 20 April 2024.

[14] Bluetooth-SIG. Security in Bluetooth Specifications[Z]. https://www.bluetooth.com/learn-about-bluetooth/key-attributes/bluetooth-security/. Accessed on: 20 April 2024.

[15] Antonioli D, Tippenhauer N O, Rasmussen K. Key negotiation downgrade attacks on bluetooth and bluetooth low energy[J]. ACM Transactions on Privacy and Security (TOPS), 2020, 23(3): 1-28.

[16] Sun J, Sun K, Li Q. Towards a believable decoy system: Replaying network activities from real system[C]. in: 2020 IEEE Conference on Communications and Network Security (CNS). 2020: 1-9.

[17] Khamaiseh S, Serra E, Li Z, et al. Detecting saturation attacks in sdn via machine learning[C]. in: 2019 4th International Conference on Computing, Communications and Security (ICCCS). 2019: 1-8.

[18] Neumann J C. The book of GNS3: build virtual network labs using Cisco, Juniper, and more [M]. No Starch Press, 2015.

[19] Yaseen M, Iqbal W, Rashid I, et al. Marc: A novel framework for detecting mitm attacks in ehealthcare ble systems[J]. Journal of medical systems, 2019, 43: 1-18.

[20] Cai H, Fang Y, Huang J, et al. Poster: Hybrid Detection Mechanism for Spoofing Attacks in Bluetooth Low Energy Networks[C]. in: Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services. 2024.

[21] Sheng Y, Tan K, Chen G, et al. Detecting 802.11 MAC layer spoofing using received signal strength[C]. in: IEEE INFOCOM 2008-The 27th Conference on Computer Communications. 2008: 1768-1776.

[22] Guo X, Li L, Xu F, et al. Expectation maximization indoor localization utilizing supporting set for Internet of Things[J]. IEEE Internet of Things Journal, 2018, 6(2): 2573-2582.

[23] Lea C, Flynn M D, Vidal R, et al. Temporal convolutional networks for action segmentation and detection[C]. in: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[24] Min E, Long J, Liu Q, et al. TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest[J]. Security and Communication Networks, 2018.

[25] Ioannou C, Vassiliou V. Network attack classification in IoT using support vector machines[J]. Journal of sensor and actuator networks, 2021, 10(3): 58.

[26] Mehmood A, Mukherjee M, Ahmed S H, et al. NBC-MAIDS: Naive Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks[J]. The Journal of Supercomputing, 2018, 74: 5156-5170.

[27] Barua A, Al Alamin M A, Hossain M S, et al. Security and privacy threats for bluetooth low energy in iot and wearable devices: A comprehensive survey[J]. IEEE Open Journal of the Communications Society, 2022, 3: 251-281.

[28] Chen X, Hao Z, Li L, et al. Cruparamer: Learning on parameter-augmented api sequences for malware detection[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 788-803.

Acknowledgements

It has been a wonderful long journey since I stepped into the MIEC.

First and foremost, I am extremely grateful to my advisors, Zhezhuang Xu and Meng Yuan, who led me to the field of wireless networks. Through more than 500 emails and numerous meetings over three years, they meticulously guided and shaped me from an eager but mostly confused freshman into an aspiring junior researcher. This thesis and all of my achievements during undergraduate studies would not have been possible without their constant guidance and encouragement. I am deeply thankful to them for teaching me not just about some aspects of networks and the art of communicating ideas, but also for cultivating my ability to think critically.

I am also immensely fortunate to have met the best teachers at MIEC, Chin Hong Wong, Ronan Reilly, Siyuan Zhan, Zhicong Chen, Lijun Wu, Lachman Tarachand, Wong Kah Hieng, Dan Chen and Yong Zhou, who have imparted knowledge and provided many invaluable advice to me. Special thanks are due to our student affairs officers, Shenghui You, Xinyi Liu, Jingxin Wang, Xiaoying Liu, Yishu Liu, and Yaping Wang, for their meticulous care and support.

It is a great honor and pleasure to work with such exceptionally talented collaborators, Jiaqi Hu, Zheng Li, and Yuchen Fang, with whom I co-founded the DefenderIoT group. I am deeply thankful for the trust and support given by our members, Shuying Liu, Jiacheng Huang, Miaolan Zhou, Hongming Chen, Xun Sun, Zhongheng Sun, Yuxuan Zheng, Wenjing Chen, and many newcomers. My journey at MIEC is coming to the end, and now it is your time.

I am also indebted to several funding agencies that supported my research and studies, including the National Undergraduate Innovation & Entrepreneurship Training Program Platform, AAAI Undergraduate Consortium, Fujian Energy Petrochemical Group, Xiamen Airlines, and the generous scholarship provided by MIEC.

