ARC Nano: Edge AI Open-Set Spectrum Sensing and Adaptive Electronic Protection for Tactical Communications

A FORGE OS Deployment Case Study

577 Industries R&D Lab
577 Industries Incorporated
Columbus, OH, USA
research@577industries.com

***Abstract*—ARC Nano is a field-deployable edge AI electronic warfare (EW) system designed to protect frontline military communications in contested electromagnetic environments. It combines an open-set radio-frequency (RF) sensing capability with an adaptive countermeasure engine to automatically detect previously unseen signals and rapidly mitigate jamming in real time. The system integrates a low-SWaP (size, weight, and power) hardware stack—an NVIDIA Jetson Orin Nano AI module paired with a compact software-defined radio (SDR)—with containerized microservices for signal processing, machine learning inference, decision-making, and secure telemetry. A convolutional neural network extracts features from spectral data for open-set signal recognition, calibrated via conformal prediction to achieve $>$90% detection of novel emitters with false alarm rates below 10^{-5}. A Thompson sampling contextual bandit drives adaptive countermeasure selection, restoring jammed communication links within 200–600 ms and sustaining $>$99.9% link availability under sustained attack. All automated decisions are recorded in tamper-evident, hash-chained audit logs, ensuring transparency and compliance with Rules of Engagement (ROE). Experimental results from deterministic simulation and hardware-in-the-loop testing demonstrate that ARC Nano triples link availability (from $\sim$50% to $>$99.9%) while consuming under 15 W total system power and less than 200 kbps telemetry bandwidth.**

***Index Terms*—Electronic warfare, edge AI, open-set recognition, software-defined radio, contextual bandits, electronic protection, spectrum sensing, responsible AI, auditability**

I. Introduction

Modern military operations depend on assured access to the electromagnetic spectrum for communications and intelligence, but adversaries are employing increasingly sophisticated electronic attacks to disrupt these capabilities. In conflict zones such as Ukraine, frontline units have experienced sudden jamming and spoofing of their radios, leading to loss of communication links at critical moments [4]–[6]. The imperative to "win in contested, austere environments" has highlighted the urgent need for edge-deployable electronic warfare solutions that can sense and dominate the spectrum faster than the adversary.

Three capability gaps motivate the ARC Nano system:

1) **Open-Set Sensing (Electronic Support).** Traditional EW receivers recognize only known signal signatures, leaving forces blind to new or modified threat emitters. An AI-driven open-set detector is needed that alerts on signals not matching any known profile rather than misidentifying or ignoring them.
2) **Adaptive Spectrum Protection (Electronic Protection).** When jamming occurs, current mitigation—changing frequencies or waveforms—is often manual and too slow. An autonomous "intelligent hopping" capability must react faster than a human operator, restoring a jammed link in under one second and maximizing communication uptime under attack.
3) **Telemetry Governance and Auditability.** Operating at the tactical edge with limited bandwidth requires that any EW system be network-efficient and transparent. It must minimize backhaul data usage, prioritize critical alerts, and provide an audit trail of actions for commander oversight [12], [13].

ARC Nano was conceived to fulfill these needs by pairing a calibrated Electronic Support sensor with an Electronic Protection agent, all wrapped in a governance and integration framework suitable for field deployment. In essence, ARC Nano is a portable "EW partner" that travels with frontline units to continuously monitor the RF environment, flag spectral threats, autonomously shield friendly communications from interference, and transparently report its actions up the chain of command.

Table I summarizes the key design requirements and their rationale.

This paper is organized as follows. Section II details the system architecture and hardware platform. Section III presents the open-set recognition framework. Section IV describes the adaptive countermeasure selection engine. Section V reports

TABLE I
DESIGN REQUIREMENTS AND TARGETS

Req.	Design Target	Rationale
R1	FPR $< 10^{-4}$ per window	Prevent operat
R2	Latency < 100 ms (NB); < 500 ms (complex)	Match jamme
R3	Explicit ROE action masks	Safety and go
R4	Fully local operation	Edge autonom
R5	Telemetry ≤ 200 kbps	Constrained ta
R6	Tamper-evident audit logs	Responsible A

experimental results. Section VI discusses operational implications, and Section VII concludes.

