PT Jurnal Cendekia Indonesia
Journal of Information Systems and Business Technology
Homepage: https://journal.jci.co.id/jisbt
Vol. 02 No. 04 (2026) Page: 58-68
E-ISSN: 3109-8886

Evidence-Gated Multi-Domain AIOps Copilot on LogLM: Leakage-Controlled Parsing, Template-Level Anomaly Classification, Root-Cause Retrieval, and Confidence-Gated Remediation

[1]Brandon Wright, [2]Ling Yun, [3]Sophia Martinez

[1]Department of Computer Science, University of Maryland, College Park, MD, USA
[2]Department of Computer Science, Rice University, Houston, TX, USA
[3]Department of Computer Science, Duke University, Durham, NC, USA

Email: [1*]b_wright_res@yahoo.com

Abstract

Operational copilots need traceable evidence and leakage-resistant evaluation. We evaluate an evidence-gated AIOps pipeline on the 2,632-record LogLM corpus. The audit found 622 parsing repetitions and aligned Apache overlap, so exact-input and connected-case groups governed every split. The nested parser reached 0.827 exact-template accuracy, versus 0.544 for regex and 0.574 for hybrid nearest-neighbor retrieval; row stratification inflated the latter to 0.908 because 47.7% of test messages had an identical training neighbor. On grouped BGL folds, word–character logistic regression reached AUROC 0.870 ± 0.026, while a fault lexicon led abnormal F1 at 0.687. Nested Platt scaling reduced Brier loss from 0.128 to 0.095 and calibration error from 0.188 to 0.044; learned models produced a 0.058 false-positive rate on negative-only Spirit. Case-grouped Apache retrieval reached overall ROUGE-L 0.261, increasing to 0.364 at 20% coverage. The findings support selective, evidence-linked assistance and human review for sensitive actions.

Keywords: AIOps, LogLM, Log Parsing, Anomaly Detection, Root-Cause Retrieval, Remediation, Selective Prediction, Data Leakage

Abstrak

Copilot operasional memerlukan bukti yang dapat ditelusuri serta evaluasi yang tahan terhadap kebocoran data. Kami mengevaluasi pipeline AIOps berbasis bukti (evidence-gated) menggunakan korpus LogLM yang terdiri atas 2.632 rekaman. Audit menemukan 622 pengulangan parsing dan tumpang tindih pada data Apache yang telah diselaraskan, sehingga kelompok exact-input (masukan identik) dan connected-case (kasus terkait) menjadi dasar dalam setiap pembagian data. Nested parser mencapai akurasi exact-template sebesar 0,827, dibandingkan dengan 0,544 untuk metode regex dan 0,574 untuk hybrid nearest-neighbor retrieval. Stratifikasi berdasarkan baris meningkatkan nilai metode terakhir menjadi 0,908 karena 47,7% pesan uji memiliki tetangga yang identik dalam data pelatihan. Pada pembagian (fold) data BGL yang dikelompokkan, regresi logistik berbasis kata–karakter mencapai AUROC sebesar 0,870 ± 0,026, sedangkan leksikon kesalahan (fault lexicon) menghasilkan skor F1 deteksi anomali sebesar 0,687. Nested Platt scaling menurunkan Brier loss dari 0,128 menjadi 0,095 serta kesalahan kalibrasi dari 0,188 menjadi 0,044. Model yang telah dilatih menghasilkan tingkat positif palsu (false-positive rate) sebesar 0,058 pada dataset Spirit yang hanya berisi data negatif. Metode retrieval Apache yang dikelompokkan berdasarkan kasus mencapai skor ROUGE-L keseluruhan sebesar 0,261 dan meningkat menjadi 0,364 pada cakupan 20%. Temuan ini mendukung pemberian bantuan yang bersifat selektif dan berbasis bukti, serta menegaskan pentingnya tinjauan manusia dalam pengambilan tindakan yang bersifat sensitif.

***Kata Kunci**: AIOps, LogLM, Parsing Log, Deteksi Anomali, Root-Cause Retrieval, Remediasi, Prediksi Selektif, Kebocoran Data.*

A. INTRODUCTION

Logs become operational evidence only after heterogeneous text is converted into stable, traceable representations. LogLM unifies 2,632 records from five tasks and nine domains [1]; Loghub and the parsing benchmark establish the cross-system diversity behind that design [2], [3]. Drain remains a practical template baseline [4]. More recent studies connect OpenStack anomalies to retrieved normal prototypes [5] and HDFS decisions to deterministic failure narratives [6]. Together, these lines of work motivate a copilot that preserves the path from message to template, score, precedent, and response, while preventing repeated messages from crossing the evaluation boundary.

Prior work treats sequence anomaly detection [7], prompt-based log analysis [8], evidence-grounded DevOps retrieval [9], and safe response verification [10] as related

Diterima Redaksi: 01-08-2026 | Selesai Revisi: 06-08-2026 | Diterbitkan Online: 07-08-2026

but separate components. That separation leaves an evaluation gap when one corpus contains duplicated templates, single-reference diagnostic text, and a negative-only domain. This study therefore asks how much row-level splitting inflates parsing and retrieval, how calibration behaves under sparse positives, and how confidence and evidence gates trade coverage for auditable output.

We contribute a leakage-controlled evaluation that (i) groups repeated parsing inputs and connected Apache cases, (ii) compares transparent parsing and anomaly baselines under those groups, (iii) couples case retrieval with similarity-based abstention and provenance checks, and (iv) separates textual remediation screening from autonomous execution. Figure 1 summarizes the flow from instruction and log evidence to a grounded response or human review.

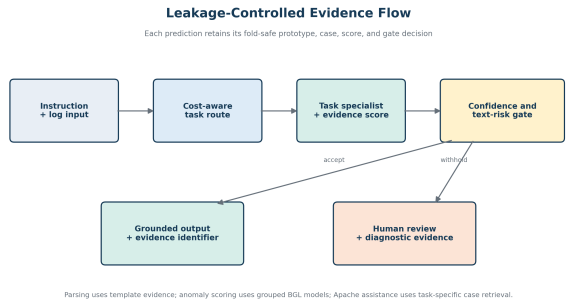

Figure 1 Evidence-Gated Pipeline. Every Branch Emits A Measurable Score And Either A Grounded Response Or Human-Review Outcome

B. METHODS

Corpus Organization And Capability Audit

The JSON array was parsed directly and validated for exactly three nonempty string fields: instruction, input, and output. Its SHA-256 digest was d575da0d529f4f57c76d022867650e3bdf973de1f2637876275fc35687b54d4e. Rows 0–899 contain three 300-row Apache tasks, rows 900–1231 contain BGL and Spirit labels, and rows 1232–2631 contain seven 200-row parsing domains. Table 1 and Figure 2 report the audit.

Table 1 Task-Level Capability Audit

Task	Rows	Unique inputs	Unique outputs	Duplicate excess	Median input words	Median output words
Interpretation	300	263	300	0	21.000	63.000
Root cause	300	266	300	0	23.000	78.000
Remediation	300	271	300	0	23.000	113.000
Anomaly	332	332	2	0	4.000	1.000
Parsing	1,400	778	134	622	6.000	6.000

Duplicate excess counts records beyond the first member of each exact triple group.

