EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

# Client-side rogue access-point detection using a simple walking strategy and round-trip time analysis

Songrit Kitisriworapan, Aphirak Jansang and Anan Phonphoem*

*Correspondence:
anan.p@ku.ac.th
Intelligent Wireless Network Groups (IWING), Department of Computer Engineering, Faculty of Engineering, Kasetsart University, 50 NgamWongWan Road, Bangkok, Thailand

**Abstract**

Traditional rogue access-point (AP) detection mechanisms are employed in network administration to protect network infrastructure and organization; however, these mechanisms do not protect end users from connecting to a rogue-AP. In this paper, a rogue-AP detection technique on the mobile-user side is proposed. By using a simple method involving walking, the round-trip time (RTT) and the modulation and coding scheme values are obtained, and a more accurate transmission rate for particular RTT values is thereby calculated. Further, the cleansed data are classified using the k-means method and the cumulative distribution function for the detection process. The results demonstrate that a rogue-AP can be detected with an F-measure value of up to 0.9. In the future, the proposed algorithm can be implemented as an application installed on mobile devices so that nontechnical users can detect rogue-APs.

**Keywords:** Rogue access-point detection, Wireless intrusion detection, WLAN security, Evil twin

## 1 Introduction

Currently, wireless network communication, such as 3G, 4G, and Wi-Fi, serves as a basic infrastructure for Internet access. Wi-Fi is the most preferable choice owing to its high data rate, low (or no) service charges, and high availability. However, Wi-Fi connections may not always be secure [1, 2]. User communication can be easily eavesdropped or spoofed by an adversary, particularly in public areas [3]. Currently, users cannot ensure whether their devices are directly connected to a legitimate or rogue access point (AP).

Normally, in a public area, several Wi-Fi APs are installed to support a large number of users. Figure 1 shows a possible attacking scenario, in which security vulnerabilities are exploited to compromise confidentiality, integrity, authentication, and availability (CIAA). The mobile phone of the adversary is used as a rogue-AP by broadcasting the same service set identifier (SSID) as that of the legitimate AP installed on the café ceiling. A number of users (U1, U2, and U3) attempt to connect to the Internet through their mobile phone. Unknowingly, their mobile device detects and connects to the rogue-AP, which has the strongest signal. Consequently, the adversary becomes an


**Fig. 1** System overview

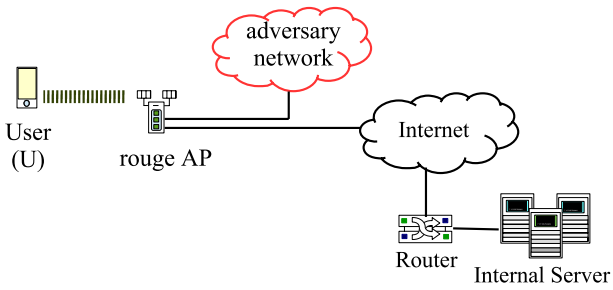
Man-in-the-middle (MITM) with the ability to spoof or modify user information or transactions.

Several approaches to counteract such attacks by deploying authentication and encryption techniques have been proposed. Since 1999, wired equivalent privacy (WEP), proposed in IEEE 802.11-1999 [4], has been a solution for encryption in Wi-Fi networks. It implements a Rivest cipher 4 (RC4) stream cipher to encrypt data-link information. However, several severe security flaws have been discovered in the protocol [1]. The major security hole is caused by the use of an initial vector (IV) and preshared key, and not by the RC4. Several vulnerable activities have been presented in [5, 6]. To mitigate WEP-related problems, the Wi-Fi Alliance proposed the concept of the temporal key integrity protocol, termed Wi-Fi protected access (WPA), in which encryption keys are changed during transmission. However, it was demonstrated in [2] that WPA vulnerabilities on the pairwise master key can lead to brute-force attacks.

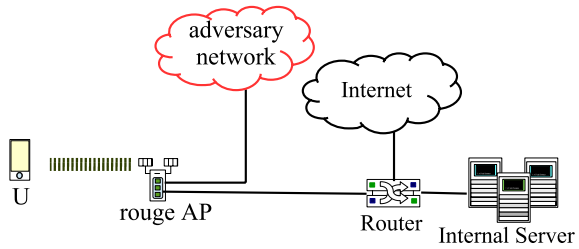
For upper-layer security, owing to certain concerns over Wi-Fi network connectivity, virtual private networking (VPN) [7] has been applied to end-to-end data encryption from the source mobile device to the destination through a public AP. However, before the VPN tunnel (or other tunnels, such as transport layer security [8] or IPsec [9]) is established, the user should first connect to either a rogue or legitimate AP. If connection is established through a rogue-AP, the rogue-AP can intentionally block the VPN tunnel, which cannot be decrypted, and allow only normal unencrypted connection, to which the user may switch, being unable to use VPN. Although data-link or upper-layer security mechanisms can be deployed, they are vulnerable to MITM attacks. In this study, we concentrate only on rogue-APs (on which the most popular MITM attacks are based) owing to their low-cost implementation [10].

As shown in Fig. 2, rogue-APs can be categorized into three types, according to their connection to the network.

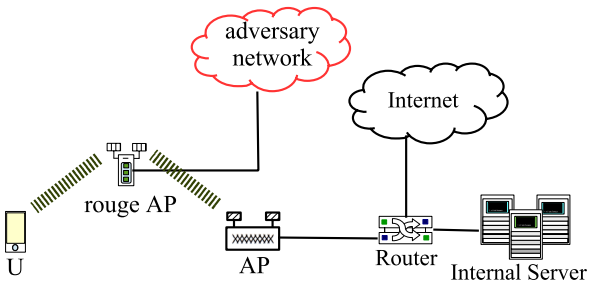
**Type 1 rogue-AP** is set up by broadcasting known SSIDs for WLAN services. Here, adversaries attempt to manipulate the connection through their own private or 3G/4G network. This type of rogue-AP can be found in public areas, such as shopping malls or tourist spots.




**a** Type 1 rogue-AP




**b** Type 2 rogue-AP




**c** Type 3 rogue-AP

**Fig. 2** Various rogue-AP categories in terms of Wi-Fi topology

**Type 2 rogue-AP** is normally deployed in organizations such as schools and universities. If a legitimate wireless network provides unsatisfactory services, internal users may illegally set up their own AP with direct connection to the LAN to gain access to some restricted services or achieve a better data rate. Although they may have no intention to attack the network, the installed AP becomes a weak point for attackers to gain unauthorized access to the internal network. However, there may be internal adversaries that intend to act as an MITM by installing the AP. Such an AP is a type 2 rogue-AP.

Additionally, adversaries without permission for LAN access can set up a **Type 3 rogue-AP** that can be located inside or outside the perimeter of the organization. Their intention is to intercept the current connection and forward it to a legitimate AP for stealing sensitive data. Type 3 rogue-APs can be easily deployed in hardware-based [11]. However, it is difficult to detect [12].

The rogue-APs can be detected in either regular (wired) or wireless LAN connections [13]. Detection for wired LAN can only be performed by the network administrator or infrastructure owners. The network administrator can use a network intrusion detection system (NIDS) [14] or hardware wireless intrusion detection system (WIDS) [15]. Nevertheless, individual users remain unaware of the legitimacy of their connections.

In a public area, several available Wi-Fi SSIDs are broadcast, and users do not know the legitimacy of these SSIDs. Additionally, they are unaware of the protection status and mechanism for the current network. Therefore, it is advisable for users to perform the detection process themselves by using their own equipment. Considering this, a simple detection process for an individual user is proposed in this study.

Normally, a user only receives the provided service, and has no privilege to access information regarding the AP or the network infrastructure. Therefore, a detection process that is independent of and nonintrusive to the infrastructure (but able to identify abnormal behavior) is proposed. This mechanism can be used by users without technical knowledge to identify rogue-APs.

In this study, a client-side rogue-AP detection method is proposed. It uses a simple walking-related mechanism to obtain the round-trip time (RTT) as well as the modulation and coding scheme (MCS). A more accurate transmission rate is calculated by using uplink and downlink packets. To detect a rogue-AP, the collected data are classified using the k-means method and the cumulative distribution function. Hence, end users can easily deploy the proposed method by themselves.

## 2 Literature review

Herein, rogue-AP detection techniques are reviewed. They can be categorized as detection on guided and unguided media.

### 2.1 Detection on guided medium

Normally, any legitimate or rogue-AP is connected to the network using a wired infrastructure of copper or fiber-optic cables passing through a switch or router. The activity of the wireless network of any user is visible in the wired network, which is also termed a guided medium and allows the network administrators to detect any suspicious behavior or type 2 rogue-APs.

Detection on guided media generally requires network-administrator privileges for observing packets and the timing differentiation between wired and wireless traffic. In [16], the difference in the maximum transmission unit of wireless network interface cards (NICs) in the wired network was used for rogue-AP identification by using the payload slicing technique.

### 2.2 Detection on unguided medium

An unguided medium refers not only to radio-frequency communication but also to infrared or ultrasonic sound waves, which are normally transmitted omnidirectionally. Both network administrators and regular end users can capture these signals and perform rogue-AP detection.

In [17], use of appropriate devices (such as LEDs) to monitor the activity of particular APs when connected to the system was proposed. Additionally, in [18], it was proposed that each wireless NIC should have a unique modulation fingerprint captured by a special device called a PARADIS sensor, which can be used in the detection mechanism.

