

Multi-Modal Correlation of Observability Signals in Cloud-Native 5G Core Networks

What Each Modality Reveals About Faults — and What It Misses

David Gherguț[1]

Supervisor(s): Nitinder Mohan[1]**, Sehan Samarakoon Mudiyanselage**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 18, 2026

Name of the student: David Gherguț
Final project course: CSE3000 Research Project
Thesis committee: Nitinder Mohan, Sehan Samarakoon Mudiyanselage, Jérémie Decouchant

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Cloud-native 5G Core networks emit metrics, logs, and distributed traces, yet faults are typically diagnosed within a single modality. We show that the *relationships between* these modalities carry fault information that single-signal analysis misses, and we use them to characterize faults in a containerized Open5GS testbed. Our method computes the change in Spearman rank correlation between cross-modal signal pairs, from a pre-fault baseline to the fault window, yielding a coupling-change metric $\Delta|\rho|$. Across 22 operational fault scenarios over seven independent deployments, and 6 security scenarios over three, the analysis surfaces reproducible coupling signatures and, more importantly, *modality blindspots* that follow the 5G interface architecture: because distributed tracing instruments only the Service-Based Interface, N2 interface partitions and N4/PFCP session faults are trace-blind and characterizable only through metrics-logs coupling, whereas a valid-request NRF flood produces no error logs. No single modality covers every fault type. We treat classification as an analytical instrument rather than a goal: cross-modal coupling features are deliberately weaker classifiers than raw per-signal features, and a SHAP analysis shows the two views rely on different modality pairs, consistent with the coupling view's value being characterization rather than accuracy. The security-fault results are preliminary, owing to the small, low-variance dataset. The contribution is a reproducible, architecture-grounded map of which modality reveals, and which is blind to, each fault.

1 Introduction

The transition to cloud-native microservice architectures has fundamentally changed how 5G Core Networks are deployed and operated. The 3rd Generation Partnership Project (3GPP) Service-Based Architecture (SBA) decomposes the network core into independent, containerized Network Functions (NFs) orchestrated on Kubernetes. While this topology delivers scalability and deployment flexibility, it introduces severe fault diagnosis challenges. When a fault occurs, its effects propagate rapidly across loosely-coupled NF dependencies, generating simultaneous anomalies across the entire system. For network operators, manually finding the root cause in this flood of telemetry is practically infeasible [11; 19], making automated fault localization critical for maintaining strict Service Level Agreements.

Modern observability stacks capture system state through three complementary telemetry modalities: continuous metrics measuring resource consumption, discrete logs recording application-level events, and distributed traces mapping inter-service request flows. Yet existing fault localization approaches typically evaluate them in isolation [20; 2; 17] or rely on opaque deep learning that sacrifices diagnostic transparency [8; 18; 4]. Critically, no prior work has systematically studied how the cross-modal correlation between metrics, logs, and traces changes across different fault types in a containerized 5G Core environment. The closest prior method, HeMiRCA [23], also couples the three modalities through Spearman rank correlation, but applies it to root-cause ranking rather than to characterizing how that coupling shifts across fault types.

This paper addresses this gap by proposing a cross-modal correlation analysis methodology for fault characterization in cloud-native 5G Core networks. We compute Spearman rank correlation between signal pairs spanning different modalities (within an NF and across selected NF pairs), comparing correlation strength during fault windows against a pre-fault baseline to yield a fault-induced coupling change metric $\Delta|\rho|$. We use this coupling view to characterize faults and locate modality blindspots. We treat classification, and a SHAP [15] analysis of which signal pairs the models rely on, as analytical tools rather than performance goals.

The central research question guiding this work is: *How can multi-modal correlation of observability signals (metrics, logs, and traces) characterize faults and expose where single-modality analysis is blind, in cloud-native 5G Core networks?* We decompose this into four sub-questions:

SQ1: How can a method be developed to identify and align cross-modal correlations within an existing observability dataset?

SQ2: How do time-alignment strategies resolve sampling-rate mismatches between continuous and event-driven signals?

SQ3: What temporal and contextual patterns emerge across the combined signals during different fault types?

SQ4: How does multi-modal correlation analysis compare to single-modality baselines in fault discriminability?

The remainder of this paper is structured as follows. Section 2 reviews background and related work. Section 3 describes the methodology. Section 4 presents the experimental results. Section 5 addresses responsible research. Section 6 discusses the findings and their limitations. Section 7 concludes.

2 Background and Related Work

2.1 5G Core Network Architecture

The 3GPP 5G Core adopts a Service-Based Architecture in which each Network Function (NF) exposes RESTful interfaces over HTTP/2. Core NFs include the Access and Mobility Management Function (AMF), which handles UE registration and mobility management; the Session Management Function (SMF), which establishes and manages PDU sessions; the User Plane Function (UPF), which forwards user-plane data packets; the Network Repository Function (NRF), which acts as a service discovery registry; and the Service Communication Proxy (SCP), which mediates inter-NF communication. In cloud-native deployments, these NFs run as Kubernetes pods, forming a dynamic topology in which a failure in one NF can cascade through dependent services. This interdependency makes fault localization considerably harder

than in monolithic systems, as the observable symptom often appears far from the actual root cause [8].

2.2 Observability Signals in Microservice Systems

The three pillars of observability provide complementary views of system state. *Metrics* are continuously sampled numerical measurements, such as container CPU usage and memory working set, collected at fixed intervals by monitoring systems such as Prometheus. *Logs* are discrete event records generated by application code, capturing error conditions and operational state changes collected by systems such as Grafana Loki. *Distributed traces* record the end-to-end lifecycle of requests across microservices, encoding per-service latency and failure points as collected by systems such as Jaeger. These modalities differ fundamentally in their sampling characteristics: metrics arrive at fixed intervals while logs and traces are event-driven and sparse. This sampling rate mismatch makes direct cross-modal comparison non-trivial without a temporal alignment strategy [5].

2.3 Fault Localization Approaches

Existing automated fault localization approaches fall broadly into three categories.

Single-modality methods analyze one telemetry type in isolation using anomaly detection or causal analysis. While tractable, these approaches miss inter-modal dependencies that are only visible when signals from different modalities are examined together. Zhang et al. [22] demonstrate that fusing system logs with monitoring metrics improves fault localization precision by over 10% over single-modality baselines in microservice deployments, providing early evidence that cross-modal analysis adds diagnostic value.

Multi-modal fusion approaches combine multiple signals to capture richer fault context. Wang et al. propose MADMM [21], which constructs separate feature graphs for metrics and logs and employs a Cross-Modal Attention-based GRU (CMA-GRU) to capture temporal dependencies between modalities. Hatami et al. [10] fine-tune a large language model on pod logs, system events, and RTT measurements from a Kubernetes-hosted 5G Core testbed, achieving significant improvements over the base model for fault classification across pod and network fault categories. Graph Neural Network (GNN) based methods model service topology alongside telemetry, but Gao et al. [8] show that a simple MLP retaining multimodal fusion capabilities without graph modeling achieves performance parity with GNN-based SOTA across five datasets, suggesting that the value of multi-modal data lies in the preprocessing and alignment rather than in the model architecture. More recent frameworks fuse all three modalities directly: Triple [18] embeds metrics and logs as node features over a trace-derived service graph and detects anomalies with a spatio-temporal graph network; FAMOS [4] couples independent per-modality feature extractors through a cross-attention mechanism; Li et al. [13] fuse the three modalities via a masked graph autoencoder; and HeMiRCA [23] aligns heterogeneous metric, log, and trace sources for fine-grained root cause analysis using anomaly-aware Spearman correlation between modalities, closely related to our coupling estimator, though applied to root cause ranking rather than per-fault characterization. These approaches achieve strong accuracy, but encode the cross-modal relationship *implicitly* inside a learned representation, requiring a separate interpreter to recover which signals drove a given decision. A complementary line of work instead feeds hand-crafted multi-modal features to conventional classifiers: Kaushik et al. [12] combine trace-log and metric features to proactively predict microservice faults with tree- and recurrence-based models. Our $\Delta|\rho|$ classifier follows this explicit-feature philosophy, but makes the cross-modal *coupling* itself, rather than per-signal magnitudes, the feature.