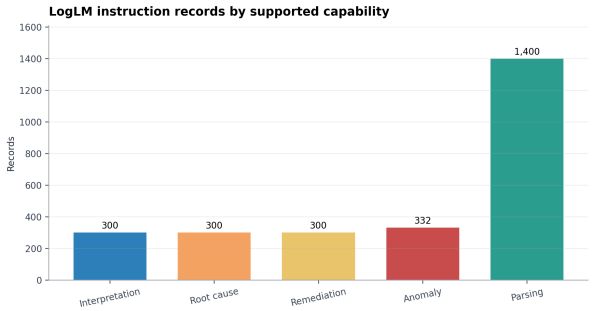

Figure 2 Distribution Of The Five Loglm Tasks.

Leakage-control protocol

All 1,400 parsing rows remained in evaluation, but identical raw inputs received one group identifier and could not cross five-fold GroupKFold boundaries. The 900 Apache rows are positionally aligned across three tasks. A union–find procedure joined each aligned case and then cases sharing an exact input; the resulting 262 connected groups governed retrieval. Row-stratified alternatives were leakage ablations, consistent with warnings that log preprocessing and split choice alter reported performance [11]. Tables 2–4 quantify repetition, class support, and overlap.

Table 2 Parsing-Domain Diversity And Repetition.

Domain	Rows	Unique inputs	Duplicate excess	Gold templates	Largest multiplicity
HDFS	200	200	0	9	1
Hadoop	200	145	55	78	17
Zookeeper	200	42	158	6	49
BGL	200	102	98	13	60
HPC	200	178	22	13	10
Linux	200	31	169	13	48
Proxifier	200	80	120	2	105

Table 3 Anomaly-Label Support By Domain.

Domain	Rows	Normal	Abnormal	Abnormal rate	Inputs with wildcard
BGL	194	161	33	0.170	71
Spirit	138	138	0	0.000	29

Spirit contains no positive label; it is used only for false-positive analysis.

Table 4 Exact-Input Overlap Across Aligned Apache Tasks.

Comparison	Exact-input overlap
Interpretation ∩ Root cause	214
Interpretation ∩ Remediation	209
Root cause ∩ Remediation	226
All three	207
Three-task union	358

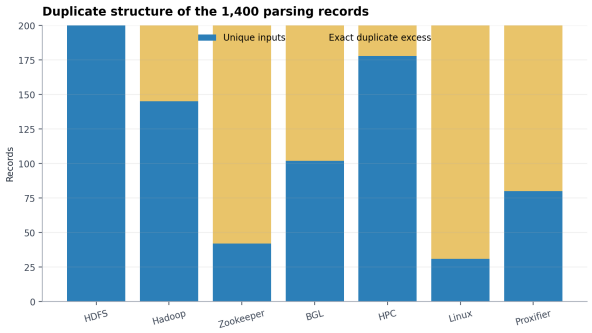

Figure 3 Unique And Repeated Parsing Records By Domain.

Log-language methods span prompt rationales [8], adaptive-cache parsing [12], joint triage and summarization [13], few-shot parsing [14], word-level attribution [15], efficient parsing [16], and deterministic provenance [17]. Together they motivate attaching a supporting record to each operator-facing output.

Seven parsers were compared. Identity returned the input unchanged. Generic regex replaced IP addresses, long hexadecimal values, Hadoop identifiers, and numbers with a wildcard; domain majority returned the most frequent training template. Word and character 1NN used TF–IDF word 1–2 grams or character 3–5 grams; hybrid averaged cosines. The gate accepted regex outputs found in the training template library, otherwise using hybrid retrieval above an inner-grouped threshold or regex fallback. Thresholds spanned 0–1 by 0.01. Drain [4], adaptive-cache LLM parsing [12], and few-shot or efficient LLM parsers [14], [16] provide the context; these controlled baselines required no external LLM call.

Primary exact-template accuracy was supplemented by normalized Levenshtein similarity, wildcard-count accuracy, and pairwise grouping precision/recall/F1. Exact McNemar tests compared the gate with regex and hybrid 1NN on paired out-of-fold decisions.

Template-Level Anomaly Models And Calibration

Anomaly detection spans normal-sequence modeling in DeepLog [7], host system-call windows [18], sequential-quantitative fusion in LogAnomaly [19], autoencoded IoT traffic [20], unstable logs [21], DDoS mitigation [22], LogBERT [23], self-supervised LogBERT-style scoring on a controlled SynHDFS benchmark [24], and CloudTrail attack stages [25].

Language-guided intrusion features [26], split-sensitive evaluations [11], lightweight intrusion models [27], and parser-free NeuralLog [28] further motivate auditable, grouped lexical baselines.

BGL contained 161 normal and 33 abnormal examples; cost-sensitive, simulated-PU, and one-class comparisons under extreme imbalance provide a cross-domain metric and threshold analogue [29]. Each message was lowercased and normalized into a skeleton that mapped wildcards, hexadecimal strings, numeric sequences, and integers to typed markers; identical skeletons formed groups. Five-fold StratifiedGroupKFold used seeds 13-22. Baselines were prevalence, a fault lexicon, and balanced logistic regression with word 1-2 gram, character 3-5 gram, or combined TF-IDF features. This compact design complements sequence and contextual detectors [7], [19], [23] and parser-free text modeling [28]. Logistic regression used C=1, liblinear optimization, and a 0.5 threshold.

Measures were balanced accuracy, macro-F1, abnormal precision/recall/F1, MCC, AUROC, average precision, Brier loss, and ten-bin ECE. Combined LR received nested three-fold sigmoid calibration inside each outer training split. Spirit was scored after fitting all BGL records; its 138 normal labels support only false positives and specificity. Paired Wilcoxon tests compared seeds. Default-risk tiering with uncertainty-driven rejection likewise illustrates that calibrated probabilities and review policy are separate decisions [30].

Apache Retrieval, Confidence, Routing, And Remediation Screen

The Apache subset places maintenance-oriented summarization [31], retrieval-quality prediction [32], retrieval-summary integration in code intelligence [33], task-separated LogEval assessment [34], and FiQA query rewriting and reranking [35] side by side. Cost-aware routing [36], incident cards [37], scientific-search evidence cards [38], and capacity briefs [39] link model choice, evidence, and presentation. Each response therefore retains its task route, held-out case, similarity, and gate outcome.

Root-cause assistance includes prior-resolution retrieval [40], evidence-structured cloud RCA [41], and CPU/network attribution [42]. LLM-generated mitigations [43], CI repair [44], and command-checked copilots [45] show the risk of action-oriented output. With one LogLM target per Apache task, we measure lexical agreement, not causal truth.

The gate draws on RAG [46], budgeted retrieval [47], citation-aware generation [48], evidence-ranked claim verification [49], calibrated abstention [50], trust-calibrated multilingual RAG [51], and Self-RAG critique [52]. Conformal risk control [53], calibrated default prediction [54], and uncertainty-aware fusion [55] separate probability quality from action policy; confidence remains a ranking signal, not a certificate.

Operational assistance creates a security boundary addressed by continual red-teaming [56], tool-approval risk cards [57], evidence-based verification [10], trajectory reliability [58], and utility-preserving refusal [59]. For multi-session operation, confidence-gated writes and time-aware forgetting limit stale memory [60], while federated preference learning with differential privacy illustrates the tenant boundary [61]. Deterministic numerical guardrails provide a further post-generation control [62]. The pipeline executes no tool and routes screened text to human review.