Another widely used detection technique, particularly for Wi-Fi networks, involves capturing and analyzing IEEE802.11 wireless frames. By changing existing wireless APs in the system to temporarily act as monitoring devices for a short period [19], the beacon information from surrounding APs is collected and compared with that in a database of known APs so that type 2 rogue-APs may be identified.

By using a wireless IDS (Wi-Fi sensor node) to capture the wireless frame type with the corresponding physical-layer properties, the fingerprint of each wireless NIC can be identified from unique crystal distortions in the clock circuit of the NIC, such as clock skew [20–22].

In [20], a wireless IDS for monitoring the time stamp for beacon frames is required by the network administrator to calculate the clock skew and compare it with those in a database. By replacing the popular jiffies [23] (Linux kernel timing) with the timing synchronization function in wireless frames, the clock skew becomes more accurate at the microsecond level. In [21, 22], it was proposed that end users can monitor beacon frames and calculate the clock skew without additional devices, and in [21], the mechanism of [22] was enhanced by adding more factors affecting the clock-skew characteristics.

However, over a period of time, wireless NICs can deteriorate; this can lead to a change in the fingerprint. Therefore, the recorded identities do not remain valid and cannot be used for long-term detection.

Conversely, by implementing user-side detection with existing wireless NICs [24–28], statistical analysis of captured frames may be more suitable for detection, particularly for type 3 rogue-APs.

Furthermore, in [24], to detect evil-twin APs (rogue-APs), the relationship between RTT and MCS was preliminary studied. However, the focus was only on the effect of the average MCS and delay; further, the evaluation was performed only by simulation, with restricted coverage area of the rogue-AP.

Evil-twin APs were defined for type 3 rogue-APs in [29], where IAT distribution models for 1-hop and 2-hop were studied. In [27, 28], by sending probe packets to an AP and a specific server, different RTTs were analyzed without considering MCS.

#### *2.2.1 1-hop and 2-hop connection analysis*

Figure 3 shows the RTT timing diagram during regular operation (wireless 1-hop), in which a user connects to a legitimate AP. The time required to prepare the ping packet is denoted by $T_1$ (processing time). Then, the user station (U) waits for a time $T_2$ to access the wireless medium. The time required for each packet to be transmitted and propagated through the medium is denoted by $T_3$ and $T_4$, respectively.

Once the packet arrives at the AP, it requires a certain time to reach the server on the LAN segment. This time comprises the processing time $T_5$, accessing time $T_6$, transmission time $T_7$, and propagation time $T_8$.

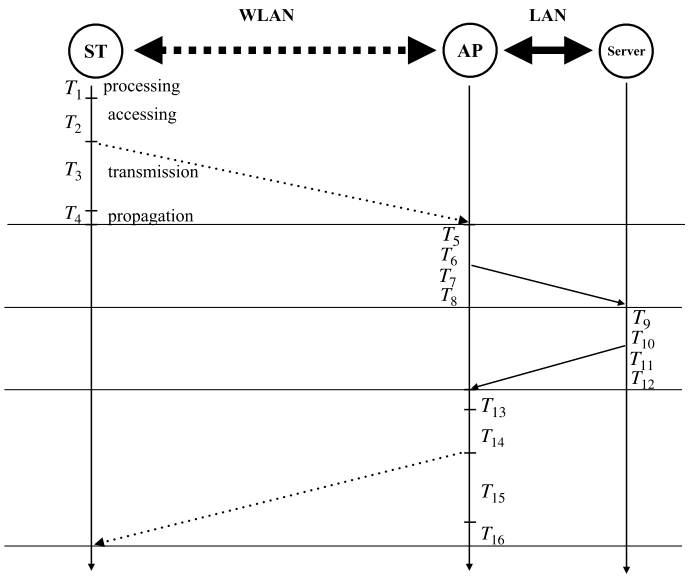



**Fig. 3** 1-Hop RTT timing diagram

On the return path, $T_9$–$T_{12}$ from the LAN segment and $T_{13}$–$T_{16}$ from the wireless segment are added.

The total RTT is as follows:

$$\Im'_L = \sum_{i=1}^{16} T_i \tag{1}$$

The processing ($T_1$, $T_5$, $T_9$, and $T_{13}$), wireless propagation ($T_4$ and $T_{16}$), LAN propagation ($T_8$ and $T_{12}$), and LAN accessing times ($T_6$ and $T_{10}$) are in the range of microseconds, and thus they are considerably small compared with the wireless access and packet transmission time. Hence, for simplicity, these timings are considered negligible in RTT calculations.

The **wireless access delay** ($T_2$ and $T_{14}$) is based on the process of distributed-coordination-function (DCF) media access control. The expected delay was analyzed in [30, 31] for the saturated case. It depends on the channel access delay, which in turn depends on the random back-off time, collision probability, success probability, and head-of-line packet transmission time. However, in our case (which is a normally unsaturated condition) [32], the expected delay $E[D]$ depends only on the mean of the uniform distribution of the contention window ($CW_{min}$):

$$E[D_{\text{unsaturated}}] \approx \tau_T + \frac{1 + CW_{min}}{2}, \tag{2}$$

where $\tau_T = 180\sigma$, and $\sigma$ represents the slot time unit for each modulation technique [33].

For example, in IEEE802.11a/g/ac, which uses orthogonal frequency-division multiplexing, $\sigma$ is 9 $\mu$s.

The **wireless packet transmission times** ($T_3$ and $T_{15}$) depend on the MCS, as shown in Eq. (3). However, the LAN packet transmission time ($T_7$ and $T_{11}$) depends on the data rate of the Ethernet technology, i.e., efficiency ($\eta = 1/(1 + 5t_{prop}/t_{trans})$, and it can be assumed to be constant.

$$T_{\text{wireless\_t}} = \frac{\text{packet size}}{\text{transmission rate(MCS)}}. \tag{3}$$

Hence, $\mathfrak{I}_L$ can be simplified to $\mathfrak{I}_L =$ accessing delay($T_2 + T_{14}$) + transmission delay($T_3 + T_{15}$), which can be rewritten as follows:

$$\mathfrak{I}_L = 2(E[D_{\text{unsaturated}}] + T_{\text{wireless\_}t}). \tag{4}$$

In the case of a type 3 rogue-AP (2-wireless hop), the RTT is shown in Fig. 4. The total ping time is increased by the accessing ($T_{R2} + T_{R3}$ ) and the transmission delay ($T_{R14} + T_{R15}$) according to the additional device, i.e., the rogue-AP. The processing and the propagation delays are negligible. The total RTT can then be expressed as follows:

$$\begin{aligned}\mathfrak{I}_R =& \mathfrak{I}_L + \text{accessing delay}(T_{R2} + T_{R3}) \\ &+ \text{transmission delay}(T_{R14} + T_{R15}) \\ &= 4(E[D_{\text{unsaturated}}] + T_{\text{wireless\_t}}).\end{aligned} \tag{5}$$

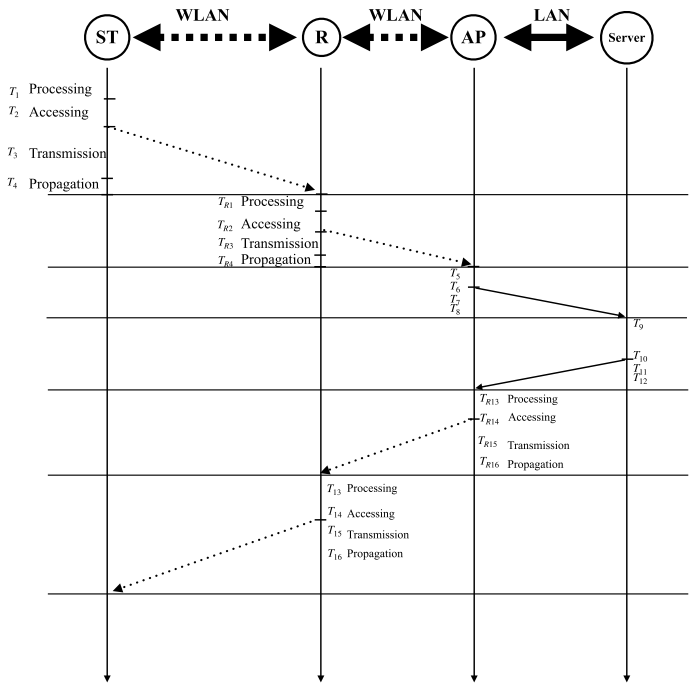



**Fig. 4** 2-Hop RTT timing diagram

## 3 Round-trip time analysis

The major concern with regard to the use of Wi-Fi, particularly in public areas, is that users do not know whether their devices are directly connected to a legitimate AP. Although certain encryption methods (such as WEP, WPA WPA-Enterprise, and WPA2) in the data link layer have been widely adopted, they may be compromised by MITM attacks.

Although Internet service or Wi-Fi hot spot providers may provide an NIDS or a WIDS, the goal of such mechanisms is to protect the infrastructure of the providers, not the end users.

To alleviate the effects of an MITM attack, the network administrator can use several monitoring tools and techniques, such as timing-behavior observation [3, 12, 26–29, 34, 35]. The administrator can send probing packets and measure their timing accuracy (transmission, propagation, and queueing delay), while they pass through devices. However, for a regular user without superuser privileges, the RTT is an interesting option for identifying a rogue-AP. The challenges of user-side detection can be found in [36].

