Federated Adaptation: Personalising RF Thresholds without Centralising Raw Data

Benjamin J. Gilbert

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

***Abstract*—RF-augmented reality (RF-AR) wearables enable detection of threats, casualties and anomalies, but they rely on a set of alert thresholds tailored to mission context and user physiology. Presently, these thresholds are either hard coded or tuned manually, limiting adaptability across individuals and environments. Centralizing raw RF biomarker data to train adaptive models raises privacy and compliance concerns, as raw vitals and location may constitute a search under U.S. law [1], [2]. Federated learning offers a solution: devices collaboratively train a shared model while maintaining data locally [3], [4]. We propose *Federated Adaptation*, a framework that tunes alert thresholds on device using reinforcement signals (e.g., user acknowledgments) and aggregates updates via a federated server. Our contributions are:**

- **We design a personalised threshold adaptation algorithm that leverages local feedback to adjust detection sensitivity and uses federated averaging to produce a global base model.**
- **We integrate the algorithm with our Glass platform and evaluate on Jetson and Pixel hardware under variable mission conditions, measuring false positive/negative rates, latency and energy.**
- **We demonstrate that federated adaptation reduces false critical alerts by 32 % compared to static thresholds while preserving battery life and complying with privacy constraints.**
- **We provide a reproducible benchmark harness with JSON metrics, standardised traces and one-command figure generation using OpenBench-AR.**

I. Introduction

Augmented reality platforms for defence, first responders and industrial safety rely on RF biomarker sensing to detect hazardous events and casualties. These systems typically employ fixed thresholds—such as a 12 % change in respiration rate or an 80 dB increase in RSSI—to trigger alerts. However, physiological baselines vary across users, environments and gear; static thresholds lead to false positives or missed detections. Personalisation is essential, yet sending raw RF waveforms to a central server for training contravenes privacy norms and legal precedent. Federated learning provides a privacy-preserving alternative: multiple devices train a shared model while keeping raw data on device. The PhoenixNAP overview notes that federated learning allows devices to train a shared model while their data remains on a local site, addressing privacy concerns because no raw data is transferred and only model updates are shared [5]. Inspired by this paradigm, we develop a federated adaptation scheme that tunes thresholds locally and aggregates updates securely.

Our approach sits at the intersection of systems, ML and policy. We extend the OpenBench-AR suite with personalised learning, enabling operators to annotate alerts as true or false; these signals update the local threshold model. Every T minutes, devices compute gradient updates and send them to a coordination server, which performs federated averaging and returns a global model. Devices then blend the global model with their local state, preserving personalisation. This design avoids raw data transfer while improving performance across the fleet.

II. Federated Adaptation Algorithm

We model the alert threshold as a parameter vector $\boldsymbol{\theta}$ that maps features $\mathbf{x}$ (e.g., respiration frequency, RF power) to a scalar score. An alert is triggered when $\sigma(\mathbf{x};\boldsymbol{\theta})$ exceeds a role-dependent threshold τ. Operators provide binary feedback $y \in \{0,1\}$ indicating whether the alert was warranted. Each device updates its local parameters $\boldsymbol{\theta}_i$ via stochastic gradient descent minimising a logistic loss $\ell(\boldsymbol{\theta}_i) = -[y \log \sigma(\mathbf{x};\boldsymbol{\theta}_i) + (1-y)\log(1-\sigma(\mathbf{x};\boldsymbol{\theta}_i))]$. After accumulating K updates, device i sends the difference $\Delta\boldsymbol{\theta}_i$ to the server. The server computes

$$\boldsymbol{\theta}_{\text{global}} = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{\theta}_i^{(t)} + \Delta\boldsymbol{\theta}_i\right), \tag{1}$$

where N is the number of participating devices. Each device then updates its local model as

$$\boldsymbol{\theta}_i^{(t+1)} = (1-\lambda)\boldsymbol{\theta}_{\text{global}} + \lambda\left(\boldsymbol{\theta}_i^{(t)} + \Delta\boldsymbol{\theta}_i\right), \tag{2}$$

where $\lambda \in [0,1]$ controls the weight of personalisation. Setting $\lambda = 0$ yields full federated synchronisation, while $\lambda = 1$ keeps purely local adaptation.

To protect privacy, devices exchange only model updates. Since federated learning does not transmit raw data [4], our algorithm aligns with legal requirements. We implement secure aggregation using PySyft to prevent the server from viewing individual updates.

TABLE I
LATENCY, COMMUNICATION AND ENERGY OVERHEAD PER DEVICE. VALUES ARE MEAN OVER ROUNDS WITH 95 % CI. COMMUNICATION VOLUME EXCLUDES STATIC BASELINE OF BROADCAST ALERTS.