The Apache experiment returned a training response rather than composing unconstrained prose. Fixed-task used the response medoid. Global word 1NN indexed instruction plus input; task-specific word, character, and hybrid 1NN

searched the matching task. Held-out responses received tokenized ROUGE-1/2/L F1. These are lexical measures, not factual or causal judgments. The retrieval-only design follows the evidence discipline of RAG [46], prior-resolution retrieval [40], provenance-aware AIOps RAG [9], [15], retrieval-summary integration [33], and hybrid reranking [35]. Five-thousand paired bootstrap resamples compared models.

Selective coverage retained the top 100%, 80%, 60%, 40%, or 20% by cosine similarity; Spearman correlation tested its relation to ROUGE-L. A grounding audit measured copied non-stopword response tokens and newly introduced numeric, path-like, address-like, or URL-like tokens. The latter indicates exposure, not error.

A supplementary router compared keyword rules, instruction TF-IDF logistic regression, and instruction-plus-log LR under five case-grouped folds; intent masking tested degraded routing. Normalized task costs were 1, 1.5, 4, 5, and 6 plus router overhead, versus 12 for the all-LLM reference. A 263-group remediation stress test compared medoid, retrieval, evidence-linked, and screened responses. The screen removed recursive deletion, unrestricted permission, security-disabling, force-kill, and destructive-database sentences, then appended rollback and validation. Artifact precision measured whether extracted numeric, path, address, and code strings appeared in the query. The screen operationalizes concerns raised by command-safe DevOps copilots [45] and evidence-based response verification [10]; evidence-link rate measured the wrapper, not factual support. These budget units are not power-aware inventory costs [63]. Because rollback language is not rollback evidence, drift-triggered retraining and reversion remain separate monitored lifecycle controls [64]. No command was executed.

C. RESULTS AND DISCUSSION

Audit findings and leakage magnitude

The corpus contained 2,010 unique triples, so 622 of 2,632 records (23.6%) were surplus exact repetitions, all in parsing. HDFS had 200 unique inputs, versus 31 for Linux and 42 for ZooKeeper. Apache had 358 distinct inputs; 207 appeared in all three tasks. Such heterogeneity, long recognized by log benchmarks [3], makes grouping part of the model definition.

Hybrid 1NN parsing rose from 0.574 under exact-input grouping to 0.908 under row stratification; 47.7% of test rows then had an identical training input. Apache hybrid ROUGE-L likewise rose from 0.261 under connected-case grouping to 0.292 under row stratification, with 18.0% identical neighbors. Figure 4 compares protocols.

Table 5.

Table 5 Five-Fold Exact-Input-Grouped Parsing Results.

Model	Exact	Edit sim.	Marker count	Group P	Group R	Group F1
Identity	0.219	0.706	0.219	1.000	0.412	0.583
Generic regex	0.544	0.865	0.678	1.000	0.762	0.865
Domain majority	0.150	0.380	0.403	0.289	0.838	0.430
Word 1NN	0.563	0.675	0.668	0.635	0.935	0.756
Char 1NN	0.552	0.698	0.784	0.620	0.919	0.741
Word+char 1NN	0.574	0.696	0.765	0.651	0.952	0.773
Nested evidence gate	0.827	0.887	0.847	0.735	0.974	0.838

All values are out-of-fold. Edit sim. is normalized Levenshtein similarity.

Table 6.

Table 6 Evidence-Gated Parser By Domain.

Model	Protocol	Exact accuracy	Identical neighbor rate
Domain majority	Exact-input-grouped	0.150	0.000
Domain majority	Row-stratified	0.263	0.000
Word 1NN	Exact-input-grouped	0.563	0.000
Word 1NN	Row-stratified	0.904	0.477
Char 1NN	Exact-input-grouped	0.552	0.000
Char 1NN	Row-stratified	0.891	0.477
Word+char 1NN	Exact-input-grouped	0.574	0.000
Word+char 1NN	Row-stratified	0.908	0.477

Table 7 Parsing Split-Protocol Ablation.

Model	Protocol	Exact accuracy	Identical neighbor rate
Domain majority	Exact-input-grouped	0.150	0.000
Domain majority	Row-stratified	0.263	0.000
Word 1NN	Exact-input-grouped	0.563	0.000
Word 1NN	Row-stratified	0.904	0.477
Char 1NN	Exact-input-grouped	0.552	0.000
Char 1NN	Row-stratified	0.891	0.477
Word+char 1NN	Exact-input-grouped	0.574	0.000
Word+char 1NN	Row-stratified	0.908	0.477

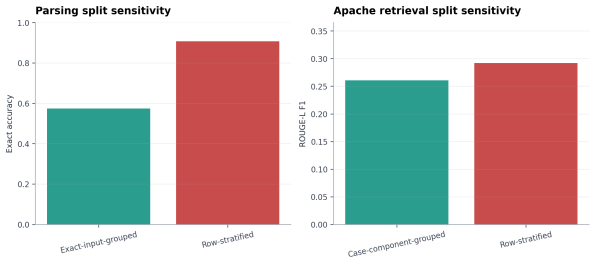

Figure 4 Leakage inflation in parsing and Apache retrieval.

Parsing Accuracy And Structural Trade-Offs

The nested gate attained 0.827 exact accuracy and 0.887 edit similarity (Table 5; Figure 5), versus 0.544 for generic regex and 0.574 for hybrid 1NN. Paired sole-correct counts were 454 versus 58 against regex ($p=3.48\times10^{-77}$) and 362 versus 8 against hybrid 1NN ($p=6.86\times10^{-96}$). The median selected cosine threshold was 0.30; known-regex, retrieval, and fallback branches achieved 1.000, 0.688, and 0.903. This illustrates why cached and retrieval parsers benefit from fallback [12], [16].

Exact gains did not dominate structural metrics. Generic regex grouping F1 was 0.865 versus 0.838 for the gate; the gate paired 0.974 recall with 0.735 precision. Exact accuracy ranged from 1.000 on ZooKeeper to 0.475 on Proxifier, where two gold templates and 105 copies of one input make results sensitive to a few decisions. Both template identity and clustering behavior are therefore necessary.

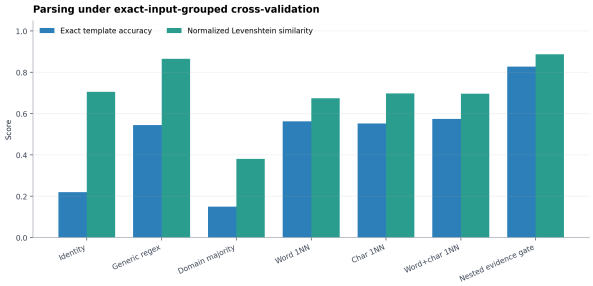

Figure 5 Grouped Parsing Comparison Across Exact Accuracy And Edit Similarity.

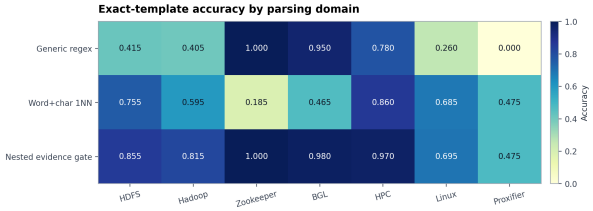

Figure 6 Per-Domain Exact Accuracy For The Principal Parsers.