In this study, a user-side probe-walking technique involving simple RTT measurement and analysis is proposed, whereby users can protect themselves from unintentionally connecting to rogue-APs. For a particular location, a user can receive only one data instant for each current AP connection. Therefore, to receive a sufficient number of timing instants for accurate analysis, the probe-walking technique is employed. For efficiently probing with less walking steps, the walking patterns are proposed according to the signal strength, relative angle, and number of walking steps further away from the currently connected AP. After sufficient data instants have been obtained, k-means clustering classification is used to identify a rogue-AP.

Herein, the simple RTT on a type 3 rogue-AP is analyzed by sending ping packets from a user device to an internal dedicated server in the LAN. In the absence of an MITM, devices along the pinging path and the server are legitimate APs. When a type 3 rogue-AP is present, an additional timing delay through the rogue-AP is introduced.

It should be noted that several channel access mechanisms have been defined. In IEEE 802.11 (1999) [4], the DCF and point coordination function (PCF) were proposed, whereas in IEEE802.11e [37], to enhance quality of service, the hybrid coordination function (HCF) was proposed, and HCF-controlled channel access (HCCA) and enhanced distributed channel access (EDCA) were defined. However, PCF, HCCA, and EDCA have not been implemented and widely used in real NICs. Therefore, only the DCF-mode channel access mechanism is considered in this study.

### 3.1 Preliminary study

In this study, simulations were used to determine the parameters of the proposed algorithm. Then, the proposed mechanism was evaluated in real testing scenarios.

The environmental parameters that affect RTT are the queueing, the transmission, and the propagation delay, which are represented by traffic load, MCS, and distance, respectively.

The RTT values received from ping that run in the application layer are not significantly differing (with mostly constant in certain microseconds) from the values received

**Table 1 NS3 parameters**

| Parameters | Value |
|---|---|
| Guard interval | 800 ns |
| Adaptive rate control algorithms | MinstrelHtWifiManager |
| Transmission power | 16.02 dBm |
| Bandwidth | 40 MHz |
| Frequency band | 5 GHz |
| Simulation time | 10 s |

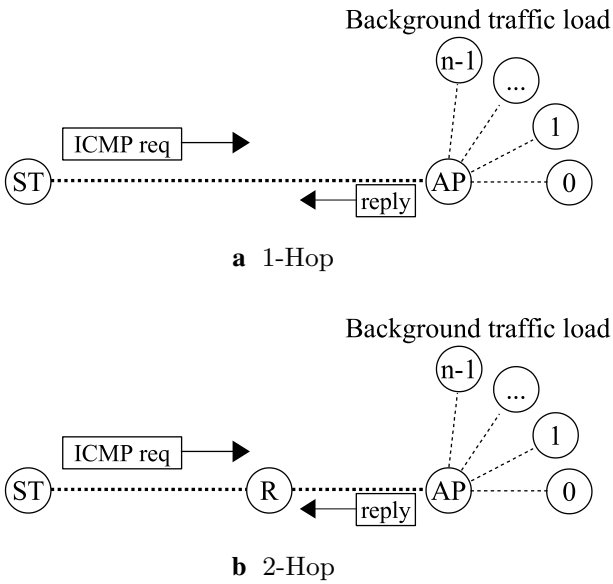



**Fig. 5** Simulation topology with background traffic load

from the MAC layer in the same machine. Moreover, implementing ping at the application layer is clearly far easier than the MAC layer.

In this work, a preliminary study of the traffic load as well as MCS effects was conducted by using the network simulator NS3 [38] version 3.29. The topologies for 1-hop and 2-hop connection are shown in Fig. 5. The simulation parameters are listed in Table 1.

In the simulation, the client machine uses the ping command to find the RTT. For each position, 100 ping packets (1472 bytes of ICMP payload) are generated at a rate of ten packets per second. The RTT is extracted from the application layer, whereas the MCS values are retrieved directly from the wireless header in the data-link frame.

### 3.2 Round-trip time and traffic load

The round-trip time primarily depends on the transmission and the access delay. The former varies according to the packet size and MCS, whereas the latter changes according to the traffic condition, which is normally unsaturated. In a saturated condition, 1-hop and 2-hop delays are closed; therefore, they cannot be used for identifying a rogue-AP. However, a saturated condition does not normally occur; as it is temporary

and exhibits excessively slow traffic, some users may decide to leave the connection, and subsequently the traffic may return to the unsaturated condition.

For the traffic load shown in Fig. 6, the user-diagram-protocol background traffic is generated by 25 users in the system. The light, medium, and heavy traffic loads are represented by 0.3, 2.4, and 24 Mbps, respectively. Note that, for a high traffic load, the RTT values for both 1-hop and 2-hop are scattered and overlapping; therefore, they are difficult to distinguish.

### 3.3 Round-trip time and modulation and coding scheme

The adaptive modulation and coding scheme (AMC) is designed for automatic MCS selection and perfectly matches the lossy connecting channel condition.

According to the IEEE 802.11 standard, from the received channel power indicator (RCPI), which is received by the NIC of the client device, the modulation and coding rate is selected as specified in Section 21.3.18.1 of [37] (receiver minimum input sensitivity). Finally, by using the parameters for VHT-MCS in Section 21.5 of [37], the MCS is selected to match the AP transmission.

It can be seen from previous research [27–29] that the average RTT is used for separating 1-hop and 2-hop without considering the MCS. Normally, to detect a rogue-AP, statistics such as mean, median, and entropy are used. However, in this study, the effects of MCS uplink and downlink are considered in the corresponding RTT, which is subsequently used for 1-hop and 2-hop classification.

Moreover, the effect of the MCS and distance is shown in Fig. 7. For IEEE802.11n, the MCS indexes of 0–7 are used for a single spatial stream, while MCS values of 8–15 are for the two spatial streams. Noticeably, IEEE 802.11ac has been deployed in our study. The MCS indexes of 0–9 are used for one or more spatial streams. Therefore, in Fig. 7a the $x$-axis should show 0–9 for the MCS values. However, throughout the experiments, no MCS index of 9 was found.

For fixed distance and variable MCS values (Fig. 7a), RTT decreases as the MCS increases. However, for a fixed MCS value with variable distance (Fig. 7b) and a

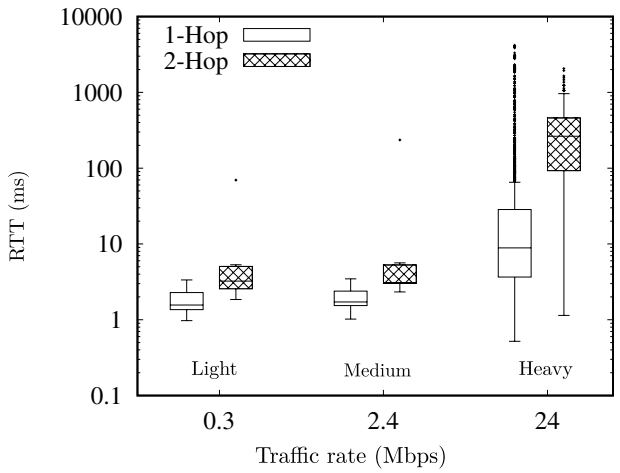
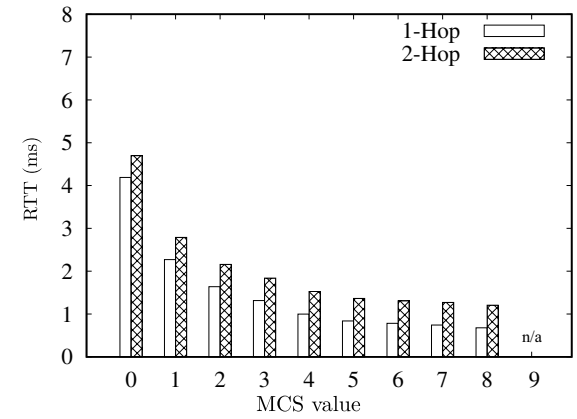
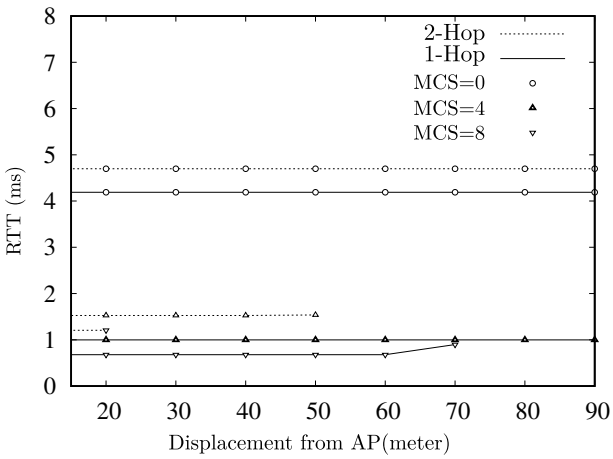



**Fig. 6** RTT analysis for various traffic loads




**a** Fixed distance with different MCS values




**b** Fixed MCS value with different distances

**Fig. 7** Effect of MCS and distance on RTT (802.11ac)

100-byte ping packet, RTT follows the MCS value but does not vary according to the distance. It should be noted that the MCS values for client-to-AP transmission might be very different from those for AP-to-client transmission. Each RTT value may be generated by various pairs of MCS values. Hence, RTT is independent of the distance and depends only on the MCS value.

