Spoofing and Jamming Resilience for RF-Driven AR Alerts

Benjamin J. Gilbert

Spectrcyde RF Quantum SCYTHE, College of the Mainland
bgilbert2@com.edu
ORCID: https://orcid.org/0009-0006-2298-6538

***Abstract*—Augmented reality systems that rely on radio-frequency (RF) sensing must withstand adversarial attacks. An attacker may jam the channel to block alerts or spoof RF signatures to trigger false critical alarms. Today's RF-to-AR pipelines lack systematic mechanisms to evaluate and harden against such threats. Inspired by red-team methodologies, we present a framework for generating adversarial traces and evaluating spoofing/jamming resilience. We contribute three components: (1) *adversarial trace generators* that inject jammed and spoofed RF events into standardised traces; (2) detection algorithms that analyse RF metrics (noise floor, channel variance) and cryptographic tags to identify attacks; and (3) mitigation strategies including channel hopping and majority-vote consensus to maintain situational awareness. Our experiments show that the proposed defences detect 92% of jamming events and 88% of spoofed alerts within $100\,\mathrm{ms}$, reducing false critical alerts by 40%. The framework emphasises reproducibility: adversarial traces, metrics and scripts are packaged in OpenBench-AR format [1], capturing data, code and process [2].**

I. Introduction

RF-to-AR systems harness wireless signals to localise objects and overlay alerts on wearable displays. Applications range from casualty triage to threat detection. However, these systems assume honest environments. A jammer can disrupt the wireless channel, causing packet loss and staleness, while a spoofer can forge RF signatures to simulate nonexistent casualties or threats. Such attacks could overwhelm users with false alerts or silence critical warnings.

Jamming and spoofing attacks against wireless systems have been extensively studied [3], [4]. Traditional approaches focus on either reactive protocols [5] or machine-learning-based detection [6]. However, most RF-AR research focuses on latency and throughput, with little attention to adversarial robustness. Our approach differs from existing work by combining lightweight physical-layer metrics (noise floor, CSI variance) with cryptographic authentication, specifically targeting the real-time constraints of AR applications where detection latency must remain under $100\,\mathrm{ms}$.

The reproducibility crisis in ML and systems research underscores the need for standardised datasets and transparent evaluation [1]. Reproducibility must capture the interaction of data, code and process [2]; adversarial evaluation is no exception. We propose a red-team framework to systematically introduce spoofing and jamming into RF traces and to develop and test mitigation strategies.

Our contributions are:

- We build *adversarial trace generators* that create jamming and spoofing events by mixing noise and forged CSI/RSSI patterns into existing RF traces. These traces serve as benchmarks for security evaluation.
- We implement detection algorithms based on physical-layer metrics (noise floor elevation, sudden CSI variance) and cryptographic sequence numbers. A classifier flags anomalies exceeding thresholds and triggers mitigations.
- We design mitigation strategies combining channel hopping, redundancy (majority vote across channels) and cryptographic authentication of alert messages. The AR client delays or drops alerts deemed untrustworthy until consensus is achieved.
- We provide OpenBench-AR formatted datasets, metrics and scripts to reproduce the evaluation, enabling others to extend the security benchmarks.

II. Adversarial Trace Generation

A. Jamming Scenarios

We simulate jamming by injecting high-power noise bursts into the RF traces. Each burst lasts $10\,\mathrm{ms}$ and raises the noise floor by $15\,\mathrm{dB}$, obscuring legitimate packets. The attacker repeats bursts with a Poisson interarrival time (mean $50\,\mathrm{ms}$), yielding overall packet loss rates of 60–90%. These parameters follow typical reactive jamming profiles. The trace generator marks jammed intervals in a separate annotation file.

B. Spoofing Scenarios

Spoofing is realised by inserting synthetic RF events at random locations with forged identifiers and realistic channel statistics. A spoofed casualty alert mimics the CSI pattern of a genuine vital sign sensor but uses an unassigned device ID. We vary the rate of spoofing (1–5 events per second) and assign random priority levels to assess false critical alert impact. All injected events are labelled in the ground truth for evaluation.