BGL Anomaly Discrimination And Probability Calibration

Table 8 BGL Repeated Grouped Anomaly Results (Mean ± SD Across Ten Seeds).

Model	Balanced acc.	Macro F1	Abnormal F1
Majority	0.500 ± 0.000	0.454 ± 0.000	0.000 ± 0.000
Lexicon	0.814 ± 0.000	0.811 ± 0.000	0.6
Word LR	0.764 ± 0.046	0.775 ± 0.042	0.6
Char LR	0.749 ± 0.020	0.725 ± 0.017	0.556 ± 0.028
Word+char LR	0.767 ± 0.024	0.761 ± 0.018	0.6

Combined word–character LR led mean AUROC (0.870 ± 0.026) and balanced accuracy (0.767 ± 0.024), but its abnormal F1 (0.605 ± 0.032) trailed word LR (0.621 ± 0.075) and the fault lexicon (0.687). It improved over character LR ($p=0.00195$), matched word LR ($p=0.275$), and trailed the lexicon ($p=0.00195$). Sparse learning improved ranking while a domain lexicon supplied a strong operating point, consistent with the importance of language-selected security features [26].

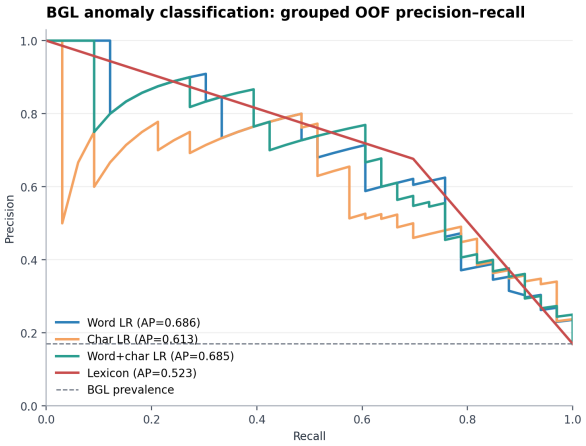

Figure 7 Primary-Seed BGL Precision–Recall Curves.

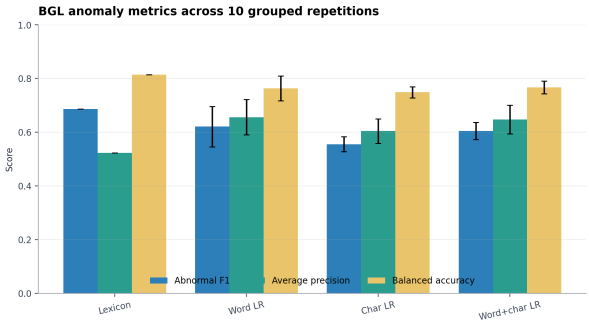

Figure 8 Variation In BGL Grouped Anomaly Metrics Across Ten Seeds.

Table 9 Spirit Negative-Only False-Positive Evaluation.

Model	Rows	Predicted abnormal	False-positive rate	Specificity
Lexicon	138	12	0.087	0.913
Word LR	138	8	0.058	0.942
Char LR	138	8	0.058	0.942
Word+char LR	138	8	0.058	0.942

Table 10 Nested Calibration At A Fixed 0.5 Threshold.

Method	Abnormal P	Abnormal R	Abnormal F1	Brier	ECE	TN/FP/FN/TP
Raw word+char LR	0.579	0.667	0.620	0.128	0.188	145/16/11/22
Nested Platt	0.769	0.303	0.435	0.095	0.044	158/3/23/10

All three learned models flagged 8 of 138 Spirit records, for false-positive rate 0.058 and specificity 0.942; the lexicon flagged 12. Nested Platt calibration reduced Brier loss from 0.128 to 0.095 and ECE from 0.188 to 0.044. At the unchanged 0.5 threshold it raised abnormal precision from 0.579 to 0.769 but lowered recall from 0.667 to 0.303. Thus calibration improved probability reliability, whereas

conformal control addresses alarm policy [65], [53]; alert cost and recall must still set the threshold. This distinction echoes calibrated tiering with explicit rejection [30], but the present threshold was estimated only from BGL.

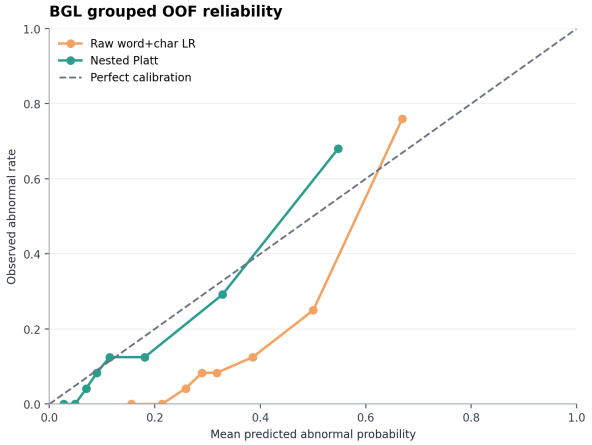

Figure 9 BGL Reliability Curves Before And After Nested Platt Scaling.

Case-Grouped Root-Cause And Remediation Retrieval

Table 11 Strict Case-Grouped Apache Retrieval.

Model	Task	Rows	ROUGE-1	ROUGE-2	ROUGE-L	Exact
Fixed-task	All	900	0.334	0.081	0.202	0.000
Global word 1NN	All	900	0.367	0.114	0.236	0.000
Task word 1NN	All	900	0.394	0.136	0.258	0.000
Task char 1NN	All	900	0.396	0.137	0.261	0.000
Task hybrid 1NN	All	900	0.397	0.138	0.261	0.000
Task hybrid 1NN	Interpretation	300	0.411	0.153	0.283	0.000
Task hybrid 1NN	Root cause	300	0.404	0.152	0.269	0.000
Task hybrid 1NN	Remediation	300	0.376	0.110	0.230	0.000

Task-hybrid retrieval achieved overall ROUGE-1/2/L of 0.397/0.138/0.261, compared with 0.334/0.081/0.202 for the fixed-task response and ROUGE-L 0.236 for global word retrieval. The hybrid ROUGE-L advantage over fixed-task was 0.0584 (95% CI [0.0526, 0.0643]); versus global retrieval it was 0.0250 [0.0203, 0.0299]. Task-character 1NN was equivalent (p=0.728). Task ROUGE-L was 0.283, 0.269, and 0.230, with no exact match. This supports provenance-focused evaluation [17] rather than a causal claim from ROUGE alone.

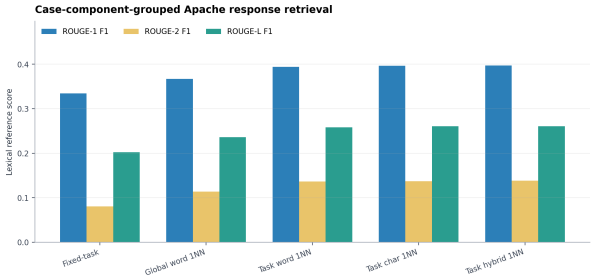

Figure 10 ROUGE Comparison For Strict Case-Grouped Apache Retrieval.

Table 12 Similarity-Gated Apache Coverage.

