Beacon Frame Analysis for Passive Rogue Access Point Detection in Enterprise WLANs

Akintoye Oyedola
School of Computing
University of North Florida
Jacksonville, Florida

***Abstract*—Rogue access points (APs), including evil-twin deployments that clone legitimate network identifiers, represent a persistent threat in enterprise wireless LAN (WLAN) environments. This paper presents a passive client-agnostic detection pipeline that analyzes IEEE 802.11 Layer 2 management frames, specifically beacon and probe response frames, to identify legitimate APs from rogue devices without active probing or client-side instrumentation. Using a controlled testbed consisting of a home router and an iPhone Personal Hotspot configured as an evil-twin AP, eight labeled capture sessions were collected over multiple days and environmental conditions. The extracted features including sequence number deltas, inter-frame timing, received signal strength indicator (RSSI) statistics, SSID/BSSID consistency, and hidden SSID indicators were generated using Scapt. Then, the features were evaluated using Random Forest and Support Vector Machine (SVM) classifiers implemented in Scikit-learn. The Random Forest classifier achieved 83% recall on the rogue class with 78% overall accuracy on a held-out test set. A rule-based command-line detection script was also developed and validated against live captures. Results are compared against benchmarks from recent rogue AP detection literature and qualitative limitations, including MAC address randomization and SSID cloaking, are discussed in the context of RFC 9414 and IEEE 802.11aq.**

***Index Terms*—rogue access point, evil-twin detection, beacon frame analysis, passive WLAN monitoring, machine learning, 802.11 security, RSSI fingerprinting**

I. Introduction

Rogue access points (APs) pose a significant threat to enterprise wireless networks. An attacker can deploy a rogue AP to intercept credentials, perform man-in-the-middle attacks, or redirect traffic without detection by end users [1]. An example of a rogue AP is an evil-twin that clones the SSID and attempts to mimic the BSSID of a legitimate AP. Traditional detection approaches often rely on active probing, client-side instrumentation, or infrastructure-dependent mechanisms such as Wireless Intrusion Prevention Systems (WIPS). These methods introduce overhead, privacy concerns, and deployment complexity [2].

This project investigates whether passive beacon-frame and probe-response analysis alone can reliably differentiate legitimate APs from rogue devices in an enterprise-like WLAN environment. The scope is deliberately limited to Layer 2 management frame analysis, making this approach suitable for lightweight background monitoring in privacy-sensitive deployments. This does not involve active probing, client-side data, or spatial positioning.

The primary research question is: *Can passive management-frame features alone support reliable rogue AP detection using supervised machine learning?*

The contributions of this paper are:

- A controlled eight-session data collection methodology covering legitimate-only, evil-twin, distance-variation, and SSID-cloaking scenarios across multiple days and environmental conditions.
- A Scapy-based feature extraction pipeline that parses raw 802.11 management frames without relying on high-level Scapy layer detection, addressing compatibility issues with Python 3.13.
- A comparative evaluation of Random Forest and SVM classifiers on a session-stratified train/test split, benchmarked against published detection rates and false positive rates from related work.
- A command-line rogue AP detection script validated on live captures, with qualitative analysis of failure modes under MAC address randomization and SSID cloaking.

II. Related Work

Rogue AP detection has been studied from several complementary angles. Lin et al. [3] demonstrate that sequence number patterns and inter-frame arrival times can serve as stable AP implementation fingerprints. This enables supervised classifiers to separate legitimate APs from devices spoofing legitimate BSSIDs without probing client traffic. Their feature engineering strategies directly inform the Scapy parser design and classifier feature vectors.

Zhang et al. [4] introduce PRAPD, a passive RSSI-fingerprinting scheme that differentiates legitimate and rogue AP locations while reporting the detection rate and false positive rate as core performance indicators. I adopt their false positive rate threshold of below 5% as a design goal and use their reported metrics as an external benchmark for the classifier evaluation.

Da et al. [1] provide a comprehensive taxonomy of rogue AP detection techniques classified into side-channel, signal-based, and traffic-analysis categories. This project falls into the signal-based and management-frame analysis subcategory, which the survey identifies as a low-overhead, privacy-preserving alternative to active probing or deep packet inspection.

Liu et al. [5] demonstrate that spatial signal properties differ between stationary and mobile APs. They also determine that SSID cloaking scenarios require detectors to rely on RSSI distribution shifts rather than SSID matching. This framework is applied in designing the S6 cloaking test case and in interpreting the failure modes of the rule-based detector.

