



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

SentinelAP: A Leakage-Controlled Rogue Wi-Fi Access Point Detection System

Shoaib Sayyed¹, Ankita Kalode², Saloni Funne³, Noman Danish⁴, Shraddha Bansod⁵, Prof. Mohammed Sajid⁶

¹⁻⁵Department of Computer Science and Engineering, Nagpur Institute of Technology, Nagpur - 441501, Maharashtra, India

⁶Guide: (Assistant Professor, Dept. of CSE, NIT Nagpur)

*E-mail: sentinelap.nit@gmail.com

ABSTRACT

Rogue access points and Evil-Twin attacks take advantage of the fact that a client normally identifies a wireless service through observable names and radio characteristics rather than an explicit ownership proof for the transmitter. This paper presents SentinelAP, a client-side detection system built around a fixed ten-feature representation and a Random Forest classifier. The scanner is designed for native Windows operation with the Windows WLAN API, a netsh fallback, and a TP-Link Archer T2U Plus adapter. A FastAPI service accepts authenticated observation batches, applies the same model artifact for server-side scoring, stores results, and supports remote monitoring. The primary corpus contains 1,108 validated observations, including 1,050 legitimate and 58 rogue cases. To reduce evaluation leakage from repeated measurements of the same AP, BSSID grouping is used for both the held-out experiment and five-fold cross-validation. The pooled out-of-fold ROC-AUC is 0.9875; at the F1-selected threshold of 0.04, precision is 0.7436, recall is 1.00, and F1 is 0.8529, with an FPR of 0.019. A separate AWID3 derivation study yields 1,814 labelled rows without inventing unsupported feature values. The main limitation is the heavy contribution of open_network to model ranking, which motivates broader attack collection and future cross-environment validation.

Keywords: Rogue AP detection, Evil Twin, Wi-Fi security, Windows WLAN API, Random Forest, grouped validation, data leakage, wireless-security machine learning.

1. Introduction

1.1 Motivation

Wireless LANs have become routine connectivity infrastructure in universities, offices, public venues, residences, and other shared environments. The same accessibility gives an attacker a low-cost opportunity to place an unauthorized access point near legitimate users. An Evil-Twin attack is especially difficult to notice because the rogue device can advertise the same SSID as a trusted network while differing in its BSSID, advertised protection, channel, or radio behaviors. A device that treats the network name as the primary identity has no direct way to establish which transmitter is operated by the legitimate owner.

Although the user may see only a familiar network name, the wireless exchange exposes several measurable properties before any application credential is supplied. A scanner can observe the BSSID, received signal level, channel, security advertisement and repeated beacon observations, then maintain short-term history for changes in signal and appearance. The engineering challenge is to combine these signals without making a simple allow-list the sole source of truth and without evaluating a model with a train-test procedure that allows the same physical AP to appear on both sides.

SentinelAP tackles both issues through a fixed ten-feature representation and an evaluation protocol that treats the BSSID as the grouping entity. Observations from one AP are therefore handled as one group during evaluation, making the experiment closer to the real question of whether the detector can recognize an AP it has not previously encountered. The resulting score can then be examined independently from the threshold chosen to trigger an operational decision.

1.2 Contributions

A Windows-based scanner that obtains BSS observations through the native WLAN interface, converts them into the fixed feature vector, performs local model inference, and can synchronize records with the FastAPI service.

A centralized feature definition shared by collection, research, and server-side inference so that feature order and semantics remain consistent between training and deployment.

A leakage-aware evaluation design that keeps complete BSSID groups together, uses five-fold grouped cross-validation, pools out-of-fold predictions, and examines several decision thresholds instead of relying on a single arbitrary cut-off.

An AWID3 compatibility study that distinguishes theoretical feature support from what individual capture files actually contain, while leaving unsupported values absent rather than manufacturing replacements.

1.3 Organization

The next section places the work in the context of Rogue-AP and wireless intrusion research. The following sections describe the architecture and feature contract, the training and evaluation procedure, the primary and external datasets, the measured results, and the resulting limitations and future directions. The article closes with the conclusion, acknowledgement, and references.