Task	Coverage	Retained	Minimum similarity	Mean ROUGE-L
All	1.000	900	0.083	0.261
All	0.800	720	0.182	0.275
All	0.600	540	0.280	0.296
All	0.400	360	0.464	0.326
All	0.200	180	0.888	0.364
Root cause	1.000	300	0.084	0.269
Root cause	0.800	240	0.188	0.279
Root cause	0.600	180	0.295	0.295
Root cause	0.400	120	0.493	0.320
Root cause	0.200	60	0.888	0.346
Remediation	1.000	300	0.083	0.230
Remediation	0.800	240	0.174	0.242
Remediation	0.600	180	0.269	0.261
Remediation	0.400	120	0.396	0.288
Remediation	0.200	60	0.836	0.346

Similarity correlated with per-record ROUGE-L (rho=0.621, p=5.40x10^-97). Retaining the most similar 20% raised overall ROUGE-L from 0.261 to 0.364; root-cause and remediation values rose to 0.346 each. The monotonic curve supports a coverage-quality control, as in calibrated unanswerable RAG [50]. Multilingual RAG work likewise separates retrieval coverage, abstention, and interface behavior [51]; none of these scores proves a diagnosis.

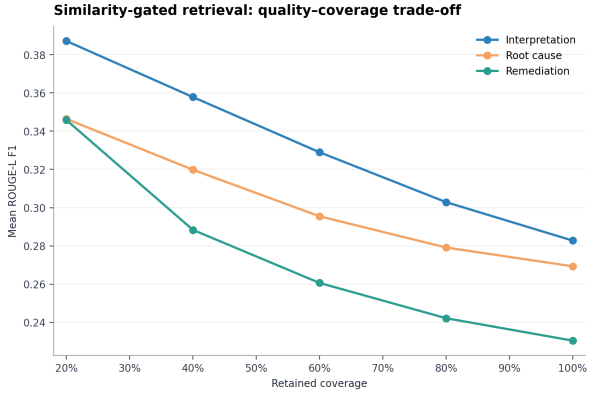

Figure 11 ROUGE-L As Selective Retrieval Coverage Decreases.

Table 13 Lexical Grounding Of Apache Reference Responses.

Task	Rows	Mean token overlap	Median overlap	Any new concrete token	Mean new token count
Interpretation	300	0.216	0.200	0.543	0.897
Root cause	300	0.145	0.121	0.530	0.893
Remediation	300	0.103	0.078	0.820	2.530

Reference text extends beyond the log. Mean non-stopword input overlap fell from 0.216 for interpretation to 0.145 for root cause and 0.103 for remediation. New concrete tokens occurred in 54.3%, 53.0%, and 82.0% of responses, respectively; remediation introduced 2.53 such tokens on average. Commands, versions, paths, or endpoints absent from the input cannot be verified from that string, so remediation requires provenance and review.

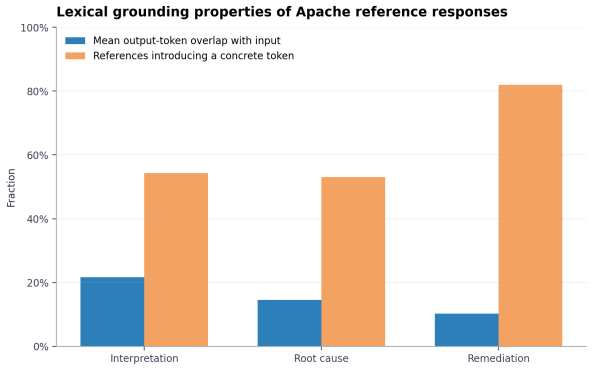

Figure 12 Input-Token Grounding And New Concrete-Token Exposure In Apache References.

Routing And Deterministic Remediation Controls

Table 14 Case-Grouped Task-Routing Quality And Analytic Budget

Router	Condition	Accuracy	Macro F1	Budget units	Budget saving
Keyword	Original intent	0.999	0.998	2.482	0.793
Keyword	Intent-masked	0.153	0.145	4.131	0.656
InstructionLR	Original intent	1.000	1.000	2.631	0.781
InstructionLR	Intent-masked	0.799	0.625	2.827	0.764
Instruction+LogLR	Original intent	0.999	0.998	2.783	0.768
Instruction+LogLR	Intent-masked	0.863	0.775	2.794	0.767

Table 15 Exact-Input-Grouped Remediation Stress Test

Method	N	ROUGE-L	Artifact Precision	Evidence Link	High-Risk Text	Rollback + Validation
Medoid	263	0.156	0.798	0.000	0.000	0.000
Word1NN	263	0.237	0.405	0.000	0.011	0.000
Char1NN	263	0.238	0.392	0.000	0.008	0.000
Evidence Rerank	263	0.229	0.763	0.000	0.019	0.000
Evidence-Linked ECR	263	0.213	1.000	1.000	0.019	0.000
Screened ECR	263	0.204	1.000	1.000	0.000	1.000

Routing was nearly deterministic when the instruction named its task: InstructionLR reached 1.000 macro-F1 at 2.63 budget units, and the keyword route reached 0.998 at 2.48. Intent masking exposed their dependence on explicit wording. Instruction-plus-log LR retained macro-F1 0.775, compared with 0.625 for instruction-only LR and 0.145 for keywords. The analytic savings locate opportunities for specialized routing [36]; they are not wall-clock measurements.

In the remediation stress test, evidence reranking reached ROUGE-L 0.229 with artifact precision 0.763. Adding an evidence-linked log sentence raised artifact precision to 1.000 and evidence-link coverage to 1.000 while reducing ROUGE-L to 0.213. The screen removed all five high-risk pattern classes and added rollback plus validation; ROUGE-L was 0.204. This mirrors the separation between command checking [45] and behavior-level refusal [59]. Numerical and source guardrails [62] would test another failure channel. The result demonstrates text filtering and provenance, not autonomous safety.

Scope And Operational Implications

Three implications follow. Splits must respect message and incident identity. Confidence should control presentation, because the parser fallback and retrieval coverage curve support different decisions. Calibration and command screening also remain separate: one improves score interpretation; the other constrains visible text. Incident cards [37] and scientific-search evidence cards [38] can expose evidence, confidence, and ranking rationale. Persistent incident memory needs gated writes and forgetting [60], with privacy constraints for distributed tenant state [61]. Trajectory reliability requires actual tool errors and recovery events [58]; drift-triggered rollback further requires recorded shift signals and reversion outcomes [64].

The corpus does not support synchronized telemetry fusion, longitudinal degradation prediction, or capacity control. Those extensions require noisy-neighbor residuals [66], calibrated demand forecasts [67], multi-horizon workload traces and risk curves [68], autonomous-cloud agents [69], cross-cloud tests [70], approximately shared-feature analysis under domain shift [71], and hybrid deployment [72]. Conformal envelopes [73] and profit-aware admission

decisions [74] require outcomes beyond log text; GPU-memory prediction [75], SSD health classification from temporal telemetry [76], and cold-start forecasting [77] likewise need time-indexed signals. Power-aware planning [63] and capacity briefs [39] describe the operating environment, not LogLM outcomes. The controlled SynHDFS benchmark [24] also cannot establish real-data transfer. Chronological incidents with aligned metrics, traces, SMART/I/O telemetry, actions, and outcomes are required.

