Tuning dependency-aware scoring for microservice anomaly detection under 16% telemetry delay: a blocked comparison

ABSTRACT
We evaluated dependency-aware scoring for microservice anomaly detection under 16% telemetry delay. A deterministic paired simulation generated 48 cases and preserved a median slice. Mean early-warning F1 changed from 0.478 to 0.532; the paired difference was +0.054 (95% interval +0.051 to +0.056). 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

Reproducible stress tests can expose weaknesses in microservice anomaly detection before expensive field collection. 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 median slice. The operating setting is mixed-load; the stress level is fixed at 16% before analysis.

2. Methods

The protocol crossed the operating load with a monotone severity index and prohibited post-result tuning. We generated 48 cases with seed 50124. 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-R05-124.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, cloud through the study Anomaly Detection and Failure Root Cause Analysis in Microservice-Based Cloud Applications. Its evidence ledger records the specific descriptors characterized, satisfaction, categorize, diagnosing, discuss, failure, survey, directions, ecosystems, discovery, ensuring, inherent, powered, causes, highlight, causal, face, identifying; 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 Explainable Anomaly Detection in Multimodal Telemetry of Banking Microservices. Its evidence ledger records the specific descriptors auditability, institutions, systemically, constitutes, multimodal, regulatory, concludes, necessary, regulated, additive, agnostic, currency, officers, operator, readable, banking, central, engines; 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 Transforming Legacy .NET Architectures into Scalable Cloud-Enabled Systems via Controlled Microservice Pattern Adoption. Its evidence ledger records the specific descriptors sustainability, transformation, destabilizing, inconsistency, independently, modernization, organizations, restructuring, transactional, academically, traceability, transforming, uncontrolled, incremental, initiatives, refactoring, replacement, synthesized; we map those archived descriptors to the telemetry delay control. For the baseline, [5] supplies abstract-backed evidence on microservice, cloud, service through the study Pattern learning for scheduling microservice workflow to cloud containers. Its evidence ledger records the specific descriptors simplification, lightweighted, substructures, incorporated, diversified, precedence, scheduling, structures, candidate, tardiness, coverage, deadline, devised, sorting, urgency, phase, personalized, challenging; we map those archived descriptors to the telemetry delay control. For the measurement, [6] supplies abstract-backed evidence on anomaly, cloud through the study Gaussian Distribution-Based Machine Learning Scheme for Anomaly Detection in Healthcare Sensor Cloud. Its evidence ledger records the specific descriptors respectively, organizing, intruders, reasons, schemes, scheme, stored, vector, harsh, body, fire, huge, improvements, activities, healthcare, throughput, gaussian, patients; we map those archived descriptors to the telemetry delay control. For the stress range, [7] supplies abstract-backed evidence on microservice, anomaly, service through the study SRdetector: Sequence Reconstruction Method for Microservice Anomaly Detection. Its evidence ledger records the specific descriptors augmentation, appropriate, forecasting, adjustment, comprising, evaluating, srdetector, timestamps, intervals, predicted, resulting, selecting, timestamp, original, strongly,

becomes, serving, weights; we map those archived descriptors to the telemetry delay control. For the error slice, [8] 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, [9] supplies abstract-backed evidence on anomaly, cloud, service through the study Cloud Network Anomaly Detection using Federated Learning and Explainable AI. Its evidence ledger records the specific descriptors synchronization, decentralized, centralized, trustworthy, detections, ransomware, understand, analysts, exposing, optimize, amounts, insider, targets, prime, explainability, evaluations, validate, denial; we map those archived descriptors to the telemetry delay control. For the baseline, [10] supplies abstract-backed evidence on microservice, anomaly, cloud through the study Unified Graph Modeling of Multi-Source Monitoring Data with Spatiotemporal Attention for Microservice Anomaly Detection. Its evidence ledger records the specific descriptors adaptively, modalities, formulate, encoded, thereby, edges, tight, msds, discriminative, spatiotemporal, outperforming, correlations, implications, attributes, baselines, dimension, instances, intricate; 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 median slice. 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, mixed-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 0.478; the intervention mean was 0.532. The paired change was +0.054, with a 95% interval from +0.051 to +0.056. In the prespecified adverse slice, the change was +0.047.

The machine-readable artifact `experiments/01-R05-124.json` contains all 48 paired records for microservice anomaly detection. Consequently, the +0.054 result can be recomputed directly under the mixed-load setting rather than inferred from prose.

5. Discussion

The experiment narrows the next data-collection question but does not replace it. For microservice anomaly detection under mixed-load, the sign of the early-warning F1 change remained positive in both the full set and the median slice. 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 median slice, 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 mixed-load generator. A real-data study should retain telemetry delay and the median slice 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. Diallo, A., & Hassan, N. A. (2026). Anomaly Detection and Failure Root Cause Analysis in Microservice-Based Cloud Applications. International Journal of Artificial Intelligence & Digital Transformation, 9, 29-33. https://doi.org/10.67228/30713315/ijaidt-v9i1p104

3. Kalivoshko, N. (2025). Explainable Anomaly Detection in Multimodal Telemetry of Banking Microservices. Universal Library of Business and Economics, 2(4), 136-144. https://doi.org/10.70315/uloap.ulbec.2025.0204015

4. Boddupally, H. L. (2026). Transforming Legacy .NET Architectures into Scalable Cloud-Enabled Systems via Controlled Microservice Pattern Adoption. https://doi.org/10.2139/ssrn.6265899

5. Li, W., Li, X., & Chen, L. (2023). Pattern learning for scheduling microservice workflow to cloud containers. https://doi.org/10.21203/rs.3.rs-3726089/v1

6. Dwivedi, R. K., Kumar, R., & Buyya, R. (2021). Gaussian Distribution-Based Machine Learning Scheme for Anomaly Detection in Healthcare Sensor Cloud. International Journal of Cloud Applications and Computing, 11(1), 52-72. https://doi.org/10.4018/ijcac.2021010103

7. Ge, H., Ji, X., Peng, F., Chen, R., Xiang, N., Zhang, K., & Wu, W. (2024). SRdetector: Sequence Reconstruction Method for Microservice Anomaly Detection. Electronics, 14(1), 65. https://doi.org/10.3390/electronics14010065

8. 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

9. Idamakanti, P. K. (2025). Cloud Network Anomaly Detection using Federated Learning and Explainable AI. International Journal on Science and Technology, 16(3), 7336. https://doi.org/10.71097/ijsat.v16.i3.7336

10. Chen, X., Shi, G., & Cao, W. (2026). Unified Graph Modeling of Multi-Source Monitoring Data with Spatiotemporal Attention for Microservice Anomaly Detection. https://doi.org/10.2139/ssrn.6504540