International Journal of Graphic Design (IJGD)
Volume 4 No. 1 April 2026
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Page 179-185
DOI: https://doi.org/10.51903/ijgd.v4i1.3703
Available online at: https://journal.stekom.ac.id/index.php/ijgd

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

Jiayi Nie[1], Ge Liu*[2], Chengliang Li[3], Tracey Zou[4]
[1]Operation Research, Columbia University, NY, USA
[2]Computer Science, USC, CA, USA
[3]Information Studies, Trine University, VA, USA
[4]Computer Science, UCB, CA, USA
Email Address: gliusde@gmail.com
*Corresponding Author

Abstract. *Site reliability engineers do not need another opaque anomaly detector as much as they need interfaces that turn noisy distributed-system logs into defensible incident decisions. This paper presents a reproducible UI/UX framework for evidence-constrained incident visualization cards. The framework converts raw and structured Loghub-2.0 logs from Hadoop, OpenStack, and ZooKeeper into cards containing an incident headline, affected service and node, anomaly timeline, log-evidence badges, suspected root cause, confidence, recommended next action, severity hierarchy, and operator decision controls. The revised evaluation uses the full available archives for the three selected systems, covering 461,898 log lines and 9,239 50-line analysis windows. A separate label-anchored sanity check uses HDFS_100k with 104,815 structured log lines and 7,940 block sessions, including 313 labeled anomalous sessions. Drain-lite normalization achieved purity of 1.000 on Hadoop, 1.000 on OpenStack, and 1.000 on ZooKeeper against Loghub-2.0 EventId references. The ensemble risk score achieved Precision@5 of 1.000 on the deterministic window-prioritization proxy for all three systems; on HDFS_100k, the same style of evidence-based scoring achieved ROC-AUC 0.730, AUPRC 0.492, and Precision@100 0.990 against external anomaly labels. The interface comparison is reported as a deterministic design proxy rather than a human-subject outcome: the card completed all nine required incident fields and produced a decision-readiness proxy of 0.956, compared with 0.146 for a raw log list and 0.501 for a plain-text summary. The results support the claim that evidence-preserving visual organization can make log analytics outputs more structured and inspectable; they do not establish improved SRE decision performance without a human evaluation.*

Keywords: *AIOps, incident visualization, UI/UX, SRE, evidence-constrained generation.*

INTRODUCTION

Modern cloud incidents unfold across schedulers, storage layers, network endpoints, container managers, API front ends, and coordination services. A single fault investigation can require the operator to read hundreds of low-level messages before deciding whether the failure is a harmless lifecycle event, an overloaded service, a broken peer connection, or a cascading availability risk. Log analytics research has produced strong methods for parsing and anomaly detection, but raw model outputs still leave a presentation problem: the incident must be organized in a form that supports the operator's next decision. The proposed artifact is an incident visualization card.

A card is a compact interface object that compresses a log window into nine decision-bearing elements: incident headline, affected service and node, anomaly timeline, log evidence badge, suspected root cause, confidence level, recommended next action, visual severity

Received: Januari 2026; Revised: February 2026; Accepted: March 2026; Published: April 2026
*Corresponding author, jiayinie8@gmail.com

hierarchy, and operator decision buttons. The card is designed for raw Hadoop, OpenStack, and ZooKeeper logs because these systems represent three common distributed operations contexts: batch job orchestration, cloud infrastructure, and cluster coordination. The empirical basis is Loghub-2.0, which provides public annotated logs for log parsing, and HDFS_100k, which provides an external anomaly-label sanity check at the HDFS block-session level (Xu et al., 2009; Zhu et al., 2023; Jiang et al., 2024). The central research question is: how can log parsing, unsupervised risk scoring, and constrained natural-language summarization be assembled into a UI component that reduces presentation complexity while keeping the evidence visible? The answer is not to hide the logs.

Instead, the interface keeps key log templates and line-range evidence attached to each suspected root cause. This is consistent with explainable analytics arguments that operators need reasons for a model output, not just a score (Ribeiro et al., 2016), and with the visual information-seeking principle of overview first, zoom and filter, and details on demand (Shneiderman, 1996). The paper makes four contributions. First, it defines a card schema for cloud incident visualization that is grounded in SRE workflow fields. Second, it reports a full-data evaluation over Hadoop, OpenStack, and ZooKeeper Loghub-2.0 archives rather than only 2,000-line benchmark slices. Third, it adds an HDFS_100k external-label check so that the risk ranking is not evaluated only against a rule-derived proxy. Fourth, it positions the UI metrics (Chen & Li, 2025) as deterministic design-proxy evidence, not as observed human task-time or error-rate improvements. The card is therefore framed as a design interface rather than as a replacement for a runbook or a validated SRE automation system. In a live incident, the operator normally moves between dashboards, raw log search, and procedural playbooks. The proposed card occupies the space between those layers. It accepts machine-generated risk signals, but it displays them as accountable visual objects with visible evidence and explicit action affordances.