An increase in signal strength causes a high MCS value, which decreases RTT. In Fig. 8, the decreasing RTT trends are the same for both 1-hop and 2-hop, and the difference increases for larger packet sizes.

The physical transmission rate (MCS-index) can be decoded from uplink and downlink packets and is used for comparisons in the table in Fig. 9 to estimate data speed. Travel to and from the speed value results in delays (RTT) data for use in illegal access points.

The proposed algorithm was tested using 80 walking patterns. Each pattern comprised three long walks, and for each of them, three short walks were performed.

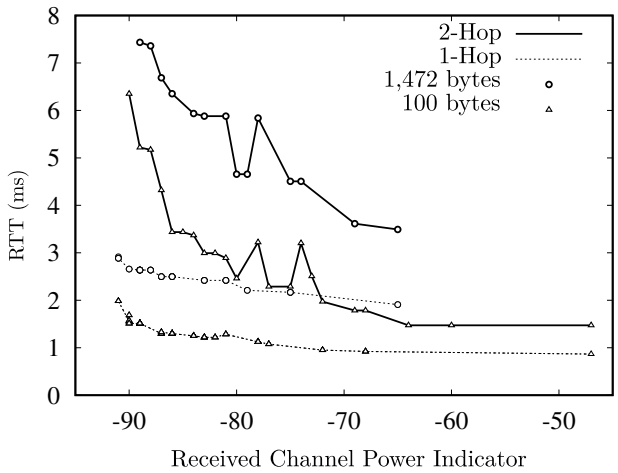



**Fig. 8** Effect of RCPI on RTT

| $MCS_{dl}$ \ $MCS_{ul}$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 13.5 | 20.25 | 27 | 33.75 | 47.25 | 60.75 | 67.5 | 74.25 | 87.75 | 96.75 |
| 1 | 20.25 | 27 | 33.75 | 40.5 | 54 | 67.5 | 74.25 | 81 | 94.5 | 103.5 |
| 2 | 27 | 33.75 | 40.5 | 47.25 | 60.75 | 74.25 | 81 | 87.75 | 101.25 | 110.25 |
| 3 | 33.75 | 40.5 | 47.25 | 54 | 67.5 | 81 | 87.75 | 94.5 | 108 | 117 |
| 4 | 47.25 | 54 | 60.75 | 67.5 | 81 | 94.5 | 101.25 | 108 | 121.5 | 130.5 |
| 5 | 60.75 | 67.5 | 74.25 | 81 | 94.5 | 108 | 114.75 | 121.5 | 135 | 144 |
| 6 | 67.5 | 74.25 | 81 | 87.75 | 101.25 | 114.75 | 121.5 | 128.25 | 144 | 150.75 |
| 7 | 74.25 | 81 | 87.75 | 94.5 | 108 | 121.5 | 128.25 | 135 | 148.5 | 157.5 |
| 8 | 87.75 | 94.5 | 101.25 | 108 | 121.5 | 135 | 144 | 148.5 | 162 | 171 |
| 9 | 96.75 | 103.5 | 110.25 | 117 | 130.5 | 144 | 150.75 | 157.5 | 171 | 180 |

$\hat{\beta_L}$ $\hat{\beta_M}$ $\hat{\beta_H}$

**Fig. 9** Grouping of average uplink and downlink transmission rates (802.11ac)

Three walking-pattern examples, namely a walking path that is only connected to the legitimate AP ($w1$), partially connected to the rogue-AP ($w2$), and fully connected to the rogue-AP ($w3$), are shown in Fig. 12. The results for $w1$, $w2$, and $w3$ are shown in Fig. 14.

## 4 Proposed method

According to preliminary result shown in Fig. 6, the graph shows the RTT for the various load conditions. For the light and medium load, the variance of RTT is quite small. However, the variance of RTT becomes considerably large for both 1-Hop and 2-Hop for the heavy-load condition. The 1-hop RTT might become larger than the 2-hop RTT. Noticeably, [27, 28] approach cannot accurately detect in such high-traffic-load or high-variance conditions. Obviously, the detection performance cannot be improved if the environment or network conditions remain the same. Hence, the motivation of the proposed algorithm is to create more chances for collecting more RTT data with various MCS values from nearby available access points by simple walking. With more RTT and MCS values from different conditions, the detection performance is improved.

The proposed protocol uses the AMC in a regular NIC. Therefore, walking is required to change the MCS; otherwise (without using AMC), it is possible to modify the wireless

NIC driver to change the MCS and collect the RTT data without performing any short walk.

The implementation is obviously feasible in all operating systems for the notebook or the mobile device. In case no driver API of the user's NIC card is available, the MCS cannot be controlled by the user and is automatically selected by the driver. This is why the proposed mechanism required the user to wander a few steps (short walk). The intension is to force the device to change MCS values and generate more data for detecting mechanism. With the availability of the driver API of the NIC card, a simple programming script can be implemented for monitoring the clear channel assessment (CCA) and controlling for required MCS values. Therefore, with no user's movement (short walk) required, the connecting data from various MCS values can promptly be collected. However, the long walk is still required for creating chances for connecting to different access points in the area.

To classify a rogue-AP effectively, the proposed protocol requires the user to move around the given area. The goal is to collect different MCS pairs and their corresponding RTT for statistical analysis. Normally, if the RCPI (in dBm) is lower than a particular threshold, the user device attempts to connect to a new available AP [39].

A walking pattern consisting of short and long walks is proposed. For each short walk, the RCPI for each location is altered, thus changing the MCS. For each particular location, the user may connect to a legitimate or rogue-AP. Therefore, a long walk increases the chance to connect to different APs.

According to [40], owing to the high packet error rate, clients associate with a new AP when they are more than 10 m away from the current AP.

In a realistic implementation, for a long walk, users only require walking approximately 10 m (10–15 walking steps), whereas a few meters (2–3 walking steps) are required for a short walk. To minimize the walking distance and shorten the probing period, an appropriate walking pattern should be defined.

Usually, a high RCPI indicates that the user is close to an AP. Small walking steps can only cause the MCS to change easily. Therefore, to collect relevant data, a few short moves should be performed. Then, the user should move further away by a distance of approximately 10–15 m from the current position and repeatedly perform a few short walks. The protocol proposes three long walks. For each long walk, three short walks should be performed, as shown in Algorithm 2.

### 4.1 1-Hop and 2-hop classification

The detection algorithm proposed in Algorithm 2 consists of two parts: k-means classification and analysis of the cumulative distribution function (CDF).

Several clustering algorithms assume independent data. However, the RTT and MCS values depend on each other. Moreover, in this study, unsupervised learning is required. Hierarchical clustering and k-means are types of unsupervised learning with dependency data. However, the dendrogram of hierarchical clustering may be inappropriate for determining the differences among clusters. Hence, the k-means algorithm is a more suitable clustering method for our scattering data.

From the experimental results shown in Fig. 7b, at all the locations with the same MCS value, the RTT is the same. Different RTT values with the same MCS indicate that a

2-hop connection occurs. Accordingly, the two-cluster k-means classification technique [41] is used.

Given that vector $\mathbb{C} = [c_1, c_2, \ldots, c_N]$ is the collection of $N$ data points, let $c_i = [rtt_i, mcs_i]$ represent a two-dimensional data point between RTT and MCS.

Let $\Omega = [\omega_1, \omega_2, \ldots, \omega_K]$ be a set of clusters. In our case, only two clusters (1-hop and 2-hop) are present. Therefore, the number of clusters $K$ is 2, implying that $\Omega = [\omega_1, \omega_2]$.

The objective function $E$ minimizes the sum of the distances from each data point to the cluster centroid:

$$E = \frac{1}{2} \sum_{k=1}^{K=2} \sum_{i \in k} \|c_i - \omega_k\|^2 \tag{6}$$

The complexity of k-means is $\mathcal{O}(Knt)$. Therefore, in our implementation, where $K = 2$ and $n$ and $t$ are the number of iterations and number of data points, respectively, the complexity is $\mathcal{O}(2nt)$.

### 4.2 Proposed rogue-AP detection algorithm

Herein, a user-side rogue detection algorithm is proposed. To collect data for the analysis, the walking pattern is defined in Algorithm 1. Once the walking pattern is defined and data collection is complete, Algorithm 2 is used for rogue-AP identification.

In Algorithm 1, a user performs three short and some long walks for collecting data from different locations. For each short walk location, 100 pings are performed from the user machine to a fixed server.

Each ping results in one RTT value. However, each ping comprises uplink and downlink packets that may be transmitted with different MCS values (0–9). Each MCS can be mapped to a data rate according to the IEEE802.11ac standard [37]. For example, for a single spatial stream with a bandwidth of 40 MHz, $f(\text{MCS} = 0) = 13.5$ Mbps. Although 100 possible pairs of uplink and downlink data rates can be averaged, as shown in Eq. (7), only 32 unique average data rates can be identified.

It should be noted that VHT-MCS can be retrieved from the packet along with other information provided by the AP, such as bandwidth, number of spatial streams, and guard intervals.