I am so blessed to have forged such meaningful friendships at Fuzhou University, which have made my time here wonderfully joyous, exciting, and loving. Thank you to Zhaolin Chen, Wenxuan Luo, Yufei Wu, Jiaxuan Zhang, and Shengbin Fu for being there for me. Of course, I shall not forget to mention the best FZU swimming team, where I left both laughter and tears.

A heartfelt thank you to my parents, my sweet litter sister and my family, whose unwavering support has empowered me to grasp new chances and pursue my dreams. So what I always strive for is to make them proud of me.

Lastly, to Linshi, the wonderful girl I met in a beautiful summer, who always believes in me ten times more than I do. I love you.

Appendices

6.1 Appendices 1: Further Information for proposed Testbed

- **BLE Devices:** A variety of commercial BLE devices, such as sensors, locks, and beacons, which represent a cross-section of typical endpoints found in BLE networks. These devices are instrumental in generating the benign traffic patterns for our datasets.
- **User Devices:** Smartphones, tablets, and computers used by end-users to interact with BLE devices. These devices are equipped with BLE capabilities to emulate regular user operations and activities within the network.
- **Network Sniffers:** Devices and software used to capture and analyze the traffic flowing through the BLE network. Examples include Wireshark for packet analysis and Ubertooth for specific BLE monitoring (Appendices 1-1).
- **Attacker Platforms:** These include custom-built software and modified hardware designed to simulate various security attacks on the BLE network, such as spoofing and denial of service (DoS) attacks. Tools in this category help test the robustness of the network's security measures.
- **Data Acquisition Systems:** These systems are configured to automatically record all network traffic, capturing essential metrics such as packet size, timing, and payload data. They are critical for gathering the raw data needed for further analysis.
- **Simulation Software:** Software tools that simulate network conditions and behaviors, which help in predicting network performance under various scenarios and in understanding potential network failures before they occur.

Appendices 1-1: Several combinations of devices to implement the network sniffer.

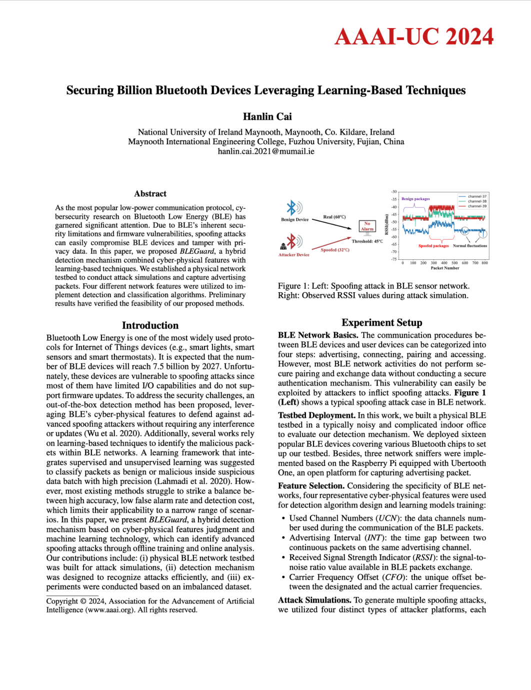
Communication Platform	Network Capture Tool	Total Cost
Raspberry Pi (Linux 5.4)	BLE-Analyzer-PRO	About $80
Raspberry Pi (Linux 5.4)	Ubertooth One	About $100
Raspberry Pi Pico (Linux 4.14)	MDBT42Q-DB-32	About $22
Google Pixel 7 (Android 13)	nRF Connect Software	(User device)
Apple MacBook (macOS 13.1)	nRF Connect Software	(User device)
Dell 7050 PC (Windows 10)	nRF Connect Software	(User device)

6.2 Appendices 2: Related Publication for this Project

[1] Securing Billion Bluetooth Devices leveraging Learning-based Technique[C]. The 38th Annual AAAI Conference on Artificial Intelligence, undergraduate consortium (AAAI 2024, Research Proposal). **First Author**. Related to the Chapter 4 (4.2).

[2] Hybrid Detection Mechanism for Spoofing Attacks in Bluetooth Low Energy Networks[C]. The 22nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys 2024, Poster). **First Author**. Related to the Chapter 4 (4.3).

Please note that I am the first author of the publications included in this appendix, which closely relate to the work discussed in this report. Due to the inclusion of these paper and their overlap with the content of this report, there may be an elevated similarity index when this document undergoes plagiarism checks. This statement serves to declare that such similarities are expected and result from the reuse of foundational work from my own published research, contributing directly to the development of this project. This declaration is made to ensure transparency and to preclude any potential concerns regarding the originality of the work presented herein.