LITERATURE REVIEW

System logs have been used for testing, debugging, diagnosis, and reliability analysis because they record observable runtime state when source code or internal metrics are unavailable (Xu et al., 2009). Loghub consolidated public log datasets for AI-driven log analytics and became a common benchmark for log parsing and anomaly detection (Zhu et al., 2023). Loghub-2.0 extended this resource with larger annotated logs and a more rigorous evaluation of log parsers, showing that parser performance remains uneven when rare events and real production distributions are considered (Jiang et al., 2024). Log parsing is the foundation for automated log analysis because raw log lines contain variable identifiers, IP addresses, task IDs, UUIDs, paths, and counters.

Drain introduced an online fixed-depth tree parser that clusters log messages into templates efficiently and accurately (He et al., 2017). Earlier work on iterative partitioning also showed that log templates can be discovered by grouping messages around invariant token structures (Makanju et al., 2009). Benchmarking studies demonstrated that parser outputs strongly affect downstream mining performance (He et al., 2016; Zhu et al., 2019). This paper uses Loghub-2.0 EventId templates as the reference and evaluates whether stable template evidence can be passed into incident cards.

Log-based anomaly detection has followed both statistical and deep-learning paths. Classical anomaly surveys define outliers as observations that deviate from expected patterns under a reference distribution (Chandola et al., 2009). Isolation Forest isolates anomalies by random partitioning (Liu et al., 2008), and Local Outlier Factor estimates local density deviation (Breunig et al., 2000). DeepLog used LSTM sequence modeling for log key prediction and diagnosis (Du et al., 2017), and LogAnomaly combined sequential and quantitative information to detect abnormal behavior (Meng et al., 2019). In the current work, the risk models remain lightweight because the research focus is the interface layer: whether measured risk signals can become readable incident cards.

Recent LLM work has shifted log analytics from detection to explanation, diagnosis, and summarization. RCACopilot used large language models for cloud incident root cause analysis, aggregating diagnostic information and generating explanatory narratives (Chen et al., 2024). LogEval framed LLM log analysis as a benchmark covering parsing, anomaly detection, fault diagnosis, and summarization (Cui et al., 2024). Other 2024 studies explored LLM-enriched log events and LLM-based parsing for anomaly detection (Fariha et al., 2024; He et al., 2024; Samanta et al., 2024). These studies motivate language-rich incident cards, but the reported generator in this paper is constrained and deterministic. It is better understood as an evidence-constrained generation layer that can be connected to an LLM in future work, not as a tested live LLM system.

The UI/UX basis (Chen & Chan, 2023) comes from visualization and human factors rather than only machine learning. Shneiderman's taxonomy emphasizes overview, filtering, and details on demand (Shneiderman, 1996), and Munzner's nested model requires a match between data abstraction, visual encoding, and user task (Munzner, 2014). Usability inspection methods argue that interfaces should be evaluated against specific tasks and error opportunities (Nielsen, 1994). NASA-TLX remains a standard workload instrument for human-subject evaluation (Hart & Staveland, 1988), but this paper does not report human NASA-TLX because no human

participants were recruited. The deterministic UI metrics are therefore presented only as design-proxy measures.

METHODS

Figure 1 summarizes the experimental pipeline. The revised study starts from full raw and structured logs, attaches service and node fields, builds fixed windows, calculates risk scores, generates evidence-constrained cards, and then evaluates two levels of evidence: a deterministic window-prioritization proxy on the three Loghub-2.0 systems and an external-label sanity check on HDFS_100k.

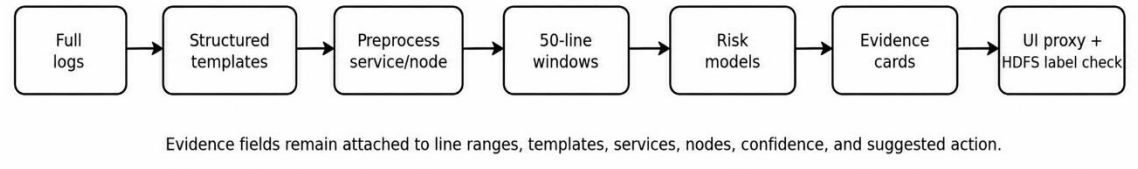

Figure 1. Experimental pipeline for evidence-constrained incident visualization cards.