To calculate a representative data rate for a ping, the data rate of uplink and downlink packets should be averaged as follows:

$$\overline{\text{rate}} = \frac{f(\text{MCS}_{ul}) + f(\text{MCS}_{dl})}{2}. \tag{7}$$

After three short walks, outliers are statistically removed from the 300 ping results for data cleansing (line 11). All tuples ($\hat{d}$) are classified into three groups: high (H), medium (M), and low (L) data rates (line 13). The H, M, and L data-rate groups are defined as greater than 121.5 Mbps, between 74.25 and 121.5 Mbps, and lower than 74.25 Mbps, respectively, as shown in Fig. 9. Then, the user moves to a new location (long walk) and repeatedly performs another three short walks until the required conditions are satisfied (line 3).

**Algorithm 1** Data gathering

1: $LW \leftarrow TRUE, N_{ShortWalk} \leftarrow 3$
2: $w_L \leftarrow 0, w_S \leftarrow 0, D \leftarrow \{\phi\}$
3: **while** $LW == TRUE$ or $w_L < 5$ **do**
4: **while** $w_S < N_{ShortWalk}$ **do**
5: pcap ← Capture wireless ping traffic
6: $rate, \overline{rtt} \leftarrow$ Extract_Rate&RTT($pcap$)
7: $D$. Insert($rate, \overline{rtt}$)
8: ShortWalk()
9: $w_S \leftarrow w_S + 1$
10: **end while**
11: $\hat{d} \leftarrow$ Data_Cleansing($D$)
12: */* group RTT into three data-rate groups {H,M,L} */*
13: $\hat{\beta}_H, \hat{\beta}_M, \hat{\beta}_L \leftarrow \hat{d}$
14: **if** No $\hat{\beta}_H$ or $\hat{\beta}_M$ exists **then**
15: Longwalk()
16: LW← TRUE
17: $w_L \leftarrow w_L + 1$
18: **else**
19: LW← FALSE
20: **end if**
21: **end while**
22: **return** $\hat{\beta}_H, \hat{\beta}_M, \hat{\beta}_L$

**Algorithm 2** Rogue identification

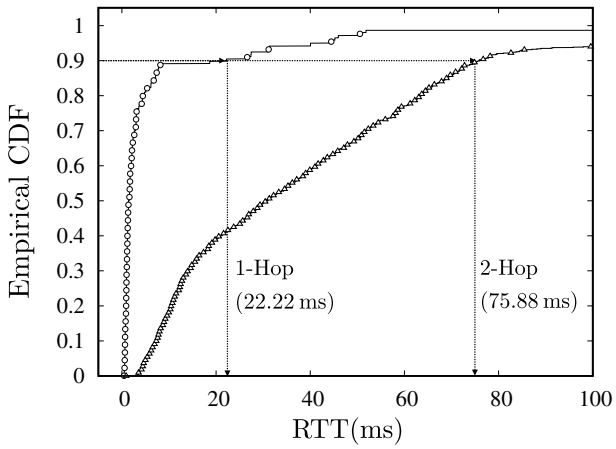
f

1: */*input parameters: $\hat{\beta}_H, \hat{\beta}_M, \hat{\beta}_L, T_\gamma, CI$ */*
2: **for** $c$ in $\{\hat{\beta}_H, \hat{\beta}_M\}$ **do**
3: */* Detect with k_means */*
4: $\gamma \leftarrow$ k_mean($c$)
5: $\gamma_{\text{upper}} \leftarrow max(\gamma)$
6: $\gamma_{\text{lower}} \leftarrow min(\gamma)$
7: $\Gamma = \frac{\gamma_{\text{upper}}}{\gamma_{\text{lower}}}$
8: **if** $\Gamma > T_\gamma$ **then**
9: **return** (rogue-AP is detected)
10: **else**
11: */* Detect with CDF */*
12: */* CI = Selected Confidence Interval */*
13: **if** CDF_Value($\hat{\beta}_L, CI$) > CDF_Value($c, CI$) **then**
14: **return** (rogue-AP is detected)
15: **end if**
16: **end if**
17: **end for**
18: **return** (No rogue-AP is detected)

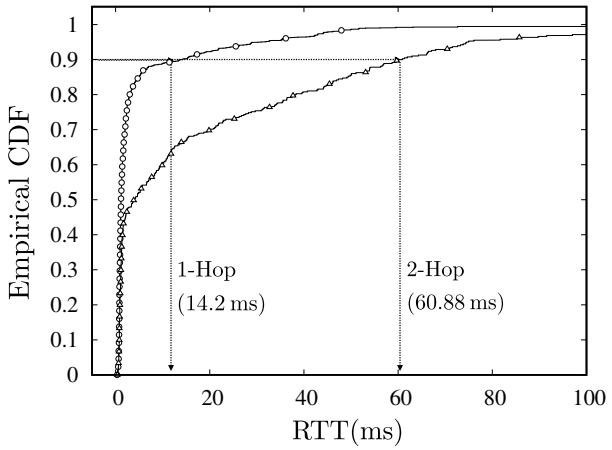
Once the data-gathering phase is completed, the tuples ($\hat{\beta}_H$, $\hat{\beta}_M$, $\hat{\beta}_L$) of the average data rate and RTT are analyzed in the rogue identification phase, as shown in Algorithm 2.

Subsequently, $\hat{\beta}_H$ is classified into two clusters according to the k-means method shown in lines 3–10. Let the distance ratio $\frac{\gamma_{\text{upper}}}{\gamma_{\text{lower}}}$ of two cluster centroids (in Y-axis) be $\Gamma$. If $\Gamma$ is greater than a threshold value $T_\gamma$, then there is a rogue-AP; otherwise, the nonexistence of a rogue-AP cannot be concluded. Therefore, the CDF of RTT is used for further investigation (lines 12–18).

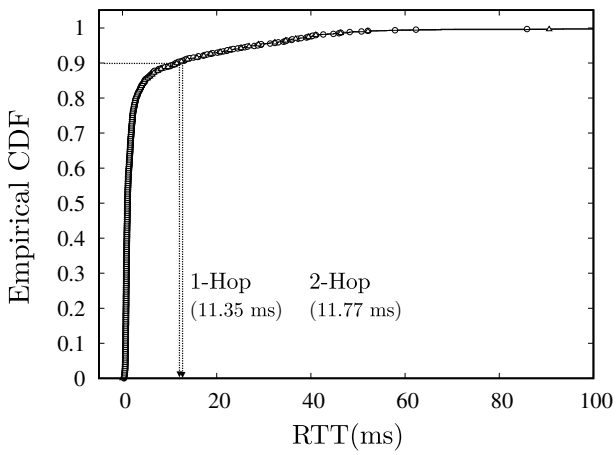
As seen from the preliminary results, for 1-hop, the CDF values for all data rates are approximately the same. However, for 2-fop, the CDF values are significantly different, particularly for low data rates, as shown in Fig. 10a.




**a** Low transmission-rate group ($\hat{\beta}_L$)




**b** Medium transmission-rate group ($\hat{\beta}_M$)




**c** High transmission-rate group ($\hat{\beta}_H$)

**Fig. 10** CDF in RTT analysis

Therefore, the CDF of the low data rate is used for confirming the existence of a rogue-AP. If the CDF value of $\hat{\beta}_L$ is greater than the empirical CDF value of $\hat{\beta}_H$, then there is a rogue-AP.

If no rogue-AP is detected with the high data rate ($\hat{\beta}_H$), then the algorithm is repeated with medium data rate ($\hat{\beta}_M$). Finally, it can be concluded with high probability that there is no rogue-AP in the area if both $\hat{\beta}_H$ and $\hat{\beta}_M$ are not detected.

## 5 Performance evaluation

A test bed was set up in a real environment to evaluate the performance of the proposed algorithm. Ping (ICMP) and sniffer software was used for gathering the RTT values on the client machine, running MacOS. Moreover, tcpdump version 4.9.2-3, running on Ubuntu 18.04, was used for extracting the MCS information from the captured files. For k-means and CDF evaluation, R version 3.4.4 was used.

The basement of the Computer Engineering building was selected as our test site. Eight legitimate Cisco APs were placed in six zones, as shown in Fig. 11. To imitate a rogue-AP scenario, two Linksys APs were employed.

All legitimate access points are in regular use, provided, and monitored by the office of computer services of the university. The channel width has been set to 40 MHz with the strict channel allocation plan according to the standard UNII-1 and UNII-3. Hence, there is no self-interference among legitimate APs in the experiments.

All legitimate APs were deployed using IEEE 802.11ac with the same SSID. According to AMC, the user device (notebook computer) would connect to the AP with the strongest received signal strength. The ping packet size was fixed at 1472 bytes.

The proposed algorithm was tested using 80 walking patterns. Each pattern comprised three long walks. For each long walk, three short walks were performed.

For statistical analysis appropriation, 20 fully legitimate AP, 30 partially rogue-AP, and 30 fully rogue-AP patterns are performed.

