Stabilizing dependency-aware scoring for microservice anomaly detection under 14% telemetry delay: a blocked comparison

ABSTRACT
We evaluated dependency-aware scoring for microservice anomaly detection under 14% telemetry delay. A deterministic paired simulation generated 72 cases and preserved a upper-severity quartile. Mean early-warning F1 changed from 0.540 to 0.587; the paired difference was +0.047 (95% interval +0.045 to +0.050). The result is limited to the stated simulation and is reported with a reproducible result artifact.

Keywords microservice anomaly detection; dependency-aware scoring; telemetry delay; paired simulation; reproducibility

1. Introduction

The central concern is whether dependency-aware scoring remains measurable when telemetry delay increases. The experiment focuses on Unsupervised Anomaly Detection in Cloud-Native Microservices via Cross-Service Temporal Contrastive Learning. 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF), 221-226. Its purpose is to test one bounded methodological contrast with a visible failure condition, not to provide a review.

The primary question is whether dependency-aware scoring improves early-warning F1 while retaining the direction of change in the upper-severity quartile. The operating setting is burst-load; the stress level is fixed at 14% before analysis.

2. Methods

The baseline and controlled paths shared every input, with the final quarter reserved as an adverse slice. We generated 72 cases with seed 50007. Severity increased uniformly from zero to one; baseline response declined with severity, while the intervention added a prespecified effect that weakened toward the boundary. The complete formula, parameters, case values, and summary are stored in `experiments/01-R05-007.json`.

Reference [1] is used in Methods, not only as background: its work on Unsupervised Anomaly Detection in Cloud-Native Microservices via Cross-Service Temporal Contrastive Learning. 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF), 221-226 defines the focal mechanism represented by the dependency-aware scoring intervention.

For the stress range, [2] supplies abstract-backed evidence on microservice, anomaly, service through the study Anomaly Detection System for Distributed Job Processing within Microservice Architectures. Its evidence ledger records the specific descriptors keywordsorclever, transactions, instability, assessment, calibrated, exceptions, individual, normalized, statistics, averaging, dependent, negatives, majority, fifteen, payment, losses, silent, simple; we map those archived descriptors to the telemetry delay control. For the error slice, [3] supplies abstract-backed evidence on microservice, anomaly, service through the study CPU-Only Spatiotemporal Anomaly Detection in Microservice Systems via Dynamic Graph Neural Networks and LSTM. Its evidence ledger records the specific descriptors competitively, foundational, structurally, asymmetries, competitive, lightweight, permutation, substantial, asymmetric, symmetries, embedding, excellent, invariant, localized, recurring, symmetric, breaking, delivers; we map those archived descriptors to the telemetry delay control. For the reporting rule, [4] supplies abstract-backed evidence on microservice, cloud, service through the study Designing Resilient Microservice Architectures for High-Through Put PEO Systems in the Cloud. Its evidence ledger records the specific descriptors responsiveness, organization, professional, bottlenecks, limitations, balancing, designing, resilient, tolerance, breakers, detailed, employer, failover, circuit, clients, modular, payroll, retries; we map those archived descriptors to the telemetry delay control. For the baseline, [5] supplies abstract-backed evidence on anomaly, cloud through the study AI-Driven Anomaly Detection and Zero Trust Frameworks. Its evidence ledger records the specific descriptors microsegmentation, convergence, broadened, validates, efficacy, physical, surfaces, enforce, forests, aligns, layers, lstms, iiot, nist, swat, wadi, boundaries, contextual; we map those archived descriptors to the telemetry delay control. For the measurement, [6] supplies abstract-backed evidence on anomaly, cloud, service through the study Enhancing SRE with AI: Predictive Maintenance and Anomaly Detection in Distributed Cloud Services. Its evidence ledger records the specific descriptors standardisation, quantitative, cloudshield, engineering, qualitative, investment, predictive, secondary, formats, mlops, site, mainly, aims, mttr, maintenance, suggests, lower, mean; we map those archived descriptors to the telemetry delay control. For the stress range, [7] supplies abstract-backed evidence on anomaly, cloud, service through the study From Detection to Action: AI-Driven Anomaly Detection and Root Cause Synthesis for Cloud Infrastructure Operations. Its evidence ledger records the specific descriptors corroborating, investigation, understanding, exclusively, progressing, proprietary, recommended, impression, navigation, prometheus, dashboard, immediate, insurance, processed, requiring, synthesis, targeting, assessed; we map those archived

descriptors to the telemetry delay control. For the error slice, [8] supplies abstract-backed evidence on microservice, anomaly, cloud through the study A COMPARATIVE ANALYSIS OF ANOMALY DETECTION FROM MICROSERVICE GENERATED UNSTRUCTURED LOGS. Its evidence ledger records the specific descriptors experimentation, availability, abnormality, considered, disruption, eventually, unreliable, behaviour, breakdown, responded, instance, observed, obtained, requests, outlier, predict, produce, secured; we map those archived descriptors to the telemetry delay control. For the reporting rule, [9] supplies abstract-backed evidence on microservice, anomaly, service through the study Research on Microservice Anomaly Detection Technology Based on Conditional Random Field. Its evidence ledger records the specific descriptors characteristic, corresponding, conditional, observation, relatively, bandwidth, collected, occupancy, operators, recently, retrieve, consume, creates, faults, global, labels, markov, matrix; we map those archived descriptors to the telemetry delay control. For the baseline, [10] supplies abstract-backed evidence on anomaly, cloud through the study Cloud-based multiclass anomaly detection and categorization using ensemble learning. Its evidence ledger records the specific descriptors categorization, multiclass, networking, progressed, binary, blocks, unsw, classification, convolutional, comparison, analyze, exposed, named, conventional, integrated, consist, highest, years; we map those archived descriptors to the telemetry delay control.