5G-specific fault diagnosis has historically relied on manual correlation of alarms, logs, and Key Performance Indicators (KPIs), an approach that is increasingly insufficient for the complexity of cloud-native 5G Core deployments. Jiang et al. [11] explicitly note this limitation and propose an automated kNN-Bagging approach on multi-source telemetry from China Telecom production 5G Core data, achieving approximately 92% root-cause localization accuracy. Tong and Jia [19] combine KPI threshold detection, alert-log clustering, and service-chain correlation for rapid root-cause localization in 5G messaging platforms. Moreira et al. [16] stress-test a Kubernetes-based 5G core with Chaos Mesh and characterize per-NF resource sensitivity on a testbed closely resembling ours, but treat each modality in isolation rather than quantifying cross-modal coupling. However, these approaches operate on proprietary production data, focus on a subset of NFs, or do not systematically study how cross-modal coupling changes as a function of fault category.

2.4 Knowledge Gap

Two gaps emerge from the above. First, existing multi-modal frameworks encode the cross-modal relationship *implicitly* inside learned models and rely on post-hoc interpreters to explain it; the coupling between modalities is never measured as a direct, inspectable quantity. Second, multi-modal studies target generic microservice benchmarks rather than 5G Core NF architectures, while 5G-specific approaches rely on single-modality or manual multi-source analysis over proprietary data. Consequently, no prior work systematically characterizes how cross-modal coupling *changes as a function of fault type* across all three modalities in a containerized 5G Core under controlled fault injection spanning a broad operational and security fault taxonomy. We address both gaps by treating the fault-induced cross-modal coupling change ($\Delta|\rho|$) as an explicit, interpretable feature, and characterizing it across 22 operational and 6 security fault scenarios on an open-source Open5GS testbed. Spearman rank correlation is the specific coupling estimator we adopt; our contribution is the systematic per-fault cross-modal characterization itself, not the choice of correlation coefficient.

3 Methodology and Experimental Setup

3.1 Testbed

Experiments are conducted on a containerized Open5GS 5G Core deployment running on Kubernetes; Open5GS is a

widely used open-source 5G core whose standalone performance has been independently characterized [9]. The deployment monitors ten components: the nine core Network Functions AMF, SMF, UPF, NRF, SCP, AUSF, UDM, UDR, and PCF, plus the MongoDB subscriber datastore. The deployment also runs NSSF, SEPP, and BSF, but these are idle in a single-operator, non-roaming, single-slice setup and are excluded from the analysis. Observability is provided by three co-deployed stacks (Figure 1). Prometheus collects metrics at 15-second scrape intervals in two complementary sub-layers: container-level *infrastructure* metrics (CPU usage and throttling, memory working set, network I/O, and UE round-trip time) capturing how hard each NF works, and Open5GS *application* KPIs (per-NF registration, session, authentication, and PFCP/N4 counters) capturing whether its 5G procedures succeed (full inventory in Appendix D). Grafana Loki aggregates structured application logs from all NF pods (error events and log volume); and Grafana Beyla collects distributed traces via eBPF instrumentation, capturing per-NF request latency (P99) as reported to Jaeger. The testbed, experiment automation, and analysis scripts are publicly available.[1] The testbed runs on two independent hardware platforms (two operational trials on the first, five on a second running Kubernetes under WSL2); seven operational and three security trials in total assess cross-platform reproducibility.

3.2 Fault Injection Scenarios

Operational faults. We inject 22 controlled fault scenarios spanning four categories, summarized in Table 1 and individually listed in Table 2. *Infrastructure* faults (6 scenarios) apply CPU and memory stress to individual NFs. *Pod crash* faults (5 scenarios, including the NRF cascade) terminate NF pods, with the NRF cascade causing downstream service-discovery failures across all NFs that depend on the registry. *Network* faults (7 scenarios) inject artificial delay, packet loss, or complete interface partitions on specific inter-NF links. *PFCP* faults (4 scenarios) disrupt the PFCP session management protocol between SMF and UPF through message flooding, deletion storms, and modification drops; PFCP is a documented denial-of-service surface in the 5G core [1]. Each fault runs for a 300-second active window, preceded by a 600-second clean pre-fault baseline and followed by a 300-second post-fault recovery window. The post-fault window is collected but not analysed here.

Security faults. We additionally inject 6 security fault scenarios across three categories: a NAS Registration Storm, four SBI HTTP/2 Flood attacks targeting the AMF, NRF, SCP, and SMF respectively, and an AUSF Authentication Exhaustion attack. These scenarios exercise documented attack surfaces of open-source 5G core network functions [3]; national risk assessments of the 5G core likewise identify denial-of-service against the HTTP/2 service-based interface as a primary attack vector [6]. Each security scenario is run for three independent trials using the same 300-second fault / 600-second pre-fault / 300-second post-fault structure.

[1] https://github.com/DavidGhergut/cse3000-5g-core-fault-experiments

Table 1: Operational fault categories and instance counts. An instance is one execution of one fault scenario in one trial; each of the 22 scenarios is repeated across 7 independent trials, yielding 154 instances in total.

Category	Scenarios	Instances (scenarios $\times$ 7 trials)
Infrastructure	6	42
Pod crash	5	35
Network	7	49
PFCP	4	28
Total (operational)	**22**	**154**
NAS Registration Storm	1	3
SBI HTTP/2 Flood	4	12
Authentication Exhaustion	1	3
Total (security)	**6**	**18**

3.3 Cross-Modal Correlation Pipeline

Each experiment produces time-series data from all three modality stacks for the pre-fault and fault windows. Both windows are divided into 30 non-overlapping 10-second bins of equal length. Within each bin, metrics are averaged, log events are counted, and trace spans are summarised as the 99th-percentile request latency per NF; we therefore use traces as a per-NF latency signal rather than as a request-level call graph. Each bin series is z-score normalised using the pre-fault baseline mean and standard deviation, anchoring all signals to a common scale regardless of modality. Using an equal-length 30-bin window for both phases ensures the pre-fault and fault correlations are estimated from series of the same length, which avoids the bias that arises when correlations computed over series of different lengths are compared.

For each pair of signals (s_a, s_b) spanning different modalities (metrics-traces, metrics-logs, or traces-logs), we compute Spearman rank correlation ρ at lag zero over both the pre-fault and fault windows. Spearman rank correlation is used in place of Pearson to avoid sensitivity to outlier bins and non-linear scaling relationships common in telemetry data. The fault-induced coupling change is then:

$$\Delta|\rho| = |\rho_{\text{during}}| - |\rho_{\text{pre}}| \tag{1}$$

A positive $\Delta|\rho|$ indicates that the two modalities became more tightly coupled during the fault relative to the pre-fault baseline. Signal pairs with fewer than six valid bins in either window are excluded.

We use this lag-zero $\Delta|\rho|$ as the characterization feature throughout; lagged, propagation-aware analysis is left to future work.