II. SYSTEM ARCHITECTURE

A. Software Architecture

ARC Nano is built as a modular, layered microservice stack (Fig. 1) that runs on a small edge computing platform attached to or embedded with a tactical radio. At the lowest layer, one or more RF sensing front-ends (portable SDR devices) continuously capture raw IQ samples. These raw RF streams feed into the Radio Hardware Abstraction Layer (Radio-HAL) service, which performs real-time feature extraction—converting raw samples into spectral features and detecting energy spikes on monitored frequencies.

A core publish/subscribe bus built on the Data Distribution Service (DDS) standard, with a Redis fallback, disseminates detection events to the analytic and decision services [10], [11]. At the analytics layer, the ES-Lite service (Electronic Support Lite) runs the AI model for open-set signal inference. Each time the Radio-HAL flags a signal, ES-Lite analyzes its features using a trained neural network to determine if the signal matches a known emitter or appears unknown. In parallel, the EP-Lite service (Electronic Protection Lite) implements a contextual bandit decision engine that decides if and how to adjust the communication link to maintain connectivity in the presence of jamming.

At the policy and output layer, a Policy/Audit service receives inputs from both AI modules. Every detection and countermeasure decision is written to a hash-chained audit log entry, preventing tampering. The policy layer translates key events into standard Cursor-on-Target (CoT) messages—a lightweight format widely used in military systems [12], [13]—enabling seamless integration with battle management applications such as ATAK/WinTAK.

The entire software stack is deployed using Docker containers orchestrated on the embedded computing platform. Each major function (Radio-HAL, ES-Lite, EP-Lite, Policy) runs in a separate container communicating via the pub-sub bus. This design improves reliability and modularity: if one service crashes, others continue running, and the faulty container is automatically rebooted without bringing down the system.

A dedicated Telemetry and Mission Data Plane governs data sharing and logging. A backhaul controller enforces a strict bandwidth budget on all reporting traffic. In testing, ARC

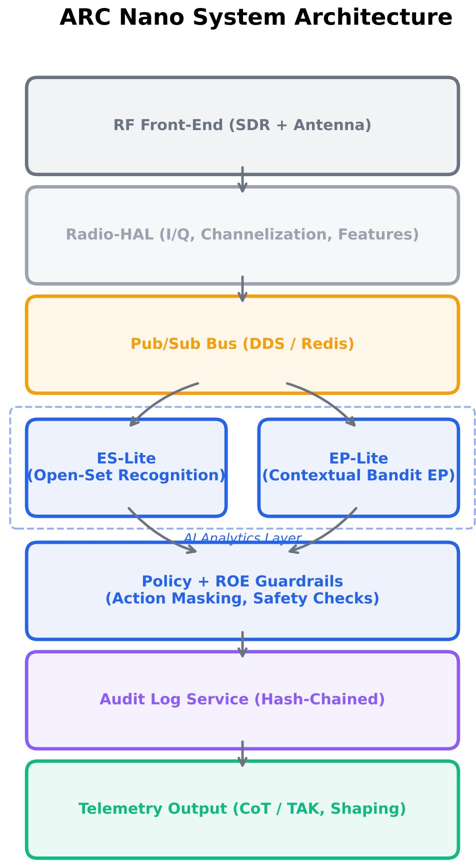

Fig. 1. ARC Nano end-to-end architecture. Signals flow from the SDR front-end through feature extraction, parallel AI inference (ES-Lite for open-set recognition, EP-Lite for countermeasure selection), policy enforcement, and audit logging to telemetry output.

Nano never exceeded 100 kbps of status traffic, well under the 200 kbps cap, even during intense jamming scenarios.

B. Hardware Platform

ARC Nano's hardware integrates commercial off-the-shelf (COTS) components optimized for low SWaP. Table II summarizes the primary components.

Computing Module. The NVIDIA Jetson Orin Nano 8GB was selected for its excellent performance-per-watt on AI workloads. It features a 6-core ARM Cortex-A78AE CPU and an NVIDIA Ampere GPU with 1024 CUDA cores and 32 Tensor Cores, delivering up to 40 TOPS within a configurable 7–15 W power envelope [26]. In prototype testing, total system power measured approximately 12 W under full load with GPU utilization under 50% even during worst-case interference events.