2. Related Work

2.1 Rogue access-point and Evil-Twin detection

Research on unauthorized and imitation access points can broadly be grouped into three operating styles. Infrastructure-oriented methods compare observed devices with an inventory of approved APs and can perform well when that inventory is complete. Client-oriented methods use information already exposed through a normal WLAN interface, such as identifiers, signal measurements, channels, and security advertisements, and therefore demand less infrastructure. Active approaches connect to a network and perform additional probes to collect application-layer evidence, but such actions require explicit authorization and introduce greater operational risk. The literature consistently shows a trade-off among coverage, deployment assumptions, and false-alarm behavior (Alotaibi and Elleithy, 2016).

2.2 Machine learning and evaluation leakage

A supervised classifier can combine several imperfect wireless indicators into one continuous suspicion score. Random Forest is well suited to the present tabular representation because the ensemble can model nonlinear relationships without requiring a linear decision boundary and can expose an attack score through aggregation of its trees (Breiman, 2001). The more subtle issue is evaluation validity. Multiple observations of one physical AP are correlated, so a row-wise random split can place related observations in both training and testing and allow the model to exploit device-specific characteristics. That form of leakage has been discussed broadly in data-mining research (Kaufman, Rosset and Perlich, 2012). Group-based validation addresses the deployment question more directly by withholding complete entities.

2.3 AWID3 as an external corpus

AWID3 provides a useful external source of wireless intrusion captures and includes several attack scenarios relevant to WLAN security. Its records are organized around individual 802.11 frames rather than the scan-round observations produced by SentinelAP. Consequently, matching column names do not guarantee matching feature semantics. The external study therefore checks feature support explicitly and keeps unsupported attributes out of the supervised experiment, rather than using the external corpus simply to increase the apparent sample size.

3. System Architecture and Design

3.1 Overall architecture

The system is divided into a Windows sensing client and a remotely accessible monitoring service. The edge component uses the TP-Link Archer T2U Plus together with the Windows WLAN interface to discover nearby BSS records. Feature state is maintained per BSSID so that the ten required values can include historical context, after which the local Random Forest is invoked. Observations are placed in a local upload queue, allowing collection to continue during temporary connectivity loss. When the server is reachable, queued records are sent in batches over HTTPS; the FastAPI backend applies the same trained estimator and stores the results for later dashboard queries.

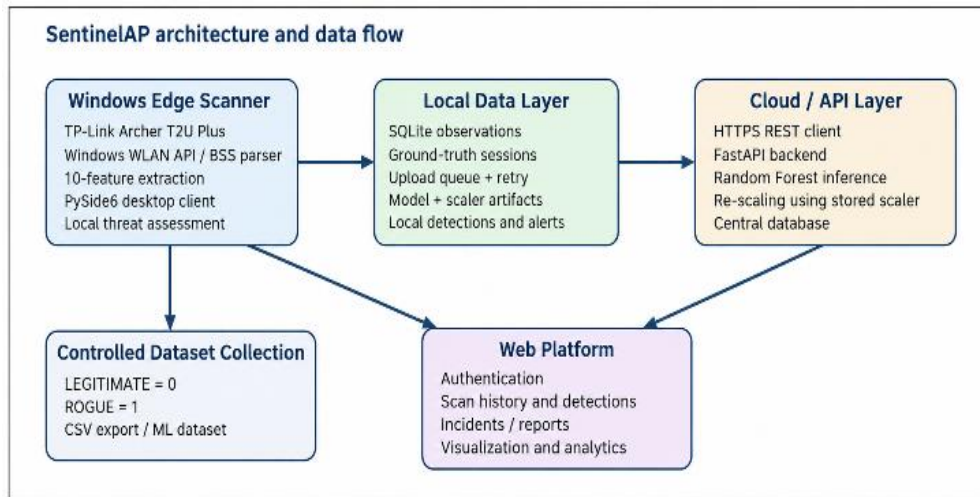


Fig. 1 - SentinelAP architecture and the movement of wireless observations from the Windows client to remote storage and web monitoring.