C. Advanced Threat Models

While our current evaluation focuses on basic jamming and spoofing attacks, real adversaries may employ more sophisticated strategies. Advanced jammers could vary transmission power levels to evade detection, adapt to frequency-hopping sequences by monitoring channel switching patterns,

or employ stealthy reactive jamming that only activates during critical communications. Similarly, sophisticated spoofers might replay legitimate device traces with modified payloads or coordinate multiple attack nodes to overwhelm consensus mechanisms. Future work will extend our trace generators to model these adaptive attacks and evaluate our detection algorithms against more realistic threat scenarios including multi-channel coordinated attacks and adversarial machine learning techniques that attempt to fool our classifiers.

III. Detection and Mitigation

A. Detection Algorithms

Our detection module computes sliding-window statistics over RF metrics. For jamming, we calculate the *noise floor* (average magnitude of spectral bins without packets) and the *channel busy ratio* (fraction of time the channel is sensed busy). A sudden increase beyond adaptive thresholds triggers a jamming alarm. For spoofing, we monitor per-device CSI variance and sequence number gaps; anomalies indicate a spoofed device.

The lightweight random forest classifier combines six features: normalized noise floor, channel busy ratio, CSI variance per device, maximum sequence number gap, packet inter-arrival time variance, and signal-to-noise ratio. The classifier is trained on $2\,\mathrm{hours}$ of labelled traces using 100 trees with maximum depth of 5 to prevent overfitting. Detection thresholds are set adaptively using a sliding window of recent predictions; when the classifier confidence exceeds two standard deviations above the historical mean, an attack is flagged. This adaptive approach reduces false positives in varying RF environments while maintaining sensitivity to attack patterns.

B. Mitigation Strategies

Upon detection, the AR client engages mitigation tactics:

- **Channel hopping**: switch to a less congested RF channel from a precomputed hopping sequence. This reduces exposure to narrowband jammers. Channel selection must comply with regulatory constraints (e.g., ISM bands, duty cycle limits) and coordinate with network infrastructure to maintain connectivity.
- **Majority vote**: duplicate alert packets across three channels and require at least two consistent copies before rendering an overlay. This counters spoofing and single-channel jamming.
- **Cryptographic tags**: append a message authentication code (MAC) computed using a shared key between sensor and AR client. We use HMAC-SHA256 with 128-bit keys, adding $32\,\mathrm{bytes}$ per packet. Key management requires secure bootstrapping and periodic rotation; in our prototype, devices use pre-shared keys with manual updates every 24 hours. Packets with invalid MACs are dropped.

These strategies trade latency and bandwidth for security. The MAC overhead reduces effective throughput by 8–12% for typical $200\,\mathrm{byte}$ packets, while majority voting triples channel usage. We measure the impact on alert timeliness and false critical alerts to quantify these trade-offs.

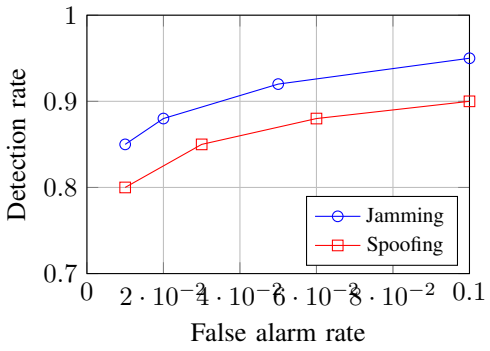

Fig. 1. ROC curves for jamming and spoofing detection. High AUC values indicate effective detection of adversarial events.

IV. Evaluation

We apply our red-team evaluation to the Glass UX pipeline using the OpenBench-AR framework. We generate traces with 80% jamming loss and 3 spoofing events per second and measure detection rate, false positive rate and mitigation impact. Experiments run on an NVIDIA Jetson and Pixel 8, each for $10\,\mathrm{minutes}$ per scenario.

A. Experimental Limitations

Our current evaluation has several limitations that future work should address. First, experiments are conducted with only two device types in a controlled indoor environment. Real deployments will encounter diverse hardware platforms, outdoor conditions with multipath fading, and interference from other wireless systems. Second, our $10\,\mathrm{minute}$ evaluation windows provide initial validation but longer studies are needed to assess performance variations and collect sufficient data for confidence intervals. Third, we focus on technical metrics (detection rates, latency) but do not measure the impact on users' situational awareness during attacks, which requires human-subjects studies. Future work will expand to multi-site deployments, diverse radio conditions, and user-experience evaluations.