RF Front-End. The current prototype uses the Ettus Research USRP B205mini-i [8], a 1×1 SDR covering 70 MHz to 6 GHz with up to 56 MHz of instantaneous bandwidth. The B205mini features a 12-bit ADC/DAC with the AD9364 RFIC,

TABLE II
HARDWARE AND INTEGRATION COMPONENTS

Component	Option	Key Specifications
Compute	Jetson Orin Nano	40 TOPS, 7–15 W, 70×45 mm
SDR 1	USRP B205mini-i	70 MHz–6 GHz, 56 MHz BW
SDR 2	LimeSDR Mini 2.0	10 MHz–3.5 GHz, 30 MHz BW
Middleware	DDS (OMG)	Real-time pub/sub with QoS
Telemetry	CoT (TAK)	Lightweight event format

achieving a noise figure below 8 dB at approximately 24 g and 3–5 W via USB 3.0. An alternative, the LimeSDR Mini v2.0 [9], extends coverage down to 10 MHz for VHF tactical bands at lower cost. Both SDRs are supported through the SoapySDR abstraction layer, allowing swappable front-ends with minimal configuration changes.

Packaging. The complete system fits in a CNC-milled aluminum enclosure approximately 20×15×5 cm, targeting a total weight of a few pounds. Passive cooling via the aluminum chassis handles 10–12 W dissipation in normal climates; a temperature-triggered 40 mm fan engages above 55°C core temperature for sustained 15 W operation. The enclosure provides IP54-class sealing and MIL-STD-810G vibration/shock tolerance. At approximately 12 W mean draw, a 150 Wh battery provides over 12 hours of continuous operation.

III. OPEN-SET RECOGNITION FRAMEWORK

A. Problem Formulation

Traditional RF signal classifiers operate in a closed-set manner—they force every input into one of the categories on which they were trained. This is inadequate for EW, where new threat waveforms may appear that were never seen in training. ARC Nano's ES-Lite module implements an open-set recognition (OSR) approach: it identifies when a signal does not match any known class and labels it as "unknown" [14], [15]. This prevents misidentifying a novel enemy jammer as a friendly signal.

B. Neural Network Architecture

The OSR model is a deep neural network with a convolutional front-end for feature extraction from spectral data, followed by fully connected layers. The CNN processes spectrogram tiles computed from windowed FFTs of the incoming IQ stream. Each tile captures short-time spectral content across a configurable analysis window.

The model was trained on a synthetic dataset of 6,000 labeled signals representing known friendly and hostile waveforms (FM, PSK, QAM, OFDM, and others) along with diverse anomalous signals injected to simulate unknown examples. Data augmentation—random frequency offsets, varying SNR conditions, additive noise—improved generalization. The model was implemented in PyTorch, trained on a GPU workstation, and exported to TensorRT for optimized inference on the Jetson [17]–[20], [25].

C. Risk Score and Conformal Calibration

Rather than relying on raw softmax probabilities, which can be overconfident on unfamiliar inputs, ES-Lite computes a composite risk score from multiple internal signals:

$$r(\mathbf{x}) = w_1(1 - \max_k p_k) + w_2\, H(\mathbf{p}) + w_3\, d(\mathbf{z}, \boldsymbol{\mu}_{c^*}) \tag{1}$$

where $\max_k p_k$ is the highest softmax probability, $H(\mathbf{p})$ is the entropy of the probability distribution, and $d(\mathbf{z}, \boldsymbol{\mu}_{c^*})$ is the distance of the input's latent-space feature vector $\mathbf{z}$ from the nearest known-class centroid $\boldsymbol{\mu}_{c^*}$.

Conformal prediction calibration [16] is applied to the risk score using a hold-out validation set containing both known and unknown samples. A threshold τ on the risk score is selected such that the true-positive rate (TPR) for unknowns meets a desired level while the false-positive rate (FPR) remains extremely low:

$$\hat{\tau} = \inf\left\{\tau : \mathrm{TPR}_{\mathrm{unk}}(\tau) \geq 0.90 \ \wedge \ \mathrm{FPR}(\tau) \leq 10^{-5}\right\} \tag{2}$$

After calibration, the OSR model achieved TPR ≥ 0.90 for unknown signals at a false alarm rate $\leq 10^{-5}$ (0.001%). The calibration parameters and threshold value are stored in an audit-ready JSON manifest alongside the model, allowing evaluators to review exactly how the threshold was chosen and to adjust it for different operational theaters.