3.2 Windows scanner

Wireless discovery is performed through the Windows WLAN API provided by wlanapi.dll, with a netsh path available when the primary API does not yield usable data. Each collection round records fields that are visible through the BSS information, including SSID, BSSID, channel, signal level, and security information. The standard collection workflow is passive and does not automatically associate with arbitrary open networks. This design keeps the scanner suitable for a conventional Windows workstation and removes the need for Linux monitor-mode tooling or a virtual machine.

3.3 Cloud service

The server component uses FastAPI and provides routes for authentication, scanner ingestion, and dashboard access. In the prototype deployment, SQLite is used for persistence, while the complete ordered feature vector is retained with the observation record. The research and server paths load the same model and scaler artifacts so that a record evaluated locally and the corresponding record evaluated remotely use the same estimator rather than two independently reimplemented classifiers.

3.4 Threat assessment

The principal learned quantity is the Random Forest attack score, denoted by $p(a)$. The application can map this score to an operator-facing severity and may combine it with deterministic evidence available in the client. Captive-portal, DNS, and TLS investigations are intentionally not presented as completed parts of the ten-feature supervised experiment; they remain future application-layer extensions that would need separate validation.

3.5 Frozen feature contract

The feature contract contains ten values in a fixed order and is reused without redefinition during collection, training, inference, and evaluation. The attributes cover identity mismatch, signal behavior, network openness, repeated observations, channel relationships, SSID resemblance, vendor information, and time since first observation. Keeping the contract in one authoritative definition reduces the risk that independently maintained scripts silently change feature meaning or ordering.

Table 1 - Ordered ten-feature representation used by SentinelAP.

#	Feature	Meaning	Domain
1	bssid_mismatch	Security advertisement differs among BSSs using the same SSID	Binary
2	signal_strength	Normalised representation of the observed RSSI	[0,1]
3	signal_anomaly	Indicator that current RSSI departs substantially from recent history	Binary
4	open_network	Binary indication that the observed network is advertised without encryption	Binary
5	beacon_count	Number of scan rounds in which the BSSID has appeared	[1,100]
6	beacon_variance	Normalised variability of RSSI over the stored observation history	[0,1]

#	Feature	Meaning	Domain
7	channel_conflict	Same SSID detected on more than one channel during a sweep	Binary
8	ssid_similarity	Similarity between the observed SSID and configured protected names	[0,1]
9	vendor_suspicious	Binary indicator derived from the BSSID OUI vendor list	Binary
10	first_seen_hours	Elapsed time since the BSSID was first observed	>=0

4. Methodology

4.1 Data acquisition and labeling

The main supervised corpus consists of SentinelAP observations gathered under controlled conditions and assigned one of two ground-truth classes: LEGITIMATE (0) or ROGUE (1). The label is provided by the operator and is not generated from the prediction of the model or from the same rule set that contributes to runtime detection. This separation keeps the learning target independent from the inference logic and makes the resulting evaluation more meaningful.

4.2 Data validation

An observation is retained for supervised learning only when the full ten-feature vector is available with numeric values and a valid label is present. For the AWID3 derivation study, missing information is not replaced with guessed constants. Source information is retained for accepted records so that dataset assembly can be traced back to the original observation file and reproduced later.

4.3 Random Forest classifier

The learning implementation is written in pure Python and contains DecisionTree, RandomForest, and StandardScaler components. Individual trees use Gini impurity and a randomly selected square-root-sized subset of features during splitting. The forest builds bootstrap samples and averages the positive-class leaf estimates to obtain the final attack score. Fixed randomization makes the resulting model reproducible when the same data and configuration are used by the research and desktop paths.

$$p(a) = (1/T) \sum I(h_t(x(a)) = 1) \quad (1)$$

For an observation a described by $x(a)$, the value $p(a)$ lies between zero and one and represents the model's attack score. Applying a threshold τ converts this continuous value into a binary decision. Threshold-dependent measures such as precision, recall, F1, FPR, and FNR describe the selected operating point, whereas ROC-AUC evaluates the quality of the ranking across all thresholds.

