Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 9, No. 1, February, 2025

Invisible Scout: A Layer 2 Anomaly System for Detecting Rogue Access Point (RAP)

Diki Arisandi [1], Nazrul M. Ahmad [1*], Subarmaniam Kannan [1]

[1] *Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia.*

Abstract

Rogue Access Points (RAPs) pose a significant security threat by mimicking legitimate Wi-Fi networks and potentially compromising sensitive data. To address this issue, this research has proposed an innovative mechanism called Invisible Scout, which uses a multi-module system to identify RAPs. This study aimed to develop and validate a mechanism capable of accurately detecting RAPs in controlled setups, real-world environments, and under de-authentication attack scenarios. The proposed system consists of four key modules: sniffer, detection, probing, and comparison. To evaluate its effectiveness, tests were conducted in controlled and open environments and under de-authentication scenarios, using decision tree models and various metrics to assess performance. The decision tree model showed promising results in the controlled setup, achieving an Area Under the Curve (AUC) score of 0.921 and classification accuracy (CA) of 0.875, indicating that the model effectively distinguished between legitimate access points and RAPs. When tested in an open environment, the model's performance improved, achieving an AUC score of 0.952 and a CA of 0.994. Furthermore, under a de-authentication attack, the model achieved an AUC score of 0.955 and a CA of 0.996. To gain a deeper understanding of RAP behaviors, linear regression analysis was conducted, revealing patterns and visualizing the existence of RAPs, which could assist in further analysis. In conclusion, the results demonstrated that the proposed mechanism was highly effective in identifying RAPs. Future research should focus on refining the detection mechanism, incorporating real-time response capabilities, and expanding testing to diverse network scenarios.

Keywords:
Anomaly;
Beacon Frames;
Client-Side Scenario;
Invisible Scout;
Layer 2;
Rogue Access Point.

Article History:

Received:	15	September	2024
Revised:	17	December	2024
Accepted:	25	December	2024
Published:	01	February	2025

1- Introduction

Telecommunication is increasingly incorporated into various aspects of modern life and significantly contributes to growth in various sectors. Important advances include wireless fidelity (Wi-Fi) technology, which has become a cornerstone of connectivity. Additionally, Wi-Fi has revolutionized wireless networking and internet access, enabling seamless communication and information exchange across devices without the limitation of physical cables. This technology has transformed how individuals and organizations interact, facilitating flexible work environments, increased productivity, and widespread access to digital resources [1]. Generally, Wi-Fi was developed based on IEEE 802.11 standards, allowing users to connect to the internet from any location without a network cable. The latest 802.11g standard, in particular, has led to the general public's widespread adoption of Wi-Fi-enabled devices [2]. Since its introduction, Wi-Fi technology has advanced rapidly, providing increasingly faster wireless connectivity for internet applications and data transmission through wireless networks. These advancements have primarily used radio frequencies in 2.4 and 5 GHz bands, commonly used in wireless devices [3]. Many Wi-Fi-enabled devices, such as desktops, mobile phones, tablets, smartphones, and other appliances, are being developed in the current era [4].

* **CONTACT**: nazrul.muhaimin@mmu.edu.my

DOI: http://dx.doi.org/10.28991/ESJ-2025-09-01-016

Wi-Fi offers many benefits to daily lives by enabling connection to the network from anywhere in the AP (access point) service area, making it useful for everyone [5]. However, despite its conveniences, Wi-Fi still poses specific security risks. For instance, attackers can be between users and connection points, intercepting data sent by a user to an intended destination [6]. According to Figure 1, when this type of attack is executed, the attacker acts as a man-in-the-middle (MITM). In this position, they can access all information that users send online. This information may include sensitive data such as personal emails, credit card information, and security credentials for business networks [7].

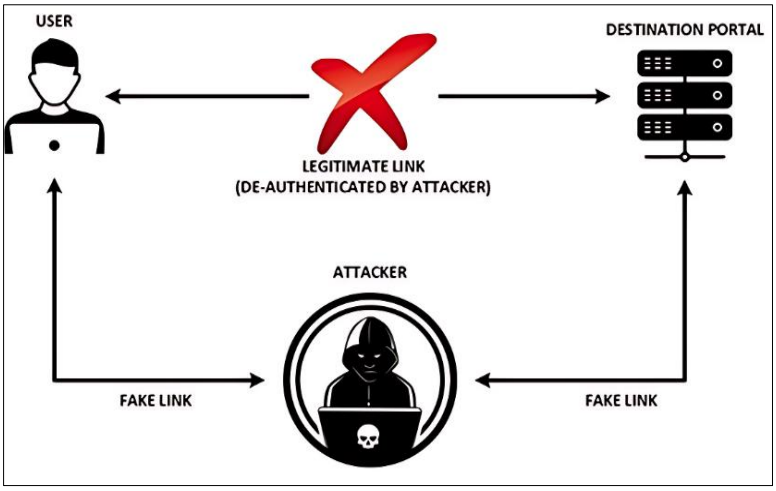


Figure 1. Man-In-the-Middle Scenario

A common malicious activity attackers use, similar to MITM attacks, is deploying a Rogue Access Point (RAP). Conceptually, RAP is a wireless AP installed on a secure network without explicit authorization from an administrator [8]. Also, it can be described as an AP created to enable attackers to establish an MITM attack. This allows attackers to intercept communication between active devices on the network, as shown in Figure 2 [9]. Traditionally, RAP has been identified using traditional detection methods. These methods typically rely on the identification of unique identifiers, such as MAC (media access control) addresses and SSID (service set identifier) [10].

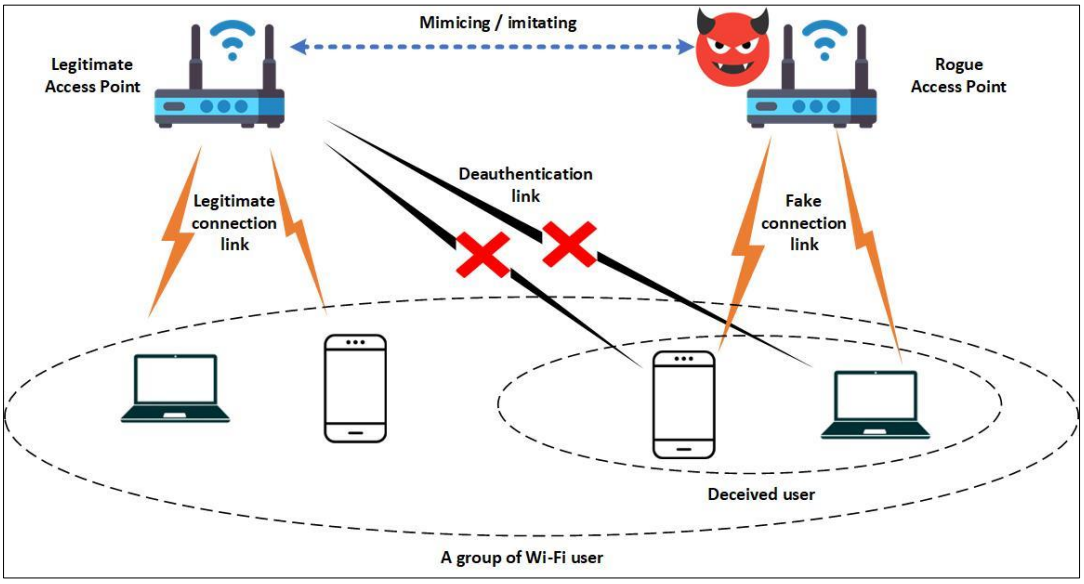


Figure 2. General Case of RAP

Since hacking tools have become more complex, it has become challenging to identify attacks from RAP using the traditional method. Traditional methods involve network administrators manually scanning the wireless environment using tools like Wi-Fi sniffers (e.g., Wireshark, NetStumbler, or Kismet) to detect and analyze all APs broadcasting in the area [11]. Another approach is for administrators to compare the discovered APs with a known list of authorized APs. APs that do not match the authorized list are flagged as potential RAPs [12]. Additionally, in some cases, RAPs may be discovered through manual inspection of the physical premises [13]. Traditional RAP detection techniques, including those based on MAC addresses and SSIDs, have become less effective as attackers adopt sophisticated methods to bypass these defenses.

While several studies have proposed RAP detection methods, many rely on complex network monitoring or require administrative control, making them unsuitable for regular users. Moreover, few approaches ensure user anonymity during detection, a critical requirement in modern network environments. Additionally, most detection methods focus on identifying RAPs but do not provide mechanisms to detect RAPs that have initiated de-authentication attacks. To address these gaps, this research proposes a lightweight, user-friendly tool, the Invisible Scout, that allows users to independently detect RAPs without needing administrative monitoring or additional hardware. Leveraging beacon frames from Wi-Fi APs, the proposed approach maintains user anonymity and secures networks against advanced RAP threats, including de-authentication attacks.

This research proposes a simple, user-friendly method for detecting and identifying RAP in Wi-Fi networks. In this context, the key features include 1) no requirement for network administration monitoring, 2) allowing Wi-Fi users to conduct detection independently, 3) not requiring connection to any AP for detection, 4) effectively securing the network from RAP, and 5) maintaining user identity confidentiality. It is essential to know that this solution significantly improves network security without compromising user convenience or privacy. Based on the proposed solution, the following contributions are made through this research.

- Since user anonymity or invisibility should be maintained, detection using layer 3 and above methods is not feasible. Therefore, the solution relies on a layer 2 method, leveraging beacon frames from each AP in the network.
- While MAC addresses and SSIDs no longer serve as the primary solutions in determining rogue or legitimate AP, other features in beacon frames are required for RAP detection.
- Administrators cannot monitor the network 24/7, and despite user invisibility, users are expected to detect RAP independently detect RAP.
- Detection signifies distinguishing whether the AP is legitimate or rogue and identifying the source of the RAP connection using existing network infrastructure or its connection. A framework that can accommodate both is required for this detection.
- In detecting RAP, users are not required to use additional hardware; only existing wireless interfaces, coupled with the lightweight prototype offered, namely invisible scout, are required.
- The invisible scout will be tested in three scenarios to determine its reliability. These include examining it in controlled environments, open spaces, and conditions where RAP has already initiated de-authentication against legitimate AP.

The remainder of the research is organized as follows. Section II primarily discusses related work, and Section III elaborates on the framework of the invisible scout and the features of layer 2 used in this exploration. Furthermore, Section IV describes the experimental design used, and Section V provides detailed results and discussions, including testing outcomes. Finally, Section VI explains the conclusion and reviews future research directions.

2- Literature Review

2-1- Introduction to Rogue Access Point (RAP)

A RAP is an unauthorized wireless access point connected to a network without the permission or knowledge of the network administrator. RAPs pose a significant security threat, as malicious actors can exploit them to intercept network traffic, steal sensitive information, or launch attacks on legitimate users [14]. With the growing reliance on wireless networks, particularly in enterprise environments, RAPs have become a major concern for organizations due to the vulnerabilities they introduce. In modern research, RAP detection has progressed beyond traditional manual methods, shifting towards automated anomaly detection, machine learning, and intrusion detection systems. These contemporary approaches focus on enhancing accuracy and scalability, especially in large and complex networks [15]. Research has explored various theoretical models to analyze beacon frames, monitor network traffic patterns, and detect anomalies indicating the presence of RAPs. The RAPs can be categorized based on their deployment and intent [16]:

- Internal RAP: This type of RAP is installed by an internal employee or user, intentionally or accidentally, without malicious intent. It often occurs when users set up wireless routers within an organization's network to bypass security policies. Even without malicious intent, internal RAPs weaken security protocols, exposing the network to potential attacks due to their lower security configurations.
- External RAP: Unlike internal RAPs, external RAPs are set up by attackers outside the organization, often to trick users into connecting to the RAP. These APs commonly mimic the SSID of legitimate networks (a tactic known as "Evil Twin" attacks). External RAPs are particularly dangerous as they can capture user data, including passwords and sensitive transactions, and enable man-in-the-middle (MITM) attacks or inject malware into the network.

- Compromised Authorized AP: This type of RAP starts as a legitimate AP but becomes compromised by an attacker. Often, attackers exploit weak security settings or default credentials to hijack the AP. Since it is a legitimate AP, compromised devices can be harder to detect. Once compromised, attackers can inject malicious traffic or use the AP as a launching point for further attacks on the network.
- Evil Twin AP: An Evil Twin AP is a type of RAP where an attacker sets up an access point with an identical SSID and security settings as a legitimate AP to deceive users into connecting to it. Once users connect, the attacker can intercept sensitive data, carry out MITM attacks, or direct users to malicious websites.
- Misconfigured AP: A misconfigured AP is a legitimate access point improperly configured due to human error or a software malfunction. This misconfiguration can classify it as RAP, as it weakens network security by using default passwords, weak encryption, or incorrect channel settings. Such APs expose the network to attacks and unauthorized access.
- Soft AP (Virtual AP): A Soft AP is created using software, such as a laptop or mobile device acting as a wireless hotspot. Although often used for convenience, these APs can become unauthorized RAPs if connected to a secure network without permission. Soft APs bypass network security controls, allowing unauthorized devices to connect and communicate within the network.

2-2- Previous Research

Many explorations have conducted research to identify RAP and address the threat it poses, while some have developed the bots as intrusion detection systems [9], but some explorations select node profiling based on predetermined profiles and criteria to distinguish between RAP and legitimate AP [17]. Concerning this, some explorations have identified RAP by detecting anomalies in recorded data packets, called packet auditing [18], providing answers. Previous research on RAP detection has been categorized in Table 1, showing that RAP identification using software is the most commonly used method due to the cost efficiency and the participation of other fields of knowledge. This method used existing hardware and additional software, while using dedicated hardware, though possible, was not often used in RAP detection [19].

In RAP detection, features from layers 1, 2, and 3 of the OSI model can be critical in identifying anomalies. Layer 1 (Physical Layer) provides features such as signal strength (RSSI) and transmission frequency, which can help detect unauthorized devices broadcasting from unusual locations or with atypical signal patterns. Layer 2 (Data Link Layer) includes features like MAC addresses, organizationally unique identifiers (OUIs), and frame types, which allow for identifying suspicious devices based on discrepancies in hardware addresses or unusual packet behavior. Layer 3 (Network Layer) focuses on IP addresses and routing information, where unauthorized access points might exhibit abnormal IP configurations or inconsistencies in routing protocols [20].

The first classification of RAP identification is using node profiling. Node profiling involves creating predefined profiles or criteria for legitimate APs in the network. These profiles could include attributes such as signal strength, MAC address, packet tracing, or other possible features. The system can distinguish between legitimate and rogue by comparing live data from APs against these profiles. Research from Jain et al. [21] highlights the threat of Evil Twin (ET) RAPs infecting Android devices before data transmission. ETGuard, a proposed real-time detection system, uses beacon frame fingerprints to identify ETs pre-association and sends de-authentication frames to block client connections. Tested in 12 scenarios, ETGuard showed high detection accuracy.