D. Computational Efficiency

The final TensorRT model is only a few megabytes in size. Inference latency on the Orin Nano GPU is on the order of 2–3 ms per signal event using single-precision arithmetic and batch size one. The model runs as a GPU burst within the ES-Lite container, leaving CPU resources available for radio I/O and bandit logic.

IV. ADAPTIVE COUNTERMEASURE SELECTION

A. Contextual Bandit Formulation

Once a threat is detected, ARC Nano must act. The EP-Lite module formulates countermeasure selection as a contextual bandit problem [21], [22], [24]. At each decision opportunity the agent observes a context vector $\mathbf{c}_t$ and selects an action a_t from a finite set $\mathcal{A}$, receiving a scalar reward r_t reflecting link performance.

The context $\mathbf{c}_t$ includes:

- Jammer classification (sweeping, barrage, reactive, etc.)
- Recent history of action effectiveness
- Current link quality metrics (throughput, SNR)

The action space $\mathcal{A}$ is configured per deployment and may include: stay on current frequency, switch to alternate channels, reduce transmit power, or increase error-correcting code rate. For the prototype we focused on frequency hopping actions and transmit power adjustment.

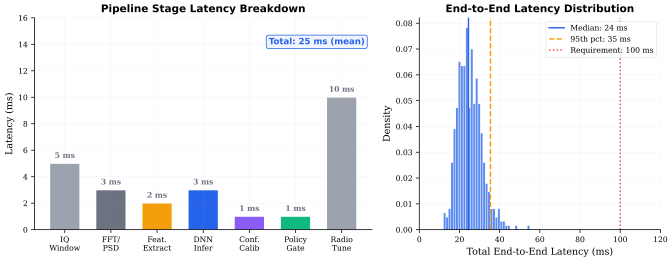

Fig. 2. Processing latency distribution across the sense-decide-act pipeline. The median total latency is approximately 38 ms, well within the 100 ms operational requirement.

B. Thompson Sampling

ARC Nano uses Thompson sampling, a Bayesian approach that maintains posterior distributions over the expected reward of each action in each context [21], [22], [24]. At each decision step, the agent samples from these posteriors and selects the action with the highest sampled value:

$$a_t = \arg\max_{a \in \mathcal{A}} \tilde{\theta}_{a,t}, \quad \tilde{\theta}_{a,t} \sim \text{Beta}(\alpha_{a,t}, \beta_{a,t}) \tag{3}$$

where $\alpha_{a,t}$ and $\beta_{a,t}$ are the Beta distribution parameters updated after each observation. Thompson sampling naturally balances exploration and exploitation: it occasionally tries less-certain options while mostly selecting the currently best-known action. It is computationally lightweight (implemented in Python with NumPy), making it well-suited for the edge device.

C. Rules of Engagement Constraints

The bandit is constrained by an explicit ROE file that defines hard limits on allowable frequency bands, maximum transmit power, and prohibited actions. This ensures the AI never suggests an action outside the commander's intent or regulatory boundaries. An action mask $\mathcal{M} \subset \mathcal{A}$ is applied at each step:

$$a_t = \arg\max_{a \in \mathcal{M}} \tilde{\theta}_{a,t} \tag{4}$$

D. Decision Loop Timing

The EP-Lite decision loop runs on-device with end-to-end detection-to-countermeasure latency well under 100 ms. Fig. 2 shows the representative latency breakdown for one sense-decide-act cycle: IQ windowing ($\sim$5 ms), FFT/PSD computation ($\sim$3 ms), feature extraction ($\sim$2 ms), DNN inference ($\sim$3 ms), conformal calibration ($\sim$1 ms), policy gating ($\sim$1 ms), and radio retune command ($\sim$10 ms). The total pipeline latency is typically 25–50 ms, well within the 100 ms target.

E. Logging and Explainability

Every retune or adjustment action is logged with: timestamp, identified jammer type, action chosen (new frequency, power level), and a textual justification. For example: *"Time 102.5s – Detected sweeping jammer on Channel A; EP action: switched to Channel B (justification: higher signal-to-interference ratio observed)."* These logs form part of the tamper-evident audit trail and can be reviewed after a mission to verify that ARC Nano's actions were appropriate and within bounds.