The rogue detection problem can be classified as an imbalanced classification. The majority class of probed access points is the legitimated AP or negative test results, while the minority class is rogue-AP or positive test result. By showing only precision

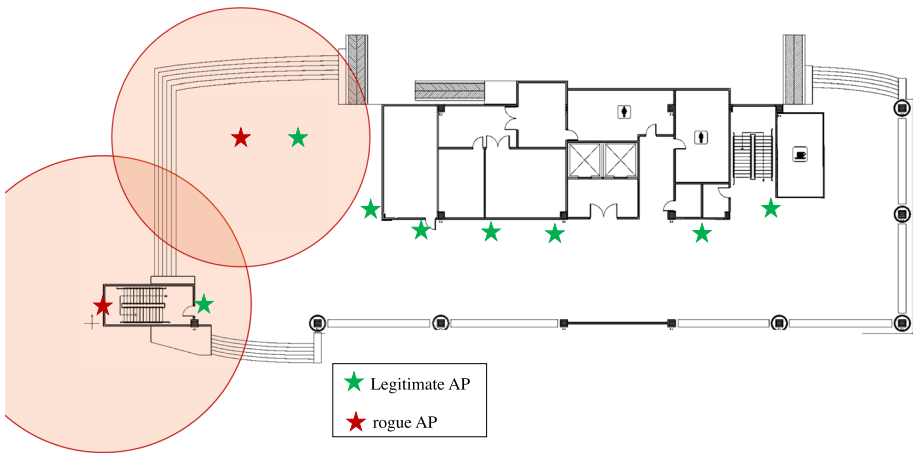



**Fig. 11** Legitimate and rogue-AP locations at the test site

or recall, the good precision with poor recall or vice versa cannot capture all properties. Hence, F-measure combines the precision and recall to a single measure that can capture both properties which is quite suitable in our case.

From the algorithm, the user's moving time is composed of two parts: the short walk and the long walk. For each long walk, the user will perform three short walks. For three short walks, the time spent is around 15 seconds, which comes from a few step walks (around 5 seconds) and 100 ping time (around 10 seconds). For each long walk, the time for the user's movement is around 8 seconds for 10–15 walking steps at the average speed of 1.4 m/s. The algorithm proposed that at most five long walks are performed (normally, three long walks are enough in the experiments). Hence, the total user's moving time becomes $5 * (8 + 15) = 115$ seconds or around 1.9 min.

Three walking pattern examples, namely a walking path that only connected to the legitimate AP ($w1$), partially connected to the rogue-AP ($w2$), and fully connected to the rogue-AP ($w3$), are shown in Fig. 12. The results for $w1$, $w2$, and $w3$ are shown in Fig. 14.

## 6 Results and discussion

For each walking pattern, the RTT results ($\hat{d}$) are scattered with the corresponding transmission rate as shown in Fig. 13. The results are then grouped as high, medium, and low, as shown in Fig. 14a.

For $w1$ (no rogue-AP, Fig. 14a), the $\Gamma$ values for all transmission-rate groups are close to 1. Similarly, for $w3$ (fully connected to the rogue-APs, Fig. 14c), only high transmission rate was detected, with a $\Gamma$ value of 1.5.

For $w2$ (partial rogue-AP: Fig. 14b), the $\Gamma$ ratio is close to 1 for the medium and high transmission-rate groups. However, the value of $\Gamma$ is approximately 2.3 only in the low transmission-rate group. The algorithm revealed that a rogue-AP was detected because the mixture of 1-hop and 2-hop connections causes the value of $\Gamma$ to be greater than 2, as derived from the test results.

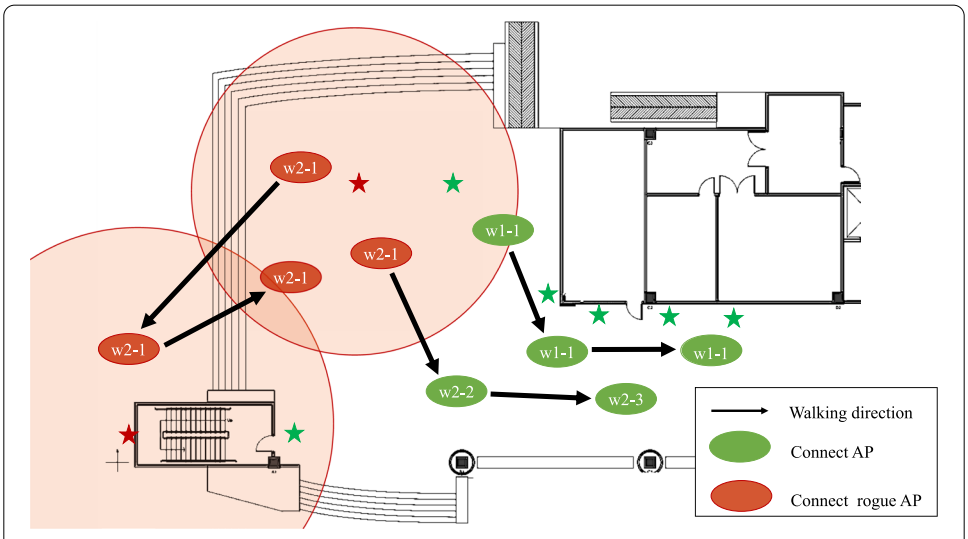



**Fig. 12** Walking-pattern examples during the test period

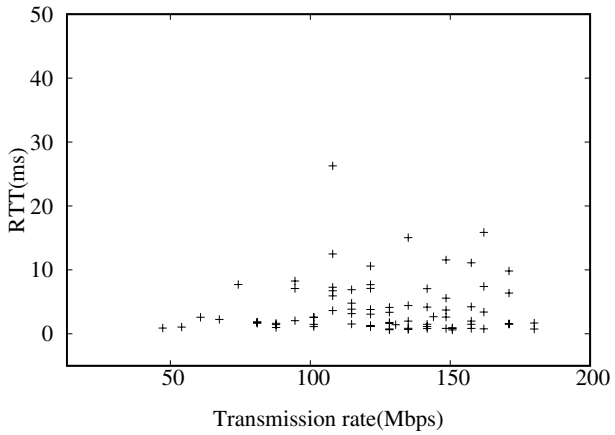



**Fig. 13** Scattering RTT values of walking pattern *w*1

From a Γ ratio close to 1, purely 1-hop or 2-hop connections can occur. The algorithm can perform additional investigation by using the CDF values. It should be noted that purely 2-hop connections are rarely observed in reality, owing to the long walk specified in the algorithm.

Experiments were conducted to confirm that the CDF values used in the algorithm are suitable for implementation. In Fig. 10, corresponding to a CDF value of 0.9, the results demonstrate that the 1-hop and 2-hop RTT values are significantly different in both the low $\hat{\beta}_L$ (Fig. 10a) and the medium $\hat{\beta}_M$ (Fig. 10b) transmission-rate groups. However, there is no difference in the RTT value for the high $\hat{\beta}_H$ (Fig. 10c) transmission-rate group.

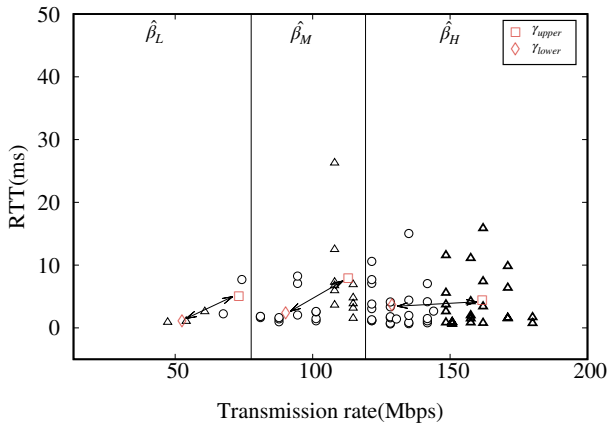
For the 80 walking patterns, the precision and recall values were extracted from the dataset to calculate the F-measure.

From the test results, the suitable setting parameters ($T_\gamma$, confidence interval (CI) of $\hat{\beta}_H$ and CI of $\hat{\beta}_L$) for Rogue identification were determined. It was demonstrated that for different fixed threshold $T_\gamma$ values of 2.5, 5.0, and 7.5 (Fig. 15), the trend of the F-measure with $T_\gamma = 7.5$ and CI of $\hat{\beta}_H = 0.75$ indicates the best performance among all testing $\hat{\beta}_H$ values (Fig. 15c).

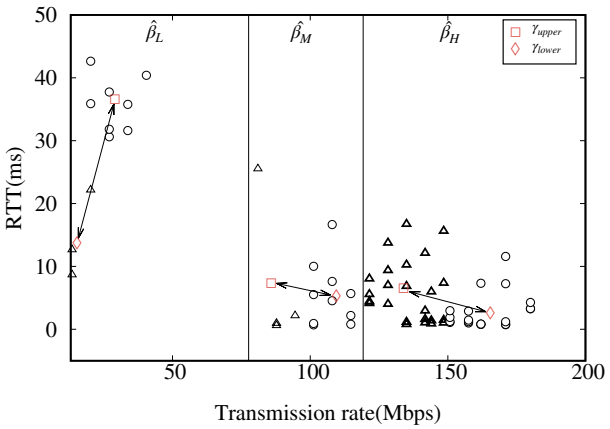
### 6.1 Complexity analysis

The proposed mechanism comprises two algorithms: data gathering and rogue identification. In the former, for each long walk, three short walks are required. The number of ICMP packets for up to five long walks is sufficient for rogue identification. Hence, the running time for this algorithm can be approximated as $\mathcal{O}(5 \times 3 \times \frac{n}{3}) \approx \mathcal{O}(5n)$, where $n$ is the number of required ICMP packets per one long walk. All collected data are then fed to the rogue identification algorithm.