D. CONCLUSIONS

Evaluation of 2,632 LogLM records produced an auditable copilot design. Exact-input grouping exposed 33.36 points of parsing inflation, and connected Apache groups prevented cross-task case reuse. The evidence gate reached 0.827 exact accuracy. On BGL, sparse features ranked anomalies well, a fault lexicon led fixed-threshold F1, and calibration improved Brier loss and ECE while reducing recall at 0.5. Similarity gating improved Apache reference agreement as coverage fell, but remediation text often introduced tokens absent from the log. Deployment should therefore preserve group-safe validation, evidence links, calibrated scores, abstention, and human approval for sensitive actions

E. REFERENCES

[1] Y. Liu et al., "LogLM: From Task-Based to Instruction-Based Automated Log Analysis," in Proc. 2025 IEEE/ACM 47th Int. Conf. Softw. Eng.: Software Engineering in Practice (ICSE-SEIP), 2025, pp. 401-412, doi: 10.1109/ICSE-SEIP66354.2025.00041.

[2] J. Zhu, S. He, P. He, J. Liu, and M. R. Lyu, "Loghub: A Large Collection of System Log Datasets for AI-Driven Log Analytics," in Proc. 34th IEEE Int. Symp. Softw. Rel. Eng. (ISSRE), 2023, pp. 355-366, doi: 10.1109/ISSRE59848.2023.00071.

[3] J. Zhu et al., "Tools and Benchmarks for Automated Log Parsing," in Proc. IEEE/ACM 41st Int. Conf. Softw. Eng.: Software Engineering in Practice (ICSE-SEIP), 2019, pp. 121-130, doi: 10.1109/ICSE-SEIP.2019.00021.

[4] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, "Drain: An Online Log Parsing Approach with Fixed Depth Tree," in Proc. IEEE Int. Conf. Web Services (ICWS), 2017, pp. 33-40, doi: 10.1109/ICWS.2017.13.

[5] Q. Xin, "Explaining OpenStack Failure-Injection Log Anomalies with Retrieved Normal Prototypes," Emerg. Inf. Sci. Technol., vol. 6, no. 2, pp. 125-146, Nov. 2025, doi: 10.18196/eist.v6i2.31232.

[6] X. Sun, Z. S. Zhong, and Q. Wu, "Retrieval-Grounded HDFS Log Anomaly Detection and Deterministic Failure Narrative Generation," J. Comput. Syst. Appl., vol. 3, no. 1, pp. 15-30, Jun. 2026, doi: 10.64229/j6d7fr94.

[7] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2017, pp. 1285-1298, doi: 10.1145/3133956.3134015.

[8] Y. Liu et al., "Interpretable Online Log Analysis Using Large Language Models with Prompt Strategies," in Proc. 32nd IEEE/ACM Int. Conf. Program Comprehension (ICPC), 2024, pp. 35-46, doi: 10.1145/3643916.3644408.

[9] Zhang, H. Rao, and D. Zhao, "Evidence-Grounded RAG for Cloud-Native DevOps: Hallucination-Resistant AIOps Question Answering over Private Operations Documents," J. Adv. Comput. Syst., vol. 4, no. 3, pp. 109-125, Mar. 2024, doi: 10.69987/JACS.2024.40308.

[10] Zheng, B. Zhang, and J. Geibel, "VerifySafe: Toxicity-Safe Agent Responses under Adversarial Prompts with Evidence-Based Self-Verification," J. Adv. Comput. Syst., vol. 4, no. 1, pp. 67-82, Jan. 2024, doi: 10.69987/JACS.2024.40106.

[11] V.-H. Le and H. Zhang, "Log-Based Anomaly Detection with Deep Learning: How Far Are We?" in Proc. IEEE/ACM 44th Int. Conf. Softw. Eng. (ICSE), 2022, pp. 1356-1367, doi: 10.1145/3510003.3510155.

[12] Z. Jiang et al., "LILAC: Log Parsing Using LLMs with Adaptive Parsing Cache," Proc. ACM Softw. Eng., vol. 1, no. FSE, pp. 137-160, 2024, doi: 10.1145/3643733.

[13] Liu, C. Li, and E. Zhang, "OpsLLM for Cloud Incident Triage: Bilingual RAG-Based Root Cause Analysis and Alert Summarization for AI Infrastructure Operations," J. Adv. Comput. Syst., vol. 4, no. 4, pp. 97-111, Apr. 2024, doi: 10.69987/JACS.2024.40408.

[14] Z. Ma, A. R. Chen, D. J. Kim, T.-H. Chen, and S. Wang, "LLMParser: An Exploratory Study on Using Large Language Models for Log Parsing," in Proc. IEEE/ACM 46th Int. Conf. Softw. Eng. (ICSE), 2024, Art. no. 99, pp. 1-13, doi: 10.1145/3597503.3639150.

[15] C. Li, J. Bai, and S. Wang, "Evidence-Chain Reliable RAG: Word-Level Hallucination Detection, Source Attribution, and Provenance Explanation for LLM Applications," J. Adv. Comput. Syst., vol. 4, no. 2, pp. 76-92, Feb. 2024, doi: 10.69987/JACS.2024.40207.

A. Zhong et al., "LogParser-LLM: Advancing Efficient Log Parsing with Large Language Models," in Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining, 2024, pp. 4559-4570, doi: 10.1145/3637528.3671810.

[16] J. Jin, "Evidence-Chain Reliable RAG: Hallucination Detection, Source Attribution, and Deterministic Provenance Explanations," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 520-533, Aug. 2025, doi: 10.51903/jtie.v4i2.535.

[17] Q. Xin, "Host-Based Intrusion Detection with System Call Sequences: Window Localization and Forensic Narratives," AVITEC, vol. 8, no. 2, pp.

325-334, Aug. 2026, doi: 10.28989/avitec.v8i2.3973.

[18] W. Meng et al., "LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs," in Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI), 2019, pp. 4739-4745, doi: 10.24963/ijcai.2019/658.

[19] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT Traffic Classification and Anomaly Detection Method Based on Deep Autoencoders," Appl. Comput. Eng., vol. 69, no. 1, pp. 64-70, 2024, doi: 10.54254/2755-2721/69/20241511.

[20] X. Zhang et al., "Robust Log-Based Anomaly Detection on Unstable Log Data," in Proc. 27th ACM Joint Eur. Softw. Eng. Conf. Symp. Foundations Softw. Eng. (ESEC/FSE), 2019, pp. 807-817, doi: 10.1145/3338906.3338931.

[21] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive Optimization of DDoS Attack Mitigation in Distributed Systems Using Machine Learning," Appl. Comput. Eng., vol. 64, no. 1, pp. 89-94, 2024, doi: 10.54254/2755-2721/64/20241350.

[22] Guo, S. Yuan, and X. Wu, "LogBERT: Log Anomaly Detection via BERT," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9534113.

[23] Q. Xin, "Self-Supervised Log Anomaly Detection with LogBERT-Style Transformers: Full Empirical Evaluation on a Reproducible SynHDFS Benchmark," J. Electr. Eng. Comput. Sci., vol. 11, no. 1, pp. 23-35, Jun. 2026, doi: 10.54732/jeecs.v11i1.3.

[24] J. Bai, S. Chen, D. Zheng, and M.-J. Kuo, "Interpretable Attack-Chain Stage Detection from AWS CloudTrail Event Sequences via Linear Models and HMM Smoothing," Inf. Electr. Electron. Eng., vol. 6, no. 1, pp. 28-43, May 2026, doi: 10.33474/infotron.v6i1.24923.