Meanwhile, Hsu et al. [22] addressed detecting rogue APs using 3G/4G connections, which evade traditional detection methods. RAP Finder (RAF) relies on reverse traceroute data for detection, eliminating the need for special hardware. The study in VanSickle et al. [23] demonstrates the ease with which malicious actors can set up RAP using tools like Aircrack-ng, Kismet, and inSSIDer. These tools effectively detect RAP through network monitoring. Similarly, Bodhe et al. [24] proposed a neuron-fuzzy method combining neural networks and fuzzy logic to secure wireless sensor networks in real-time. In Jang et al. [25], PrAP-Hunter detects hardware-based RAPs (PrAPs) by introducing intentional channel interference, achieving near-perfect accuracy.

The other study from Hsu et al. [26] proposes Legal Access Point Finder (LAF) for Wi-Fi users, a passive solution that identifies legitimate APs by analyzing TCP packet forwarding without active probing. Shrivastava et al. [27] introduced EvilScout, using Software-Defined Networking (SDN) to detect RAPs based on IP-prefix analysis, showing high accuracy in real-world tests. Similarly, Lu et al. [28] proposed BiRe, which detects Evil Twin attacks by monitoring TCP SYN-ACK packets, achieving 100% detection accuracy. Research from Agyemang et al. [9] targets WiFi-enabled

IoT devices, proposing a lightweight, real-time algorithm based on information-theoretic principles to distinguish between legitimate and RAP. At the same time, Igarashi et al. [29] improved detection in unstable traffic environments using Address Resolution Protocol (ARP) failures. Lastly, Bello & Kanu [30] exposed vulnerabilities in GSM networks, such as base station spoofing and IMSI catching, using open-source tools to demonstrate the ease of attacks due to one-way authentication.

The second classification, packet auditing, involves analyzing and monitoring data packets transmitted across the network to detect anomalies, such as unusual packet headers, unexpected traffic flows, or irregular transmission patterns, which could indicate the presence of a RAP. Wi-Fi networks are particularly vulnerable to impersonation attacks from RAPs mimicking legitimate devices' SSIDs and MAC/IP addresses. To counter these threats, researchers propose various detection methods. For instance, Liu et al. [31] leveraged channel state information (CSI) to identify rogue devices based on non-linear phase errors, achieving 96% accuracy and a false alarm rate below 2%. Meanwhile, Lu et al. [32] addressed evil twin attacks (ETAs) using Special Length Frames Arrival Time (SLFAT), a client-side method that detects ETAs by monitoring frame arrival times.

In mobile scenarios, Kitisriworapan et al. [33] suggested using round-trip time (RTT) and transmission rates to detect RAPs, yielding an F-measure of 0.9. Sankhe et al. [34] reduced Wi-Fi latency and improved spectrum efficiency with CSI scan, which embeds discovery information in regular AP transmissions. For passive device identification, Delgado et al. [35] introduced the IRID framework, using machine learning to distinguish devices with over 99% accuracy. For practical RAP detection, Korolkov & Kutsak [36] used RSSI-based cluster analysis and trilateration, localizing RAPs with a 1.5-meter error margin. To address privacy concerns, Alyami et al. [17] demonstrated how encrypted Wi-Fi traffic can reveal IoT device information with 95% accuracy. Lu et al. [37] proposed PEDR, which improves detection accuracy by mitigating phase error drift in CSI.

Machine learning is also employed by Kim et al. [38] to analyze RTT values, where the Decision Tree classifier showed the highest accuracy. In IoT environments, Yang et al. [39] used DL-PEDR, a deep-learning approach, to achieve a 96.6% RAP detection rate. Additionally, Liu & Papadimitratos [40] leveraged Wi-Fi positioning to detect RAPs by identifying inconsistent RSSI measurements. For enhanced security, Jing et al. [41] combined zero-trust architectures with radio frequency fingerprint (RFF) authentication, achieving 99% accuracy. Finally, Zhang et al. [42] presented a deep-learning RFF framework for IoT device authentication, with a 90.23% RAP detection rate and reduced time overhead by 58%.

The last classification is bot setup. In this context, "bots" refer to software agents that are part of an intrusion detection system (IDS) designed to monitor network traffic for malicious activities like RAPs. These bots automatically analyze network behaviors and flag suspicious patterns that indicate the presence of a RAP. The bots continuously scan the network for threats and can respond in real-time by alerting administrators or taking actions to mitigate the threat. Typically, these bots operate in a distributed or centralized manner, depending on the network configuration.

The vulnerability of Wi-Fi networks to RAP attacks, including evil twin attacks where attackers spoof legitimate access points to deceive clients, is a significant concern. To address this, Jain et al. [21] proposed a Discrete Event System (DES)-based Intrusion Detection System (IDS), which provides a scalable and cost-effective solution for detecting these attacks. In response to the challenge of covering multiple locations, Hsu et al. [22] introduced a UAV-based detection system that leverages the high mobility of drones combined with software-defined radio (SDR) to enhance detection efficiency and coverage. Additionally, White & Sjelin [43] tackled the issue of rogue software updates in JavaScript packages, where malicious code is concealed within legitimate updates, by presenting RogueOne—a system that utilizes differential data-flow analysis and abstract interpretation to detect these updates with up to seven times greater accuracy than other systems, reducing false positives significantly.

The various methods for detecting RAP in wireless networks signified server-side and client-side detection methods. Layer 2 methods, typically MAC-based, often enabled anonymous detection. Moreover, several advanced methods have been proposed, including SLFAT, which uses machine learning to identify malicious packet arrivals, and PrAP-Hunter, which interferes with AP communication to detect RAP with high accuracy. Other important methods included combining fuzzy logic with neural networks, leveraging SDN capabilities with EvilScout, and using lightweight algorithms for IoT devices. Additionally, research showed the potential of received-signal-strength-based localization and ARP failure detection under MAC address duplication. Despite some methods requiring specialized hardware, many innovative solutions, such as IRID for secured device identification and profiling IoT devices through encrypted Wi-Fi traffic, showed high detection accuracy and efficacy in various scenarios.

Table 1. Comparison of Previous Work

No.	Research	Actor	Anonymous Probing	Dedicated Hardware	Classification	Aim Features	Detection
1	Jain et al. [21]	Client	No	No	Node profiling	Layer 2	Immediate
2	Liu et al. [31]	Server	Yes	No	Packet auditing	Layer 1	Data-oriented
3	Hsu et al. [22]	Client	No	No	Node profiling	Layer 3	Immediate
4	Lu et al. [32]	Client	Yes	No	Packet auditing	Layer 2	Data-oriented
5	Selvarathinam et al. [44]	Server	No	No	Bots setup	Layer 2	Data-oriented
6	Wang et al. [45]	Server	Yes	Yes	Bots setup	Layer 1	Immediate
7	VanSickle et al. [23]	Client	Yes	No	Node profiling	Layer 2	Immediate
8	Kitisriworapan et al. [33]	Client	No	No	Packet auditing	Layer 3	Data-oriented
9	Bodhe et al. [24]	Client	Yes	No	Node profiling	Layer 2	Data-oriented
10	Sankhe et al. [34]	Server	Yes	No	Packet auditing	Layer 1	Data-oriented
11	Jang et al. [25]	Client	Yes	Yes	Node profiling	Layer 2	Immediate
12	Hsu et al. [26]	Client	No	No	Node profiling	Layer 3	Immediate
13	Delgado et al. [35]	Server	Yes	Yes	Packet auditing	Layer 2	Data-oriented
14	Shrivastava et al. [27]	Client	Yes	No	Node profiling	Layer 2	Immediate
15	Lu et al. [28]	Client	No	No	Node profiling	Layer 2	Immediate
16	Agyemang et al. [9]	Client	Yes	No	Node profiling	Layer 2	Immediate
17	Igarashi et al. [29]	Client	Yes	No	Node profiling	Layer 2	Immediate
18	Korolkov & Kutsak [36]	Client	Yes	No	Packet auditing	Layer 2	Data-oriented
19	Alyami et al. [17]	Client	Yes	Yes	Packet auditing	Layer 2	Data-oriented
20	Lu et al. [37]	Client	Yes	No	Packet auditing	Layer 1	Data-oriented
21	Kim et al. [38]	Client	No	No	Packet auditing	Layer 3	Data-oriented
22	Yang et al. [39]	Client	Yes	No	Packet auditing	Layer 1	Data-oriented
23	Bello and Kanu [30]	Client	Yes	Yes	Node profiling	Layer 1	Immediate
24	Liu & Papadimitratos [40]	Client	Yes	No	Packet auditing	Layer 1	Data-oriented
26	Sofaer et al. [46]	Server	No	No	Bot setup	Layer 3	Immediate
27	Jing et al. [41]	Server	No	No	Packet auditing	Layer 1	Data-oriented
28	Zhang et al. [42]	Client	Yes	Yes	Packet auditing	Layer 1	Data-oriented

3- Proposed Methods

This research leveraged the National Institute of Standards and Technology (NIST) Cybersecurity Framework due to the comprehensive method to identify, protect, detect, respond to, and recover from various cybersecurity threats [47]. Given the complexity of RAP and other cybersecurity risks, this systematic method thoroughly addressed all critical factors [43]. Additionally, the NIST framework was widely recognized as an industry-standard in public and private sectors, facilitating risk management and supporting best practices. NIST provided valuable guidance for implementing cybersecurity measures aimed explicitly at RAP identification and mitigation [48].

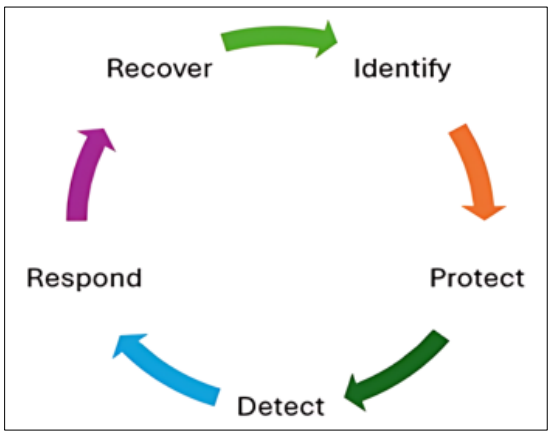


Figure 3. NIST Framework

In the identify phase, a new strategy to uncover the presence of RAP was essential. This phase used software-based identification to maintain efficiency when probing surrounding APs, eliminating the specialized hardware requirement. Additionally, an anonymous method using anomaly detection was preferred to identify irregularities. Anonymous probing ensured that users or clients were not required to pair with any AP on the network. Moreover, the client-side scenario offered several advantages over relying solely on network administrators to identify RAP. Public spaces, such as the MMU Hostel, the Melaka Sentral bus station, the postgraduate (PG) corridor, and KLIA1 airport, were used for data collection. These locations were selected due to their ease of access and high probability of being detected by a wireless device as having a significant amount of APs. The goals were to assess the complexity of the wireless network environment and locate any possible multiple APs.

In the protect phase, anonymous probing was selected to improve client security for several important reasons, which included preventing RAP from being alerted, which was crucial as RAP, often set up by attackers, monitored network traffic and retaliated when probing was detected. In this context, anonymous probing also protected the personal identification of clients by preventing the exposure of device-specific identifiers that malicious actors could use to track and aim clients. The probing ensured client anonymity, reducing the risk of being aimed at by RAP. Furthermore, anonymous probing allowed clients to gather information about nearby APs without direct interaction, enabling a safer evaluation of potential security threats without drawing attention.

Client participation in detecting RAP improved security by leveraging the ability to spot unusual Wi-Fi networks, providing crucial information that might otherwise go undetected. Consequently, customers could promptly notify network administrators of unauthorized AP, enabling swift threat mitigation. Clients' mobility offered a broader detection range, uncovering unlawful networks in areas administrators might miss. Additionally, the customer's diverse device connections provided valuable data for developing comprehensive security measures. This method also increased client awareness about security risks and promoted best practices for safe Wi-Fi usage. Beyond the client-side scenario, the software-based method required a new perspective. Unlike previous explorations in Table 1, this proposed method combined node profiling and packet auditing. Node profiling compared legitimate AP profiles stored in the database with suspected RAP, while packet auditing determined anomalies or differences in the beacon frames. This method aimed to help clients anonymously and passively distinguish between legitimate AP and RAP profiles based on differences in the beacon frames obtained.

Three major categories of features were used to detect anomalies in beacon frames and identify RAP, including layer 1, 2, and 3 features, as shown in Figure 4. Layer 3 features, including Inter-Arrival Time (IAT), Round-Trip Time (RTT), IP address, TCP handshake, traffic analysis/Quality of Service (QoS), and encrypted packet streams, were not feasible for anonymous and passive RAP detection. Moreover, this feasibility was due to the features being only accessible until the client was associated with AP, compromising the customer's identity.

Layer 1 features, such as radio signals and Received Signal Strength Indicator (RSSI) elements, enabled anonymous and passive RAP identification. Previous explorations used RSSI and radio signals to locate AP and identify legitimate or rogue features. However, despite the advantage of maintaining client confidentiality, using RSSI values for distance mapping had limitations due to susceptibility to noise, multipath fading, and interference, leading to significant fluctuations in received signal strength.

The research focused on layer 2 features, which operated at the Data Link layer of the OSI model and could be obtained through anonymous and non-anonymous probing. IAT, SSID, RTT, and security information were obtained through prominent or active probing. Additionally, some other layer 2 features could not be easily imitated and might show the identity of AP. These features included retry bit, IBSS (Independent Basic Service Set) status, Addresses 1-4, and Organizationally Unique Identifier (OUI) number/manufacturer ID. Moreover, this research used the features as essential information to identify RAP.

The retry bit contained an essential clue for identifying RAP attacks, and every frame transmitted for the first time had this bit set to 0. When the frame was lost (no acknowledgment was received or de-authenticated), it was retransmitted with the bit set to 1. Furthermore, IBSS status provided information concerning the connection used by RAP when launching an attack, using a private connection such as a modem or existing infrastructure.

Address 1-4 provided important information for RAP detection, with 1 containing the client MAC address, 2 being BSSID, 3 containing the MAC address of AP, and 4 being unused. In the case of RAP, the attacker cloned the MAC address of the genuine AP, enabling the address 2 and 3 fields to be easily manipulated. It should be acknowledged that some RAP cases used address 4 while retransmitting the packet. OUI, or manufacturing ID, was also used to detect the presence of RAP, as OUI was a manufacturing code that RAP could not modify. Moreover, there were differences in OUI value between a legitimate AP and RAP because it was distinctive.