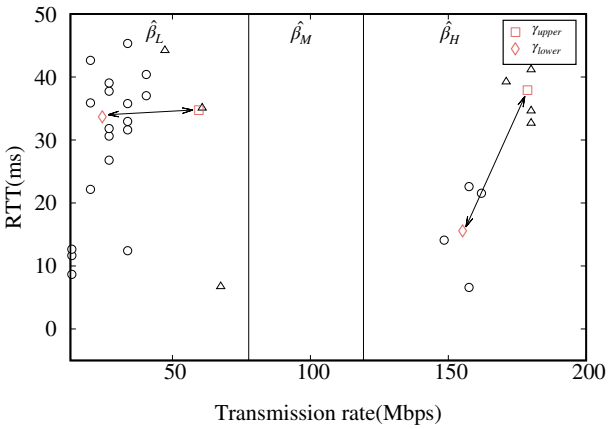
In the rogue identification algorithm, two out of three data-rate groups ($\hat{\beta}_H, \hat{\beta}_M$) are selected for performing k-means. The running time of two-group k-means is $\mathcal{O}(2 \times \frac{2}{3}(5n) \times t)$, where $t = 5$ is the selected number of iterations that is the same order as before. Then, the CDF analysis consists of sorting and array search with the $\mathcal{O}(n \log n + 1)$. Hence, the total running time is $\mathcal{O}(2 \times \frac{2}{3}(5n) \times 5 + n \log n + 1) \approx \mathcal{O}(n \log n)$




**a** Only connected to legitimate AP ($w1$)



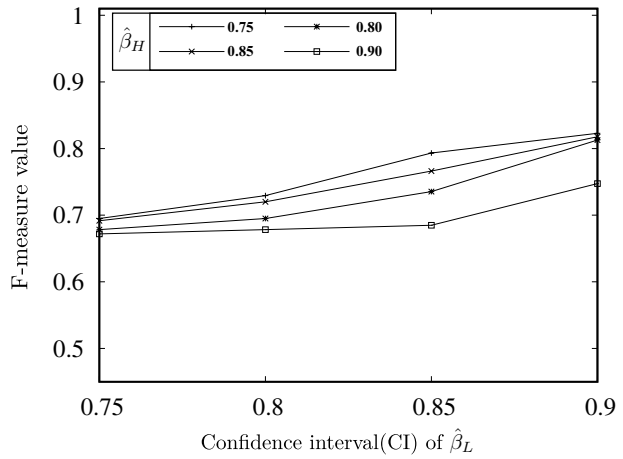

**b** Partially connected to the rogue-AP ($w2$)



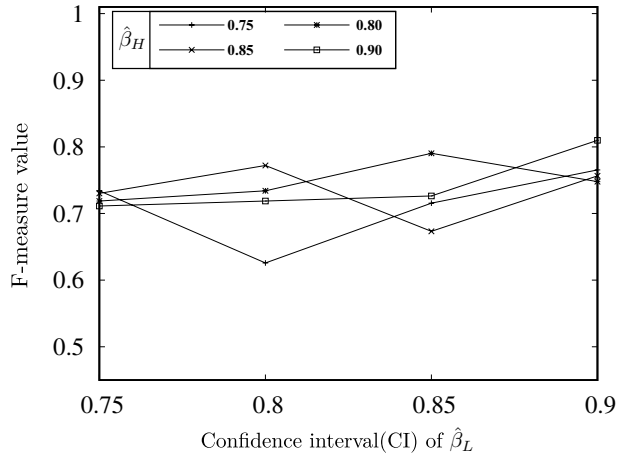

**c** Fully connected to the rogue-APs ($w3$)

**Fig. 14** Distance ratio of two cluster centroids in each data-rate group for selected walking paths

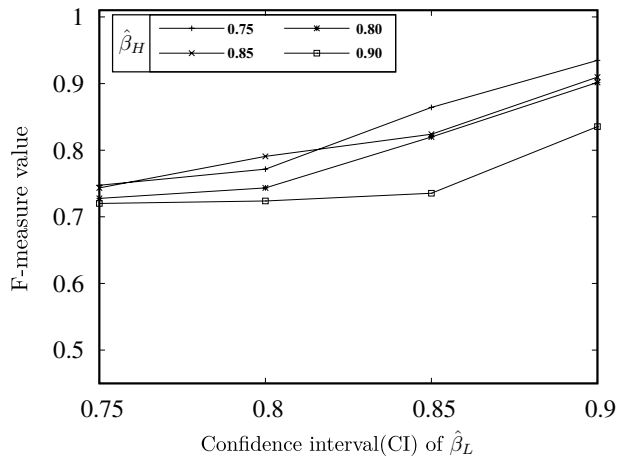
Previous client-side rogue-detection techniques [24, 27–29] were compared with the proposed mechanism, as shown in Table 2. For Yang et al. [29], the calculation is based on $n$ data points and involves determining the argument of the minimum (argmin).




**a** Fixed threshold $T_\gamma = 2.5$




**b** Fixed threshold $T_\gamma = 5.0$




**c** Fixed threshold $T_\gamma = 7.5$

**Fig. 15** Detection evaluation from various $T_\gamma$ values

**Table 2 Running-time analysis**

| Method | Running time |
|---|---|
| Yang et al. [29] | $\mathcal{O}(n^2 \log n)$ |
| Kitisriworapan et al. [24] | $\mathcal{O}(n^2 \log n)$ |
| Han et al. [27, 28] | $\mathcal{O}(n^2)$ |
| Ours | $\mathcal{O}(n \log n)$ |

Hence, assuming the best searching algorithm, the running time is approximately $\mathcal{O}(n^2 \log n)$.

In Kitisriworapan et al. [24], in $n$ walking steps, the highest MCS and RTT are determined with a complexity of $\mathcal{O}(n(2n \log n))$, followed by MCS searching involving different RTT, with a complexity of $\mathcal{O}(n)$. The running time is therefore $\mathcal{O}(n(2n \log n) + n) \approx \mathcal{O}(n^2 \log n)$.

In the study by Han et al. [27, 28], based on both probe requests and DNS packets, the algorithm consists of three parts: rogue detection, outlier filtering, and standard deviation calculation. With $n$ rounds of rogue detection, the running time is $\mathcal{O}(n)$. For outlier filtering, priority queue search is used. Based on [42], the running time can be derived as $\mathcal{O}(n \log n)$. For the standard deviation computation used for estimating the threshold, the running time is $\mathcal{O}(n^2)$. Hence, the overall complexity is $\mathcal{O}(n^2 + n \log n + n) \approx \mathcal{O}(n^2)$.

## 7 Conclusion

Normally, with the same transmission rate, the RTTs of 1-hop and 2-hop Wi-Fi connections are significantly different. However, in this study, our preliminary investigations demonstrated that owing to the AMC, the transmission rate might frequently change at a location. Therefore, the one-way transmission rate alone cannot be used to represent the corresponding RTT value. Accordingly, in the proposed algorithm, the average uplink and downlink MCS values are used to calculate the transmission rate.

The proposed mechanism uses a simple walking data-gathering method for the RTT and MCS values. Subsequently, all RTT values with the average transmission rate are categorized into three groups (high, medium, and low). Each group is classified into two clusters by the k-means method. By comparing the distance ratio of two cluster centroids on the Y-axis with the threshold value, a rogue-AP can be detected once this ratio is greater than the threshold; otherwise, a conclusion cannot be made. Therefore, the CDF of the RTT is used for further investigation. The results demonstrate that, with suitable CI setting values for high and low transmission-rate groups, rogue-APs can be detected with an F-measure value up to 0.9.

Obviously, the proposed mechanism requires extra work from the user. However, within 2 min from just a few steps movement around the area where the user will become more confident to connect to the available untrusted Wi-Fi access point for 1 or 2 h, the trade-off becomes worth it.

In the future, the proposed algorithm could be implemented as a mobile application. Nontechnical users would connect to any available hot spot with high confidence by performing the rogue-AP detection themselves. Moreover, in public locations, Wi-Fi services may be offered by different providers using the same channel configuration,

and interference may cause various delays depending on the channel access condition. Hence, the surrounding interference signal should be considered in the RTT analysis.

**Abbreviations**

AP: Access point; AMC: Adaptive modulation and coding; CDF: Cumulative distribution function; CIAA: Confidentiality, integrity, authentication, and availability; DCF: Distributed coordination function; EDCA: Enhanced distributed channel access; HCCA: HCF-controlled channel access; HCF: Hybrid coordination function; ICMP: Internet control message protocol; IV: Initial vector; LED: Light-emitting diode; MCS: Modulation and coding scheme; MITM: Man-in-the-middle; NIC: Network interface card; NIDS: Network intrusion detection systems; PCF: Point coordination function; RC4: Rivest cipher 4; RCPI: Received channel power indicator; RTT: Round-trip time; SSID: Service set identifier; TKIP: Temporal key integrity protocol; VPN: Virtual private network; WEP: Wired equivalent privacy; WIDS: Wireless intrusion detection systems; WPA: Wi-Fi protected access.

**Acknowledgements**

Not applicable.

**Authors' contributions**

SK, AJ, and AP contributed to the design and implementation of the research, the analysis of the results, and the writing of the manuscript. SK conducted the simulations and the experiments. All authors read and approved the final manuscript..