[25] Y. Li and S. Lu, "Language-Guided Feature Selection for DDoS and Intrusion Detection on CICIDS2017," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 284-305, Apr. 2025, doi: 10.51903/jtie.v4i1.531.

[26] S. Lu and D. Zhou, "TinyLLM-Assisted Intrusion Detection for Real-Time IoT Networks," J. Adv. Comput. Syst., vol. 4, no. 8, pp. 72-87, Aug. 2024, doi: 10.69987/JACS.2024.40809.

[27] V.-H. Le and H. Zhang, "Log-Based Anomaly Detection without Log Parsing," in Proc. 36th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE), 2021, pp. 492-504, doi: 10.1109/ASE51524.2021.9678773.

[28] J. Jin, T. Huang, and S. Lu, "Cost-Sensitive Learning, Simulated PU Learning, and One-Class Autoencoding for Extreme-Imbalance Credit Card Fraud Detection," J. Adv. Comput. Syst., vol. 4, no. 6, pp. 64-73, Jun. 2024, doi: 10.69987/JACS.2024.40605.

[29] Y. Chen, Y. Zhang, D. Chau, and M. Sherman, "Credit Card Default Risk Tiering with Probability Calibration and Uncertainty-Driven Rejection: A Reproducible Study on the UCI Credit Card Clients Dataset," J. Adv. Comput. Syst., vol. 3, no. 4, pp. 31-47, Apr. 2023, doi: 10.69987/JACS.2023.30403.

[30] W. Meng et al., "LogSummary: Unstructured Log Summarization for Software Systems," IEEE Trans. Netw. Service Manag., vol. 20, no. 3, pp. 3803-3815, Sep. 2023, doi: 10.1109/TNSM.2023.3236994.

[31] R. Zhang, Z. Wen, C. Wang, C. Tang, P. Xu, and Y. Jiang, "Quality Analysis and Evaluation Prediction of RAG Retrieval Based on Machine Learning Algorithms," arXiv:2511.19481, 2025, doi: 10.48550/arXiv.2511.19481.

[32] Y. Li, "Findable then Explainable: Retrieval-Summary Integration for Code Intelligence on a Lightweight CodeSearchNet Subset," J. Adv. Comput. Syst., vol. 4, no. 7, pp. 65-82, Jul. 2024, doi: 10.69987/JACS.2024.40706.

[33] T. Cui et al., "LogEval: A Comprehensive Benchmark Suite for LLMs in Log Analysis," Empirical Softw. Eng., vol. 30, Art. no. 173, 2025, doi: 10.1007/s10664-025-10701-6.

[34] S. Meng, J. Chen, and I. Zheng, "LLM-Inspired Offline Reranking for Financial Search: Query Rewriting, Hybrid Retrieval, and Listwise Relevance Ranking on FiQA," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 361-378, Apr. 2026, doi: 10.51903/jtie.v5i1.537.

[35] C. Li, G. Liu, and Z. Zhao, "Cost-Aware LLM-Style Routing for AIOps Log Analysis: Log Parsing, Anomaly Detection, Fault Diagnosis, and Incident Summarization on LogEval Task Files," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 91-103, 2026, doi: 10.51903/jtie.v5i2.538.

[36] J. Nie, G. Liu, C. Li, and T. Zou, "Evidence-Constrained Incident Visualization Cards for Distributed Cloud Logs: A UI/UX Framework for Turning Hadoop, OpenStack, and ZooKeeper Logs into Actionable SRE Design Interfaces," Int. J. Graph. Des., vol. 4, no. 1, pp. 179-185, Apr. 2026, doi: 10.51903/ijgd.v4i1.3703.

[37] J. Jin, "LLM-Style Evidence Cards for Scientific Search Interfaces: A UI/UX Design Framework for Retrieval Transparency, Ranking Trust, and Visual Evidence Hierarchy," Int. J. Graph. Des., vol. 3, no. 2, pp. 397-414, Oct. 2025, doi: 10.51903/ijgd.v3i2.3698.

[38] Liu, S. He, and H. Wong, "LLM-Compatible Visual Brief Cards for AI Infrastructure Capacity Dashboards: A UI/UX Framework for Turning Forecast Risk into Graphic Design Decisions," Int. J. Graph. Des., vol. 3, no. 1, pp. 196-213, May 2025, doi: 10.51903/ijgd.v3i1.3723.

[39] Wang et al., "LogExpert: Log-Based Recommended Resolutions Generation Using Large Language Model," in Proc. IEEE/ACM 46th Int. Conf. Softw. Eng.: New Ideas and Emerging Results (ICSE-

NIER), 2024, pp. 42-46, doi: 10.1145/3639476.3639773.

[40] Y. Chen et al., "Automatic Root Cause Analysis via Large Language Models for Cloud Incidents," in Proc. 19th ACM Eur. Conf. Comput. Syst. (EuroSys), 2024, pp. 674-688, doi: 10.1145/3627703.3629553.

[41] G. Liu, S. He, and I. Liu, "LLM-Augmented Multi-Source Root Cause Attribution for CPU and Network Faults in Microservices," J. Adv. Comput. Syst., vol. 3, no. 6, pp. 39-57, Jun. 2023, doi: 10.69987/JACS.2023.30604.

[42] T. Ahmed, S. Ghosh, C. Bansal, T. Zimmermann, X. Zhang, and S. Rajmohan, "Recommending Root-Cause and Mitigation Steps for Cloud Incidents Using Large Language Models," in Proc. IEEE/ACM 45th Int. Conf. Softw. Eng. (ICSE), 2023, pp. 1737-1749, doi: 10.1109/ICSE48619.2023.00149.

[43] H. Zhang, "LLM-Driven CI Failure Diagnosis and Automated Repair: From GitHub Actions Logs to Patch Recommendation," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 190-214, Apr. 2025, doi: 10.51903/jtie.v4i1.484.

[44] B. Zhang, X. Sun, G. Liu, and B. Zhou, "LLM-Style DevOps Copilot for Cloud-Native Troubleshooting: Retrieval-Augmented Runbook Generation and Command-Safety Evaluation," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 104-118, 2026, doi: 10.51903/jtie.v5i2.534.

[45] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 9459-9474.

[46] W. Su, S. Chen, and C. Zhao, "Budgeted Multi-Hop Retrieval Agent for Compositional Question Answering: A Retrieval-Policy Evaluation on the Official MultiHop-RAG Benchmark," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 649-662, Dec. 2025, doi: 10.51903/jtie.v4i3.543.

[47] T. Gao, H. Yen, J. Yu, and D. Chen, "Enabling Large Language Models to Generate Text with Citations," in Proc. 2023 Conf. Empirical Methods Natural Language Processing (EMNLP), pp. 6465-6488, doi: 10.18653/v1/2023.emnlp-main.398.

[48] W. Su, S. Chen, and E. Qian, "Narrative-Aware Scientific Claim Verification Agent with Evidence Ranking for ClimateCheck," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 327-340, Apr. 2026, doi: 10.51903/jtie.v5i1.549.

[49] Z. S. Zhong, J. Chen, E. Zhong, and X. Sun, "Evidence-Calibrated RAG for Unanswerable Question Answering: Retrieval Coverage, Abstention Calibration, and Hallucination-Proxy Analysis on SQuAD 2.0," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 502-520, Aug. 2025, doi: 10.51903/jtie.v4i2.536.