4.4 Leakage-controlled splitting

BSSID is the unit used to control leakage. All records belonging to one BSSID remain together during the held-out split and during five-fold cross-validation. The held-out experiment reserves 20 percent of the data and contains 188 observations from five APs that are absent from training. For the primary evaluation, the predictions generated for every out-of-fold observation are combined, ensuring that each record is assessed by a model that did not train on that BSSID.

4.5 Class balancing and reproducibility

Because rogue observations form a minority of the primary corpus, the training portion is balanced by repeating minority examples only after the grouped split has been created. The validation and test records are left untouched. StandardScaler is fitted exclusively on the training data. The main configuration uses 50 trees, depth 8, seed 20260101, and a 20 percent held-out fraction; the fitted model and scaler are retained as reusable artifacts.

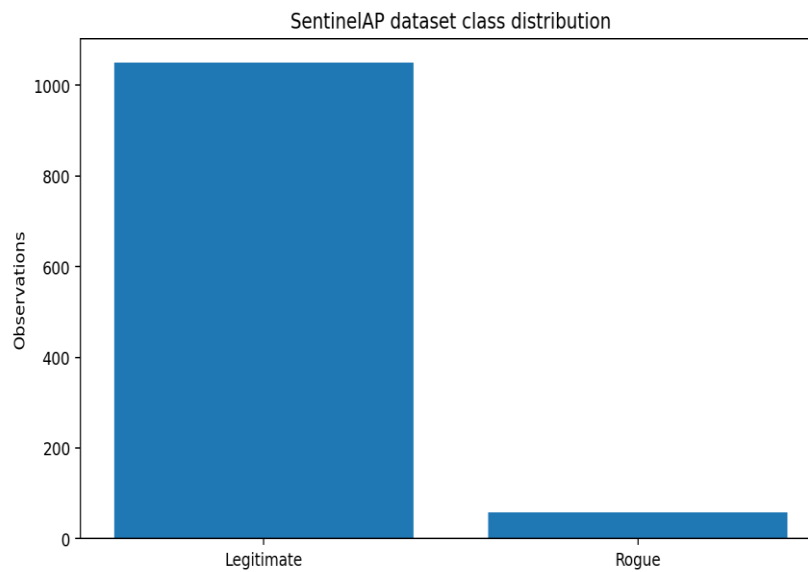
5. Dataset and Derivability Study

5.1 SentinelAP observation set

The validated SentinelAP corpus contains 1,108 observations, of which 1,050 are normal and 58 are labelled as attacks. The positive class therefore represents approximately 5.2 percent of the dataset. The records are assembled from the bundled and legacy training files together with four Windows export files. Across these sources, 23 distinct BSSID groups are available for the leakage-aware evaluation.

Table 2 - Composition of the primary validated SentinelAP observation set.

Source	File	Rows kept
bundled_training_csv	network.csv	146
legacy_training_csv	network.csv	146
windows_export_observations	SentinelAP_Observations_2026-09-20.csv	204
windows_export_observations	SentinelAP_Observations_2026-09-20_2.csv	204
windows_export_training	SentinelAP_Training_2026-09-20.csv	204
windows_export_training	SentinelAP_Training_2026-09-20_2.csv	204
Total		1,108

**Fig. 2 - Class composition of the validated SentinelAP observations.**

5.2 AWID3 feature derivability

AWID3 is treated as an external compatibility study rather than as a direct test set for the SentinelAP classifier. Because its records describe individual frames, some SentinelAP attributes can be reproduced directly, some can only be approximated under documented assumptions, and others depend on scan-level or temporal information that the capture does not provide. The derivability analysis therefore reports two outcomes: a CEILING assessment describing the theoretical support of the dataset type, and an ACHIEVED assessment based on the actual file examined.

Table 3 - External AWID3 derivability assessment for the SentinelAP features.