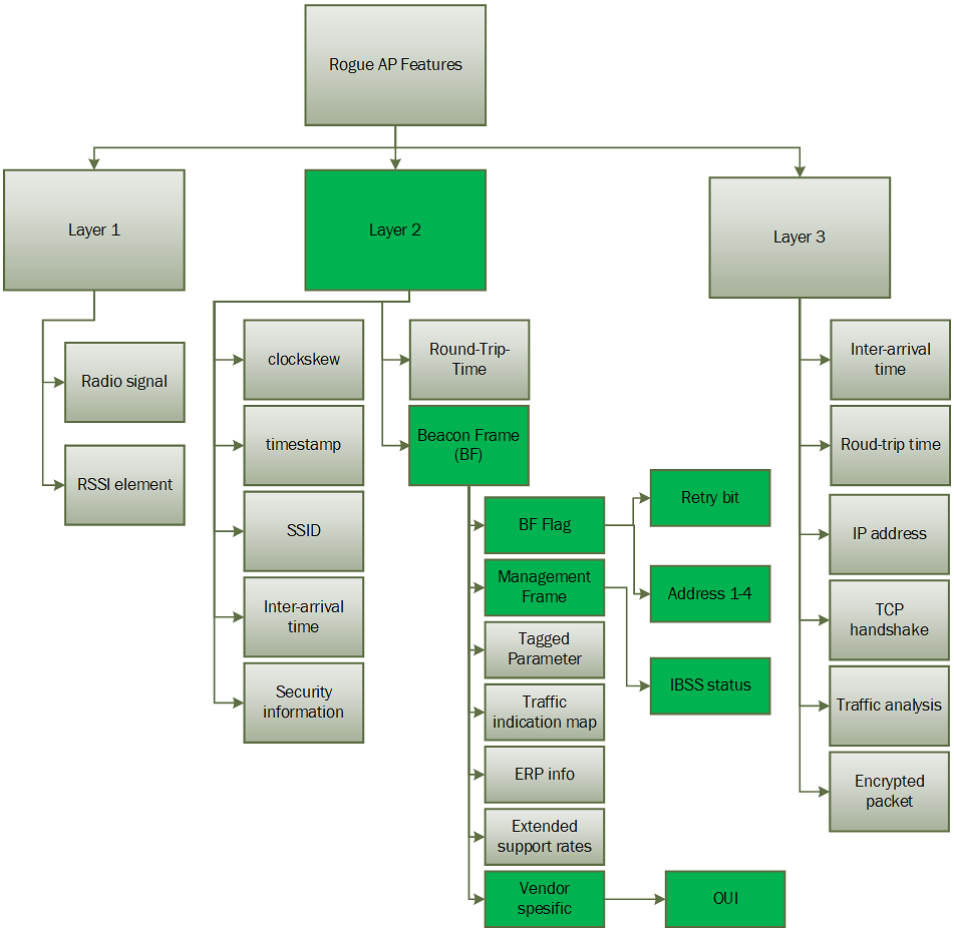


Figure 4. Preferred Features (Green Boxes)

The rogue detection phase using beacon frame anomalies consisted of two phases. The first phase was a legitimacy check, which aimed to determine whether AP was rogue or legitimate, and the second phase was to identify the connection used by RAP. Specifically, initial investigation showed that many RAPs used existing network connections. However, there were also a small number of RAPs providing individual connections with more limited resources to access victims' data.

The proposed algorithm for detecting RAP consisted of two stages, as shown in Figure 5. The client conducted all processes anonymously to avoid association with any AP. The first stage was to check the legitimacy of AP using monitor mode, and the process commenced with probing requests. When there is no response after five attempts, AP might be rogue. However, when AP responded, the frame response was captured, and features such as Addresses 1-4, OUI number, retry bit, and IBSS status were extracted and saved. Another request was then sent to the same network to verify AP. When the features from both checks matched, AP was legitimate; otherwise, it was rogue.

In the second stage of the detection process, the connection source of RAP was determined. This determination included checking two parameters, namely retry bit and IBSS status. Specifically, when the retry bit was set to 1, and IBSS status showed ESS, it implied that RAP was using existing infrastructure. Meanwhile, when the retry bit was 0, and the IBSS status was set to independent/IBSS, it showed that RAP was operating on a private connection.

To validate the effectiveness of the proposed identification algorithm for detecting RAP through anomaly detection in beacon frames, examinations were conducted under various conditions in three scenarios: controlled setup, real environments, and a de-authentication attack as part of the response and recovery phase. Additionally, these environments featured multiple legitimate APs and one or more RAPs. Wi-Fi adapters capable of capturing and analyzing beacon frames were used. The primary inputs for the examination were beacon frames transmitted by both rogue and legitimate APs, collected continuously in the testing period. Furthermore, preprocessing of the collected data included extracting relevant features from beacon frames, including address fields (Addresses 1-4), OUI numbers, retry bits, and IBSS status.

Anomaly detection algorithms in Invisible Scout were applied to the features using Scapy and Python to identify anomalies indicative of RAP. The method distinguished between rogue and legitimate by comparing the detected anomalies from the initial and subsequent responses. Moreover, performance evaluation included visualizing the results to show the method's effectiveness in identifying RAP. The acceptance criteria focused on accurately identifying RAP with minimal false positives. In addition, acknowledging the potential influence of environmental factors on testing outcomes, iterative testing was performed to refine the algorithm and ensure the toughness across various conditions.

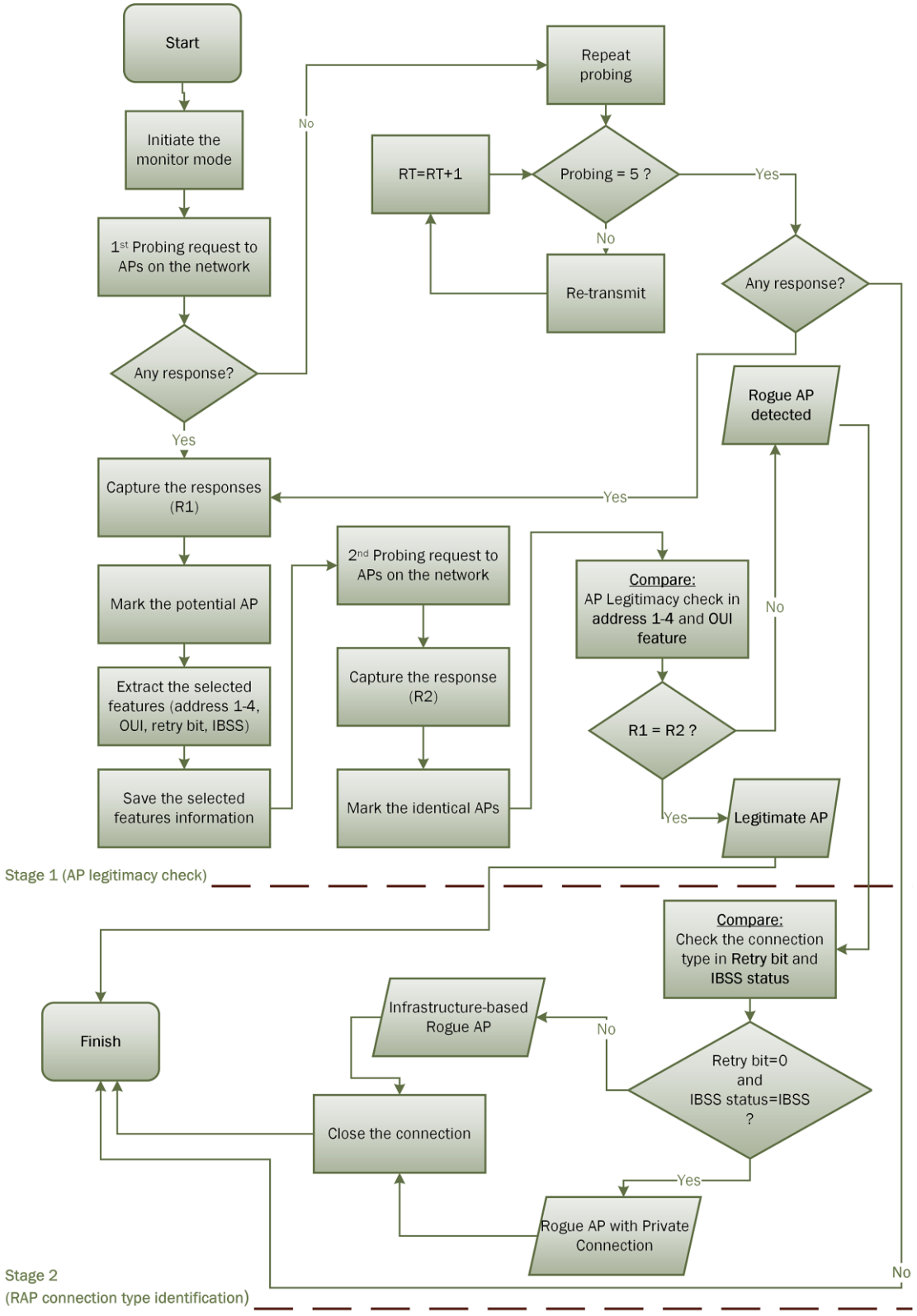


Figure 5. The Whole Anomaly Identification Algorithm for Invisible Scout

4- Result and Discussion

4-1- System Design

The proposed mechanism of the invisible scout is shown in Figure 6. The Invisible Scout system for RAP identification uses a modular approach with four key components: Sniffer, Detection, Probing, and Comparison, all working together to ensure accurate RAP detection. The system begins with the Sniffer module, which passively monitors the network by capturing beacon frames from both legitimate and rogue sources. It gathers critical features

such as Address 1-4, OUI Number, Retry Bit, and IBSS Status from any AP within its range. Once the information is collected, it is forwarded to the Detection and Comparison modules for further analysis. The sniffer's continuous monitoring ensures that all APs, whether legitimate or suspicious, are detected.

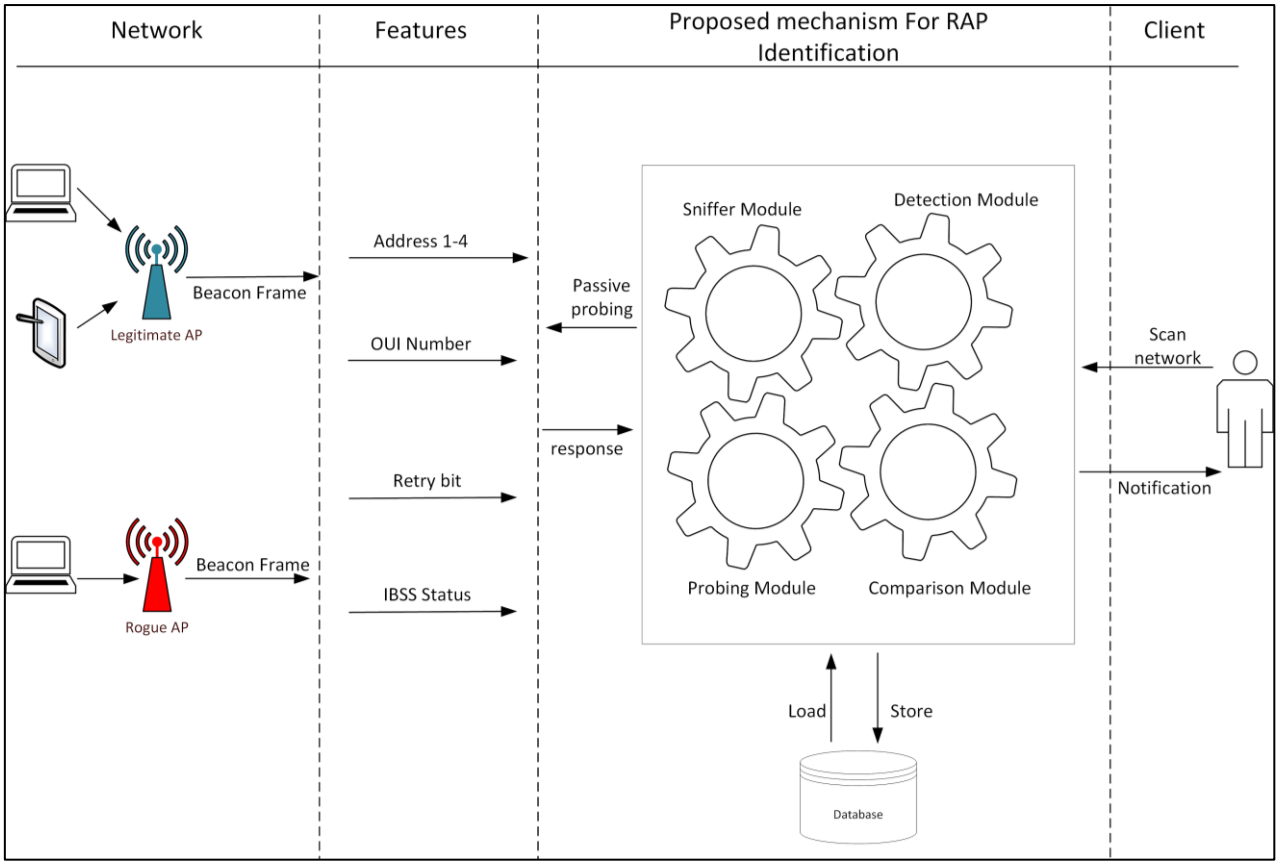


Figure 6. The Proposed Mechanism for Invisible Scout

The Detection module processes the data received from the sniffer and analyzes the extracted features to assess the authenticity of the APs. By comparing the AP's characteristics with a database of known legitimate APs, the Detection module can identify potentially suspicious access points that exhibit signs of rogue behavior. If an AP is flagged as suspicious, the Detection module triggers further verification by passing the data to the Comparison module for more thorough inspection. The Comparison module plays a critical role by validating the Detection module's findings. It compares the suspicious AP's features and those of known legitimate APs stored in the database. RAPs often manipulate specific features to imitate legitimate APs, but if the AP fails this comparison, it is flagged as a RAP. The Comparison module then informs the Probing module to conduct additional active verification.

The Probing module initiates active measures to confirm the legitimacy of flagged APs by sending probe requests to the suspected RAPs. Legitimate APs typically respond predictably, whereas RAPs may exhibit anomalies in their responses. The Probing module gathers these response patterns and shares the results with the Detection and Comparison modules, further refining the RAP identification process. Once the entire process is complete, the results are stored in a central database and accessible to the client. Clients can scan the network anytime and receive notifications regarding potential rogue APs. This interactive system ensures that users remain constantly informed of any network threats.

Based on the proposed mechanism, the process of identifying RAP by invisible scout is shown in Figure 7. The process commenced with the "Get Response" step, which included collecting data or responses from the network. Furthermore, this data was later used in the "AP Detection" and "AP Comparison" steps, where AP was detected and compared against known or legitimate AP. The "Scan Network" step was also included, showing that the network was actively scanned for AP. Probing was used to analyze the surrounding AP in the network, and sniffing was performed to gather more information about AP. Moreover, the collected data was then stored, as shown by the "Store Data" step. The final steps included obtaining notifications, viewing RAP, and legitimate AP, which were presented to the client. This process allowed the client to be informed about the presence of RAP and perform necessary actions.