3.4 Random Forest Classifier

To quantify whether $\Delta|\rho|$ features are diagnostic of fault category, we train a Random Forest (RF) classifier on per-instance feature vectors, where each instance corresponds to one execution of one fault scenario in one trial. Each instance is represented as a vector whose entries are the $\Delta|\rho|$ value of one cross-modal signal pair, i.e. a metric, log, or trace signal of one NF paired with a different-modality signal of any

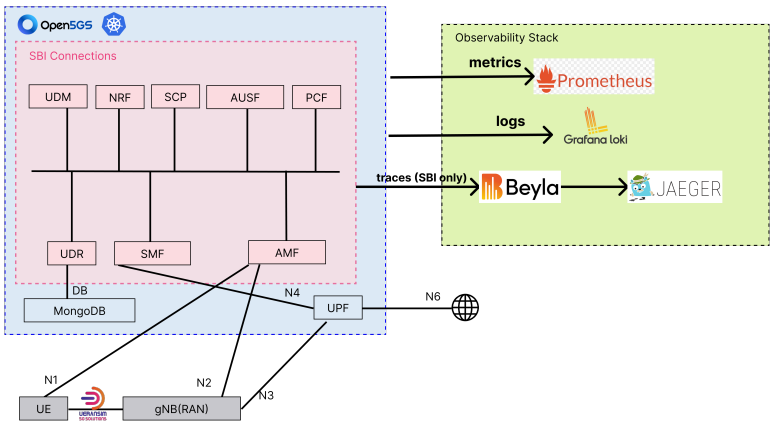

Figure 1: Open5GS 5G Core testbed and observability overlay. The control-plane NFs share the HTTP/2 SBI bus, while the UPF carries the user plane (N3/N4/N6). Prometheus and Loki collect metrics and logs from all pods; Beyla traces only the SBI, so the non-SBI interfaces (N1, N2, N3, N4) are not traced, making faults on them trace-blind.

Table 2: The 22 operational and 6 security fault scenarios, grouped by the category label used for classification. The NRF cascade is triggered by terminating the NRF pod and is therefore grouped with pod crash.

Category	**Fault scenarios**
Operational faults	
Infrastructure	CPU stress (AMF, SCP, AUSF, NRF); memory pressure (UPF, AMF)
Pod crash	Pod kill (AMF, SMF, UDM, MongoDB); NRF cascade (NRF pod kill)
Network	Network delay (gNB-AMF, NRF, SCP); link partition (AMF-SCP); N2 partition (AMF-gNB); packet loss (UPF, UPF infrastructure)
PFCP	Session establishment flood; session deletion storm; modification drop; modification duplication (all SMF-UPF PFCP)
Security faults	
NAS storm	NAS Registration Storm (UERANSIM mass registration)
SBI flood	HTTP/2 flood on the SBI of AMF, NRF, SCP, SMF
Auth. exhaustion	AUSF Authentication Exhaustion

NF. Retaining only pairs that are non-zero in at least 20% of instances (a sparsity filter that removes pairs absent from most faults) yields a **180**-dimensional cross-modal feature vector per operational instance. Correlation-derived quantities are well established as discriminative inputs for classification [7], which motivates feeding $\Delta|\rho|$ directly to a classifier. We use $n = 200$ trees, balanced class weights, and $\sqrt{p}$ features per split. Because the dataset is modest in size (154 operational instances, 18 security instances), we apply both Leave-One-Out Cross-Validation (LOOCV) and a leave-one-trial-out protocol (holding out an entire deployment) to obtain accuracy estimates and to test cross-deployment generalization. All reported accuracies and macro-F1 scores are means over 10 random-seed repetitions of the classifier, with standard deviations where shown.

As a strong baseline for the classification comparison (Table 5), we also build a *raw signal* feature vector that ignores cross-signal coupling entirely: for every monitored metric, log, and trace signal, drawn from the same inventory that feeds the coupling features (Appendix D) so the two models are compared on identical underlying data, we compute four shape statistics over the fault window, mean, standard deviation, maximum absolute value, and linear slope, augmented with span-count, error-rate, and latency-percentile summaries for trace signals, giving a 379-dimensional vector per instance. We report two metrics throughout: overall *accuracy* (Acc., the fraction of instances classified correctly) and *macro-averaged F1*, the unweighted mean of the four per-class F1 scores, where each F1 is the harmonic mean of that class's precision and recall. Because the operational classes are imbalanced (network 49, infrastructure 42, pod crash 35, PFCP 28 instances), macro-F1 weights every fault category equally and is the more conservative of the two.

We position the RF classifier explicitly as an analytical instrument rather than a production fault detector: its role is to test which feature families carry diagnostic value for each fault category. We use the Random Forest for its interpretability rather than for peak accuracy: logistic regression and gradient boosting in fact score higher (Appendix A), confirming that the conclusions do not depend on the classifier. To assess the discriminative value of cross-modal $\Delta|\rho|$, we compare against within-modality ablation baselines that use only same-modality pairwise $\Delta|\rho|$ features (metrics-metrics, logs-logs, or traces-traces pairs respectively), and against the raw signal features defined above.

4 Results

4.1 Modality Blindspots: Which Faults Each Modality Can See

Existing anomaly detection for the 5G core operates within a *single* modality: event logs [20], per-NF KPI time series [2], or network traffic [17]. We ask which faults a single modality cannot see, and why. The answer lies in the 5G core's interface architecture, not in a generic claim that the modalities complement each other: *which* modality goes blind depends on the fault, so covering every fault type requires more than one modality. We report each pattern under a strict protocol: it must replicate across all independent-deployment trials (seven for operational faults, three for security) and be distinguishable from the other faults taken as a baseline.

Tracing follows the SBI, so non-SBI faults are trace-blind. The 5G core is a heterogeneous-interface system: control-plane NFs communicate over the HTTP/2 Service-Based Interface (SBI), while the AMF reaches the radio access network over N2/NGAP and the SMF controls the UPF over N4/PFCP, neither of which is HTTP. Beyla's eBPF tracer instruments only the SBI, so faults on the other interfaces produce no *direct* trace anomaly *by construction*: a structural blindspot, not a dataset artefact. Two fault families fall on these non-SBI interfaces, and the blindspot shapes each differently: one loses the trace signal entirely, while the other has it fall silent in a way that, set against a surge in error logs, becomes diagnostic in itself.

Interface partitions (N2/NGAP): a relational signature. The N2 partition disconnects the AMF from the gNB at the RAN–core boundary; because N2 does not transit the SBI, it produces no direct trace anomaly, and what characterizes it is a *coupling* signature. The two interface-partition faults (AMF–SCP and the N2 AMF–gNB partition) drive the traces–logs coupling sharply negative as the partitioned NF's SBI span volume collapses while its error logs surge: as a group, the partition faults decouple more strongly than the non-partition faults in every one of the seven trials (mean difference -0.19; sign test $p = 0.016$; Figure 2). The size of this difference varies across trials (-0.09 to -0.33), but its direction is always the same; we therefore rely on the fact that the partition faults consistently decouple more than the rest, not on the exact size of the gap. The discriminative information is in the *relationship*: from logs alone one sees errors, from traces alone a volume drop, but it is the direction of the coupling change, the two signals pulling apart rather than rising together, that distinguishes a connectivity fault from a load fault. A single-modality monitor sees a symptom; the cross-modal view names the fault class.

PFCP session faults (N4): required coverage. PFCP faults between the SMF and UPF run over N4 (UDP), again off the SBI. They manifest as strong metrics–logs coupling (PFCP protocol-error logs coinciding with UPF resource changes) and near-zero metrics–traces coupling, because the protocol never reaches the tracer (Figure 3, PFCP row). They are therefore characterizable only by the metrics–logs pair: here multi-modal investigation is not an enhancement but a requirement for covering the user-plane control interface.

Security attacks shift the blindspot to logs. The blindspot is not always the trace modality. The NRF SBI flood is a *valid-request* attack: it floods the NRF with well-formed service-discovery queries that the NRF answers normally, producing *zero* error logs across all three trials, against hundreds to tens of thousands for every other attack (Table 8). A flat log signal cannot couple with any other modality, so the NRF flood is characterizable only through metrics (NRF resource exhaustion) and traces (SBI span-volume explosion), and is invisible to the log-based detection common in 5G-core practice [20]. The authentication-exhaustion and NAS registration-storm attacks, by contrast, generate explicit rejection logs and are log-visible. This log blindspot is the security-side mirror of the trace blindspot seen for operational interface partitions: which modality goes blind is attack-specific, so multi-modal coverage is required here too.