[50] Y. Chen and H. Xu, "Trust-Calibrated Multilingual RAG for Humanitarian Information Platforms: Empirical Evaluation on OMoS-QA for Migration Information Access," Int. J. Graph. Des., vol. 4, no. 1, pp. 141-164, Apr. 2026, doi: 10.51903/ijgd.v4i1.3552.

A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," in Proc. 12th Int. Conf. Learn. Representations (ICLR), 2024.

[51] N. Angelopoulos, S. Bates, A. Fisch, L. Lei, and T. Schuster, "Conformal Risk Control," in Proc. 12th Int. Conf. Learn. Representations (ICLR), 2024.

[52] Jin, T. Huang, and S. Lu, "A Model-Risk-Friendly Probability of Default Workflow: Calibration, Distribution-Free Uncertainty Quantification, and SHAP Explanations on the UCI Credit Card Default Dataset," J. Adv. Comput. Syst., vol. 4, no. 6, pp. 74-85, Jun. 2024, doi: 10.69987/JACS.2024.40606.

[53] Q. Xin, "Uncertainty-Aware Late Fusion for 3D Perception (Confidence Calibration + Fusion Rule Learning)," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 215-238, Apr. 2025, doi: 10.51903/jtie.v4i1.485.

[54] Zheng, C. Li, and H. Davidson, "Continual Red-Teaming for In-the-Wild Jailbreaks via Online Guardrail Updates and Guardrail Distillation," J. Adv. Comput. Syst., vol. 3, no. 2, pp. 35-49, Feb. 2023, doi: 10.69987/JACS.2023.30203.

[55] W. Su, H. Rao, and E. Ma, "Privacy and Data-Integrity Risk Cards for LLM Agents: A UI/UX Design Framework for Secure Human Oversight under Prompt-Injection Attacks," Int. J. Graph. Des., vol. 4, no. 1, pp. 186-191, Apr. 2026, doi: 10.51903/ijgd.v4i1.3699.

[56] Z. S. Zhong, C. Li, and H. Rao, "Trajectory Reliability Prediction for Generalist AI Agents: Tool-Use Failure Analysis and Success Forecasting on ZClawBench," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 341-360, Apr. 2026, doi: 10.51903/jtie.v5i1.539.

[57] Zheng and C. Li, "Behavior-Level Jailbreak Resistance via Multi-Stage Refusal + Utility Preservation," J. Adv. Comput. Syst., vol. 4, no. 1, pp. 83-99, Jan. 2024, doi: 10.69987/JACS.2024.40107.

[58] X. Sun, Y. Lu, and J. Chen, "Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting," J. Adv. Comput. Syst., vol. 3, no. 8, pp. 9-24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[59] M.-J. Kuo, D. Zheng, and J. Hires, "Federated Topic-Preference Learning for Knowledge-Grounded Chat with Differential Privacy," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 385-401, Aug. 2025, doi: 10.51903/jtie.v4i2.502.

[60] Z. Li, K. Zhang, and A. Wong, "Numerical-Reasoning Guardrails for a Quant Research Assistant: A Compact Reproducible Benchmark Using SEC and FRED Data," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 75-90, Aug. 2026, doi: 10.51903/jtie.v5i2.541.

[61] S. He, J. Nie, and C. Li, "Power-Aware Inventory Planning for AI Infrastructure Using Job-Level Forecasting and LLM Workload Explanations," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 341-359, Apr. 2026, doi: 10.51903/jtie.v5i1.548.

[62] H. Zhang, "DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models," J. Adv. Comput. Syst., vol. 3, no. 7, pp. 24-40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[63] Q. Xin, "Log Anomaly Detection with Conformal Alert Control and Evidence-Grounded Incident Ticket Generation," AVITEC, vol. 8, no. 2, pp. 247-264, Aug. 2026, doi: 10.28989/avitec.v8i2.3974.

[64] Nie and D. Zheng, "Noisy-Neighbor-Aware VM Degradation Risk Modeling with Unsupervised Residual Fusion," J. Adv. Comput. Syst., vol. 4, no. 4, pp. 112-123, Apr. 2024, doi: 10.69987/JACS.2024.40409.

[65] S. He, H. Tu, and I. Liu, "Safe PD Capacity Forecasting with Time-Series Foundation Models and Calibrated Uncertainty for Heterogeneous GPU Clusters," J. Adv. Comput. Syst., vol. 3, no. 4, pp. 48-66, Apr. 2023, doi: 10.69987/JACS.2023.30404.

[66] S. Zhao, J. Bai, and D. Roberson, "Multi-Horizon GPU Demand Forecasting with Workload Semantics and Operational Risk Curves: An Empirical Study on Alibaba Clusterdata GPU Trace," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 544-571, Dec. 2025, doi: 10.51903/jtie.v4i3.498.

[67] Shetty et al., "Building AI Agents for Autonomous Clouds: Challenges and Design Principles," in Proc. 15th ACM Symp. Cloud Computing (SoCC), 2024, pp. 99-110, doi: 10.1145/3698038.3698525.

[68] S. He, X. Chang, and E. Sun, "Cross-Cloud Transfer Learning for AI Training Capacity Forecasting under Workload and Topology Distribution Shift," J. Adv. Comput. Syst., vol. 4, no. 1, pp. 100-120, Jan. 2024, doi: 10.69987/JACS.2024.40108.

[69] Z. S. Zhong, X. Pan, and Q. Lei, "Bridging Domains with Approximately Shared Features," in Proc. 28th Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR, vol. 258, pp. 559-567, 2025.

[70] Q. Xin, "Hybrid Cloud Architecture for Efficient and Cost-Effective Large Language Model Deployment," J. Inf. Syst. Informatics, vol. 7, no. 3, pp. 2182-2195, Sep. 2025, doi: 10.51519/journalisi.v7i3.1170.

[71] S. Chen, S. He, and E. Sun, "Risk-Bounded GPU Resource Oversubscription via Conformal Demand Envelopes in Production AI Clusters," J. Adv. Comput. Syst., vol. 4, no. 5, pp. 119-134, May 2024, doi: 10.69987/JACS.2024.40509.

[72] S. Zhao, Y. Ren, and X. Chang, "Profit-Aware Spot GPU Admission Control with Cost-Sensitive Loss and Evidence-Grounded Policy Memos for AI Workload Supply-Demand Matching," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 45-59, Aug. 2026, doi: 10.51903/jtie.v5i2.545.

[73] Wang, Z. Wen, R. Zhang, P. Xu, and Y. Jiang, "GPU Memory Requirement Prediction for Deep Learning Task Based on Bidirectional Gated Recurrent Unit Optimization Transformer," in Proc. 5th Int. Conf. Artif. Intell., Virtual Reality and Visualization (AIVRV), 2025, doi: 10.1109/AIVRV67401.2025.11350369.

[74] Z. Wen, R. Zhang, and C. Wang, "Optimization of Bi-Directional Gated Loop Cell Based on Multi-Head Attention Mechanism for SSD Health State Classification Model," in Proc. 6th Int. Conf. Electron. Commun. Artif. Intell. (ICECAI), Chengdu, China, 2025, doi: 10.1109/ICECAI66283.2025.11171441.

[75] S. He, C. Li, and H. Rao, "Few-Shot Cold-Start Workload Forecasting for New AI Inference Tenants with Time-Series Foundation Models," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 306-324, Apr. 2025, doi: 10.51903/jtie.v4i1.546.