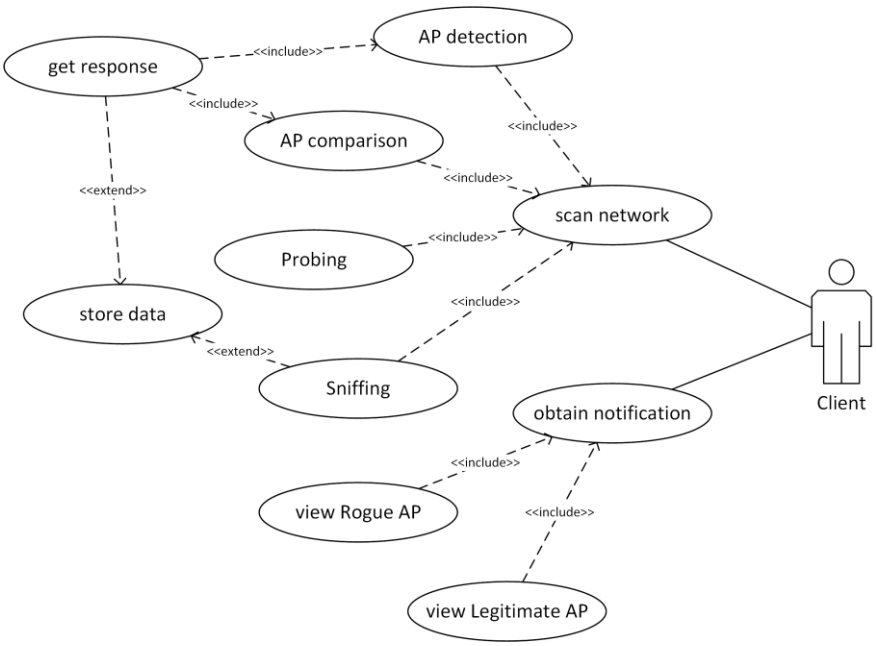


Figure 7. The Identification Process of RAP for Invisible Scout

4-2- Testing Scenario

This research included examining three cases. The first case involved a controlled setting in the postgraduate research lab at the Faculty of Information Science and Technology, Multimedia University. The second case involved a real-world setting in several public areas, including a university area. The third case also involved a real-world setting in a public area but with the addition of a de-authentication attack on several randomly selected legitimate APs. This attack may forcibly disconnect clients from an AP by sending fraudulent de-authentication frames, as shown in Figure 12 [49]. Within this framework, the controlled environment allowed for precise control and detailed analysis, as shown in Figure 8. Meanwhile, the real-world environment provided an understanding of how the phenomenon behaved outside controlled conditions. In the controlled setup, a RAP machine, a legitimate AP, and two clients (one surveillance and one regular) were used, as seen in Figure 8. Specifically, the RAP machine simulated an unauthorized AP, the legitimate AP represented an authorized Wi-Fi network, and the clients attempted to connect to any available network, with the surveillance client monitoring network activity and security vulnerabilities.

Figure 8. Controlled Setup

In the second and third experiments, conducted in an open environment, several locations were selected, including the postgraduate (PG) corridor, MMU Hostel, Melaka Sentral bus station, and KLIA1 airport. These locations were selected for their accessibility and the likelihood of discovering a substantial number of APs using a wireless device. By experimenting in these areas, the aim was to assess the complexity of the wireless network environment and the potential presence of multiple APs. The Free Space Path Loss (FSPL) Model was used to estimate the effective distance between the AP and the user for effective distance in all testing scenarios.

$$FSPL = 20log_{10}(d) + 20log_{10}(f) + 20log_{10}\left(\frac{4\pi}{c}\right) \quad (1)$$

FSPL was measured in decibels (*dB*), and variable *d* represented the distance between AP and the user in meters (*m*). *f* was the signal frequency in Hertz (*Hz*), and the speed of light (*c*) was approximately 3×10^8 meters per second (*m/s*).

Using Equation 1, detecting more than ten wireless devices broadcasting beacon frames during the initial scanning at each location, as shown in Figure 9, using a Wi-Fi analyzer, showed a dense network environment with various APs competing for connections. However, using monitor mode to detect wireless devices and beacon frames showed more than 30 AP at each location from both 2.4 GHz and 5 GHz frequencies.

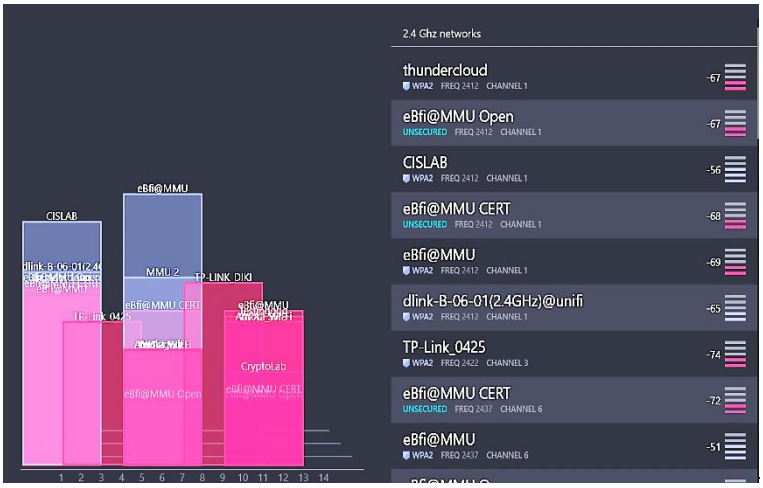


Figure 9. Initial Wi-Fi Scanning for Open Environment

RAP was initialized with desired settings by developing and configuring the hostapd module, which managed RAP and was set up with parameters such as SSID, channel, and encryption settings. Subsequently, after initialization, RAP was broadcasted by running the hostapd module as a service or background process, continuously sending beacon frames with AP information. After clients connected, the module handled the authentication and association, and when the MITM module was enabled, it performed attacks such as ARP spoofing on connected clients. Additionally, a function created beacon frames containing fields such as SSID, BSSID, channel, and encryption details, reviewing these steps in the pseudocode provided in algorithm 1.

Algorithm 1. Deploying RAP and Perform Broadcasting

```
# Start RAP and begin broadcasting
function start_rogue_ap()
    run Hostapd as a service or background process
while Hostapd is running
    beacon_frame = create_beacon_frame
    broadcast beacon frame to network
    if there are clients connected
        handle_client_connection
    end if
    wait for some time
end while
end function

# Create a beacon frame with the AP information
function create_beacon_frame(ap_info)
    frame = new BeaconFrame()
    frame.setSSID(ap_info.ssid)
    frame.setMACAddress(ap_info.mac_address)
    frame.setChannel(ap_info.channel)
    frame.setEncryption(ap_info.encryption)
    frame.setTimestamp(current_time)
    frame.setInterval(beacon_interval)
    // Add other necessary beacon frame fields
    return frame
end function

# Clean up and stop the rogue AP and MITM module
function stop_rogue_ap()
    stop Hostapd service
end function
```

Some clients, acting as invisible scouts, conducted probing activities to analyze the surrounding network area. The image in Figure 10 was a simple diagram showing the probe request and response process in computer networking. After performing RAP identification and probing activities, the next step included sniffing to gather information about any AP on the network, legitimate or rogue. The pseudocode provided in Algorithm 2 started by setting up a sniffer environment and importing necessary libraries such as scapy for packet sniffing and mysql.connector for database connectivity. Moreover, immediately after the required modules were imported, the code initialized a connection to the MySQL database, which stored information about anomalous beacon frames detected during the surveillance process by the client.

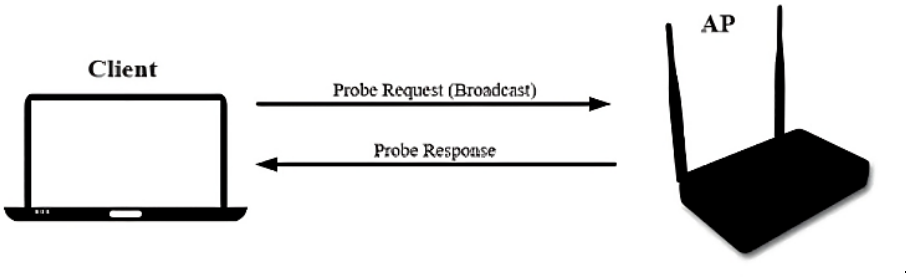


Figure 10. Probing Process Diagram

Algorithm 2 defined four functions to detect anomalies in different fields of beacon frames. These functions included 'check_address_fields' verified when Address 1-4 fields conformed to standards, 'check_oui' extracted OUI from BSSID and checked when it was approved, 'check_retry_bit' ensured retry bit in Frame Control field was set correctly, and 'check_ibss_status' checked IBSS status in the beacon frame. Additionally, the main sniffing loop continuously captured beacon frames using Scapy, running the four anomaly detection functions on each frame and flagging any that failed the checks as anomalous. These anomalous frames were stored in the MySQL database for further analysis. Relating to this process, the pseudocode also included a section for reporting and analysis, allowing for querying, analyzing, and reporting on the stored anomalous beacon frames to identify the presence of RAP on the network.

Algorithm 2. Sniffing, Extracting, and Analyzing Beacon Frames

```
# Set up the sniffer
Import libraries (scapy, mysql.connector)
Initialize MySQL connection

# Define anomaly detection functions
function check_address_fields(beacon_frame):
    Check if Address 1-4 fields are valid and conform to standards
    return True if valid, False otherwise

function check_oui(beacon_frame):
    Extract OUI from BSSID
    Check if OUI is on the approved list
    return True if approved, False otherwise

function check_retry_bit(beacon_frame):
    Check if the Retry bit in the Frame Control field is set incorrectly
    return True if set correctly, False otherwise
function check_ibss_status(beacon_frame):
    Check if the IBSS status is set correctly
    return True if set correctly, False otherwise

# Main sniffing and analysis loop
while True:
    Sniff for beacon frames using scapy
    for each beacon frame:
        if not check_address_fields(beacon_frame):
            Flag as anomalous
        if not check_oui(beacon_frame):
            Flag as anomalous
        if not check_retry_bit(beacon_frame):
            Flag as anomalous
        if not check_ibss_status(beacon_frame):
            Flag as anomalous
    Store in MySQL database

# Reporting and analysis
Query MySQL database for anomalous beacon frames
    Analyze and report on detected anomalies
```

As the character of RAP was to copy a legitimate AP, the code in Algorithm 3 served as a configuration file for the hostapd module, a user-space daemon software for wireless AP. For research purposes, this code was used to replicate the behavior of RAP from a legitimate AP. Furthermore, the code sets the parameters for creating a Wi-Fi AP using a specified interface (wlan0) and driver (nl80211). The operation mode was set to 'a' for AP mode, showing that the device functioned as an AP. The mode further specified the channel (channel=36) on which to operate, determining the frequency band for wireless communication.

SSID was the Wi-Fi network's name broadcasted to users' devices and was set to "MMU2" in this example. Additionally, the configuration included settings for enabling the 802.11n standard (ieee80211n=1) to provide faster Wi-Fi speeds. Security settings were also configured, specifying a WPA/WPA2 passphrase (wpa_passphrase= MySecurePassword) to encrypt the network traffic. The authentication algorithm (auth_algs) was set to 1, showing the use of WPA authentication, while encryption protocols (wpa, wpa_key_mgmt, rsn_pairwise) were defined to ensure secured communication between the AP and connected devices.

Algorithm 3. A RAP Mimicking Legitimate AP

```
# Set the interface to use
interface=wlan0

# Set the driver to use
driver=nl80211

# Set the operation mode (a = AP mode)
hw_mode=a

# Set the channel to use
channel=36
# Set the SSID (Wi-Fi network name)
ssid=MMU2

# Enable 802.11n
ieee80211n=1

# Set the WPA/WPA2 passphrase
wpa_passphrase=MySecurePassword

# Use WPA authentication
auth_algs=1
# Specify encryption/authentication
wpa=2
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
```

When a client attempted to connect to an available AP through Wi-Fi, the customer would first enable a Wi-Fi adapter. However, this process became vulnerable in the presence of RAP among legitimate APs in the network. Attackers intentionally deployed RAPs to confuse clients, often leading customers to associate with the wrong AP, including RAPs with stronger signals. Following the discussion, this association unwittingly granted attackers access to sensitive information or credentials. Multiple SSIDs labelled 'MMU 2' were visible, deliberately acting as RAP (painted in yellow), as shown in Figure 11. Clients were deceived by the perceived benefits of RAP, such as improved signal strength and lack of authentication requirements. However, unknown to the client, associating with RAP exposed all transmitted data, potentially compromising valuable or confidential information to malicious interception.

We developed based on the second scenario for the third scenario but added a de-authentication attack, which we launched on random legitimate APs, as shown in Figure 12. In this scenario, we allowed the RAP to perform its attack to deceive clients, with the goal that the invisible scout could distinguish between the legitimate AP and the RAP, even after the RAP had attacked the legitimate AP. Based on the algorithm 4 for the de-authentication attack, The interface refers to the network interface used for sending packets, typically a wireless interface like 'wlan0'. The target_access_point_mac is the MAC address of the access point we intend to disrupt, while the target_client_mac is the client's MAC address connected to this access point. Packet_count denotes the number of de-authentication packets to be sent. The send_deauth_packets function is responsible for repeatedly sending these de-authentication packets. The deauthentication_frame function creates a de-authentication frame with the necessary parameters, such as the destination, source, BSSID, and reason code, to facilitate the attack.