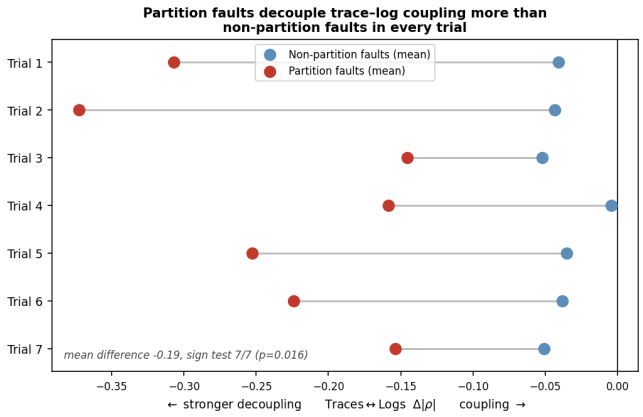

Figure 2: Per-trial traces–logs coupling change $\Delta|\rho|$: mean of the interface-partition faults (red) versus mean of all non-partition faults (blue), for each of the seven trials. A negative value means the trace and log signals decoupled during the fault. The partition group decouples more strongly in every trial (mean difference -0.19, sign test $p = 0.016$), as the partitioned NF's spans collapse while its logs persist.

4.2 Cross-Modal Coupling Signatures by Fault Category

Stepping back from individual faults, the mean $\Delta|\rho|$ per fault category gives each category a distinct cross-modal signature. Because the metrics modality spans two sub-layers, *infrastructure* metrics (container resource usage) and *application* metrics (Open5GS protocol KPIs), we resolve the metrics-logs and metrics-traces channels into these sub-layers (Figure 3), averaged across all NF pairs and seven operational trials. This separation is what the Open5GS metrics add to the characterization: it exposes *whether* a fault couples to logs through resource pressure or through procedure failure, a distinction the resource metrics alone cannot make.

The application sub-layer carries a procedure-failure coupling that resource metrics miss. For *pod crash* faults the application-metrics-logs coupling is the single strongest cell in the matrix ($\Delta|\rho| = +0.134$), well above the infrastructure-metrics-logs coupling for the same faults ($+0.085$). A pod crash is therefore, in coupling terms, primarily a *procedure* failure rather than a resource-pressure event, and only the ap-

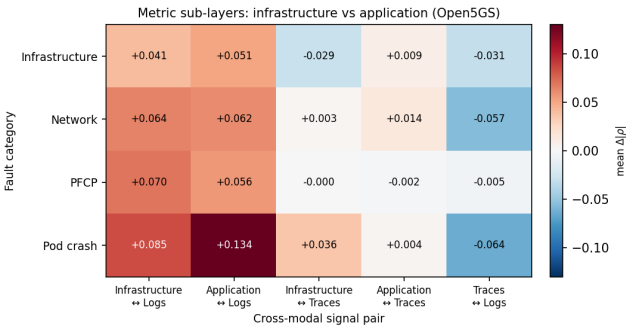

Figure 3: Mean $\Delta|\rho|$ per fault category, with metrics split into infrastructure (container-resource) and application (Open5GS KPI) sub-layers (7 trials; NRF cascade folded into pod crash). Application-logs coupling is strongest for pod crash ($+0.134$), above infrastructure-logs ($+0.085$); network and PFCP couple comparably across both sub-layers. Both sub-layers couple near-zero to traces, so the application KPIs do not close the trace blindspot.

plication sub-layer (the Open5GS KPIs) renders it as such.

For load and protocol faults the two sub-layers couple comparably. *Network* faults (infrastructure-logs $+0.064$, application-logs $+0.062$) and *PFCP* faults (infrastructure-logs $+0.070$, application-logs $+0.056$) elevate both sub-layers to a similar degree: the error logs rise alongside both the resource metrics and the application counters, so neither sub-layer dominates. *Infrastructure* faults are the most balanced (infrastructure-logs $+0.041$, application-logs $+0.051$) but weakest overall, diluted because stress is applied to one NF at a time and averaged across all ten.

The application sub-layer does not rescue the trace blindspot. Application-metrics-traces coupling is near-zero for every category ($\leq +0.014$), no stronger than the infrastructure-traces channel: adding procedure-level KPIs does not recover the structural trace gap of Section 4.1, because the missing information lives on non-SBI interfaces, not in any metric sub-layer. Traces-logs coupling remains negative across all categories, most strongly for pod crash (-0.064) and network (-0.057), reflecting the interface-partition decoupling averaged into those categories.

4.3 Temporal Alignment of Heterogeneous Signals

Binning all three modality streams into 10-second intervals effectively resolves the sampling-rate mismatch: Prometheus metrics (15-second scrape) produce approximately two samples per bin on average, while log and trace events are aggregated as counts and P99 latency respectively within each bin. Z-score normalisation against the pre-fault baseline (the matching 300-second, 30-bin window) removes per-NF and per-signal scale differences, enabling direct Spearman rank correlation across modality pairs with very different absolute magnitudes (e.g., CPU fractions vs. millisecond latency values). The 10-second width balances two competing requirements: enough bins per window (30 over the 300-second fault window) for stable rank-correlation estimates, and fine enough resolution to capture event-driven log and trace bursts. Larger bins would reduce the sample count available to each correlation estimate, while smaller bins would leave many trace and log bins empty. To confirm this choice is not an artefact, we re-ran the full pipeline at bin widths from 5 to 30 seconds: classification accuracy is essentially unchanged, and the per-category coupling signatures retain the sign and relative ordering of Figure 3 (pod crash strongest, infrastructure weakest), with absolute $\Delta|\rho|$ magnitudes scaling with bin width as expected from the changing bin count.

4.4 Classification as an Analytical Instrument

Classifying fault type from raw signals. We classify the four operational fault categories using raw per-signal shape features (the mean, standard deviation, max-abs, and slope of each z-scored signal). A Random Forest reaches **85.5% accuracy** and 0.86 macro-F1 under leave-one-trial-out (LOTO), which holds out an entire deployment and is therefore the honest cross-deployment estimate, across 154 instances and 7 trials, far above the 32% majority-class baseline. The result is not specific to the Random Forest: under LOTO, logistic regression and gradient boosting score even higher (90.9% and 92.9%). The explicit $\Delta|\rho|$ coupling features classify substantially less accurately (56% LOOCV, 61% LOTO); the full coupling-feature comparison and ablation are reported in Appendix A, the security-fault classification in Appendix B, and a data-sufficiency learning curve in Appendix C, since the role of the coupling features is characterization (Section 4.1) rather than classification.

Which modalities carry the classification signal. Table 3 decomposes the raw-feature classifier by modality subset. Metrics alone reach 84.3% (LOTO), already close to the full model, while logs (60.7%) and traces (74.7%) are individually far weaker. Combining modalities yields at most a marginal gain: adding traces to metrics improves accuracy by 1.9 percentage points (84.3% to 86.2%), whereas adding logs yields no improvement: metrics+logs (83.5%) does not exceed metrics alone, and the all-modality result (85.5%) does not exceed metrics+traces. The diagnostic signal for these operational faults is therefore carried almost entirely by metrics, with traces contributing marginally and logs redundantly; multi-modal fusion does not meaningfully raise classification accuracy. This is consistent with the injected faults showing up mainly in the metrics (Section 6) and reinforces that the value of the cross-modal view lies in characterization (Section 4.1) rather than classification.