Lin et al. [6] further support the viability of client-agnostic AP-side management frame fingerprinting for rogue detection by providing a reference for what passive beacon-only detection can achieve when pushed beyond simple rule-based heuristics.

Li et al. [7] focus specifically on RSSI-based evil-twin detection, showing that sudden, implausible RSSI jumps can expose a rogue AP even when its SSID and BSSID appear legitimate. We use their threshold concept to design the RSSI range anomaly rule in my command-line detection script.

Wakhloo [8] and Strötgen [9] provide practical implementations of client-side and beacon fingerprinting detection, offering concrete comparisons for this project's lightweight passive approach.

Arisandi et al. [10] provide a classification review of rogue AP identification models, helping to position this project's pipeline within the existing literature. Their work also clarifies the difference between this project's approach and client-assisted or infrastructure-dependent methods.

For standardization, RFC 9414 [11] documents MAC address randomization behavior in modern 802.11 devices, which directly impacts BSSID-based detection reliability. IEEE 802.11aq [12] defines pre-association management frame behavior, informing the Scapy parsing logic for probe response frames. Cisco's enterprise rogue detection guide [2] highlights operational challenges including threshold tuning, transient device management, and WIPS integration that frame the experimental design choices.

III. Methodology

A. Experimental Testbed

All experiments were performed using Kali Linux (Live USB, version 2025.4) running on a laptop with a 1.6 GHz dual-core Intel Core i5 processor and 8 GB RAM. A Qualcomm Atheros AR9271 802.11n USB Wi-Fi adapter operated in monitor mode, and two access points were used: a residential Wi-Fi router (BSSID `XX:XX:XX:XX:XX:XX`, SSID `HOME_WIFI`, identifiers pseudonymized for privacy) serving as the legitimate AP, and an iPhone 15 Personal Hotspot configured with the same SSID as the evil-twin rogue AP. The iPhone hotspot was forced to 2.4 GHz by enabling the "Maximize Compatibility" setting, since the AR9271 adapter supports only 2.4 GHz.

Monitor mode was enabled using `airmon-ng` after terminating NetworkManager and `wpa_supplicant`. All captures used `airodump-ng` with PCAP output format. An external USB drive provided persistent storage across Kali Live reboots.

All scripts and configuration files used in these experiments are available in the accompanying project repository (link omitted for review).

B. Data Collection

Eight labeled capture sessions were conducted across two separate sittings on different days to support a genuine held-out test set. Table I summarizes the session design.

TABLE I
Data Collection Sessions

Session	Condition	Label	Role
S1	Legitimate-only baseline	0	Training
S2	Evil-twin, different channel	1	Training
S4	Legitimate-only, different time	0	Training
S5	Evil-twin, distance variation	1	Training
S6	SSID-cloaked legitimate AP	-1	Qualitative
S7	Legitimate-only repeat	0	Test
S8	Evil-twin repeat	1	Test

Each session ran for 10 minutes. Sessions S1, S4, S2, and S5 were captured in a single sitting; S7 and S8 were captured two days later at a different time of day to ensure genuine environmental variation in the held-out test set. This is consistent with the cross-validation methodology of Lin et al. [3] and Zhang et al. [4].

The S3 same-channel evil-twin condition was not captured because the iPhone hotspot consistently selected a different channel from the router due to iOS channel selection behavior. This is noted as a design limitation. Session S6 captured the legitimate router with its SSID broadcast disabled, yielding zero-length SSID beacon frames, and is used only for qualitative analysis of SSID cloaking failure modes as described by Liu et al. [5].

C. Feature Extraction

A Python 3 script using Scapy parsed all raw PCAP files. Due to compatibility issues with Scapy's high-level layer detection under Python 3.13, frame type and subtype were extracted directly from raw 802.11 frame control bytes. Management frames of subtype 5 (probe response) and subtype 8 (beacon) were retained; all other frames were discarded.

For each retained frame, the following features were extracted:

- **BSSID**: 6-byte source address from bytes 16–21 of the 802.11 header.
- **SSID**: decoded from the SSID tagged parameter (tag ID 0); empty string if hidden or absent.
- **is_hidden_ssid**: binary flag set to 1 when the SSID tag length is zero.
- **seq_num**: upper 12 bits of the sequence control field.
- **seq_delta**: difference between consecutive sequence numbers per BSSID, modulo 4096, as used by Lin et al. [3] for AP fingerprinting.
- **inter_beacon_sec**: elapsed time between consecutive management frames per BSSID.