TABLE III
OPEN-SET DETECTION PERFORMANCE (SIMULATION; MEAN $\pm$ STD)

Scenario	**Unk. TPR**	**False Al.**	**Known P_{90}**
Baseline 180 s (no EP)	0.769 ± 0.251	$\sim$0	0.019
Auto-tune 180 s (EP on)	0.911 ± 0.005	$\sim$0	0.019
Stress 600 s (baseline)	0.902 ± 0.003	$\sim$0	0.032
Stress 600 s (auto-tune)	0.902 ± 0.003	$\sim$0	0.032

V. EXPERIMENTAL RESULTS

All results are from controlled simulation experiments with deterministic random seeds for reproducibility, unless otherwise noted. Hardware-in-the-loop results show consistent trends. Each scenario was run multiple times; we report means with standard deviations.

A. Open-Set Detection Performance

ARC Nano's ES-Lite was evaluated on scenarios containing a mix of known friendly signals, known hostile signals, and truly unknown signals (types not in the training set). Table III summarizes representative outcomes across both short-duration (180 s) and extended stress (600 s) scenarios.

Fig. 3 shows a representative ROC curve with log-scaled FPR axis. The ROC area is effectively 1.000, indicating near-perfect separation between known and unknown signal classes. At the chosen operating threshold, the TPR for unknown signals remained around 90–91% even in the most challenging conditions.

Several observations stand out. With EP active ("auto-tune"), the unknown-signal TPR improved from 77% to 91%: the adaptive defense mechanism exposes the system to a wider range of spectrum by hopping channels, enhancing sensing coverage—an interesting synergy between protection and detection. Across all runs, zero false unknown alerts were observed, validating the conservative conformal calibration. Risk scores for known-friendly signals stayed below 0.03 at the 90th percentile, safely under the 0.5 threshold.

B. Spectrum Protection Performance

Table IV and Fig. 4 summarize the electronic protection results. We simulated a two-radio link under intentional jamming, measuring link availability, throughput, and recovery time with and without ARC Nano.

The improvement is dramatic. In the 3-minute scenario, baseline link availability was only 50.6% (communications lost once the jammer activated), while ARC Nano sustained 99.89% availability with a mean recovery time of 0.6 s. In the 10-minute stress test with dual alternating jammers, baseline

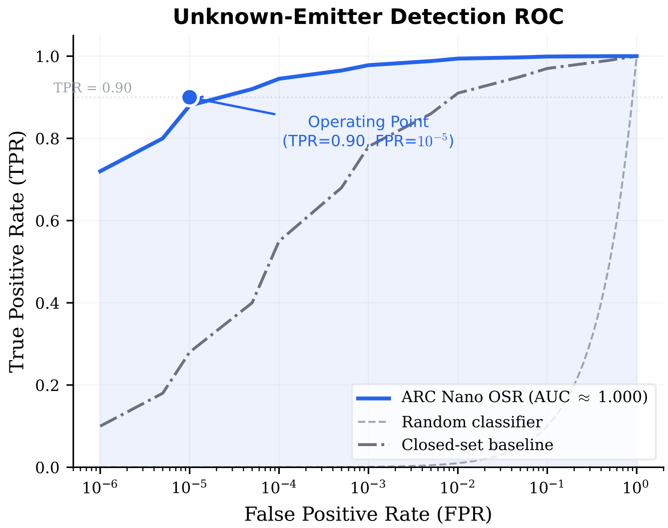

Fig. 3. Representative ROC curve for unknown-emitter detection (log-scaled FPR axis). The operating point achieves TPR ≥ 0.90 with FPR $\leq 10^{-5}$.