What cross-modal coupling adds beyond classification. That raw per-signal features out-classify $\Delta|\rho|$ is unsurprising: raw deviations directly capture how much each signal changes, while $\Delta|\rho|$ captures something more indirect, how the relationship between two signals changes. The value of $\Delta|\rho|$ lies elsewhere. The findings in Section 4.1 show that cross-modal $\Delta|\rho|$ reveals patterns in how the signals relate that looking at each signal on its own cannot: the reproducible trace-log decoupling under interface partitions (Figure 2) is a property of the *relationship* between two modalities, which no single signal shows and which appears only when their coupling is measured. Conversely, a fault that simply changes how large the signals get, such as the system-wide spike from the NRF cascade, is already captured by the raw per-signal features, which is exactly why $\Delta|\rho|$ does not win on classification accuracy. The security faults are a lim-

Table 3: Operational raw-feature modality ablation (Random Forest, n=200; 154 instances across 7 trials, 4 categories). Accuracy and macro-F1 by modality subset under leave-one-instance-out (LOOCV) and leave-one-trial-out (LOTO). Logs add nothing on top of metrics (metrics+logs $\approx$ metrics) and traces add only $+1.9$ pp; metrics+traces is the strongest subset (86.2%). Means over 10 random seeds. Majority-class baseline 31.8%.

		LOOCV		LOTO	
Modality subset	**#feat**	Acc.	F1	Acc.	F1
Majority baseline	—	31.8	0.12	31.8	0.12
Metrics	248	83.4	0.85	84.3	0.86
Logs	60	57.5	0.57	60.7	0.60
Traces	71	72.7	0.72	74.7	0.73
Metrics + Logs	308	82.9	0.84	83.5	0.85
Metrics + Traces	319	85.4	0.87	**86.2**	**0.87**
Logs + Traces	131	72.1	0.71	73.6	0.71
All modalities	379	85.3	0.86	85.5	0.86

iting case: both raw and coupling features separate the three attack classes at near-ceiling accuracy (Appendix B), so there too the cross-modal contribution is coverage and characterization, the modality blindspots of Section 4.1, rather than classification accuracy.

SHAP and InterSHAP emphasize different cross-modal pairs. Figure 4 compares two ways of asking which cross-modal pairs a model relies on. Both agree on what matters *least*: metrics–traces ranks bottom in each, a further sign of the SBI-only trace blindspot. They differ at the top because they measure different things. SHAP on the $\Delta|\rho|$ model identifies which coupling change best separates the fault classes, ranking logs–traces highest, the trace-log decoupling under interface partitions (Section 4.1). SHAP interaction values on the raw model ("InterSHAP") [14] identify which raw signals the classifier combines, ranking metrics–logs highest, because metric and log values both rise sharply during faults while trace latency stays flat. That the two point to different pairs is expected: one scores an engineered coupling feature, the other a joint use of raw signals. We read only the rough ordering; with a small dataset and near-flat raw interactions, the comparison is illustrative.

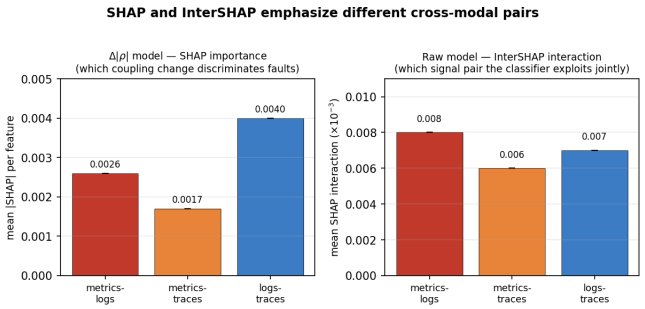

Figure 4: Two SHAP-based views of cross-modal importance; both rank metrics–traces lowest (the SBI-only trace blindspot). **Left:** SHAP on the $\Delta|\rho|$ model ranks logs–traces highest (the partition decoupling). **Right:** SHAP interaction (InterSHAP) on the raw model ranks metrics–logs highest, though the raw interactions are near-flat. The two measure different things, so they need not agree; illustrative given the dataset size.

5 Responsible Research

Experimental environment and ethical considerations. All fault injection experiments are conducted on a dedicated testbed running on privately owned hardware (two independent hardware platforms). No production network infrastructure or end-user traffic is involved at any point. The Open5GS core and UERANSIM simulator operate in a fully isolated Kubernetes cluster with no external SIM credentials or live spectrum usage. No personal data is collected, processed, or stored: the telemetry streams (Prometheus metrics, Loki logs, Jaeger traces) contain only container-level resource measurements, internal NF application logs, and synthetic RPC latency values generated by the simulated UE. Fault injection itself (CPU stress, memory pressure, packet manipulation, pod termination, and traffic flooding) is applied only within this isolated environment and carries no risk of disruption to third-party systems or infrastructure.

Reproducibility. The methodology is designed to be reproducible by any researcher with access to an Open5GS Kubernetes deployment. The cluster setup, fault injection, experiment automation, and analysis scripts are publicly available.[2] The full telemetry dataset is available as a release of the experiment repository.[3] Each experiment is parameterised through a structured directory layout (fault name, trial number, pre/during/post windows) and a `timeline.json` file recording exact window boundaries, enabling deterministic replay of the analysis from raw telemetry. Reported accuracy figures use leave-one-instance-out (LOOCV) and leave-one-trial-out cross-validation, averaged over 10 runs with different random seeds, so the numbers do not depend on a single seed. To avoid selective reporting, every fault scenario and trial collected is included in the analysis, with no runs excluded.

Seven operational trial runs are used (two on one hardware platform, five via WSL2) to assess cross-platform reproducibility of fault signatures. The use of seven independent trials as training data for the classifier implicitly tests whether cross-modal coupling signatures are consistent enough across hardware platforms to be discriminative; in particular, the leave-one-trial-out protocol (Table 5) holds out an entire deployment, and the preserved accuracy indicates that signatures generalise across platforms rather than overfitting to a single trial. The 56% LOOCV accuracy on the combined 154-instance operational dataset (using $\Delta|\rho|$ features) suggests moderate cross-trial consistency for the four-category classification task.

Use of generative AI. Generative AI (Claude Code, Anthropic) was used as a coding and writing aid; it was not relied on to generate the research ideas, experimental design, or conclusions. Full details of its use are given in Appendix E.

[2] https://github.com/DavidGhergut/cse3000-5g-core-fault-experiments

[3] https://github.com/DavidGhergut/cse3000-5g-core-fault-experiments/releases/tag/dataset-v1

Limitations. The testbed uses a small-scale deployment (ten monitored components, ten simulated UEs via UERANSIM) that does not reflect the scale or traffic diversity of a production 5G Core. The 22 operational and 6 security fault types were selected to cover representative failure modes; faults not included in this study (hardware failures, storage faults, complex multi-fault scenarios) may exhibit cross-modal coupling patterns not captured by the current feature space. The cross-modal coupling signatures are specific to the injected fault types and do not generalise to unseen ones: under leave-one-fault-out (LOFO) validation the model categorises held-out fault types at 23%, below the majority baseline (Appendix A). The trace modality is further limited in two respects: Beyla's eBPF instrumentation captures only HTTP/2 SBI traffic, so UDP-based PFCP/GTP-U (UPF) and N2/NGAP signalling produce no spans; and we use the spans only as a per-NF P99 latency signal rather than a request-level call graph (Open5GS routes inter-NF calls indirectly through the SCP, and Beyla's distributed context propagation was not enabled, so spans are not stitched across NFs). Conclusions about the limited added value of traces are therefore scoped to per-NF SBI latency and should not be read as a general statement about distributed tracing. Most of the faults we inject (CPU and memory stress, pod kills, PFCP and network floods) show up strongly in the metrics. Because of this, the result that metrics alone are enough to classify the faults may partly reflect the kinds of faults we chose. Faults whose main symptom does not appear in the metrics, such as silent data corruption or logic errors that only show up in the logs or traces, could give the other modalities more value. Building a fault set that is more evenly spread across the three modalities would be a good next step.