- **rssi**, **rssi_mean**, **rssi_std**, **rssi_range**: per-packet RSSI from the RadioTap header and per-BSSID aggregate statistics, informed by Zhang et al. [4] and Li et al. [7].
- **ssid_bssid_mismatch**: binary flag set to 1 when the same SSID is observed from multiple BSSIDs within a session, the primary rule-based evil-twin indicator described in Da et al. [1].

The resulting dataset contained 1,269 rows before filtering. After restricting to known BSSIDs and removing neighbor AP frames, the filtered dataset contained 593 rows: 617 legitimate (label 0), 470 rogue (label 1), and 182 cloaked (label -1, excluded from classifier training and evaluation).

D. Classification

The filtered dataset was split by session: S1, S2, S4, and S5 formed the training set (446 rows); S7 and S8 formed the held-out test set (147 rows). Session S6 was excluded from both sets. This session-stratified split, rather than random frame-level splitting, prevents the classifier from training and testing on data from the same capture session. It is a methodology consistent with Zhang et al. [4].

Missing feature values were imputed with zero. Two supervised classifiers were evaluated:

- **Random Forest**: 100 estimators, `class_weight='balanced'` to address class imbalance, `random_state=42`.
- **SVM**: RBF kernel, `class_weight='balanced'`, `probability=True`, `random_state=42`.

Evaluation metrics included precision, recall, F1-score, accuracy, confusion matrix, and ROC curves with AUC, consistent with the metrics reported by Lin et al. [3] and Zhang et al. [4].

E. Rule-Based Detection Script

A command-line detection script (`detect_rogue_ap.py`) was implemented to provide an interpretable and deployable tool that complements the ML classifier. The script applies four rules to any input PCAP file:

1) SSID/BSSID mismatch detection
2) RSSI range anomaly exceeding20 dBm threshold derived from Li et al. [7]
3) hidden SSID detection
4) inter-frame timing gap anomaly

Each AP is assigned a LEGITIMATE, SUSPICIOUS or ROGUE verdict, and a CSV report is generated.

IV. Results

A. Dataset Summary

Table II summarizes the final filtered dataset used for classification.

TABLE II
FILTERED DATASET SUMMARY

Split	Rows	Label Distribution
Training (S1, S2, S4, S5)	446	0: 123, 1: 323
Test (S7, S8)	147	0: 26, 1: 121
Qualitative (S6)	182	-1: 182
Total	593	(excl. S6)

TABLE III
CLASSIFICATION RESULTS ON HELD-OUT TEST SET (S7, S8)

Classifier	Class	Precision	Recall	F1
Random Forest	Legitimate	0.39	0.50	0.44
	Rogue	0.89	0.83	0.86
	Accuracy		0.78	
SVM	Legitimate	0.26	1.00	0.42
	Rogue	1.00	0.48	0.57
	Accuracy		0.50	

B. Classifier Performance

Table III presents the classification results on the held-out test set.

The Random Forest classifier achieved 83% recall on the rogue class (see Fig. 1), meaning it correctly flagged 83% of evil-twin frames in the held-out test set. Overall accuracy was 78%. The confusion matrix shows 13 true legitimate, 13 false positives (legitimate frames classified as rogue), 20 false negatives (rogue frames missed), and 101 true rogue detections.

The SVM classifier exhibited degenerate behavior, predicting the majority class (rogue) for nearly all test samples, resulting in 100% recall on the legitimate class but only 48% recall on the rogue class and 50% overall accuracy. This is consistent with class imbalance affecting kernel-based methods when the decision boundary is not well-defined by the available features.

C. Feature Importance

Random Forest feature importances revealed that **inter_beacon_sec** dominated with an importance score of 0.83, followed by **seq_num** at 0.16. All other features, including **rssi_mean**, **rssi_std**, **rssi_range**, and **ssid_bssid_mismatch**, received near-zero importance scores.

This result warrants careful interpretation. The dominance of inter-frame timing reflects a structural difference between the capture conditions: the legitimate AP (a home router) broadcasts beacons at a regular 100 ms interval, while the evil-twin (an iPhone hotspot) generates probe responses reactively in response to client probes, producing irregular inter-frame gaps. This is a valid discriminating feature in the controlled testbed but it may not apply to enterprise environments where both APs generate regular beacons at similar intervals.