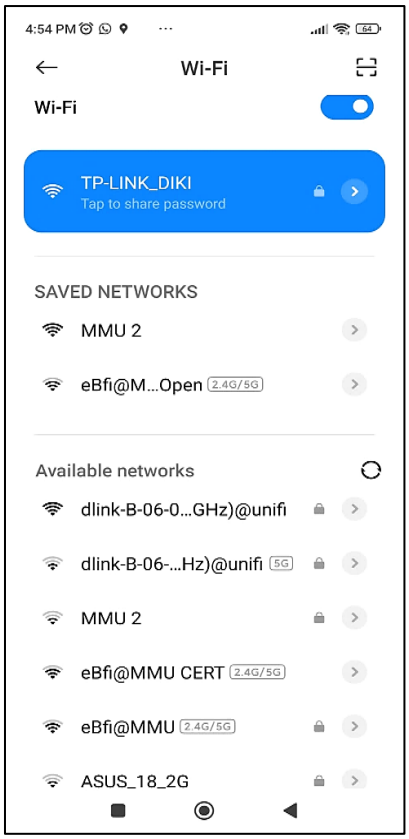


Figure 11. RAP Setup Seen By Client

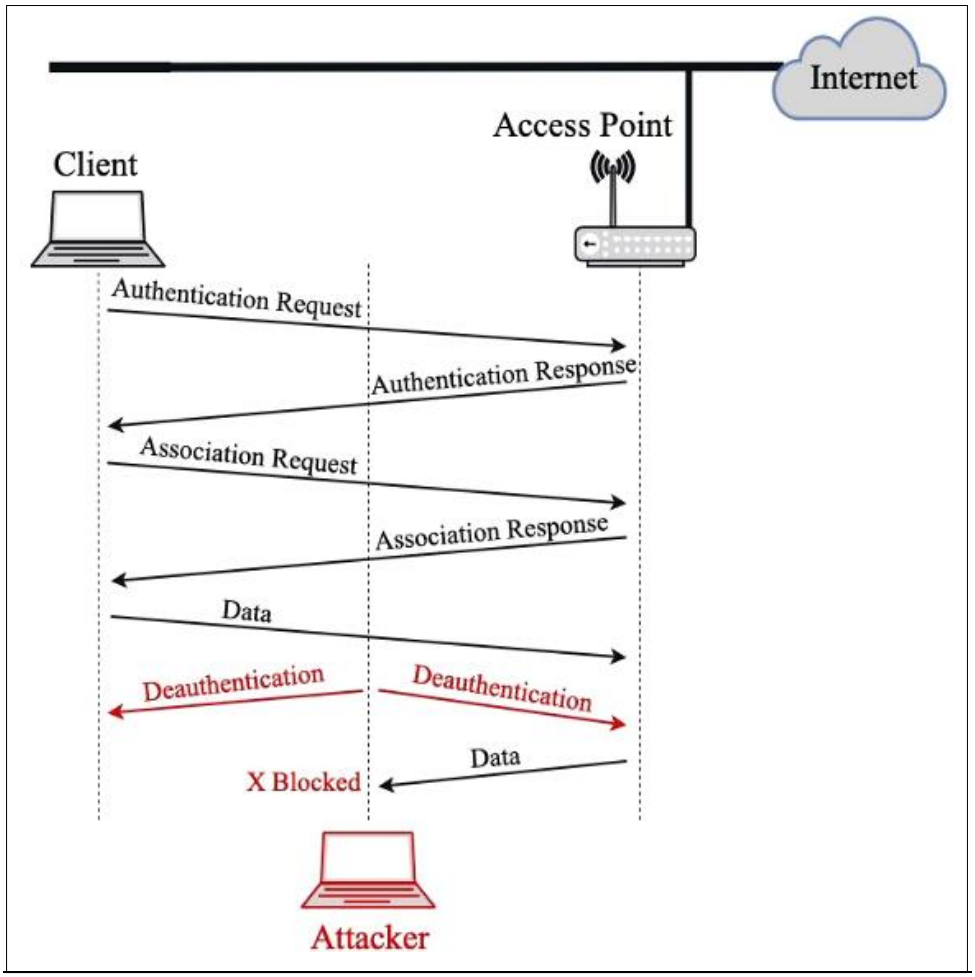


Figure 12. Deauthentication Attack Scenario

Algorithm 4. Deauthentication Attack

```
begin
    set interface to 'wlan0'
    set target_access_point_mac to 'xx:xx:xx:xx:xx:xx'
    set target_client_mac to 'yy:yy:yy:yy:yy:yy'
    set packet_count to 1000

    function send_deauth_packets:

        for i from 1 to packet_count:
            create deauth_packet to deauthentication_frame
            send deauth_packet using the interface
            print "de-authentication packet sent", i, "of", packet_count
            wait for a short interval

    function deauthentication_frame(ap_mac, client_mac):
        frame = {
            "type": "management",
            "subtype": "deauthentication",
            "destination": client_mac,
            "source": ap_mac,
            "bssid": ap_mac,
            "reason_code": 7  # class 3 frame received from nonassociated sta
        }
        return frame

    call send_deauth_packets
end
```

4-3- Testing Result and Evaluation

As mentioned in point B, testing was conducted in three scenarios: a controlled setup, an open environment, and an open environment with a de-authentication attack. In all scenarios, the invisible scout could distinguish between RAPs and legitimate APs, marked in blue, as seen in Figure 13. We conducted several experiments across the three defined scenarios to collect and evaluate identification results using the decision tree algorithm, confusion matrix, linear regression, and evaluation metrics.

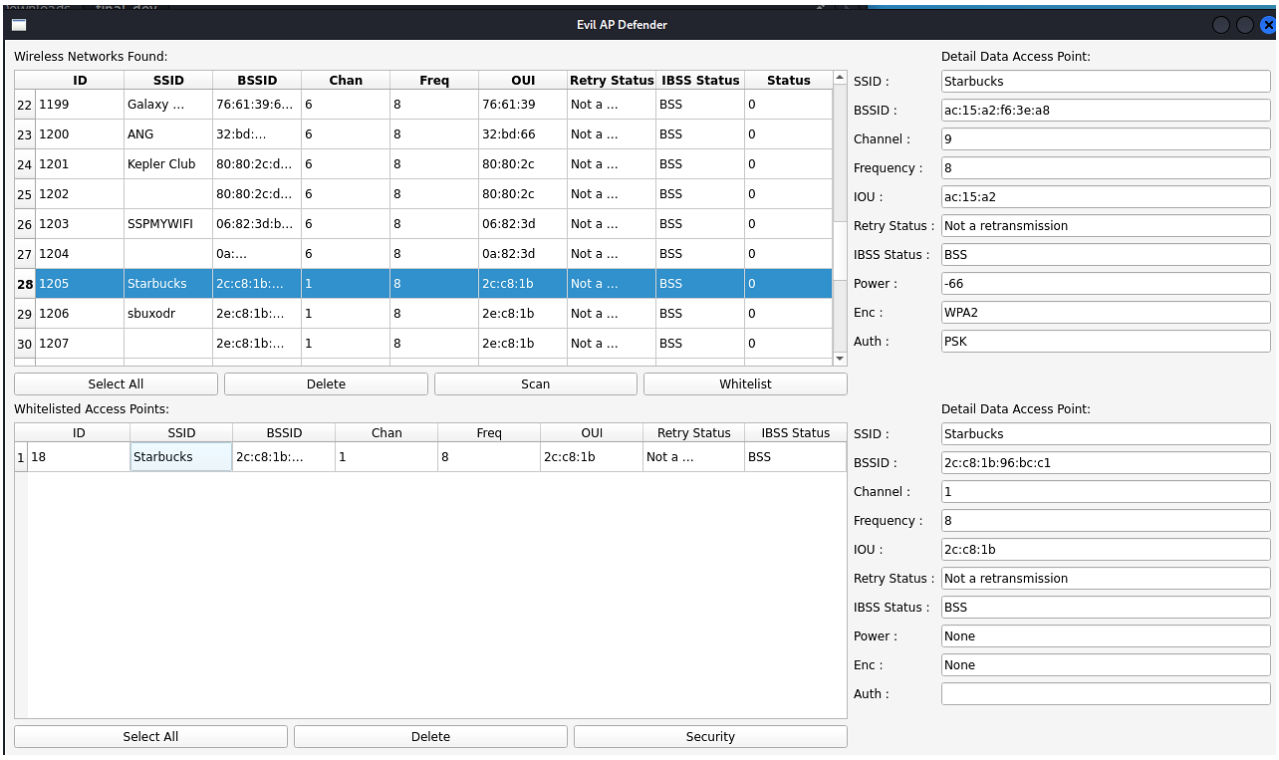


Figure 13. The Main Interface of Invisible Scout

The decision tree model was selected for this study due to its interpretability and effectiveness in handling datasets with categorical features, such as those used to identify RAPs. The decision tree's strength lies in providing clear insights into feature importance. It is critical to distinguish legitimate APs from RAPs based on parameters like OUI Number, Retry Bit, and IBSS Status. Additionally, decision trees can handle non-linear relationships, making them effective in complex scenarios across various network environments.

Other classification algorithms were considered, including Random Forest, Support Vector Machine (SVM), and Logistic Regression. Random Forest improves accuracy by averaging multiple decision trees, reducing overfitting, but it is computationally intensive and less interpretable than a single decision tree. SVM is powerful for high-dimensional data, especially when decision boundaries are clear, but its lack of interpretability and more significant computational requirements make it less ideal for this study. While simple and efficient for binary classification, Logistic Regression is less effective at capturing complex relationships in multi-class problems like distinguishing different types of APs. Ultimately, the decision tree model provided a desirable balance between simplicity, interpretability, and accuracy. Its visual structure enabled easier identification of key features and how they contributed to RAP detection, while its relatively low computational cost and firm performance in the study's evaluation metrics made it the optimal choice for RAP identification in this research.

In the decision tree, we utilize Information Gain (IG) in the Decision Tree algorithm (2) to select the most informative attributes to separate the data.

$$IG(D,A) = Entropy(D) - \sum_{\vartheta \epsilon values(A)} \left(\frac{|D_\vartheta|}{D} . Entropy(D_\vartheta) \right) \quad (2)$$

where IG(D, A) represents the Information Gain of attribute A concerning the dataset D. Entropy (D) measures the uncertainty or impurity within the entire dataset D. The term $\vartheta \in$ values(A) refers to each possible value of attribute A. $D\vartheta$ denotes the subset of the dataset D that has the value ϑ for attribute A. $\frac{|D_\vartheta|}{D}$ is the proportion of the dataset D that corresponds to the value ϑ for attribute A.

Figure 14 illustrates the decision tree results for the controlled setup, where network APs were categorized as either "legitimate" or "rogue" (unauthorized). The initial node, labelled "legitimate_ap," demonstrates that 52% of the instances were classified as legitimate APs, reflecting a relatively balanced dataset essential for developing a reliable model. The decision tree is divided into two main branches: one for legitimate APs and the other for rogue APs (RAPs). On the left branch, the decision tree model successfully identifies all instances as legitimate APs with 100% accuracy. This high accuracy likely stems from a critical distinguishing feature—presumably "legitimate_ssid"—which effectively differentiates legitimate APs from rogue ones. This feature at the root of the decision tree underscores its significance in discerning authorized from unauthorized network access.

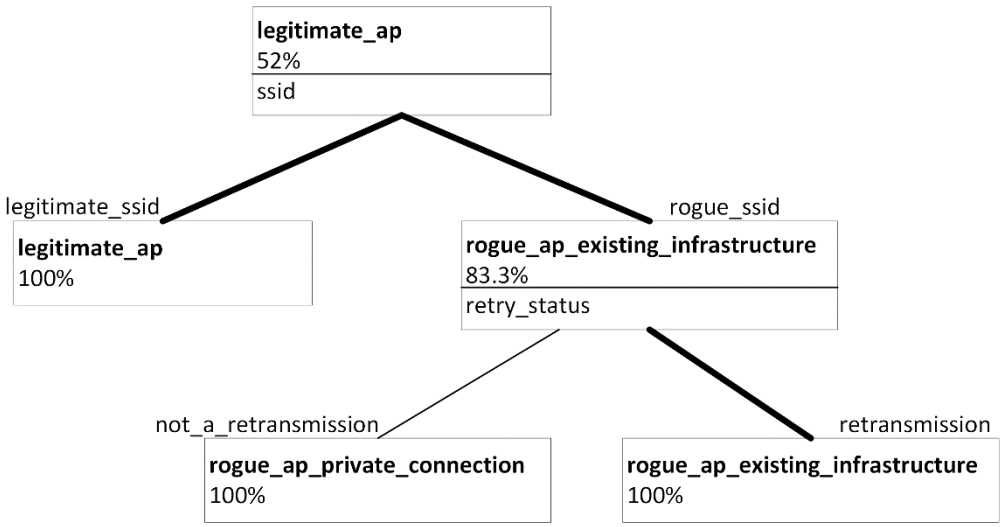


Figure 14. The Decision Tree Result for Controlled Setup

The right branch of the tree, addressing RAPs, showcases greater complexity, with the "rogue_ssid" node demonstrating an 83.3% success rate in classifying RAPs. Although this is a strong performance, the 16.7% misclassification rate suggests a need for deeper analysis to refine the model's sensitivity to specific rogue characteristics. This branch is further divided based on the "retry_status," distinguishing different rogue behaviors. Specifically, APs with a "retransmission" status are uniformly classified as "rogue_ap_existing_infrastructure," implying that these are unauthorized APs leveraging the existing network infrastructure. Conversely, those without retransmission are identified as "rogue_ap_private_connection," indicating separate, potentially external network connections, like 4G modems. This nuanced classification by "retry_status" is crucial for security measures, offering insights into the operational methods of rogue APs and aiding in developing targeted security protocols.

A confusion matrix complements the decision tree above. Additionally, the metrics provided an overview of the model's classification performance on test data, as shown in Figure 15.

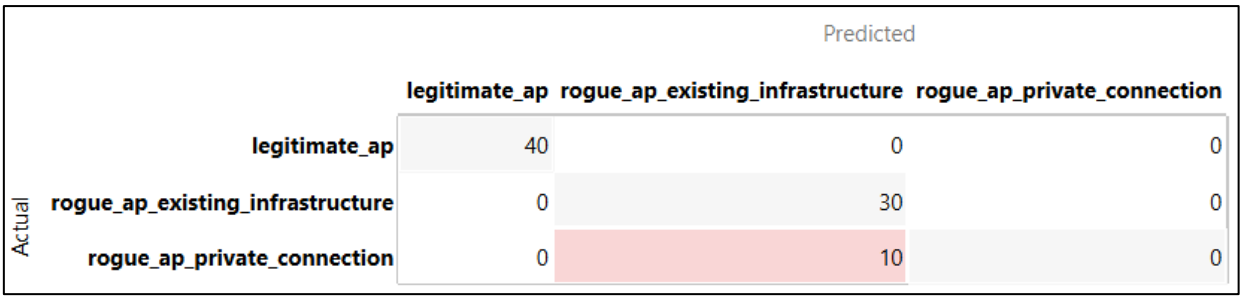


Figure 15. The Confusion Matrix For Controlled Setup

The confusion matrix results reveal that while the classification model excelled in identifying "legitimate_ap" and "rogue_ap_existing_infrastructure" with perfect accuracy, it faced significant challenges with the "rogue_ap_private_connection" class. All ten instances of "rogue_ap_private_connection" were misclassified as "rogue_ap_existing_infrastructure," indicating a critical area for improvement in the model's performance. This misclassification can be attributed to several interrelated factors.

Firstly, substantial feature overlap may exist between "rogue_ap_private_connection" and "rogue_ap_existing_infrastructure." The features the decision tree uses, such as SSID and retry status, must be sufficiently distinctive to differentiate between these two classes. RAPs using private connections, such as 4G modems, exhibit similar characteristics to those utilizing existing infrastructure, complicating the model's ability to distinguish between them accurately. Secondly, the data representation might play a significant role in the misclassification. With only ten instances of "rogue_ap_private_connection," the model may have insufficient examples to learn the distinguishing patterns effectively. This small sample size could lead to insufficient generalization, where the model performs well on familiar data but needs help with unseen instances.