6 Discussion

Cross-modal correlation as fault characterization, not classification. The classification results (Table 3; full coupling comparison in Appendix A) reveal an important nuance: raw signal features (85-86%) substantially outperform cross-modal $\Delta|\rho|$ (56-61%) for operational fault classification, a gap that holds under both leave-one-instance-out and leave-one-trial-out cross-validation. However, cross-modal $\Delta|\rho|$ separates the security faults perfectly and provides characterization that raw per-signal statistics cannot: relational signatures such as the reproducible trace-log decoupling under interface partitions show up only when the coupling *between* modalities is measured, even for faults that are just a matter of signal size, which per-signal statistics already capture. Our methodology does not compete with production classifier approaches such as those of Gao et al. [8] or Wang et al. [21], which are designed and evaluated for high-accuracy fault localization on large-scale benchmark datasets. We use the RF classifier to answer the question: *which cross-modal signal relationships carry diagnostic value for each fault type?*

When does cross-modal correlation add the most value? Our results suggest a clear pattern: cross-modal $\Delta|\rho|$ adds the most value for faults that reshape the *relationship* between modalities rather than merely the magnitude of individual signals. For localized faults whose effect is a change in signal magnitude (e.g., CPU stress on a single pod, or the NRF cascade's system-wide effect), raw per-signal features already provide discriminative information and cross-modal coupling adds little. For faults that change the relationship between modalities, most clearly the interface partitions, which collapse trace-log coupling as one NF's spans vanish while its logs surge, the cross-modal coupling change is a fundamentally different signal that any single-modality analysis misses entirely.

Security faults: strong signal but confounded accuracy. The 100% LOOCV accuracy on security faults warrants a caveat. The security dataset has 18 instances across 3 categories, with far more cross-modal features than instances, a severely underdetermined problem (features $\gg$ instances). Additionally, all three trials run on the same hardware platform and the fault injection scripts are deterministic, meaning that trial-to-trial variation is minimal. The 100% accuracy likely reflects genuine separability of the security fault classes: the attacks differ markedly in observable behaviour, with error-log volumes ranging from zero for the NRF flood to tens of thousands of lines for the others (Table 8). It is also partially an artefact of the small, low-variance dataset. These results should be treated as preliminary evidence of discriminability rather than a generalizable accuracy claim.

Comparison with related work. Hatami et al. [10] finetune an LLM on pod logs, system events, and RTT measurements from a Kubernetes 5G Core testbed, showing strong classification results. Zhang et al. [22] demonstrate that fusing logs with metrics improves localization precision over single-modality baselines. Our work is complementary: rather than optimising classification accuracy, we provide a lightweight, interpretable methodology for quantifying the degree to which each modality pair changes during different fault types, enabling targeted multi-modal analysis for operators who must reason about which signals to examine first during an incident.

7 Conclusions and Future Work

This paper presented a cross-modal correlation methodology for fault characterization in cloud-native 5G Core networks. By computing Spearman rank correlation between metrics, logs, and trace signals binned into synchronized 10-second intervals and comparing fault-window coupling against a pre-fault baseline ($\Delta|\rho|$), we derived interpretable fault signatures across 22 operational scenarios over seven independent trials, and 6 security scenarios over three, in an Open5GS Kubernetes testbed. Binning resolves the modalities' sampling-rate mismatch, z-score normalization anchors them to a common scale, and Spearman correlation is robust to outlier bins and non-linear telemetry, yielding consistent signatures across both hardware platforms.

Cross-modal analysis reveals coupling signatures and modality blindspots, reproducible across all seven trials, that are invisible to single-modality inspection: metrics-logs strengthens for every fault category (Figure 3), interface partitions reproducibly collapse trace-log coupling as the partitioned NF's spans flatline while its logs surge, and metrics-traces couples weakly throughout, reflecting the SBI-only

scope of the tracer. Cross-modal $\Delta|\rho|$ features outperform all within-modality alternatives for operational classification (56% vs. 34-47%) and separate the security faults perfectly, while raw per-signal features classify operational faults more accurately (85-86% under leave-one-trial-out), confirming that the contribution of cross-modal correlation lies in characterization depth rather than raw classification power.

Future work. Extending the dataset to additional platforms and production-scale traffic would test whether the coupling signatures hold beyond controlled conditions. Modelling how faults propagate across modalities over time, for instance with causal methods such as Granger causality, is a natural next step. Finally, jointly exploiting the $\Delta|\rho|$ and raw representations, and reconstructing the inter-NF call graph from the SBI endpoints already in the span data, could improve both classification and topology-aware trace characterization.

References

[1] George Amponis, Panagiotis Radoglou-Grammatikis, Thomas Lagkas, Wissam Mallouli, Ana Cavalli, Dimitris Klonidis, Evangelos Markakis, and Panagiotis Sarigiannidis. Threatening the 5G core via PFCP DoS attacks: the case of blocking UAV communications. *EURASIP Journal on Wireless Communications and Networking*, 2022.

[2] Yeon-Jea Cho, Hong-Jae Lee, A-Sol Choi, Do-Young Kwak, and Jemin Chung. A framework for AI-based large-scale time-series anomaly detection in 5G core networks. In *2025 16th International Conference on Information and Communication Technology Convergence (ICTC)*, 2025.

[3] Filippo Dolente, Rosario G. Garroppo, and Michele Pagano. A vulnerability assessment of open-source implementations of fifth-generation core network functions. *Future Internet*, 16(1):1, 2024.

[4] Chiming Duan, Yong Yang, Tong Jia, Guiyang Liu, Jinbu Liu, Huxing Zhang, Qi Zhou, Ying Li, and Gang Huang. FAMOS: Fault diagnosis for microservice systems through effective multi-modal data fusion. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 2613–2625. IEEE, 2025.

[5] Mostafa Farshchi, Jean-Guy Schneider, Ingo Weber, and John Grundy. Metric selection and anomaly detection for cloud operations using log and metric correlation analysis. *Journal of Systems and Software*, 137:531–549, 2018.

[6] Federal Office for Information Security (BSI). 5G risk analysis – detailed analysis: 5G core network. Risk analysis, Bundesamt für Sicherheit in der Informationstechnik, 2025. Version 1.0.

[7] Yun Fu, Shuicheng Yan, and Thomas S. Huang. Correlation metric for generalized feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12), 2008.

[8] Fei Gao, Ruyue Xin, Xiaocui Li, and Yaqiang Zhang. Are GNNs actually effective for multimodal fault diagnosis in microservice systems? *arXiv preprint arXiv:2501.02766*, 2025.

[9] Jan Erik Håkegård, Henrik Lundkvist, Ashish Rauniyar, and Peter Morris. Performance evaluation of an open source implementation of a 5G standalone platform. *IEEE Access*, 12, 2024.

[10] Parsa Hatami, Ahmadreza Majlesara, Ali Majlesi, and Babak Hossein Khalaj. Automated fault detection in 5G core networks using large language models. *arXiv preprint arXiv:2512.19697*, 2025.

[11] Xiaofeng Jiang, Peihua Yu, Xuechun Yan, Qiang Zeng, Hua Zhang, Zhaoxia Liang, and Hui Zhou. Fault root cause location based on machine learning in 5G core network. In *Third International Conference on Green Communication, Network, and Internet of Things (CNIoT 2023)*, volume 12814, page 128140B. SPIE, 2023.

[12] Neha Kaushik, Harish Kumar, and Vinay Raj. Proactive fault prediction in microservices applications using trace logs and monitoring metrics. In *2024 International Conference on Progressive Innovations in Intelligent Systems and Data Science (ICPIDS)*. IEEE, 2024.

[13] Yingke Li, Jing Han, Yongqian Sun, Binpeng Shi, and Zican Gong. A root cause analysis framework for microservice systems with multimodal data. *ZTE Communications*, 23(4):110–119, 2025.

[14] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, 2020.

[15] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[16] Rodrigo Moreira, Larissa F. Rodrigues Moreira, and Flávio de Oliveira Silva. Performance evaluation and threat mitigation in large-scale 5G core deployment. *arXiv preprint arXiv:2507.17850*, 2025.