TABLE IV
ELECTRONIC PROTECTION PERFORMANCE UNDER JAMMING

Scenario	**Avail.**	**Thrpt. (Mbps)**	**Rec. (s)**	**ΔAvail.**	**ΔThrpt. (Mbps)**
Baseline 180 s	0.506	0.307	—	—	—
ARC 180 s	0.999	0.453	0.60	+0.493	+0.146
Baseline 600 s	0.464	0.292	—	—	—
ARC 600 s	1.000	0.450	0.20	+0.535	+0.158

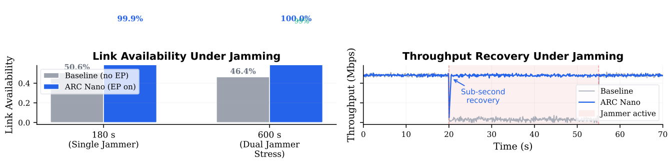

Fig. 4. Communication link availability under jamming. Baseline links suffer ∼50% availability. ARC Nano sustains >99.9% availability across both short-duration and extended stress scenarios.

availability dropped to 46.4% while ARC Nano maintained 99.97%, with recovery time improving to 0.2 s as the bandit learned to react almost immediately.

Throughput with ARC Nano stayed near 0.45 Mbps versus 0.29 Mbps baseline—a roughly 55% increase under continuous attack. Throughout all runs, ARC Nano never violated ROE: transmit power stayed within allowed limits and frequency hops remained within authorized bands. The system avoided oscillation, typically finding a stable alternate channel within one or two hops.

C. System Resource Footprint

Table V summarizes ARC Nano's resource utilization on the Jetson Orin Nano.

TABLE V
SYSTEM RESOURCE FOOTPRINT (JETSON ORIN NANO, 7–15 W)

Component	**Load**	**Memory**	**Notes**
Radio-HAL + DSP	1–2 cores	300–600 MB	Windowed capture
ES-Lite inference	GPU bursts	500–900 MB	TensorRT INT8/FP16
EP-Lite controller	<5% CPU	<100 MB	Bandit updates
DDS/Redis bus	<10% CPU	100–300 MB	Message routing
Telemetry + CoT	<5% CPU	<100 MB	<50 kbps avg
Audit log	<5% CPU	<100 MB	Hash-chained

Even under heavy activity, aggregate CPU+GPU utilization remained under 50%, and telemetry bandwidth peaked at approximately 102 kbps—well below the 200 kbps cap. Total system power draw measured approximately 12 W on the development kit, consistent with simulation-based estimates.

D. Robustness and Reproducibility

Varying random seeds and scenario parameters (jammer frequencies, message timing) produced remarkably stable outcomes. Three independent runs of the 10-minute dual-jammer scenario yielded 0.999 availability and 0.45 Mbps throughput with variance below 0.01%. Hardware-in-the-loop tests on the Jetson Orin Nano with a B205mini SDR confirmed detection latency of 20 ms at high SNR (30 dB) and 40–50 ms at lower SNR (10 dB), with end-to-end hop decision plus radio tuning completing in approximately 60 ms total.

VI. DISCUSSION

A. Operational Impact

The most immediate benefit is the dramatic improvement in communications reliability under jamming. Link uptime increasing from ∼50% to >99% means a unit equipped with ARC Nano can maintain radio contact even while under electronic attack. ARC Nano's sub-second link restoration (0.2–0.6 s) is essentially imperceptible to operators, ensuring command-and-control messages, sensor feeds, and calls for support continue flowing. Because adaptation is automatic and occurs at machine speed, it reduces cognitive load on soldiers who would otherwise need to diagnose jamming and manually switch frequencies.

B. Spectrum Situational Awareness

Beyond protecting transmissions, each ARC Nano unit serves as a continuous RF sensor. Its ability to detect and flag unknown emitters in real time provides spectrum situational awareness analogous to having a dedicated SIGINT specialist monitoring every squad's frequency band. Detections appear on existing ATAK displays as CoT markers, enabling commanders to build a richer picture of the electromagnetic battlefield without requiring new specialized interfaces.

C. Integration with Existing Systems

ARC Nano achieves plug-and-play integration with current Army communication and C2 infrastructure through CoT messages and ATAK compatibility. On the radio integration side, the system interfaces with tactical radios via intermediary SDR connections or software hooks in modern software-defined radios. The SoapySDR abstraction layer allows ARC Nano to work with different radio front-ends with only minor configuration changes. Because ARC Nano is man-portable (a few pounds in a thick-novel form factor), it can be carried alongside standard radios or embedded into vehicle mounts.