AAAI-UC 2024

Securing Billion Bluetooth Devices Leveraging Learning-Based Techniques

Hanlin Cai
National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland
Maynooth International Engineering College, Fuzhou University, Fujian, China
hanlin.cai.2021@mumail.ie

Abstract

As the most popular low-power communication protocol, cybersecurity research on Bluetooth Low Energy (BLE) has garnered significant attention. Due to BLE's inherent security limitations and firmware vulnerabilities, spoofing attacks can easily compromise BLE devices and tamper with privacy data. In this paper, we proposed *BLEGuard*, a hybrid detection mechanism combined cyber-physical features with learning-based techniques. We established a physical network testbed to conduct attack simulations and capture advertising packets. Four different network features were utilized to implement detection and classification algorithms. Preliminary results have verified the feasibility of our proposed methods.

Introduction

Bluetooth Low Energy is one of the most widely used protocols for Internet of Things devices (e.g., smart lights, smart sensors and smart thermostats). It is expected that the number of BLE devices will reach 7.5 billion by 2027. Unfortunately, these devices are vulnerable to spoofing attacks since most of them have limited I/O capabilities and do not support firmware updates. To address the security challenges, an out-of-the-box detection method has been proposed, leveraging BLE's cyber-physical features to defend against advanced spoofing attackers without requiring any interference or updates (Wu et al. 2020). Additionally, several works rely on learning-based techniques to identify the malicious packets within BLE networks. A learning framework that integrates supervised and unsupervised learning was suggested to classify packets as benign or malicious inside suspicious data batch with high precision (Lahmadi et al. 2020). However, most existing methods struggle to strike a balance between high accuracy, low false alarm rate and detection cost, which limits their applicability to a narrow range of scenarios. In this paper, we present *BLEGuard*, a hybrid detection mechanism based on cyber-physical features judgment and machine learning technology, which can identify advanced spoofing attacks through offline training and online analysis. Our contributions include: (i) physical BLE network testbed was built for attack simulations, (ii) detection mechanism was designed to recognize attacks efficiently, and (iii) experiments were conducted based on an imbalanced dataset.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Figure 1: Left: Spoofing attack in BLE sensor network. Right: Observed RSSI values during attack simulation.

Experiment Setup

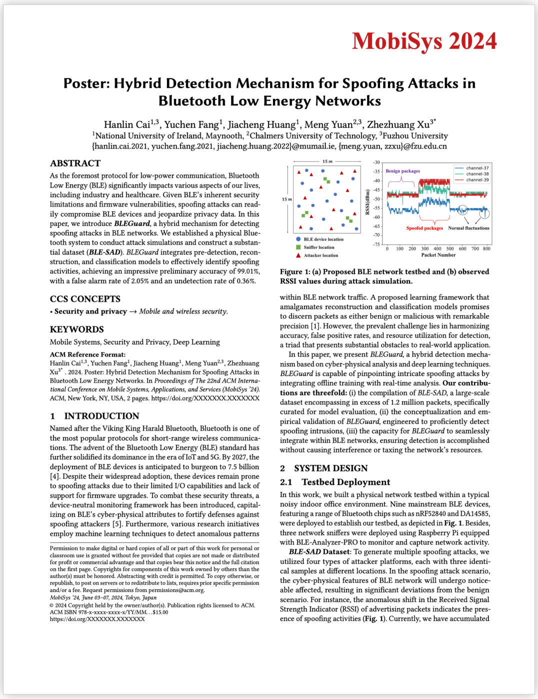
BLE Network Basics. The communication procedures between BLE devices and user devices can be categorized into four steps: advertising, connecting, pairing and accessing. However, most BLE network activities do not perform secure pairing and exchange data without conducting a secure authentication mechanism. This vulnerability can easily be exploited by attackers to inflict spoofing attacks. **Figure 1 (Left)** shows a typical spoofing attack case in BLE network.

Testbed Deployment. In this work, we built a physical BLE testbed in a typically noisy and complicated indoor office to evaluate our detection mechanism. We deployed sixteen popular BLE devices covering various Bluetooth chips to set up our testbed. Besides, three network sniffers were implemented based on the Raspberry Pi equipped with Ubertooth One, an open platform for capturing advertising packet.

Feature Selection. Considering the specificity of BLE networks, four representative cyber-physical features were used for detection algorithm design and learning models training:

- Used Channel Numbers (*UCN*): the data channels number used during the communication of the BLE packets.
- Advertising Interval (*INT*): the time gap between two continuous packets on the same advertising channel.
- Received Signal Strength Indicator (*RSSI*): the signal-to-noise ratio value available in BLE packets exchange.
- Carrier Frequency Offset (*CFO*): the unique offset between the designated and the actual carrier frequencies.