[17] Júnia Maísa Oliveira, Jônatan Almeida, Erik de Britto e Silva, Larissa Ferreira Rodrigues Moreira, Rodrigo Moreira, Flávio O. Silva, Daniel F. Macedo, and José Marcos Nogueira. Anomaly detection employing a 5G core data analytics framework. In *2024 IEEE 13th International Conference on Cloud Networking (CloudNet)*, 2024.

[18] Rui Ren, Yang Wang, Fengrui Liu, Zhenyu Li, and Gaogang Xie. Triple: The interpretable deep learning anomaly detection framework based on trace-metric-log of microservice. In *2023 IEEE/ACM 31st International Symposium on Quality of Service (IWQoS)*. IEEE, 2023.

[19] Ying Tong and Xiang Jia. Intelligent fault root cause localization for 5G message platform. In *2024 6th In-*

ternational Conference on Electronics and Communication, Network and Computer Technology (ECNCT), pages 172–176. IEEE, 2024.

[20] Quang Vinh Tran, Viet Quan Dinh, Ngoc Thanh Do, Quoc Trung Le, and Thu Huong Truong. An efficient lightweight anomaly detection for 5G core network. In *2024 IEEE International Conference on Consumer Electronics (ICCE)*, 2024.

[21] Peipeng Wang, Xiuguo Zhang, Zhiying Cao, and Zihan Chen. MADMM: microservice system anomaly detection via multi-modal data and multi-feature extraction. *Neural Computing and Applications*, 36:15739–15757, 2024.

[22] Qixun Zhang, Tong Jia, Zhonghai Wu, Qingxin Wu, Lichun Jia, Donglei Li, Yuqing Tao, and Yutong Xiao. Fault localization for microservice applications with system logs and monitoring metrics. In *2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pages 149–155. IEEE, 2022.

[23] Zhouruixing Zhu, Cheryl Lee, Xiaoying Tang, and Pinjia He. HeMiRCA: Fine-grained root cause analysis for microservices with heterogeneous data sources. *ACM Transactions on Software Engineering and Methodology*, 2024.

A Coupling-Feature Classification

Table 5 compares the explicit $\Delta|\rho|$ coupling features against raw signal features under both protocols: raw features classify substantially better (85–86%) than cross-modal $\Delta|\rho|$ (56–61%), and accuracy is preserved under leave-one-trial-out, indicating cross-deployment generalization rather than within-trial leakage. The gap is not model-specific (Table 6): logistic regression (90.9%) and gradient boosting (92.9%) score even higher than the Random Forest under leave-one-trial-out, while $\Delta|\rho|$ stays far lower across classifiers. Concatenating the two feature sets (559 dimensions) does not improve on raw alone and slightly degrades it (Combined, Table 5: 84.2% vs. 85.5%), so the coupling change adds no classification signal beyond what the raw magnitudes already capture.

Feature-count robustness. The raw model operates in a $p > n$ regime (379 features, 154 instances). To check it does not simply overfit the high-dimensional space, we apply leakage-free, fold-wise feature selection: within each leave-one-trial-out fold we rank features by Random Forest importance on the training trials *only*, retain the top-ranked subset, retrain, and score the held-out deployment. A small, importance-selected subset of features reaches the same accuracy as the full set, and that subset is mostly metric features, consistent with the metric-dominated modality ablation (Table 3). **Generalization to unseen fault types.** A complementary leave-one-fault-out (LOFO) protocol holds out an entire fault scenario (all seven trials) and classifies it from the other 21. Accuracy falls below the majority baseline for every category (Table 4): the classifier recognises the specific faults it was trained on rather than learning what defines each category, so its role is to characterize the studied faults, not to detect unseen ones.

Table 4: Leave-one-fault-out (LOFO) accuracy: each fault scenario is held out entirely and classified from the others (raw features, RF). Accuracy is below the 32% majority baseline for every category, so the classifier does not generalise to unseen fault types.

Held-out category	**Accuracy (%)**
Infrastructure	24
Network	18
PFCP	29
Pod crash	23
Overall	**23**
Majority baseline	32

Table 5: Classification accuracy and macro-F1 under two cross-validation protocols, for the explicit cross-modal coupling features ($\Delta|\rho|$), raw signal features (mean, std, max-abs, slope of each signal), and their concatenation (Combined). LOOCV holds out a single fault instance; LOTO holds out an entire deployment and is the honest cross-deployment estimate. RF (n=200 trees), 154 operational instances across 7 trials, 4 categories; accuracies are mean $\pm$ std over 10 random seeds (F1 std $\leq$0.02). The majority-class baseline always predicts the most frequent class (network).

	LOOCV		**LOTO**	
Feature set	Acc. (%)	F1	Acc. (%)	F1
Majority-class baseline	32	0.12	32	0.12
Cross-modal $\Delta\|\rho\|$	55.9$\pm$2.2	0.54	61.4$\pm$2.8	0.60
Raw signal features	85.3$\pm$1.8	0.86	85.5$\pm$0.9	0.86
Combined (raw + $\Delta\|\rho\|$)	84.0$\pm$1.6	0.85	84.2$\pm$1.3	0.85

Table 6: Raw-feature operational classification across classifiers, leave-one-trial-out (LOTO, the honest cross-deployment protocol). Logistic regression and gradient boosting match or exceed the Random Forest, so the result does not depend on the choice of classifier. 154 instances across 7 trials, 4 categories.

Classifier	**LOTO accuracy (%)**
Majority-class baseline	32
Random Forest (n=200)	85.5
Logistic regression	90.9
Gradient boosting	92.9

Table 7 breaks the operational $\Delta|\rho|$ classifier down by feature subset. Combining all cross-modal coupling (56% LOOCV) is above every within-modality channel ($\leq$47%) and above two of the three individual cross-modal pairs, and comparable to the

strongest (metrics-logs 54%): the coupling signal is spread across modality pairs rather than concentrated in one. This shows where the coupling signal lies, not how accurately the model classifies; on accuracy the raw features still dominate.

Table 7: Operational $\Delta|\rho|$ modality ablation (Random Forest, n=200; 154 instances, 7 trials, 4 categories). LOOCV and leave-one-trial-out (LOTO) accuracy by coupling-feature subset: within-modality pairs, individual cross-modal pairs, and all cross-modal pairs combined. Means over 10 seeds.

Feature subset	**#feat**	**LOOCV**	**LOTO**
Majority baseline	—	31.8	31.8
Metrics–metrics	30	46.6	52.5
Logs–logs	4	40.5	40.0
Traces–traces	7	34.2	35.5
Metrics–logs	107	54.1	56.9
Metrics–traces	62	38.4	41.8
Logs–traces	11	49.8	51.0
All cross-modal	180	**55.9**	**61.4**

B Security-Fault Classification

On the 3-class security task, cross-modal $\Delta|\rho|$ reaches 100% (Table 9); single-modality $\Delta|\rho|$ appears weaker (metrics 81%, logs 89%, traces 66%), but this weakness reflects the $\Delta|\rho|$ representation, not a real limitation of those modalities: with *raw* features, every single modality already classifies the security faults at near-ceiling accuracy (Table 10). The task is trivially separable (18 instances, hundreds of features, very distinct attack types), so these accuracies are preliminary, and the security contribution is the coverage/blindspot characterization of Section 4.1, not classification.

Table 8: Error-log volume during each security attack (mean over three trials). The NRF SBI flood is a valid-request attack: it produces *zero* error logs, while every other attack produces hundreds to tens of thousands. A flat log signal cannot couple with any modality, so this is the security-side mirror of the operational trace blindspot: the NRF flood is characterizable only through metrics and traces.