In the next experiment conducted in an open environment, including scenarios involving a de-authentication attack, the Gini Index was utilized to assess the impurity or irregularity within the compiled dataset. This approach allowed for a systematic data evaluation, facilitating the identification of the most effective splits at each node in the decision tree. By leveraging the Gini Index, the method ensured that the optimal decision boundaries were determined, enhancing the model's ability to differentiate between legitimate AP and RAPs accurately under varying conditions.

$$GINI(D) = 1 - \sum_{i=1}^{c}(Pi)^2 \quad (3)$$

where *Pi* was the proportion of class *i* in the dataset *D*, and *c* represented the number of classes. Additionally, entropy measurement was used to assess uncertainty in the dataset based on the results of testing in an open environment. Where *Pi* represented the probability of class *i*,

$$Entropy\ (D) = -\sum_{i=1}^{c} Pi\ log_2(Pi) \quad (4)$$

Given the large amount of AP data, linear regression was used to identify outliers. These outliers showed the presence of rogue data,

$$y = b_0 + b_1x_1 + b_2x_2 \quad (5)$$

where *y* was the dependent variable, while x_1 and x_2 represented the independent variables. The term b_0 represented *y*-intercept, which was the value of *y* when both x_1 and x_2 were 0. Moreover, coefficient b_1 showed the alteration in *y* for a one-unit change in x_1, holding x_2 constant, and b_2 represented the alteration in *y* for a one-unit change in x_2, holding x_1 constant.

The decision tree in Figure 16, which utilizes GINI impurity (3) and entropy measurements (4), effectively categorized APs into two primary groups: legitimate and rogue. At the tree's root, the "legitimate_ap" node achieved an 89.5% classification rate, indicating that most access points in the open environment were classified as legitimate. This high classification rate suggests that the network environment is predominantly secure, with most APs behaving by standard security protocols. The left branch of the tree, which further analyzed legitimate connections, emphasizes the critical role of the "ssid" (Service Set Identifier) in identifying network legitimacy. The "ssid" feature, as shown in previous research, is often crucial to distinguish between legitimate and rogue APs, given that unauthorized devices may attempt to mimic legitimate SSIDs to deceive users or devices into connecting to them. The 100% classification rate at the leaf node "legitimate_ap" further underscores this scenario's prevalence of legitimate access points.

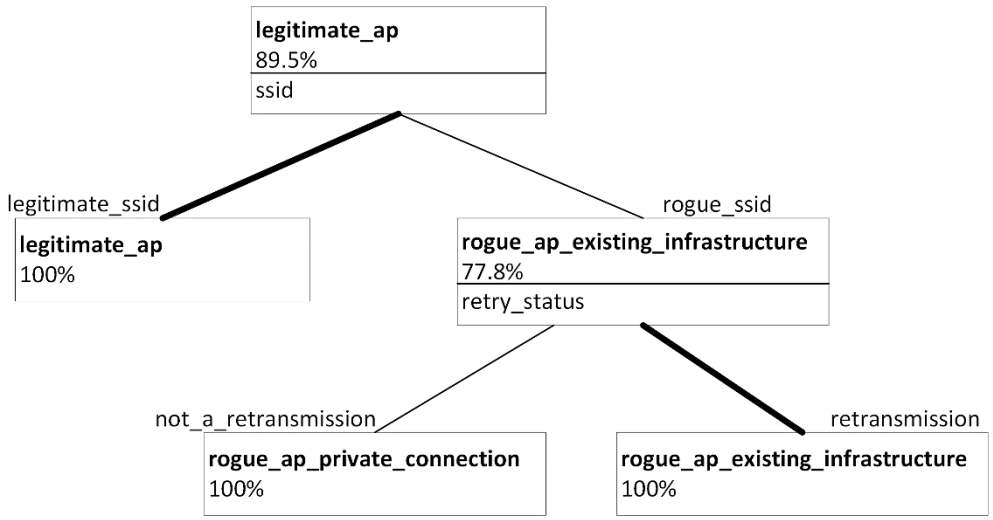


Figure 16. The Decision Tree Result For Open Environment

The decision tree's right branch reveals significant vulnerabilities in the network due to the presence of RAPs. Notably, the "rogue_ap_existing_infrastructure" node shows that 77.8% of the instances were classified as rogue APs, leveraging the existing network infrastructure. This suggests that a substantial number of unauthorized devices are infiltrating the network, aligning with common RAP attack strategies. RAPs connecting through the existing infrastructure blend seamlessly into the network, making them particularly challenging to detect through conventional security measures. This is a concern because these RAPs can provide unauthorized internet access or act as intermediaries to steal sensitive information.

The "retry_status" node plays a crucial role in refining the detection process, as a high rate of retransmissions often signifies repeated attempts to establish or maintain a connection, a behavior frequently observed in RAPs as they try to bypass network security protocols. Legitimate APs generally do not exhibit such behaviors, reinforcing the value of retry rates as a diagnostic feature for identifying RAPs. At the leaf nodes, the model achieves a perfect classification rate (100%) for both "rogue_ap_private_connection" and "rogue_ap_existing_infrastructure." This indicates that the decision tree is highly effective in detecting RAPs, whether using external private connections, like 4G modems, or embedded within the existing network. RAPs utilizing private connections are particularly concerning because they operate outside the controlled network infrastructure, making them more challenging to detect and potentially facilitating attacks such as man-in-the-middle. Similarly, the perfect classification of RAPs exploiting existing infrastructure demonstrates the model's capability in identifying these devices, which pose significant risks to network integrity, including data breaches, unauthorized access, and malware propagation.

Figure 16 presents a scatter plot illustrating the relationship between RAPs and the x-axis variable, as depicted in Figure 17. The equation y = 0,1708x + 106,66. indicates a positive correlation between the x-axis variable and the "status_ap" variable for the blue data points. A clear separation among these points suggests distinct AP categories. The upper points, appearing as outliers, represent potential RAPs, indicating unauthorized use of existing infrastructure or private connections.

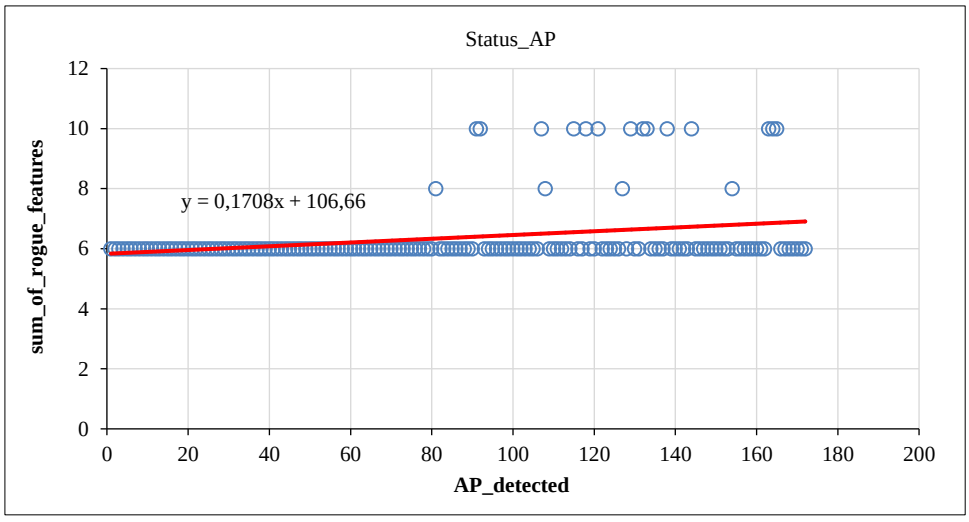


Figure 17. The Regression Chart for AP Legitimacy Status For Open Environment

The scatter plot reveals two clusters: a dense lower cluster likely representing legitimate APs and a more scattered upper cluster indicating RAPs. The higher "sum_of_rogue_features" scores in the upper cluster distinguish rogue APs from legitimate ones, reinforcing the model's effectiveness in detecting rogue activity. The x-axis ("AP_detected") reflects varying network densities, while the y-axis ("sum_of_rogue_features") highlights the distinction between legitimate and RAPs. The gap could serve as a threshold for classification, aiding in the detection process. The variability in the upper cluster suggests a range of RAP behaviors, including the use of private connections or existing infrastructure. Meanwhile, the slight spread in the lower cluster points to minor variations among legitimate APs, supporting the model's consistency in identifying them.

The heatmap in Figure 18 highlights critical differences between legitimate APs and RAPs across several features: SSID, Address, OUI, Retry Status, and IBSS Status. Legitimate APs consistently display green checks, indicating alignment with expected network norms, while RAPs, particularly those utilizing existing infrastructure or private connections, show red crosses in most categories, signifying deviations. SSID and OUI are critical identifiers where RAPs diverge sharply, allowing them to be quickly flagged as anomalous. RAPs exploiting existing infrastructure attempt to mimic certain network behaviors but still display significant discrepancies in SSID and Address. In contrast, RAPs in private connections show broader anomalies, especially in Retry and IBSS Status. When combined with regression analysis, the heatmap suggests that focusing on specific features like SSID, OUI, and IBSS Status enhances the accuracy of RAP detection, even in environments with many APs. These features allow for precise identification of rogue devices despite their attempts to blend into the network.

Class	SSID	Address	OUI	Retry Status	IBSS status
legitimate_AP	✓ 154	✓ 154	✓ 154	✓ 158	✓ 168
RAP existing infrastructure	✗ 18	✗ 18	✗ 18	✗ 14	✓ 168
RAP private connection	✗ 18	✗ 18	✗ 18	✓ 158	✗ 4

Figure 18. The Heatmap of RAP Detection for Open Environment

The classification model from Figure 16 was reviewed by this confusion matrix, as shown in Figure 19. The confusion matrix illustrates the performance of a classification model designed to detect various types of access points (APs) in a network. The model categorizes APs into three classes: legitimate_ap, rogue_ap_existing_infrastructure, and rogue_ap_private_connection. In the matrix, rows represent actual classes, while columns represent predicted classes, with diagonal elements indicating correct predictions and off-diagonal elements showing misclassifications. The model accurately identified 154 legitimate APs, 14 RAPs on existing infrastructure, and 4 RAPs with private connections, with no misclassifications noted. This results in a 100% accuracy rate, demonstrating the model's ability to distinguish between the three categories perfectly.

		Predicted			
		legitimate_ap	rogue_ap_existing_infrastructure	rogue_ap_private_connection	Σ
Actual	legitimate_ap	154	0	0	154
	rogue_ap_existing_infrastructure	0	14	0	14
	rogue_ap_private_connection	0	0	4	4
	Σ	154	14	4	172

Figure 19. The Confusion Matrix For Open Environment

The confusion matrix in Figure 19 evaluates the performance of the classification model from Figure 16, which was designed to detect various types of APs within a network. The model categorizes APs into three distinct classes: "legitimate_ap," "rogue_ap_existing_infrastructure," and "rogue_ap_private_connection." As with standard confusion matrices, the rows represent actual class labels, while the columns indicate predicted class labels. Diagonal elements correspond to correct predictions, and off-diagonal elements would indicate misclassifications. In this case, the model accurately identified 154 legitimate APs, 14 RAPs leveraging existing infrastructure, and four rogue APs using private connections. Remarkably, no misclassifications were recorded, resulting in a perfect 100% accuracy rate, showcasing the model's ability to distinguish between the three categories precisely.

This classification performance indicates several key insights. First, the model's accuracy across all categories demonstrates that the features chosen for classification are highly discriminative and effective at differentiating between

legitimate and rogue. The distribution of APs reveals that legitimate APs are the most common (154 instances, 89.5%), while RAPs, with 14 (8.1%) utilizing existing infrastructure and 4 (2.3%) relying on private connections. Notably, the absence of off-diagonal elements in the confusion matrix means no false positives or negatives. This lack of misclassification suggests the model is robust and unbiased, even when faced with the less common "rogue_ap_private_connection" category. From a security perspective, the model's ability to accurately distinguish between RAPs using existing infrastructure and those with private connections is critical for tailoring effective responses. Such precision would significantly enhance a network's threat detection and response capabilities, as it ensures confidence in the model's predictions, supporting automated security measures to protect against unauthorized access.

For the following scenario, the decision tree visualization in Figure 20 demonstrates a classification model for identifying various types of APs within a network in a de-authentication attack case. Starting at the root node, which contains 156 total samples, the model classifies 140 samples as legitimate APs with an accuracy of 89.7%, based on the "ssid" feature. This first split is crucial, as it effectively separates the majority of legitimate APs from potential rogue ones. The left branch, labelled "legitimate_ssid," includes 140 samples correctly identified as legitimate APs with 100% accuracy, leading to a leaf node with no further splits. This high accuracy on the left branch indicates that the "ssid" feature alone can classify all legitimate APs in this dataset, highlighting its importance in the model.

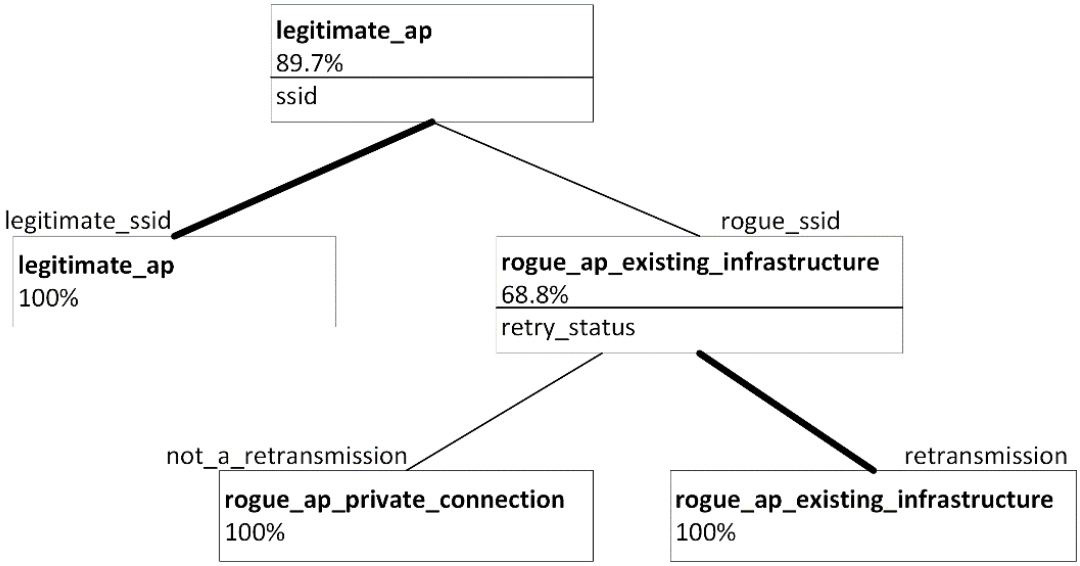


Figure 20. The Decision Tree Result For Open Environment With Deauthentication Attack

On the right branch, the model classifies the remaining 16 samples of APs as rogue, further dividing them based on the "retry_status" feature. This split generates two child nodes: The left child node, which represents APs with no retransmission, classifies five samples as rogue APs using a private connection, achieving 100% accuracy. The right child node, representing APs with retransmission, correctly classifies 11 samples as RAPs using existing network infrastructure with 100% accuracy. The "retry_status" feature is a highly effective discriminator for distinguishing between these two types of rogue APs. This perfect subclassification indicates that the model is well-suited for identifying RAP subtypes, such as those exploiting existing network resources or using private connections. This is critical in scenarios involving de-authentication attacks. This relatively simple decision tree, with just two split levels, underscores the high discriminative power of the selected features. The tree's structure implies that RAPs involved in de-authentication attacks are more likely to exploit existing infrastructure, as evidenced by the higher number of instances (11) classified in this category than RAPs using private connections (5). This insight benefits network security, as it helps prioritize monitoring and detecting RAPs based on their connection behavior.