**Funding**

This research has been partially supported by Kasetsart University Research and Development Institute, Ref.2017/2560: P-Y(D)144.60.

**Availability of data and materials**

The datasets used or analyzed in this study are available from the corresponding author on reasonable request.

**Competing interests**

The authors declare that they have no competing interests.

Received: 11 December 2019 Accepted: 13 November 2020
Published online: 11 December 2020

**References**

1. N. Borisov, I. Goldberg, D. Wagner, Intercepting mobile communications: the insecurity of 802.11, in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking* (ACM, 2001), pp. 180–189
2. E. Tews, M. Beck, Practical attacks against WEP and WPA, in *Proceedings of the Second ACM Conference on Wireless Network Security. WiSec '09* (ACM, New York, NY, USA, 2009), pp. 79–86. https://doi.org/10.1145/1514274.1514286
3. O. Nakhila, M.F. Amjad, E. Dondyk, C. Zou, Gateway independent user-side wi-fi Evil Twin Attack detection using virtual wireless clients. Comput. Secur. **74**, 41–54 (2018). https://doi.org/10.1016/j.cose.2017.12.009
4. I.W. Group, et al., Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: higher-speed physical layer extension in the 2.4 GHz band. ANSI/IEEE Std 802.11 (1999)
5. V. Kumkar, A. Tiwari, P. Tiwari, A. Gupta, S. Shrawne, Vulnerabilities of Wireless Security protocols (WEP and WPA2). Int. J. Adv. Res. Comput. Eng. Technol. IJARCET **1**(2), 34 (2012)
6. Aircrack-ng: Aircrack-ng is a complete suite of tools to assess WiFi network security (2018). https://www.aircrack-ng.org
7. P. Knight, C. Lewis, Layer 2 and 3 virtual private networks: taxonomy, technology, and standardization efforts. IEEE Commun. Mag. **42**(6), 124–131 (2004). https://doi.org/10.1109/MCOM.2004.1304248
8. E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.3 (2018). https://tools.ietf.org/html/rfc8446
9. Frankel, Krishnan, IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap (2011). https://tools.ietf.org/html/rfc6071
10. A. Bartoli, E. Medvet, F. Onesti, Evil twins and WPA2 enterprise: a coming security disaster? Comput. Secur. **74**, 1–11 (2018). https://doi.org/10.1016/j.cose.2017.12.011
11. R. Jang, J. Kang, A. Mohaisen, D. Nyang, Catch me if you can: Rogue access point detection using intentional channel interference. IEEE Trans. Mob. Comput. **19**(5), 1056–1071 (2019)
12. Y. Song, C. Yang, G. Gu, Who is peeping at your passwords at starbucks? 2014; to catch an evil twin access point, in *2010 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 323–332 (2010). https://doi.org/10.1109/DSN.2010.5544302
13. R. Beyah, A. Venkataraman, Rogue-access-point detection: challenges, solutions, and future directions. IEEE Secur. Priv. **9**(5), 56–61 (2011)
14. H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: a comprehensive review. J. Netw. Comput. Appl. **36**(1), 16–24 (2013). https://doi.org/10.1016/j.jnca.2012.09.004
15. S.V. Radhakrishnan, A.S. Uluagac, R. Beyah, GTID: a technique for physical device and device type fingerprinting. IEEE Trans. Depend. Secure Comput. **12**(5), 519–532 (2015). https://doi.org/10.1109/TDSC.2014.2369033
16. C.D. Mano, A. Blaich, Q. Liao, Y. Jiang, D.A. Cieslak, D.C. Salyers, A. Striegel, RIPPS: rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning. ACM Trans. Inf. Syst. Secur. **11**(2), 2–1223 (2008). https://doi.org/10.1145/1330332.1330334

17. V. Roth, W. Polak, E. Rieffel, T. Turner, Simple and effective defense against evil twin access points, in *Proceedings of the First ACM Conference on Wireless Network Security* (ACM, 2008), pp. 220–235
18. V. Brik, S. Banerjee, M. Gruteser, S. Oh, Wireless device identification with radiometric signatures, in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking* (ACM, 2008), pp. 116–127
19. S. Srilasak, K. Wongthavarawat, A. Phonphoem, Integrated wireless rogue access point detection and counterattack system, in *International Conference on Information Security and Assurance, 2008. ISA 2008* (IEEE, 2008), pp. 326–331
20. S. Jana, S.K. Kasera, On fast and accurate detection of unauthorized wireless access points using clock skews. IEEE Trans. Mob. Comput. **9**(3), 449–462 (2010)
21. F. Lanze, A. Panchenko, B. Braatz, T. Engel, Letting the puss in boots sweat: detecting fake access points using dependency of clock skews on temperature, in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security. ASIA CCS '14* (ACM, New York, NY, USA, 2014), pp. 3–14. https://doi.org/10.1145/2590296.2590333
22. C. Arackaparambil, S. Bratus, A. Shubina, D. Kotz, On the reliability of wireless fingerprinting using clock skews, in *Proceedings of the Third ACM Conference on Wireless Network Security* (ACM, 2010), pp. 169–174
23. K. Salah, A. Kahtani, Performance evaluation comparison of Snort NIDS under Linux and Windows Server. J. Netw. Comput. Appl. **33**, 6–15 (2010). https://doi.org/10.1016/j.jnca.2009.07.005
24. S. Kitisriworapan, A. Jansang, A. Phonphoem, Evil-twin detection on client-side, in *2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (IEEE, 2019), pp. 718–721
25. C.-M. Chen, Y.-H. Chen, Y.-H. Lin, H.-M. Sun, Eliminating rouge femtocells based on distance bounding protocol and geographic information. Expert Syst. Appl. **41**(2), 426–433 (2014). https://doi.org/10.1016/j.eswa.2013.07.068
26. F.-H. Hsu, C.-S. Wang, Y.-L. Hsu, Y.-P. Cheng, Y.-H. Hsneh, A client-side detection mechanism for evil twins. Comput. Electr. Eng. **59**, 76–85 (2015)
27. H. Han, B. Sheng, C.C. Tan, Q. Li, S. Lu, A measurement based rogue AP detection scheme, in *INFOCOM 2009* (IEEE, 2009), pp. 1593–1601. https://doi.org/10.1109/INFCOM.2009.5062077
28. H. Han, B. Sheng, C.C. Tan, Q. Li, S. Lu, A timing-based scheme for rogue ap detection. IEEE Trans. Parallel Distrib. Syst. **22**(11), 1912–1925 (2011). https://doi.org/10.1109/TPDS.2011.125
29. C. Yang, Y. Song, G. Gu, Active user-side evil twin access point detection using statistical techniques. IEEE Trans. Inf. Forensics Secur. **7**(5), 1638–1651 (2012). https://doi.org/10.1109/TIFS.2012.2207383
30. B.-J. Kwak, N.-O. Song, M. Miller, Performance analysis of exponential backoff. IEEE/ACM Trans. Netw. **13**(2), 343–355 (2005)
31. T. Sakurai, H.L. Vu, Mac access delay of IEEE 802.11 dcf. IEEE Trans. Wirel. Commun. **6**(5), 1702–1710 (2007). https://doi.org/10.1109/TWC.2007.360372
32. L. Dai, X. Sun, A unified analysis of IEEE 802.11 DCF networks: stability, throughput, and delay. IEEE Trans. Mob. Comput. **12**(8), 1558–1572 (2013)
33. G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J. Sel. Areas Commun. **18**(3), 535–547 (2000). https://doi.org/10.1109/49.840210
34. O. Nakhila, E. Dondyk, M.F. Amjad, C. Zou, User-side wi-fi evil twin attack detection using SSL/TCP protocols, in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)* (2015), pp. 239–244. https://doi.org/10.1109/CCNC.2015.7157983
35. F. Lanze, A. Panchenko, I. Ponce-Alcaide, T. Engel, Hacker's toolbox: detecting software-based 802.11 evil twin access points, in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)* (2015), pp. 225–232. https://doi.org/10.1109/CCNC.2015.7157981
36. B. Alotaibi, K. Elleithy, Rogue access point detection: taxonomy, challenges, and future directions. Wirel. Pers. Commun. **90**(3), 1261–1290 (2016). https://doi.org/10.1007/s11277-016-3390-x
37. IEEE Standard for Information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements—part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012), 1–3534 (2016). https://doi.org/10.1109/IEEESTD.2016.7786995
38. G.F. Riley, T.R. Henderson, The ns-3 Network Simulator, 15–34 (2010)
39. IEEE Standard for Information technology—local and metropolitan area networks—specific requirements—part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs. IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007), 1–244 (2008). https://doi.org/10.1109/IEEESTD.2008.4544755
40. M.-D. Dianu, J. Riihijärvi, M. Petrova, Measurement-based study of the performance of IEEE 802.11 ac in an indoor environment, in *2014 IEEE International Conference on Communications (ICC)* (IEEE, 2014), pp. 5771–5776
41. Y.P. Raykov, A. Boukouvalas, F. Baig, M.A. Little, What to do when k-means clustering fails: a simple yet principled alternative algorithm. PLoS ONE **11**(9), 1–28 (2016). https://doi.org/10.1371/journal.pone.0162259
42. S. Raschka, STAT 479: Machine Learning Lecture Notes (2018)


## Publisher's Note