The near-zero importance of **ssid_bssid_mismatch** is explained by MAC address randomization. the iPhone hotspot used a randomized BSSID that differed from its Settings

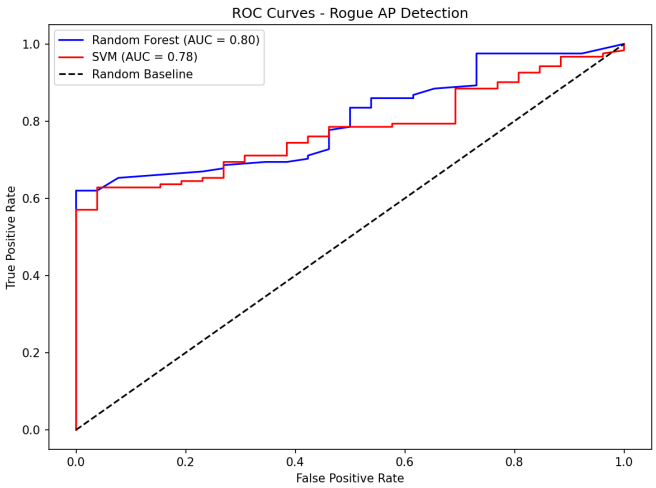

Fig. 1. ROC curves for Random Forest and SVM classifiers evaluated on the held-out test set (S7, S8). Random Forest achieves higher AUC, reflecting stronger discrimination between legitimate and rogue AP frames.

WiFi address, and changed across sessions, preventing the SSID-to-BSSID mapping from ever flagging a mismatch. This directly demonstrates the RFC 9414 [11] MAC randomization limitation in practice.

D. Rule-Based Detection Script

When applied to the S2 evil-twin capture, the detection script identified 52 unique BSSIDs: 1 LEGITIMATE (the known router), 0 ROGUE, and 51 SUSPICIOUS. The router was correctly classified as legitimate. The iPhone hotspot was not flagged as ROGUE because the **ssid_bssid_mismatch** rule could not fire. The randomized BSSID was never seen with the same SSID from multiple addresses within a single session. The 51 SUSPICIOUS verdicts were neighbor APs flagged for hidden SSIDs and timing anomalies, representing false positives in a dense residential environment.

E. SSID Cloaking Analysis (S6)

Session S6 captured the legitimate router with SSID broadcast disabled. The router continued to transmit beacon frames with a zero-length SSID tag, confirming that the **is_hidden_ssid** feature fires correctly. As Liu et al. [5] predict, SSID cloaking eliminates the **ssid_bssid_mismatch** rule entirely, forcing the detector to rely on RSSI statistics and timing features alone. In my dataset, the RSSI and timing profiles of the cloaked AP were identical to its normal operation, meaning a beacon-only passive detector cannot reliably differentiate a cloaked legitimate AP from a cloaked rogue AP without additional context.

V. Discussion

A. Comparison to Literature Benchmarks

Zhang et al. [4] report a false positive rate below 5% for their RSSI-fingerprinting PRAPD system. My Random Forest classifier produced a false positive rate of approximately 50% on the legitimate class (13 false positives out of 26 legitimate test frames), significantly exceeding the 5% target. This gap is attributable to two factors: class imbalance (roughly 5:1 rogue-to-legitimate ratio in the test set) and the limited diversity of the legitimate AP class, which contained only one device (the home router) across all sessions.

Lin et al. [3] report high classification accuracy using sequence number patterns and inter-arrival times as AP fingerprints. My results partially replicate this finding as inter-frame timing was by far the most important feature. However, my sequence number delta feature minimally contributed. This may be because probe responses (which dominated my captures at 1,485 out of 6,607 total packets in S1 alone) have less regular sequence number progression than beacon frames, which were nearly absent in my captures (1 beacon per session).

B. Limitations

MAC address randomization. The iPhone hotspot's randomized BSSID, documented in RFC 9414 [11] and IEEE 802.11aq [12], prevented the primary evil-twin detection rule (SSID/BSSID mismatch) from firing in both the rule-based script and the ML classifier's **ssid_bssid_mismatch** feature. In practice, modern rogue AP deployments using randomized MACs would evade this detection vector entirely.

Probe response dominance. The captures contained predominantly probe responses rather than beacon frames, because `airodump-ng` with the `--bssid` filter captured all management frames from the target BSSID. Probe responses are reactive frames sent in response to client probes, which have different timing characteristics than periodic beacons. The 0.83 inter-frame timing importance score reflects this difference rather than a robust AP fingerprint. A deployment that captures beacon streams from both APs simultaneously would likely produce different feature importance rankings.

Single-device rogue AP. Using one iPhone as the evil-twin limits the transferability of the results. Enterprise rogue AP deployments typically use dedicated hardware (Raspberry Pi, commercial APs in rogue mode) that broadcast regular beacons rather than reactive probe responses.