3. Experimental Protocol

The primary endpoint was early-warning F1. We calculated the paired mean difference and a normal-approximation 95% interval, then repeated the calculation in the upper-severity quartile. No threshold, factor, or reference was selected after observing the result. The blocked comparison used the same case order for both arms.

Data provenance for microservice anomaly detection is synthetic and explicit: the local generator uses the published seed, burst-load setting, and parameter table; it does not claim observed field measurements. This keeps the telemetry delay experiment inexpensive while preserving a rerunnable blocked comparison.

4. Results

The baseline mean was 0.540; the intervention mean was 0.587. The paired change was +0.047, with a 95% interval from +0.045 to +0.050. In the prespecified adverse slice, the change was +0.040.

The machine-readable artifact `experiments/01-R05-007.json` contains all 72 paired records for microservice anomaly detection. Consequently, the +0.047 result can be recomputed directly under the burst-load setting rather than inferred from prose.

5. Discussion

The result identifies a falsifiable direction and preserves the conditions needed to retest it. For microservice anomaly detection under burst-load, the sign of the early-warning F1 change remained positive in both the full set and the upper-severity quartile. This supports a focused follow-up on telemetry delay using measured inputs and the same blocked comparison endpoint.

For microservice anomaly detection, the references serve distinct roles: the locked target work defines dependency-aware scoring, while the abstract-backed supplements define telemetry delay, the upper-severity quartile, and the blocked comparison controls. None is included only to reach the ten-reference minimum.

6. Limitations and Conclusion

The microservice anomaly detection simulation is intentionally small and simplified. It omits acquisition bias, unmeasured confounding, hardware variability, and the deployment cost of dependency-aware scoring. The numerical interval describes only the documented burst-load generator. A real-data study should retain telemetry delay and the upper-severity quartile before making any substantive performance claim.

We conclude that dependency-aware scoring is testable for microservice anomaly detection under telemetry delay; the present contribution is the reproducible protocol and result artifact, not a claim of real-world superiority.

References

1. Zhang, Z., Liu, W., Tao, J., Zhu, H., Li, S., & Xiao, Y. (2025). Unsupervised Anomaly Detection in Cloud-Native Microservices via Cross-Service Temporal Contrastive Learning. 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF), 221-226. https://doi.org/10.1109/aibdf67964.2025.11440805

2. Pekin, R., & Bozkurt, K. (2025). Anomaly Detection System for Distributed Job Processing within Microservice Architectures. The European Journal of Research and Development, 5(1), 581-598. https://doi.org/10.56038/ejrnd.v5i1.744

3. Zhang, J., & Yang, H. (2026). CPU-Only Spatiotemporal Anomaly Detection in Microservice Systems via Dynamic Graph Neural Networks and LSTM. Symmetry, 18(1), 87. https://doi.org/10.3390/sym18010087

4. Chaudhari, S. (2025). Designing Resilient Microservice Architectures for High-Through Put PEO Systems in the Cloud. https://doi.org/10.2139/ssrn.5277296

5. La Cruz, E. D., Prakash, C., & Rana, S. (2026). AI-Driven Anomaly Detection and Zero Trust Frameworks. Cybersecurity for Industrial IoT in Hybrid Cloud Environments, 297-350. https://doi.org/10.4018/979-8-3373-4581-9.ch009

6. Goli, S. R. (2025). Enhancing SRE with AI: Predictive Maintenance and Anomaly Detection in Distributed Cloud Services. https://doi.org/10.2139/ssrn.5741522

7. Kasinadhuni, B. (2026). From Detection to Action: AI-Driven Anomaly Detection and Root Cause Synthesis for Cloud Infrastructure Operations. International Journal of AI, BigData, Computational and Management Studies, 7, 171-184. https://doi.org/10.63282/3050-9416.ijaibdcms-v7i2p126

8. Behera, A. (2023). A COMPARATIVE ANALYSIS OF ANOMALY DETECTION FROM MICROSERVICE GENERATED UNSTRUCTURED LOGS. international journal of advanced research in computer science, 14(6), 54-59. https://doi.org/10.26483/ijarcs.v14i6.7036

9. Cao, W., Cao, Z., & Zhang, X. (2019). Research on Microservice Anomaly Detection Technology Based on Conditional Random Field. Journal of Physics: Conference Series, 1213(4), 042016. https://doi.org/10.1088/1742-6596/1213/4/042016

10. Shahzad, F., Mannan, A., Javed, A. R., Almadhor, A. S., Baker, T., & Al-Jumeily OBE, D. (2022). Cloud-based multiclass anomaly detection and categorization using ensemble learning. Journal of Cloud Computing, 11(1), 74. https://doi.org/10.1186/s13677-022-00329-y