Security attack	**Error-log lines (during)**
NAS registration storm	815
SBI flood:AMF	4,793
SBI flood:NRF	**0**
SBI flood:SCP	4,773
SBI flood:SMF	22,960
Authentication exhaustion	13,928

Table 9: Security-fault ablation, $\Delta|\rho|$ coupling features: LOOCV accuracy by feature subset (18 instances, 3 categories, 3 trials; RF n=200, means over 10 seeds). The log-coupling pairs (metrics-logs, logs-traces) separate the three classes perfectly; metrics-traces does not. Not directly comparable to the 4-class operational task. Preliminary (small, low-variance dataset).

Feature subset	**LOOCV accuracy**
Majority-class baseline	67%
Metrics-only $\Delta\|\rho\|$	81%
Logs-only $\Delta\|\rho\|$	89%
Traces-only $\Delta\|\rho\|$	66%
Metrics-Logs $\Delta\|\rho\|$	100%
Metrics-Traces $\Delta\|\rho\|$	78%
Logs-Traces $\Delta\|\rho\|$	100%
All cross-modal $\Delta\|\rho\|$	**100%**

Table 10: Security-fault classification with *raw* per-signal features (Random Forest, n=200; 18 instances, 3 classes, 3 trials). Every single modality already classifies at near-ceiling accuracy, confirming the task is trivially separable; results are preliminary. Majority-class baseline 67%.

Modality subset	**LOOCV**	**LOTO**
Majority baseline	67	67
Metrics	100	100
Logs	100	100
Traces	94	100
All	100	100

C Data Sufficiency: Learning Curve

Figure 5 plots held-out-trial (LOTO) accuracy against the number of training deployments. The raw model is still rising at six training trials (+3.4 pp at the last step), so it is instance-limited rather than at a ceiling, whereas the $\Delta|\rho|$ model is near saturation, flattening well below the raw model. More deployments would raise raw accuracy with diminishing returns; adding more features would not, since the model already has far more features than instances.

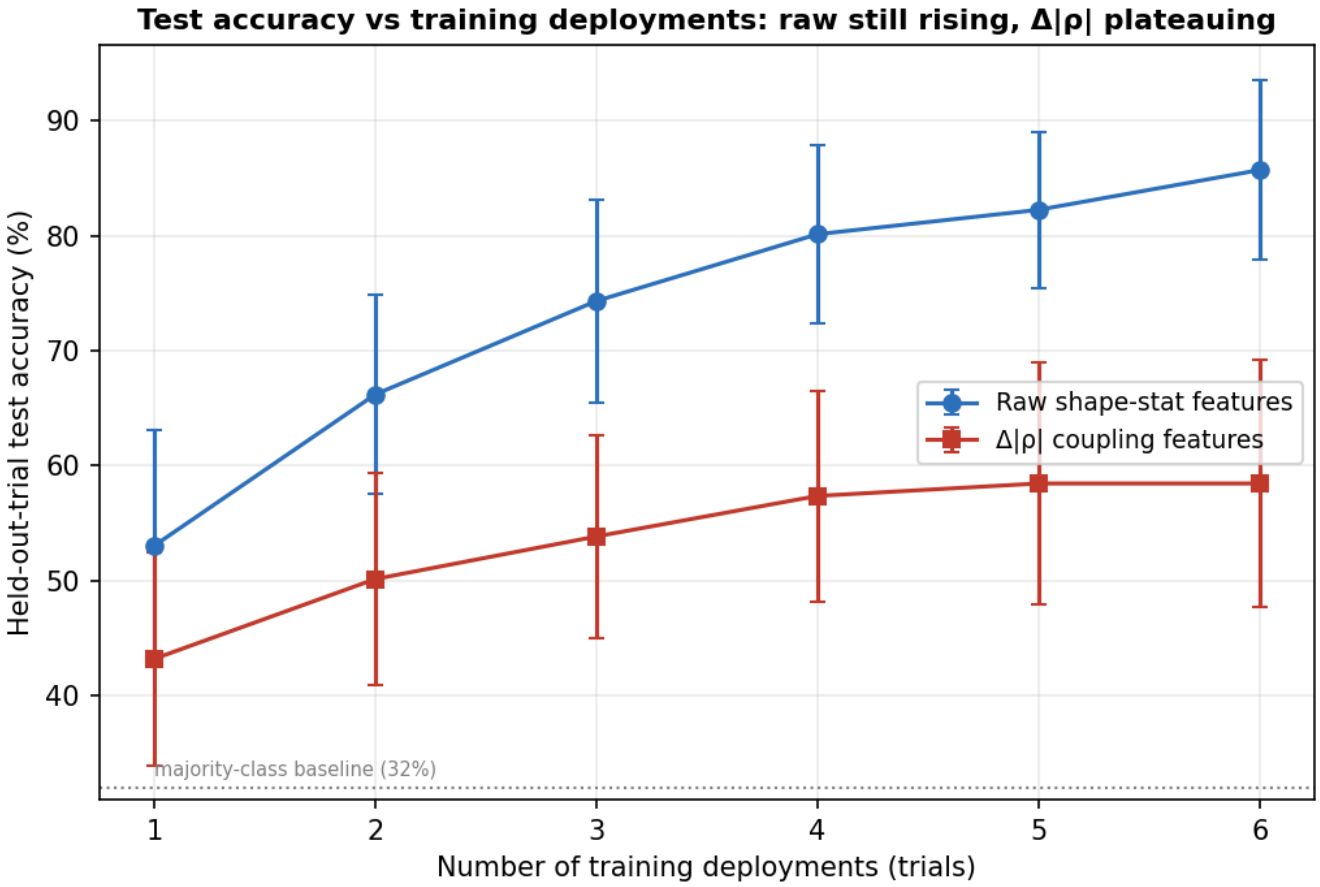

Figure 5: LOTO accuracy versus the number of training deployments. The raw model is still rising at six trials; the $\Delta|\rho|$ model is near saturation. 154 instances, 7 trials.

D Telemetry Signal Inventory

Table 11 lists every telemetry signal used in the correlation study and the classifiers, grouped by modality. Within the metrics modality we distinguish two sub-layers: *infrastructure* metrics (container-level resource usage) and *application* metrics (Open5GS protocol KPIs). The cross-modal $\Delta|\rho|$ features pair signals of *different* modalities, while the raw classifier summarises each signal on its own by four shape statistics (mean, standard deviation, maximum absolute value, and linear slope) over the fault window.

Table 11: Telemetry signals per modality. Infrastructure and application are the two sub-layers of the metrics modality; logs and traces are separate modalities. Signals are collected per network function (NF) unless noted; traces are emitted only for the SBI (service-based interface) request path.

Modality	**Sub-layer**	**Signals (per NF)**
Metrics	Infrastructure	CPU usage, CPU throttling, memory working set, network RX rate, network TX rate, UE round-trip time
	Application	*AMF:* registration requests / successes / failures, authentication failures / rejects, active sessions, registered subscribers, RAN-UE count, gNB count, paging requests. *SMF:* PDU-session requests / successes, session count, N4 session establishments / reports, active bearers, QoS flows, active UEs. *UPF:* N4 session establishments, QoS flows, session count. *PFCP:* active sessions, active peers. *GTP:* node failures
Logs	—	Error-event count, log volume
Traces	—	Request latency (mean, P50, P95, P99), span count, error rate

E Use of Generative AI

The generative-AI tool used in this project was Claude Code (Anthropic).

Code and analysis. Claude Code was used to develop and debug the experiment-automation, classification, correlation, and plotting scripts in the public repository, prompted with specific, isolated instructions, for example: "add a leave-one-trial-out split to this classifier and report mean accuracy over ten seeds," or "reposition this matplotlib legend so it does not overlap the bars." The resulting scripts were run and checked by the author, and all reported numbers were reproduced from the dataset.

Writing. Claude Code was used to improve the clarity and structure of the manuscript, simplify dense phrasing, and assist with LaTeX formatting. The author reviewed and edited the suggested text and did not adopt output that was not understood or verified.

The research questions, experimental design, interpretation of results, and conclusions are the author's own.