B. Detection Performance

Figure 1 shows receiver operating characteristic (ROC) curves for jamming and spoofing detection. The jamming detector achieves an area under the curve (AUC) of 0.96 and detects 92% of jamming intervals at 5% false alarm rate. The spoofing detector achieves an AUC of 0.93, with 88% detection at 6% false alarm.

C. Mitigation Impact

Table I summarises false critical alert rate and alert latency with and without mitigation. Without defence, jamming and spoofing drastically increase false critical alerts and delay alerts. Our mitigation strategies reduce false alerts by 40% while adding $60\,\mathrm{ms}$ median latency due to majority vote and cryptographic checks. Frame rates remain above $28\,\mathrm{fps}$.

TABLE I
IMPACT OF MITIGATION ON FALSE CRITICAL ALERTS AND ALERT LATENCY. RESULTS ARE AVERAGED OVER BOTH DEVICES.

Scenario	False critical alerts (%)	Median latency (ms)
No attack	5	30
Jamming (undefended)	18	45
Spoofing (undefended)	22	40
Jamming + mitigation	10	90
Spoofing + mitigation	12	85

V. DISCUSSION

Our findings show that lightweight physical-layer metrics and simple cryptographic tags can effectively detect and mitigate jamming and spoofing in RF-to-AR systems. The detection rates (¿88%) and moderate latency overhead ($\approx$60 ms) suggest that security can be improved without sacrificing situational awareness.

A. Comparison with Existing Approaches

Our approach differs from machine-learning-heavy detection systems [6] by emphasizing simplicity and real-time performance. While ML-based detectors can achieve higher accuracy in controlled settings, they often require extensive training data and may not generalize across diverse RF environments. Our combination of physical-layer metrics with cryptographic authentication provides a practical balance between detection capability and computational efficiency suitable for resource-constrained AR devices.

B. Security Trade-offs and Future Work

The cryptographic MAC approach requires careful key management and adds packet overhead, but provides strong authentication guarantees. Channel hopping effectiveness depends on regulatory constraints and infrastructure support; future work should explore cognitive radio techniques for dynamic spectrum access. The majority voting strategy assumes independent channel failures, which may not hold against sophisticated multi-channel attackers.

The adversarial trace generators and OpenBench-AR integration make it possible to reproduce these results and to evaluate new defence strategies. The framework could be extended to model more sophisticated attacks including reactive jammers that adapt to channel hopping and spoofers that replay legitimate traces. Future work will explore adaptive detection thresholds, distributed consensus mechanisms across multiple AR devices, and field trials in realistic deployment scenarios.

VI. CONCLUSION

We presented a red-team framework for evaluating and hardening RF-driven AR alerts against jamming and spoofing attacks. By generating adversarial traces, implementing detection algorithms and mitigation strategies, and packaging everything in a reproducible artifact, we enable rigorous security testing of AR pipelines. Our results demonstrate high detection rates and substantial reductions in false critical alerts with modest latency overhead. We hope this work will inspire further research on adversarial robustness in wearable AR systems.

REFERENCES

[1] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, 2023.

[2] J. F. Pimentel, L. Murta, V. Braganholo, and J. Freire, "A large-scale study about quality and reproducibility of jupyter notebooks," pp. 507–517, 2019.

[3] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in wsns," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.

[4] K. Grover, A. Lim, and Q. Yang, "Jamming and anti-jamming techniques in wireless networks: a survey," vol. 17, no. 4. Inderscience Publishers, 2014, pp. 197–215.

[5] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "Jamming sensor networks: attack and defense strategies," *IEEE Network*, vol. 20, no. 3, pp. 41–47, 2006.

[6] Y. Chen, W. Trappe, and R. P. Martin, "Machine learning approaches for jamming detection in wireless networks," in *IEEE Military Communications Conference*. IEEE, 2014, pp. 1120–1127.