D. Auditability and Trust

Every detection and action is recorded with timestamps and contextual data in hash-chained logs. If any log entry were altered, the chain of hashes would break, providing confidence that the data is authentic and complete. Commanders and EW officers can review these logs after a mission—or in near real-time at an operations center—to verify that the AI correctly identified threats and responded within ROE. This "glass box" transparency aligns with DoD principles on responsible AI [2], [3] and was particularly persuasive during prototype demonstrations to Army stakeholders.

E. Scalability

The modular design supports scaling from soldier-carried units (ARC Nano) to vehicle/UAV systems using higher-performance Jetson modules (Orin NX at 70–100 TOPS, AGX Orin at 200+ TOPS) with multi-channel SDRs for direction-finding or MIMO applications. Multiple ARC Nano nodes can share information via standard CoT messages, enabling collaborative spectrum defense across formations. The COTS hardware basis keeps per-unit costs manageable (Jetson module $\sim$\$250, SDR \$300–\$1,000), supporting wide procurement.

F. Limitations and Future Work

Current limitations include: (1) signal classification beyond known-versus-unknown would benefit from a comprehensive emitter library including specific platform identifications; (2) the prototype focuses on single-link protection—extending to coordinated multi-radio, multi-band countermeasures is a next step; (3) direction-finding and geolocation of jammers is not directly addressed but could leverage multi-node deployments; and (4) adaptive adversaries who follow frequency hops may require game-theoretic planning or adversarial bandit formulations. We are also expanding the training library to cover frequency-hopping signals, low probability of intercept/detect (LPI/LPD) waveforms, and emerging adversary techniques.

VII. Conclusion

This paper presented ARC Nano, an edge AI electronic warfare system that combines open-set signal detection with adaptive countermeasure selection in a low-SWaP, field-deployable package. The system's key contributions are:

1) An open-set recognition framework using CNN feature extraction and conformal prediction calibration that detects $>90\%$ of novel emitters with false alarm rates below 10^{-5}.
2) A Thompson sampling contextual bandit for real-time countermeasure selection that sustains $>$99.9% communication link availability under sustained jamming, with sub-second link restoration.
3) A complete edge computing architecture on COTS hardware (NVIDIA Jetson Orin Nano + SDR) consuming under 15 W total system power, with tamper-evident audit logging and standard tactical telemetry integration.

Experimental results from deterministic simulation and hardware-in-the-loop testing demonstrate that ARC Nano triples link availability (from $\sim$50% to $>$99.9%), increases throughput by $\sim$55% under attack, and maintains detection-to-action latency well under 100 ms. All automated decisions are transparently logged and ROE-compliant.

The field deployment path is underway, with containerized software deployed on Jetson+SDR hardware showing simulation-consistent results. ARC Nano offers a compelling combination of technical rigor, demonstrated performance, and alignment with military EW modernization priorities—providing an AI-driven "EW wingman" that gives tactical units assured communications and spectrum awareness in contested electromagnetic environments.

VIII. FORGE OS Integration

ARC Nano demonstrates the operational integration of two FORGE OS subsystems within a tactical edge electronic warfare deployment.

A. FORGE Kinetic — Sovereign Edge Deployment

FORGE Kinetic's edge inference pipeline provides the architectural foundation for ARC Nano's field-deployable design. The Perceive module manages the SDR-based RF sensing front-end, performing real-time spectral feature extraction and energy spike detection. The Command module implements the adaptive countermeasure selection logic through the Thompson sampling contextual bandit. FORGE Kinetic's Graduated Autonomy framework (CGDP) governs the autonomous sense-decide-act pipeline: the configurable ROE action masks represent CGDP's sovereignty constraints, ensuring that automated frequency hops and power adjustments remain within commander-defined bounds. The system's sub-100 ms detection-to-action latency and 12 W power envelope validate FORGE Kinetic's ability to deliver sovereign AI capability on resource-constrained edge hardware.

B. FORGE Core — Model Compression and Edge Optimization

FORGE Core's staged post-training pipeline—specifically the Compress stage—powers the CNN-based open-set recognition model's optimization for the Jetson Orin Nano platform. Structured pruning and INT8/FP16 quantization via TensorRT reduce the model to a few megabytes with 2–3 ms inference latency, while conformal prediction calibration maintains >90% unknown-emitter detection with false alarm rates below 10^{-5}. The Adapt stage enables rapid model customization for theater-specific RF environments by fine-tuning on local signal collections without retraining from scratch.