Figure 21 illustrates the relationship between the number of detected access points (AP_detected) and the sum of rogue features (sum_of_rogue_features), with a red trend line represented by the equation $y = 0.0071x + 5.7911$. While this line indicates a weak positive correlation, suggesting that the sum of rogue features slightly increases as more APs are detected, the scatter plot reveals a more complex data pattern with two distinct clusters. The lower cluster, densely packed around a y-value of 7, likely represents legitimate APs exhibiting no rogue feature counts across varying numbers of detected APs. In contrast, the upper cluster, centred around a y-value of 8 and 10, indicates RAPs that consistently show higher rogue feature counts, regardless of network density. This clear separation between clusters highlights the system's ability to effectively differentiate between normal APs and RAPs.

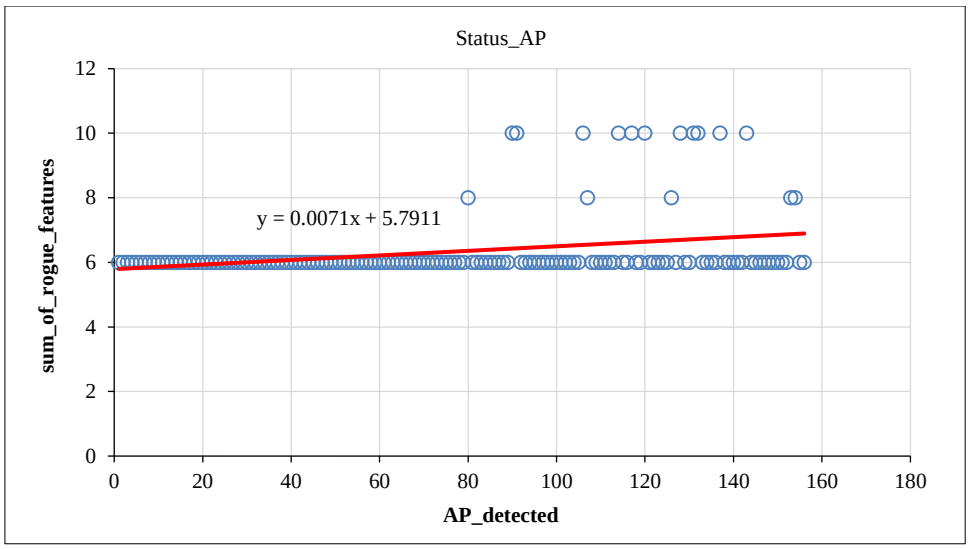


Figure 21. The Regression Chart for AP Legitimacy Status For Open Environment With Deauthentication Attack

Vertical alignments in the scatter plot, particularly at x-values around 80, 100, 120, and 140, suggest repeated measurements in environments with the same detected APs but differing rogue feature counts (Figure 22). These alignments point to variability in the network environment or edge cases where APs exhibit some but not all rogue characteristics. Additionally, outliers between the two main clusters may represent the consistency of the RAP cluster, and the separation between normal and rogue APs across different network densities demonstrates the robustness of the detection method in distinguishing RAPs.

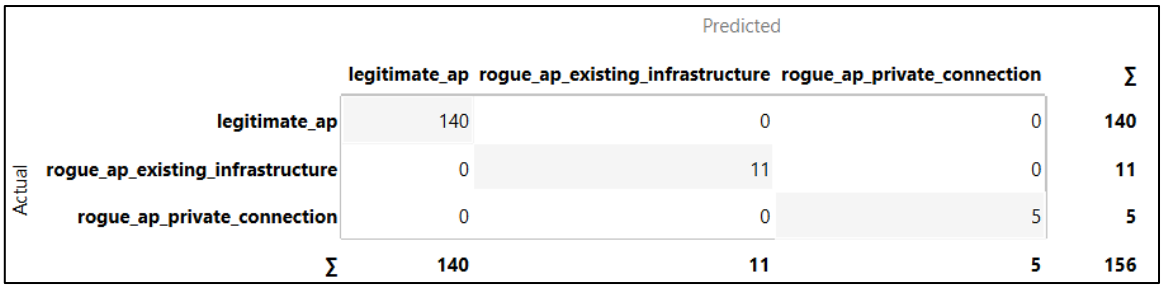
Class	SSID	Address	OUI	Retry Status	IBSS status
legitimate_AP	✓ 140	✓ 140	✓ 140	✓ 145	✓ 151
RAP existing infrastructure	✗ 16	✗ 16	✗ 16	✗ 11	✓ 151
RAP private connection	✗ 16	✗ 16	✗ 11	✓ 145	✗ 5

Figure 22. The Heatmap of RAP Detection for Open Environment Open Environment With Deauthentication Attack

The heatmap above visually compares legitimate and RAPs in existing infrastructure and private connections across several key factors: SSID, Address, OUI, Retry Status, and IBSS Status in a de-authentication attack case. Legitimate APs maintain consistency across all parameters, as the green checks indicate, reflecting adherence to network norms. In contrast, RAPs show red crosses in critical fields such as SSID, Address, and OUI, signaling significant deviations. RAPs using existing infrastructure exhibit some overlap with legitimate APs in IBSS status, suggesting an attempt to mimic legitimate behavior. However, the differences in Retry Status (11 vs. 145) and key identifiers like SSID and OUI underscore clear distinctions. RAPs operating through private connections display even more significant discrepancies, particularly in Retry Status and IBSS Status (5 vs. 151), making them more distinguishable. The heatmap underscores that critical features like SSID and OUI remain pivotal in differentiating legitimate APs from rogue ones. The substantial differences in Retry and IBSS values further highlight that RAPs exhibit more isolated network behavior, making detection easier, especially in private connections.

The classification model from Figure 20 was evaluated using the confusion matrix shown in Figure 23. The confusion matrix assesses the performance of the decision tree model for rogue access point (RAP) identification, revealing perfect classification results across all categories. The matrix shows that all 140 legitimate APs were correctly classified as legitimate, with no misclassifications into rogue AP categories. Additionally, all 11 instances of rogue APs using existing infrastructure and all five RAPs using private connections were accurately identified, resulting in 100% classification accuracy for each class. This flawless performance suggests that the model has effectively captured the distinguishing features between legitimate APs and the two types of rogue APs, demonstrating its capacity to differentiate accurately within the dataset.

With legitimate APs comprising 89.7% of the instances, RAPs using existing infrastructure making up 7.1%, and RAPs using private connections representing 3.2%, the model performs exceptionally well on the minority classes. This suggests that the model's feature selection and training are robust, as it accurately identifies both types of RAPs without any false positives or negatives. Such performance is invaluable in real-world scenarios, where the precise identification of network threats is crucial. The absence of misclassifications underscores the model's effectiveness. Analyzing the model's confidence scores for each prediction in future work could provide additional insights, especially for potential edge cases, and further support its readiness for real-world implementation in network environments.



Actual \ Predicted	legitimate_ap	rogue_ap_existing_infrastructure	rogue_ap_private_connection	∑
legitimate_ap	140	0	0	140
rogue_ap_existing_infrastructure	0	11	0	11
rogue_ap_private_connection	0	0	5	5
∑	140	11	5	156

Figure 23. The Confusion Matrix For Open Environment With Deauthentication Attack

In comparing the decision tree and confusion matrix results, the controlled environment yielded significantly fewer AP data than the open environment. This limited dataset likely restricted the decision tree model's ability to fully capture the variability of APs, leading to a slightly lower performance than other scenarios. With fewer APs to analyze, the model may have struggled to differentiate between legitimate APs and the various types of RAPs due to the lack of diversity in the beacon frames captured. This reduced variety in the controlled setup can also lead to overfitting, where the model performs well under specific test conditions but needs more robustness for more dynamic and variable environments.

In contrast, the open environment provided a much larger and more diverse dataset, including APs with different characteristics, signal strengths, and interference levels. This abundance of data enabled the decision tree model to generalize more effectively and adapt to the complexities of real-world conditions. The increased volume and diversity of APs improved the model's ability to distinguish between legitimate APs and different types of RAPs, resulting in improved classification accuracy. The richer dataset from the open environment contributed to the enhanced performance of the Invisible Scout system, showcasing its ability to transition successfully from controlled to open environments.

In this study, we also employed several evaluation metrics to assess the performance of our model. AUC (Area Under the ROC Curve) reflects the model's ability to distinguish between RAPs and legitimate APs, with higher values indicating better discrimination. Classification Accuracy (CA) provides an overall percentage of correctly classified instances but may be less informative when a class imbalance exists. We used the F1 score to complement accuracy, which balances precision and Recall, mainly when false positives or negatives are critical. Precision measures the proportion of actual RAPs among all predicted RAPs, focusing on minimizing false alarms. At the same time, Recall indicates the proportion of actual RAPs that were correctly identified, minimizing missed detections. The MCC (Matthews Correlation Coefficient) offers a more balanced metric, considering all aspects of the confusion matrix (true positives, true negatives, false positives, and false negatives). The corresponding evaluation parameters are calculated using the equations provided below.

$$CA = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

$$Precison = \frac{TP}{TP+FP} \tag{7}$$

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

$$F1 = 2 \times \frac{Precision\ x\ Recall}{Precision+Recall} \tag{9}$$

$$MCC = \frac{(TP\ x\ TN)-(FP\ x\ FN)}{\sqrt{(TP+FN)(TP+FN)(TN+FP)(TN+FP)}} \tag{10}$$

Precision is crucial in handling false positives (legitimate APs classified as RAPs), as higher precision values correspond to fewer false alarms. For example, as shown in Table 2, in our open environment with a de-authentication attack, the model achieved a precision of 0.990, demonstrating its effectiveness in reducing false positives. On the other hand, false negatives (RAPs incorrectly classified as legitimate APs) are managed by optimizing Recall. High recall values, such as 0.996 in the same scenario, indicate that the model effectively identifies nearly all RAPs, minimizing missed detections. The F1 score further supports this by balancing precision and Recall, providing a reliable measure of overall performance across different environments.

Table 2. Evaluation Metrics For All Scenarios

Scenario	AUC	CA	F1	Precision	Recall	MCC
1 – Controlled Setup	0.921	0.875	0.821	0.781	0.875	0.803
2 – Open Environment	0.952	0.994	0.991	0.988	0.994	0.971
3 – Opent Environment With Deauthentication Attack	0.955	0.996	0.993	0.990	0.996	0.975

The evaluation metrics in Table 2 offer a clear view of the decision tree model's performance in identifying RAPs across three scenarios using Invisible Scout. In the controlled setup, the model performed well with an AUC of 0.921, but this was the lowest among the scenarios, suggesting the model may be less robust in simpler environments. A CA of 0.875 and an F1 score of 0.821 indicate solid performance, though there is room for improvement in classification accuracy. Precision at 0.781 shows occasional misclassification of legitimate APs, while Recall at 0.875 reflects the model's ability to detect most RAPs. The MCC of 0.803 points to a good but not flawless classification performance.

In the open environment, the model's performance improved significantly. The AUC increased to 0.952, showing better discrimination between RAPs and legitimate APs, and the CA rose to 0.994, indicating strong generalization in real-world conditions. With an F1 score of 0.991, the model balanced precision (0.988) and Recall (0.994), minimizing false positives and negatives. The MCC of 0.971 underscores the model's reliability in this scenario.

The third scenario, involving a de-authentication attack, showed the model's best performance across all metrics. The AUC of 0.955 and CA of 0.996 reflect the model's robustness under attack. The F1 score of 0.993, precision of 0.990, and Recall of 0.996 demonstrate near-perfect performance, with minimal misclassification of legitimate APs and almost flawless detection of RAPs. The MCC of 0.975 highlights the model's reliability even in challenging conditions.

The system's performance under de-authentication attacks presented unique challenges compared to other scenarios due to the disconnections and re-authentications of APs. This process involved our own tool and public APs, making it necessary to carefully time the attack scenario to avoid disrupting any Wi-Fi users. The attack introduced noisy data, leading to more significant fluctuations in beacon frame features. Despite this, the RAP detection model demonstrated resilience, mainly using the decision tree algorithm. By focusing on key features in the beacon frame, as mentioned earlier, which remained relatively stable during such attacks, the model was still able to detect RAPs.

When comparing performance across different scenarios, the system excelled in the controlled setup, where the environment was stable and predictable, although some misclassifications still occurred. In the open environment, performance dipped slightly due to more APs and increased noise, but the system adapted by focusing on critical features such as Address 1-4 and the Retry Bit. The de-authentication attack scenario, however, proved more challenging, taking longer to complete compared to both the controlled setup and the open environment. Nonetheless, the core beacon frame features remained unaffected by the attack, allowing the model to detect RAPs successfully.

Overall, the model's performance improves in more complex and challenging environments, such as open environments and those with de-authentication attacks. While the controlled setup showed promising results, there is room for further refinement to enhance the model's precision and Recall in such settings. Future efforts could address these challenges to achieve more consistent and robust results across all scenarios. We also compare the highest accuracy rates with several previous studies that similarly used layer 2 as the basis for their features, as shown in Table 3.

Table 3. Results and Comparison With Other Studies

No.	Research	Actor	Anonymous Probing	Dedicated Hardware	Classification	Highest Accuracy	Detection
1	Jain et al. [21]	Client	No	No	Node profiling	Not mentioned	Immediate
2	Lu et al. [32]	Client	Yes	No	Packet auditing	1.000	Data-oriented
3	Selvarathinam et al. [43]	Server	No	No	Bots setup	0.947	Data-oriented
4	VanSickle et al. [23]	Client	Yes	No	Node profiling	Not mentioned	Immediate
5	Bodhe et al. [24]	Client	Yes	No	Node profiling	0.930	Data-oriented
6	Jang et al. [25]	Client	Yes	Yes	Node profiling	1.000	Immediate
7	Delgado et al. [35]	Server	Yes	Yes	Packet auditing	0.990	Data-oriented
8	Shrivastava et al. [27]	Client	Yes	No	Node profiling	Not mentioned	Immediate
9	Lu et al. [28]	Client	No	No	Node profiling	Not mentioned	Immediate
10	Agyemang et al. [9]	Client	Yes	No	Node profiling	Not mentioned	Immediate
11	Igarashi et al. [29]	Client	Yes	No	Node profiling	0.920	Immediate
12	Korolkov & Kutsak [36]	Client	Yes	No	Packet auditing	Not mentioned	Data-oriented
13	Alyami et al. [17]	Client	Yes	Yes	Packet auditing	0.960	Data-oriented
14	Our research	Client	Yes	No	Node profiling and packet auditing	0.996	Immediate

5- Conclusion

This research introduces Invisible Scout, a pioneering multi-module system designed to detect the RAPs through its integrated sniffer, detection, probing, and comparison modules. The system's effectiveness was demonstrated through rigorous testing in a controlled environment, achieving high classification accuracy (CA), precision, and Recall. Notably, the decision tree model achieved an AUC score of 0.921 and a CA of 0.875, indicating a solid capability to differentiate between legitimate access points and RAPs. However, the system encountered challenges distinguishing between certain RAP types, highlighting the need for further refinement and development. In the open environment, the system maintained robust performance with an AUC score of 0.952 and a CA of 0.994, successfully identifying RAPs that utilized existing infrastructure. The system performed exceptionally well under the de-authentication attack scenario, achieving its highest accuracy with an AUC score of 0.955 and a CA of 0.996. This scenario, while challenging, demonstrated the model's resilience, and linear regression analysis provided valuable insights into RAP behaviors and distribution patterns. Despite its strengths, the confusion matrix revealed difficulties in accurately identifying legitimate access points with private connections in the initial setup. Future work will focus on enhancing the detection mechanism to address the identified challenges, particularly improving the differentiation between various types of RAPs. This includes integrating additional features and exploring advanced machine learning techniques, such as ensemble or deep learning, to enhance the model's accuracy and robustness. Moreover, integrating real-time detection and automated response capabilities will be prioritized to provide immediate alerts and effective mitigation against RAP threats. Invisible Scout aims to advance as a comprehensive solution for securing wireless networks from RAP-related vulnerabilities by addressing these aspects.