Feature	Ceiling	Achieved
signal_strength	Exact	Exact
open_network	Exact	Exact
vendor_suspicious	Exact	Exact
signal_anomaly	Partial	Partial
beacon_variance	Partial	Partial
first_seen_hours	Partial	Partial
bssid_mismatch	Partial	Not derivable

Feature	Ceiling	Achieved
channel_conflict	Partial	Not derivable
ssid_similarity	Partial	Not derivable
beacon_count	Not derivable	Not derivable

The external derivation process yields 1,814 labelled records, consisting of 1,725 normal and 89 attack rows. These rows are retained for audit and compatibility analysis, but they do not become part of the primary supervised evaluation when required feature values are unavailable. The resulting separation prevents incomplete external data from creating an artificial advantage in the reported classifier performance.

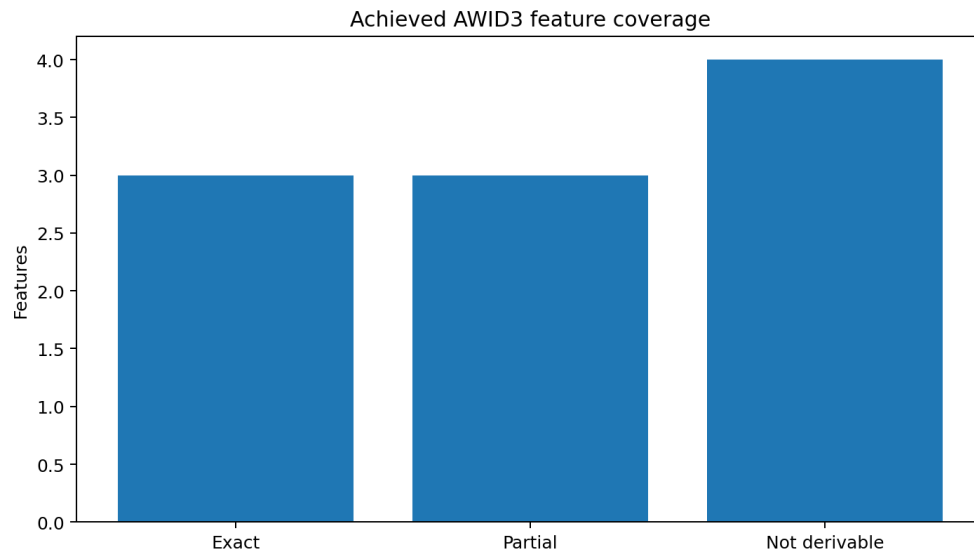


Fig. 3 - Feature support achieved during AWID3 derivation.

6. Evaluation and Results

6.1 Held-out grouped evaluation

The grouped held-out experiment contains 188 observations from five access points that were excluded from training; 186 are normal and two are attacks. With thresholds of 0.30, 0.50, and 0.75, the small attack subset receives zero positive predictions, even though the ROC-AUC remains 0.9892. This illustrates that a strong ranking statistic does not determine the operational threshold. At the F1-selected threshold of 0.26, precision is 0.3333, recall is 1.0, F1 is 0.5, and FPR is 0.0215.

6.2 Grouped five-fold cross-validation

The main cross-validation result is obtained by pooling the out-of-fold predictions produced by five BSSID-grouped folds. This procedure gives a ROC-AUC of 0.9875. At the threshold that maximizes F1 (0.04), precision reaches 0.7436, recall is 1.00, and F1 is 0.8529; the false-positive rate is 0.019 and the false-negative rate is zero. The resulting confusion matrix contains 58 true positives, 20 false positives, 1,030 true negatives, and no false negatives. Because three individual folds contain no positive AP group, pooled out-of-fold performance is the more informative headline statistic.

Table 4 - Leakage-controlled performance on grouped evaluation splits.