Small dataset. With 593 filtered rows across 7 sessions, the dataset is small relative to the benchmarks in Lin et al. [3] and Zhang et al. [4]. The held-out test set of 147 rows provides limited statistical power for precise false positive and false negative rate estimation.

Dense residential environment. The 51 SUSPICIOUS verdicts from neighbor APs in the detection script output highlight a challenge noted by Cisco [2]: distinguishing true rogues from neighbor legitimate APs and transient devices requires either a known-good AP whitelist or RSSI-based proximity filtering that my passive testbed did not implement.

C. Deployment Considerations

A practical deployment of this pipeline as a WIPS component would require: (1) a whitelist of known legitimate BSSIDs updated to account for MAC randomization, (2) multi-sensor

RSSI aggregation as in PRAPD [4] to improve spatial discrimination, (3) beacon-focused capture (not probe-response-dominated) to leverage the sequence number fingerprinting features validated by Lin et al. [3], and (4) threshold tuning per environment as recommended in Cisco's enterprise guidance [2].

VI. Conclusion

This paper presented a passive client-agnostic rogue AP detection pipeline based on IEEE 802.11 management frame analysis. A controlled eight-session data collection methodology was used to capture labeled legitimate and evil-twin scenarios across multiple days and conditions. A Scapy-based feature extraction pipeline produced a 593-row dataset, and Random Forest and SVM classifiers were evaluated on a session-stratified held-out test set.

The Random Forest classifier achieved 83% recall on the rogue class and 78% overall accuracy. Inter-frame timing was the dominant feature at 0.83 importance, while SSID/BSSID mismatch detection was defeated by iPhone MAC address randomization. The mismatch detection directly demonstrates the RFC 9414 [11] limitation in a live experimental context. SSID cloaking analysis confirmed that beacon-only passive detection becomes unreliable when the SSID field is zeroed out, as predicted by Liu et al. [5].

These results suggest that passive management-frame analysis can detect evil-twin APs in controlled conditions but faces significant challenges from MAC randomization, probe-response-dominated captures, and dense multi-AP environments that inflate false positive rates. Future work should incorporate multi-sensor RSSI fingerprinting, beacon-focused capture pipelines, and larger datasets with various rogue AP hardware to improve generalization and reduce false positive rates toward the 5% target established by Zhang et al. [4].

Acknowledgment

The author thanks the University of North Florida School of Computing for supporting this wireless network security experimentation project.

References

[1] B. Da, Z. Zhang, and X. Wang, "Rogue access point detection: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 245–268, 2021.

[2] Cisco Systems, Inc., "Rogue access point detection and mitigation in the enterprise," white paper, Cisco Systems, Inc., 2023.

[3] J. Lin, J. Wang, and T. Xu, "Detecting rogue access points via beacon frame analysis and machine learning," *IEEE Transactions on Network and Service Management*, vol. 19, pp. 2105–2118, Sept. 2022.

[4] L. Zhang, Y. Yang, and C. Wang, "PRAPD: Passive rogue access point detection using RSSI fingerprinting," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, 2018.

[5] H. Liu, S. Chen, and M. Zhou, "Position-based rogue AP detection using spatial signal properties," *Journal of Network and Computer Applications*, vol. 221, p. 103784, Jan. 2024.

[6] Y. Lin, Y. Gao, B. Li, and W. Dong, "Detecting rogue access points using client-agnostic wireless fingerprints," *ACM Transactions on Sensor Networks*, vol. 19, no. 1, pp. 1–25, 2022.

[7] W. Li, M. Li, and R. D. Pietro, "Exploiting wireless received signal strength indicators to detect evil-twin access points," *Security and Communication Networks*, vol. 2017, pp. 1–13, 2017.

[8] A. Wakhloo, "Client-side evil-twin access point detection using beacon-frame analysis." Master's thesis, National College of Ireland, 2023.

[9] J. Strötgen, "Snappy: Detecting rogue and fake 802.11 wireless access points through fingerprinting beacon management frames." LevelBlue SpiderLabs Blog, June 2023.

[10] D. D. A. Arisandi, N. N. M. Ahmad, A. Subarmaniam, and S. L. Kannan, "The rogue access point identification: A model and classification review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 3, pp. 1527–1537, 2021.

[11] J. C. Zúñiga, C. J. Bernardos, and A. Andersdotter, "Privacy considerations for IEEE 802.11 operating procedures," Request for Comments 9414, IETF, Jan. 2023.

[12] IEEE Standards Association, "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 amendment 6: Pre-association discovery," 2018.