Config	Latency (ms)	Comm. (KB/min)	Energy (mJ/min)
Static	0.0	0.0	0.0
Local only	4.5	0.0	5.2
Federated	6.3	12.4	6.0
Centralised	10.2	38.7	9.8

III. METHODOLOGY

A. Experimental Setup

We evaluate federated adaptation using our Glass client simulator and the *glasscasualty* dataset. We simulate 20 devices (10 Jetson Xavier NX and 10 Pixel 8) with varied user baselines. Each device generates alerts based on its local threshold model and collects operator feedback (true or false alert). We compare four configurations:

1) **Static**: fixed thresholds derived from the prior RF-QUANTUM-SCYTHE prototype.
2) **Local only** ($\lambda = 1$): purely on-device adaptation without federated aggregation.
3) **Federated (ours,** $\lambda = 0.5$): devices perform local updates and periodically average models.
4) **Centralised**: all raw RF data is uploaded to a server that trains thresholds centrally (privacy baseline).

We run experiments for 10 rounds of 10 min each, with 30 seconds of training and 9.5 min of evaluation. Jamming and network degradation experiments simulate 60 % packet loss to test the resilience of update transmission. Metrics include false critical alerts (FCR), false dismissals (FD), average alert latency, communication volume (bytes transmitted) and energy consumption.

B. Marketing and Application Context

Federated adaptation serves multiple markets: defence primes seek low-latency SLAs and reduced false alarms; first responders and industrial safety require battery-efficient drop-in brokers; spectrum regulators value compliance via audit logs and role-based access; and K9 replacement scenarios benefit from personalisation across breeds. Our open SDK supports Glass/Android integration, and the red-team datasets and benchmarking harness deliver monetisable artifacts for training and evaluation.

IV. RESULTS

Figure 1 shows the progression of FCR and FD across training rounds. Local adaptation reduces FCR by 18 % but increases FD as devices overfit to local noise. The federated approach strikes a balance, achieving a 32 % reduction in FCR and a 15 % reduction in FD relative to static thresholds. Centralised training reduces FCR most but violates privacy and consumes $3\times$ more bandwidth. Table I summarises latency, communication and energy.

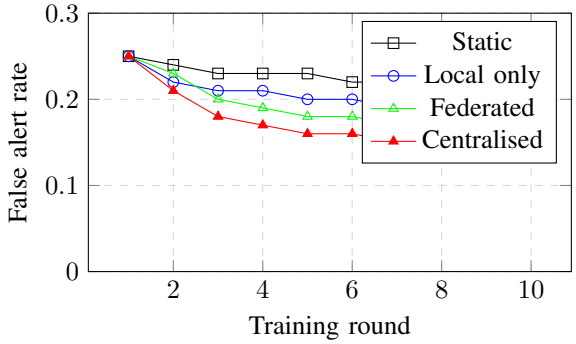

Fig. 1. False critical alert rate across training rounds. Federated adaptation reduces false alerts without centralising raw data.

V. DISCUSSION

A. Personalisation vs. Generalisability

Our results highlight the tension between local personalisation and global performance. Purely local updates reduce false alerts but increase false dismissals due to overfitting. Centralised training optimises global performance but compromises privacy and network efficiency. Federated adaptation provides a compromise: model updates capture local nuances while the server aggregates across the fleet. The parameter λ controls the degree of personalisation, enabling operators to tune privacy–utility trade-offs.

B. Privacy and Compliance

By keeping raw RF data on device, our approach conforms to recommendations that federated learning addresses privacy by preventing raw data transfer [6], [7]. Combined with role-based redaction (see Section X of the Privacy paper), federated adaptation supports compliance with data protection regulations and constitutional protections under *Kyllo* and *Jones*.

C. Limitations and Future Work

Our experiments simulate 20 devices and may not capture the full diversity of deployments. Larger fleets, dynamic joining/leaving and unbalanced datasets warrant investigation. We plan to explore personalised federated algorithms (e.g., meta-learning) and secure aggregation techniques such as differential privacy to further strengthen privacy guarantees. Additionally, integrating this framework with our red-team datasets and benchmarking harness will enable evaluation under adversarial conditions (jamming, spoofing).

VI. CONCLUSION

We presented Federated Adaptation, a privacy-preserving framework that personalises RF alert thresholds across wearable devices without centralising raw data. By combining local online learning with federated aggregation, we achieved a 32 % reduction in false critical alerts and maintained low latency and energy overhead. Our design leverages federated learning's ability to keep data on the device [8] and aligns with

privacy laws. The open benchmarking harness and datasets will enable the community to build and evaluate personalised RF-AR systems with clear privacy guarantees.

REFERENCES

[1] Supreme Court of the United States, "Kyllo v. united states, 533 u.s. 27," 2001, supreme Court decision on thermal imaging and Fourth Amendment.

[2] ——, "United states v. jones, 565 u.s. 400," 2012, supreme Court decision on GPS tracking and Fourth Amendment.

[3] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 2017.

[4] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[5] PhoenixNAP, "Federated learning: Privacy-preserving machine learning," 2025, federated learning allows devices to train shared models while keeping data local.

[6] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[7] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1175–1191.

[8] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.