Evaluation	Threshold	Accuracy	Precision	Recall	F1	FPR	ROC-AUC
Pooled grouped CV	0.04	0.9819	0.7436	1.0000	0.8529	0.0190	0.9875
Held-out grouped	0.26	0.9787	0.3333	1.0000	0.5000	0.0215	0.9892

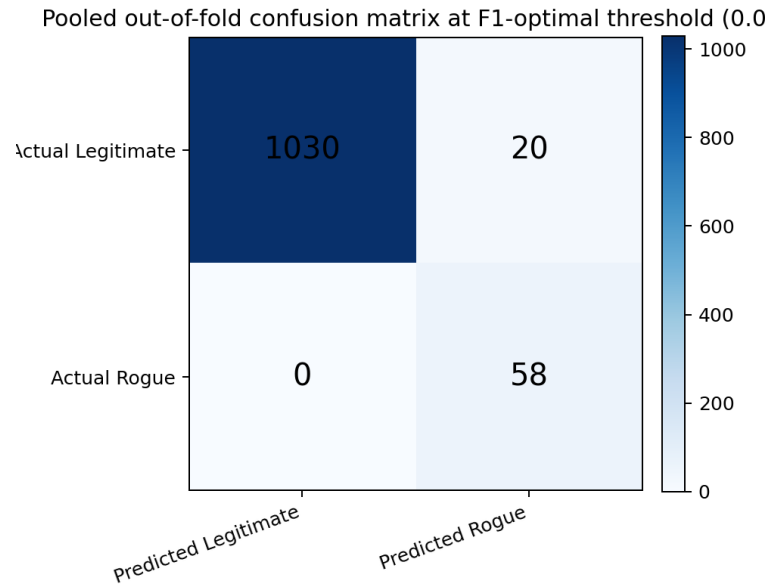


Fig. 4 - Confusion matrix obtained from pooled out-of-fold predictions at the selected threshold.

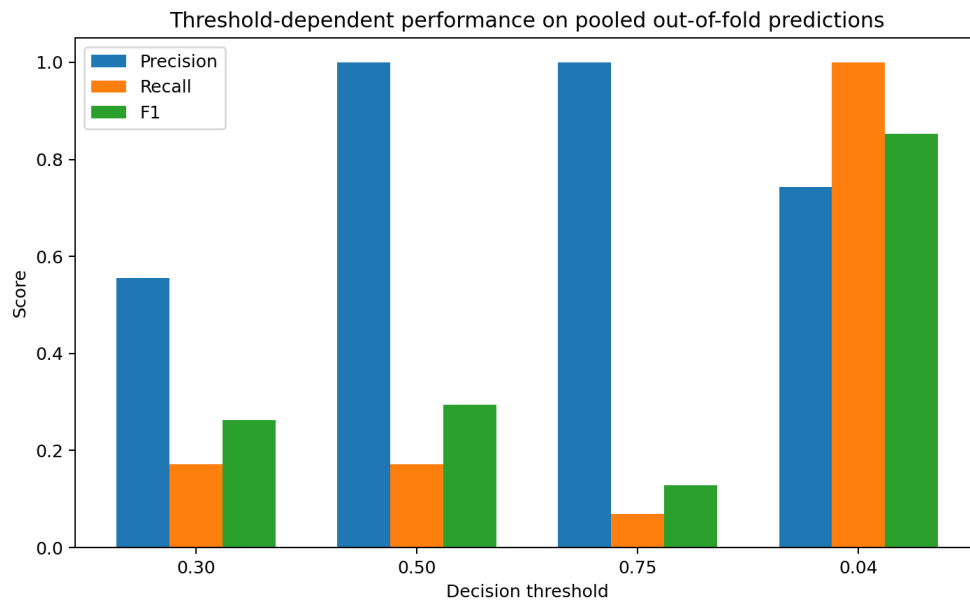


Fig. 5 - Threshold-dependent precision, recall, and F1 values.