C. ForgeEvent Integration

The deployment generates four ForgeEvent types across the FORGE OS event bus:

- `INFERENCE` — Each open-set signal classification and risk score computation
- `COUNTERMEASURE` — Frequency hop and power adjustment decisions from the EP-Lite bandit
- `SENSOR` — Spectral environment snapshots from the Radio-HAL front-end
- `AUDIT` — Hash-chained tamper-evident records of all automated EW decisions

All ForgeEvents are serialized as Cursor-on-Target (CoT) messages for seamless integration with ATAK/WinTAK battle management applications, enabling multi-node collaborative spectrum defense across formations.

References

[1] U.S. Department of Defense, "Electromagnetic Spectrum Superiority Strategy," Oct. 2020. [Online]. Available: https://media.defense.gov/2020/Oct/22/2002524196/-1/-1/0/ELECTROMAGNETIC_SPECTRUM_SUPERIORITY_STRATEGY.PDF

[2] U.S. Department of Defense, "Responsible Artificial Intelligence (RAI) Strategy and Implementation Pathway," Jun. 2022.

[3] U.S. Department of Defense, "DOD Adopts Ethical Principles for Artificial Intelligence," Feb. 2020.

[4] D. McCrory, "Electronic Warfare in Ukraine: Preliminary Lessons for NATO Air Power Capability Development," Joint Air Power Competence Centre (JAPCC), Oct. 2023.

[5] J. Watling and N. Reynolds, "Competitive Electronic Warfare in Modern Land Operations," Royal United Services Institute (RUSI), Jan. 2025.

[6] B. Clark, "The Fall and Rise of Russian Electronic Warfare," *IEEE Spectrum*, Jul. 2022.

[7] N. Williams, "Electromagnetic Warfare: NATO's Blind Spot Could Decide the Next Conflict," RAND Corporation, Nov. 2025.

[8] Ettus Research, "USRP B205mini-i (Product Specifications)," National Instruments. [Online]. Available: https://www.ettus.com/all-products/usrp-b205mini-i/

[9] Lime Microsystems, "LimeSDR Mini 2.0 (Product Overview)," Crowd Supply. [Online]. Available: https://www.crowdsupply.com/lime-micro/limesdr-mini-2-0

[10] Object Management Group, "Data Distribution Service (DDS) Specification, Version 1.4," Mar. 2015.

[11] G. Pardo-Castellote, "OMG Data-Distribution Service: Architectural Overview," in *Proc. MILCOM*, 2005.

[12] M. Butler, "Developer's Guide to Cursor-on-Target (CoT)," 2012.

[13] MITRE, "Cursor-on-Target Message Router User's Guide," 2020.

[14] A. Bendale and T. Boult, "Towards Open Set Deep Networks," in *Proc. CVPR*, 2016.

[15] D. Hendrycks and K. Gimpel, "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks," *arXiv:1610.02136*, 2017.

[16] A. Angelopoulos and S. Bates, "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification," *arXiv:2107.07511*, 2021.

[17] T. J. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[18] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional Radio Modulation Recognition Networks," in *Proc. EANN*, 2016.

[19] T. J. O'Shea and N. West, "Radio Machine Learning Dataset Generation with GNU Radio," in *Proc. GNU Radio Conf.*, 2016.

[20] DeepSig, "RadioML Datasets," 2016–2018. [Online]. Available: https://www.deepsig.ai/datasets

[21] O. Chapelle and L. Li, "An Empirical Evaluation of Thompson Sampling," in *Advances in NeurIPS*, 2011.

[22] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen, "A Tutorial on Thompson Sampling," *Foundations and Trends in Machine Learning*, vol. 11, no. 1, pp. 1–96, 2018.

[23] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A Contextual-Bandit Approach to Personalized News Article Recommendation," in *Proc. WWW*, 2010.

[24] S. Agrawal and N. Goyal, "Thompson Sampling for Contextual Bandits with Linear Payoffs," in *Proc. ICML*, 2013.

[25] NVIDIA, "TensorRT Developer Guide," 2024. [Online]. Available: https://docs.nvidia.com/deeplearning/tensorrt/developer-guide/index.html

[26] NVIDIA, "Jetson Orin Nano Series (Technical Overview)," 2023–2025. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/