Attack Simulations. To generate multiple spoofing attacks, we utilized four distinct types of attacker platforms, each



MobiSys 2024

Poster: Hybrid Detection Mechanism for Spoofing Attacks in Bluetooth Low Energy Networks

Hanlin Cai[1,3], Yuchen Fang[1], Jiacheng Huang[1], Meng Yuan[2,3], Zhezhuang Xu[3*]
[1]National University of Ireland, Maynooth, [2]Chalmers University of Technology, [3]Fuzhou University
{hanlin.cai.2021, yuchen.fang.2021, jiacheng.huang.2022}@mumail.ie, {meng.yuan, zzxu}@fzu.edu.cn

ABSTRACT

As the foremost protocol for low-power communication, Bluetooth Low Energy (BLE) significantly impacts various aspects of our lives, including industry and healthcare. Given BLE's inherent security limitations and firmware vulnerabilities, spoofing attacks can readily compromise BLE devices and jeopardize privacy data. In this paper, we introduce ***BLEGuard***, a hybrid mechanism for detecting spoofing attacks in BLE networks. We established a physical Bluetooth system to conduct attack simulations and construct a substantial dataset (***BLE-SAD***). *BLEGuard* integrates pre-detection, reconstruction, and classification models to effectively identify spoofing activities, achieving an impressive preliminary accuracy of 99.01%, with a false alarm rate of 2.05% and an undetection rate of 0.36%.

CCS CONCEPTS

- **Security and privacy** → *Mobile and wireless security.*

KEYWORDS

Mobile Systems, Security and Privacy, Deep Learning

ACM Reference Format:
Hanlin Cai[1,3], Yuchen Fang[1], Jiacheng Huang[1], Meng Yuan[2,3], Zhezhuang Xu[3*] . 2024. Poster: Hybrid Detection Mechanism for Spoofing Attacks in Bluetooth Low Energy Networks. In *Proceedings of The 22nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '24).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Named after the Viking King Harald Bluetooth, Bluetooth is one of the most popular protocols for short-range wireless communications. The advent of the Bluetooth Low Energy (BLE) standard has further solidified its dominance in the era of IoT and 5G. By 2027, the deployment of BLE devices is anticipated to burgeon to 7.5 billion [4]. Despite their widespread adoption, these devices remain prone to spoofing attacks due to their limited I/O capabilities and lack of support for firmware upgrades. To combat these security threats, a device-neutral monitoring framework has been introduced, capitalizing on BLE's cyber-physical attributes to fortify defenses against spoofing attackers [5]. Furthermore, various research initiatives employ machine learning techniques to detect anomalous patterns within BLE network traffic. A proposed learning framework that amalgamates reconstruction and classification models promises to discern packets as either benign or malicious with remarkable precision [1]. However, the prevalent challenge lies in harmonizing accuracy, false positive rates, and resource utilization for detection, a triad that presents substantial obstacles to real-world application.

In this paper, we present *BLEGuard*, a hybrid detection mechanism based on cyber-physical analysis and deep learning techniques. *BLEGuard* is capable of pinpointing intricate spoofing attacks by integrating offline training with real-time analysis. **Our contributions are threefold:** (i) the compilation of *BLE-SAD*, a large-scale dataset encompassing in excess of 1.2 million packets, specifically curated for model evaluation, (ii) the conceptualization and empirical validation of *BLEGuard*, engineered to proficiently detect spoofing intrusions, (iii) the capacity for *BLEGuard* to seamlessly integrate within BLE networks, ensuring detection is accomplished without causing interference or taxing the network's resources.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MobiSys '24, June 03–07, 2024, Tokyo, Japan
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Figure 1: (a) Proposed BLE network testbed and (b) observed RSSI values during attack simulation.

2 SYSTEM DESIGN

2.1 Testbed Deployment

In this work, we built a physical network testbed within a typical noisy indoor office environment. Nine mainstream BLE devices, featuring a range of Bluetooth chips such as nRF52840 and DA14585, were deployed to establish our testbed, as depicted in **Fig. 1**. Besides, three network sniffers were deployed using Raspberry Pi equipped with BLE-Analyzer-PRO to monitor and capture network activity.

***BLE-SAD* Dataset**: To generate multiple spoofing attacks, we utilized four types of attacker platforms, each with three identical samples at different locations. In the spoofing attack scenario, the cyber-physical features of BLE network will undergo noticeable affected, resulting in significant deviations from the benign scenario. For instance, the anomalous shift in the Received Signal Strength Indicator (RSSI) of advertising packets indicates the presence of spoofing activities (**Fig. 1**). Currently, we have accumulated

Appendices 2-1: Related Publication for this Project