6.3 Naive row-level split as a leakage contrast

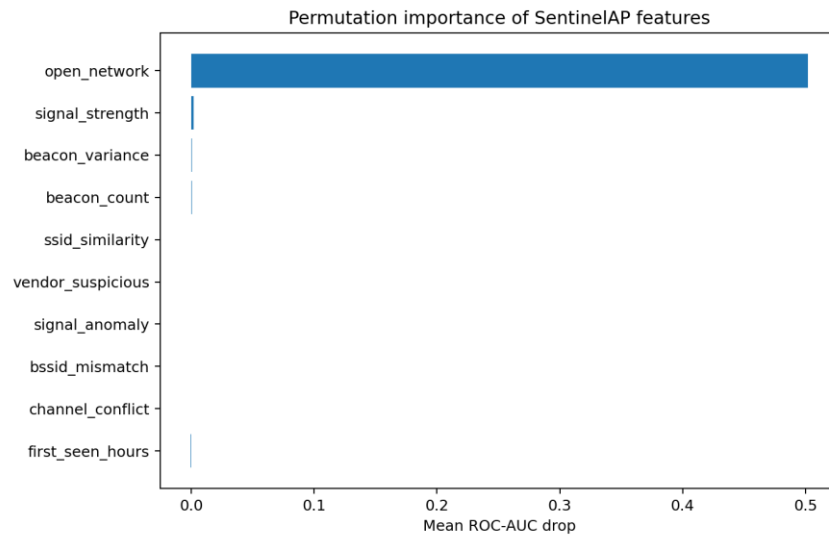
When the same records are divided by a conventional stratified row-level split, the evaluation becomes artificially perfect: accuracy, precision, recall, F1, specificity, balanced accuracy, Matthews correlation coefficient, and ROC-AUC all reach 1.0. No change has been made to the model; only the splitting rule differs. Since observations from the same AP can be present in both partitions, the model is partly recognising entities it has already encountered. The contrast demonstrates why BSSID grouping is essential for the intended deployment scenario.

6.4 Permutation importance and latency

Feature permutation was used to measure how much the ROC-AUC changes when one input is shuffled while the others remain unchanged. The effect is dominated by open_network, whose permutation lowers ROC-AUC by about 0.5021. The next largest change is only 0.0021, and several features produce zero measured change in the current forest. This concentration indicates that the present attack examples are strongly associated with the open-network condition and exposes an important weakness of the current dataset.

Table 5 - Permutation importance expressed as the change in ROC-AUC.

Feature	Mean ROC-AUC drop
open_network	0.5021
signal_strength	0.0021
beacon_count	0.0005
beacon_variance	0.0005
bssid_mismatch	0.0000
signal_anomaly	0.0000
channel_conflict	0.0000
ssid_similarity	0.0000
vendor_suspicious	0.0000
first_seen_hours	-0.0003

**Fig. 6 - Feature importance measured through permutation-induced ROC-AUC reduction.**

The 188-row held-out batch requires about 17.9 ms for inference, equivalent to approximately 0.095 ms per AP. At the collection rate used in the experiments, computation by the classifier is therefore not the main source of delay; acquiring the wireless observations itself remains the dominant part of the scan cycle.

7. Discussion, Limitations and Future Work

7.1 What the evaluation protocol changes

The difference between row-level and grouped evaluation is a consequence of how the test population is defined, not evidence that the classifier changes behavior between runs. The deployment problem concerns previously unseen access points, so the evaluation must keep the physical entity out of training. The lower grouped score is therefore more informative than the perfect row-level result for assessing generalization.

7.2 Dataset dependence of the model

The strongest limitation is the model's dependence on open_network. In the current collection, legitimate and attack records are separated to a large extent by whether the observed AP is open. That pattern is real for the sampled environment, but it does not prove that the classifier has learned the more subtle characteristics of an Evil Twin. A rogue AP using the same WPA2 protection as its legitimate counterpart may therefore be considerably harder to identify. Future data collection should deliberately include such security-matched attacks while varying channel, signal, timing, and identity features.

7.3 External data portability

The AWID3 experiment also highlights a portability issue: shared column names do not imply shared feature semantics. By recording exact, partial, and unsupported derivations separately, the study avoids presenting inferred values as observations. This makes the external corpus valuable for validating the data-processing methodology without allowing incomplete external records to influence the primary supervised performance claim.

7.4 Application-layer analysis

Application-layer inspection could strengthen detection by examining suspicious captive portals, DNS manipulation, or certificate behavior. Those capabilities are not included in the validated ten-feature model reported here. Integrating them would require a separate, authorized test workflow, additional observations, and an evaluation protocol designed specifically for application-layer evidence.