6- Declarations

6-1- Author Contributions

Conceptualization, D.A. and N.M.A.; methodology, S.K.; software, D.A.; validation, N.M.A. and S.K.; formal analysis, N.M.A.; investigation, D.A.; resources, N.M.A.; data curation, S.K.; writing—original draft preparation, D.A.; writing—review and editing, N.M.A.; visualization, S.K.; supervision, N.M.A.; project administration, S.K.; funding acquisition, N.M.A. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6-3- Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6-4- Acknowledgements

The authors are grateful to the Faculty of Information Science and Technology (FIST) at Multimedia University (Melaka Campus) for their support and to all CICC/Thundercloud lab members for their contributions and assistance in this research.

6-5- Institutional Review Board Statement

Not applicable.

6-6- Informed Consent Statement

Not applicable.

6-7- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Pahlavan, K., & Krishnamurthy, P. (2021). Evolution and Impact of Wi-Fi Technology and Applications: A Historical Perspective. International Journal of Wireless Information Networks, 28(1), 3–19. doi:10.1007/s10776-020-00501-8.

[2] Reshef, E., & Cordeiro, C. (2022). Future Directions for Wi-Fi 8 and Beyond. IEEE Communications Magazine, 60(10), 50–55. doi:10.1109/MCOM.003.2200037.

[3] Tian, L., Santi, S., Seferagić, A., Lan, J., & Famaey, J. (2021). Wi-Fi HaLow for the Internet of Things: An up-to-date survey on IEEE 802.11ah research. Journal of Network and Computer Applications, 182. doi:10.1016/j.jnca.2021.103036.

[4] Oughton, E. J., Lehr, W., Katsaros, K., Selinis, I., Bubley, D., & Kusuma, J. (2021). Revisiting Wireless Internet Connectivity: 5G vs Wi-Fi 6. Telecommunications Policy, 45(5), 102127. doi:10.1016/j.telpol.2021.102127.

[5] Wu, C., Wang, B., Au, O. C., & Liu, K. J. R. (2022). Wi-Fi Can Do More: Toward Ubiquitous Wireless Sensing. IEEE Communications Standards Magazine, 6(2), 42–49. doi:10.1109/MCOMSTD.0001.2100111.

[6] Chatzoglou, E., Kambourakis, G., & Kolias, C. (2022). How is your Wi-Fi connection today? DoS attacks on WPA3-SAE. Journal of Information Security and Applications, 64, 103058. doi:10.1016/j.jisa.2021.103058.

[7] Karbasi, A. H., & Shahpasand, S. (2020). A post-quantum end-to-end encryption over smart contract-based blockchain for defeating man-in-the-middle and interception attacks. Peer-to-Peer Networking and Applications, 13(5), 1423–1441. doi:10.1007/s12083-020-00901-w.

[8] Arisandia, D., Ahmad, N. M., & Kannan, S. (2022). A Detection Technique Using Dual Authentication Stages Framework for Rogue Access Point Identification. IOP Conference Series: Earth and Environmental Science, 1083(1). doi:10.1088/1755-1315/1083/1/012091.

[9] Agyemang, J. O., Kponyo, J. J., Klogo, G. S., & Boateng, J. O. (2020). Lightweight rogue access point detection algorithm for WiFi-enabled Internet of Things(IoT) devices. Internet of Things (Netherlands), 11, 100200. doi:10.1016/j.iot.2020.100200.

[10] Hu, J., Li, Y., Cui, Y., & Bu, L. (2021). A Technical Survey on Approaches for Detecting Rogue Access Points. Smart Innovation, Systems and Technologies, 190, 169–174. doi:10.1007/978-981-15-5697-5_20.

[11] Hasan, Md. T., Hossain, Md. R., & Pathan, A.-S. K. (2021). Protecting Regular and Social Network Users in a Wireless Network by Detecting Rogue Access Point. Securing Social Networks in Cyberspace, 255–275, CRC Press, Boca Raton, United States. doi:10.1201/9781003134527-16.

[12] Wofford, P. (2020). Rogue Access Points: The Threat to Public Wireless Networks. Master Thesis, Utica College, Utica, United States.

[13] Khodadady, N. B. (2024). A Study on the Effectiveness of Offensive Wi-Fi Network Security Management. Ph.D. Thesis, Colorado Technical University, Colorado Springs, United States.

[14] Lovinger, N., Gerlich, T., Martinasek, Z., & Malina, L. (2020). Detection of wireless fake access points. 2020 12th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT), 113–118. doi:10.1109/icumt51630.2020.9222455.

[15] Kim, M., Kwon, S., Elmazi, D., Lee, J. H., Barolli, L., & Yim, K. (2020). A Technical Survey on Methods for Detecting Rogue Access Points. Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2019, Advances in Intelligent Systems and Computing, 994. Springer, Cham, Switzerland. doi:10.1007/978-3-030-22263-5_21.

[16] Patel, K. C., & Patel, A. (2022). Taxonomy and Future Threat of Rogue Access Point for Wireless Network. Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development, INDIACom 2022, 679–688. doi:10.23919/INDIACom54597.2022.9763150.

[17] Alyami, M., Alharbi, I., Zou, C., Solihin, Y., & Ackerman, K. (2022). WiFi-based IoT Devices Profiling Attack based on Eavesdropping of Encrypted WiFi Traffic. Proceedings - IEEE Consumer Communications and Networking Conference, CCNC, 385–392. doi:10.1109/CCNC49033.2022.9700674.

[18] Satam, P., & Hariri, S. (2021). WIDS: An Anomaly Based Intrusion Detection System for Wi-Fi (IEEE 802.11) Protocol. IEEE Transactions on Network and Service Management, 18(1), 1077–1091. doi:10.1109/TNSM.2020.3036138.

[19] Arisandi, D., Ahmad, N. M., & Kannan, S. (2021). The rogue access point identification: A model and classification review. Indonesian Journal of Electrical Engineering and Computer Science, 23(3), 1527–1537. doi:10.11591/ijeecs.v23.i3.pp1527-1537.

[20] Coll, E. (2023). The OSI Layers and Protocol Stacks. Teracom Training Institute, Las Vegas, United States.

[21] Jain, V., Laxmi, V., Gaur, M. S., & Mosbah, M. (2019). ETGuard: Detecting D2D attacks using wireless Evil Twins. Computers and Security, 83, 389–405. doi:10.1016/j.cose.2019.02.014.

[22] Hsu, F. H., Hsu, Y. L., & Wang, C. S. (2019). A solution to detect the existence of a malicious rogue AP. Computer Communications, 142–143, 62–68. doi:10.1016/j.comcom.2019.03.013.

[23] VanSickle, R., Abegaz, T., & Payne, B. (2019). Effectiveness of tools in identifying rogue access points on a wireless network. KSU Proceedings on Cybersecurity Education, Research and Practice, 5, 1-11.

[24] Bodhe, A. S., Dhanrao, P., Sangle, A., & Jagdisha, N. (2020). Design secure WSN with advancement in finding rouge access point with soft computing tools. Advances in Parallel Computing, 37, 543–551. doi:10.3233/APC200200.

[25] Jang, R., Kang, J., Mohaisen, A., & Nyang, D. (2020). Catch me if you can: Rogue access point detection using intentional channel interference. IEEE Transactions on Mobile Computing, 19(5), 1056–1071. doi:10.1109/TMC.2019.2903052.

[26] Hsu, F. H., Wang, C. S., Ou, C. W., & Hsu, Y. L. (2020). A passive user-side solution for evil twin access point detection at public hotspots. International Journal of Communication Systems, 33(14), 1–16,. doi:10.1002/dac.4460.

[27] Shrivastava, P., Jamal, M. S., & Kataoka, K. (2020). EvilScout: Detection and Mitigation of Evil Twin Attack in SDN Enabled WiFi. IEEE Transactions on Network and Service Management, 17(1), 89–102. doi:10.1109/TNSM.2020.2972774.

[28] Lu, Q., Jiang, R., Ouyang, Y., Qu, H., & Zhang, J. (2020). BiRe: A client-side Bi-directional SYN reflection mechanism against multi-model evil twin attacks. Computers and Security, 88. doi:10.1016/j.cose.2019.101618.

[29] Igarashi, K., Kato, H., & Sasase, I. (2021). Rogue Access Point Detection by Using ARP Failure under the MAC Address Duplication. 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 1469–1474. doi:10.1109/pimrc50174.2021.9569473.

[30] Bello, N., & Kanu, O. (2023). Penetration Testing of Gsm Network Using Man-in-the-Middle Attack. JES. Journal of Engineering Sciences, 0(0), 0–0. doi:10.21608/jesaun.2023.226718.1249.

[31] Liu, P., Yang, P., Song, W. Z., Yan, Y., & Li, X. Y. (2019). Real-time Identification of Rogue WiFi Connections Using Environment-Independent Physical Features. Proceedings - IEEE INFOCOM, 190–198. doi:10.1109/INFOCOM.2019.8737455.

[32] Lu, Q., Qu, H., Ouyang, Y., & Zhang, J. (2019). SLFAT: Client-Side Evil Twin Detection Approach Based on Arrival Time of Special Length Frames. Security and Communication Networks, 2718741. doi:10.1155/2019/2718741.

[33] Kitisriworapan, S., Jansang, A., & Phonphoem, A. (2020). Client-side rogue access-point detection using a simple walking strategy and round-trip time analysis. Eurasip Journal on Wireless Communications and Networking, 252. doi:10.1186/s13638-020-01864-5.

[34] Sankhe, K., Jaisinghani, D., & Chowdhury, K. (2020). CSIscan: Learning CSI for Efficient Access Point Discovery in Dense WiFi Networks. IEEE 28th International Conference on Network Protocols (ICNP), 1–12. doi:10.1109/icnp49622.2020.9259360.

[35] Delgado, O., Kechtban, L., Lugan, S., & Macq, B. (2020). Passive and active wireless device secure identification. IEEE Access, 8, 83312–83320. doi:10.1109/ACCESS.2020.2991649.

[36] Korolkov, R. Y., & Kutsak, S. V. (2021). Received-signal-strength-based approach for detection and 2D indoor localization of evil twin rogue access point in 802.11. International Journal of Safety and Security Engineering, 11(1), 13–20. doi:10.18280/ijsse.110102.

[37] Lu, Q., Li, S., Zhang, J., & Jiang, R. (2022). PEDR: Exploiting phase error drift range to detect full-model rogue access point attacks. Computers and Security, 114, 102581. doi:10.1016/j.cose.2021.102581.

[38] Kim, D., Shin, D., & Shin, D. (2018). Unauthorized Access Point Detection Using Machine Learning Algorithms for Information Protection. 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 1876–1878. doi:10.1109/trustcom/bigdatase.2018.00284.

[39] Yang, Z., Lu, Q., Zhang, H., Chen, F., & Xian, H. (2024). Eliminating Rogue Access Point Attacks in IoT: A Deep Learning Approach With Physical-Layer Feature Purification and Device Identification. IEEE Internet of Things Journal, 11(8), 14886–14900. doi:10.1109/JIOT.2023.3345378.

[40] Liu, W., & Papadimitratos, P. (2024). Position-based Rogue Access Point Detection. Proceedings - 9th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2024, 436–442. doi:10.1109/EuroSPW61312.2024.00055.

[41] Jing, W., Peng, L., Fu, H., & Hu, A. (2024). An Authentication Mechanism Based on Zero Trust With Radio Frequency Fingerprint for Internet of Things Networks. IEEE Internet of Things Journal, 11(13), 23683–23698. doi:10.1109/JIOT.2024.3385989.

[42] Zhang, B., Zhang, T., Ma, Y., Xi, Z., He, C., Wang, Y., & Lv, Z. (2024). A Low-Latency Approach for RFF Identification in Open-Set Scenarios. Electronics, 13(2), 384. doi:10.3390/electronics13020384.

[43] White, G. B., & Sjelin, N. (2022). The NIST Cybersecurity Framework. Research Anthology on Business Aspects of Cybersecurity, 39–55, IGI Global, Hershey, United States. doi:10.4018/978-1-6684-3698-1.ch003.

[44] Selvarathinam, N. S., Dhar, A. K., & Biswas, S. (2019). Evil Twin Attack Detection using Discrete Event Systems in IEEE 802.11 Wi-Fi Networks. 2019 27th Mediterranean Conference on Control and Automation (MED), 316–321. doi:10.1109/med.2019.8798568.

[45] Wang, J., Juarez, N., Kohm, E., Liu, Y., Yuan, J., & Song, H. (2019). Integration of SDR and UAS for Malicious Wi-Fi Hotspots Detection. 2019 Integrated Communications, Navigation and Surveillance Conference (ICNS), 1–8. doi:10.1109/icnsurv.2019.8735296.

[46] Sofaer, R. J., David, Y., Kang, M., Yu, J., Cao, Y., Yang, J., & Nieh, J. (2024). RogueOne: Detecting Rogue Updates via Differential Data-flow Analysis Using Trust Domains. Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, 1–13. doi:10.1145/3597503.3639199.

[47] Syafrizal, M., Selamat, S. R., & Zakaria, N. A. (2020). Analysis of Cybersecurity Standard and Framework Components. International Journal of Communication Networks and Information Security, 12(3), 417–432. doi:10.17762/ijcnis.v12i3.4817.

[48] Taherdoost, H. (2022). Understanding Cybersecurity Frameworks and Information Security Standards—A Review and Comprehensive Overview. Electronics (Switzerland), 11(14), 2181. doi:10.3390/electronics11142181.

[49] Korolkov, R., Kutsak, S., & Voskoboinyk, V. (2021). Analysis of deauthentication attack in IEEE 802.11 networks and a proposal for its detection. Bulletin of VN Karazin Kharkiv National University, Series "Mathematical modeling. Information technology. Automated control systems", 50, 59-71. doi:10.26565/2304-6201-2021-50-06.