7.5 Threshold selection

The F1-maximising threshold of 0.04 should be understood as an operating choice for this dataset rather than a universal detector setting. In practice, the chosen threshold should reflect the relative cost of missed attacks and false alarms at a particular site, and it may need to be recalibrated when the class balance or deployment environment changes.

7.6 Reproducibility

Reproduction follows a fixed sequence: validate and assemble the SentinelAP observations, generate the auditable AWID3 derivation, train the forest and scaler, then run the grouped held-out and five-fold evaluations. Random seed 20260101 controls the relevant stochastic operations, including splitting, minority oversampling, bootstrap sampling, random feature selection, and permutation procedures. Latency is measured repeatedly and summarized to reduce sensitivity to transient system load.

The research outputs include machine-readable records for dataset composition, feature order, model settings, metrics, and saved artifacts. The FastAPI service can load the same trained model used by the Windows client, allowing local and server-side scoring to share one estimator implementation. This common path improves auditability because differences between environments are less likely to come from duplicated inference logic.

8. Conclusion

The study presents SentinelAP as a Windows-oriented detector that combines a fixed ten-feature wireless representation, a Random Forest classifier, local alerting, and cloud-assisted monitoring. Its most important methodological choice is to perform evaluation at the BSSID-group level so that repeated measurements of one AP do not cross the train-test boundary. With this protocol, pooled out-of-fold ROC-AUC reaches 0.9875, while the F1-oriented threshold gives precision 0.7436, recall 1.00, and F1 0.8529.

The experiments also expose the limits of the current evidence. Perfect separation under a row-wise split should not be interpreted as generalization to new APs. In addition, permutation analysis shows that the classifier is strongly influenced by `open_network`, indicating a need for attack examples that maintain stronger security settings and differ along other radio attributes. The AWID3 study arrives at a complementary conclusion: an external dataset should contribute to evaluation only when its values have the same semantic meaning as the features used by the deployed detector.

Overall, SentinelAP provides a reproducible baseline for client-side Rogue-AP and Evil-Twin research and makes its evaluation assumptions explicit. The current evidence supports the detector for the distribution represented in the collected data; it should not be interpreted as a universal solution to wireless impersonation. Further work should enlarge the positive class, collect data across multiple environments, add authorized application-layer measurements, and study threshold selection under realistic operational costs.

Acknowledgements

The authors are grateful to the Department of Computer Science and Engineering, Nagpur Institute of Technology, for the academic and technical support provided throughout the SentinelAP project. We also acknowledge the broader wireless-security research community and the maintainers of the public AWID/AWID3 resources used in the external feature-derivability work.

Frozen feature notation

The paper uses the following notation: a represents an observed access point or BSS identified by its BSSID; $x(a)$ is the ordered ten-feature vector for that observation; $p(a)$ is the model's attack score; G denotes the BSSID used as the grouping entity; $\hat{y}$ is the threshold classification; and τ is the selected decision threshold.

9. References

- Alotaibi, A., & Elleithy, K. (2016). Rogue Access Point Detection: Taxonomy, Challenges, and Future Directions. *Wireless Personal Communications*, 90(3), 1261–1290.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.
- Chatzoglou, E., Kambourakis, G., Kolias, C., & Smiliotopoulos, C. (2022). Pick Quality Over Quantity: Expert Feature Selection and Data Preprocessing for 802.11 Intrusion Detection Systems. *IEEE Access*, 10, 64761–64784.
- Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8), 861–874.
- IEEE. (2021). IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—LAN/MAN Specific Requirements—Part 11: Wireless LAN MAC and PHY Specifications (IEEE Std 802.11-2020).
- Kaufman, S., Rosset, S., & Perlich, C. (2012). Leakage in Data Mining: Formulation, Detection, and Avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4), 1–21.
- Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2016). Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. *IEEE Communications Surveys & Tutorials*, 18(1), 184–208.
- Roberts, D. R., Bahn, V., Ciuti, S., et al. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8), 913–929.
- Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta*, 405(2), 442–451.
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, 3(1), 32–35.
- Ramírez, S. (2018–). FastAPI. Open-source web framework.