Data and Preprocessing

The main experiment used three full Loghub-2.0 systems: Hadoop, OpenStack, and ZooKeeper. These archives contain 179,993, 207,632, and 74,273 log lines, respectively. The structured files provide LineId, Content, EventId, and EventTemplate. Raw logs were parsed to recover timestamp, severity, component, service, and node fields. Table 1 reports the resulting dataset characteristics. Preprocessing was identical across systems. Timestamps were parsed from the raw files; comma millisecond separators in Hadoop and ZooKeeper were normalized to dot separators. Severity levels were mapped to weights: INFO, DEBUG, and TRACE received 0; WARN and WARNING received 1; ERROR received 2; and FATAL or CRITICAL received 3. Services and nodes were derived by deterministic rules from components, log-file prefixes, IP addresses, hostnames, and thread names.

Parsing, Risk Scoring, and Card Generation

The parsing experiment measured Drain-lite normalization against the Loghub-2.0 EventId reference. The objective was not to outperform full Drain, but to verify that template evidence could be recovered consistently enough to populate evidence badges. For risk scoring, each system was divided into fixed 50-line windows. Window features combined EventId counts, severity mass, warning and error counts, rare-template hits, risk-keyword hits, and repeated-template concentration.

International Journal of Graphic Design (IJGD)
Volume 4 No. 1 April 2026
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Page 179-185
DOI: https://doi.org/10.51903/ijgd.v4i1.3703
Available online at: https://journal.stekom.ac.id/index.php/ijgd

Table 1. Full dataset characteristics for the three evaluated Loghub-2.0 archives.

Dataset	Rows evaluated	Unique templates	Unique services/components	Unique nodes	Time span	INFO/ DEBUG/ TRACE	WARN/ WARNING	ERROR/ FATAL/ CRITICAL	Top-5 template share	Template entropy (bits)
Hadoop	179993	236	6	2412	2 days 02:31:02.077000	168043	11419	531	0.448	5.101
OpenStack	207632	48	3	2069	2 days 16:23:33.875000	204502	3118	12	0.640	3.756
Zookeeper	74273	89	6	60	26 days 17:44:46.623000	25149	48507	617	0.700	3.783

Table 2. Reproducible method parameters used by the revised experiments.

Stage	Setting	Value
Input data	Main archives	Full Hadoop, OpenStack, and ZooKeeper Loghub-2.0 raw and structured files
External label check	HDFS_100k	Structured HDFS log lines grouped by BlockId and merged with anomaly_label.csv
Log parsing	Drain-lite normalization	UUID/IP/path/block/job/task/container/numeric identifiers replaced with <*>
Windowing	Window size	50 log lines; all rows used; incomplete final window retained
Risk features	Window features	EventId counts, severity counts, service concentration, rare-template hits, and risk-keyword hits
Isolation Forest	Parameters	n_estimators=120 for window ranking; contamination=0.12; random_state=42
Local Outlier Factor	Parameters	5 to 20 neighbors for window ranking; contamination=0.12
Text-distance score	Parameters	TF-IDF centroid distance over window text; max_features=1000
Card generator	Fields	Headline, service/node, timeline, evidence badges, root cause, confidence, action, severity, decision buttons

Received: Januari 2026; Revised: February 2026; Accepted: March 2026; Published: April 2026
*Corresponding author, jiayinie8@gmail.com

International Journal of Graphic Design (IJGD)
Volume 4 No. 1 April 2026
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Page 179-185
DOI: https://doi.org/10.51903/ijgd.v4i1.3703
Available online at: https://journal.stekom.ac.id/index.php/ijgd

The deterministic high-risk proxy used in the three Loghub-2.0 systems marks windows in the top quartile of risk relevance or windows containing ERROR, FATAL, or CRITICAL messages. This proxy is an explicit prioritization target for ranking windows; it is not treated as a production incident label. Four scores were then calculated: template-frequency z-score, Isolation Forest, Local Outlier Factor, and TF-IDF centroid distance. The ensemble score averaged normalized model signals and a normalized risk-relevance prior

The card generator selected the top three ensemble windows per dataset. For each selected window, it populated the nine fields required by the card schema. Evidence badges came from the most frequent EventTemplates in the window. Root-cause phrases and recommended actions came from a fixed evidence library keyed to observed services and phrases such as ResourceManager contact errors, Nova lifecycle states, API request concentration, ZooKeeper GOODBYE messages, and quorum peer exceptions. This rule-locked design keeps the generated cards reproducible and evidence-grounded.

The HDFS_100k label check grouped structured HDFS log lines by BlockId, merged those block sessions with anomaly_label.csv, and evaluated the same family of evidence-based scores against the external Normal/Anomaly labels. This check was added to reduce reliance on the deterministic window proxy alone. The UI comparison used three interface conditions: a raw log list, a plain-text summary, and the visual incident card. The comparison is a deterministic interface proxy calculated from visible fields, evidence localization, action availability, and confidence. It does not represent observed SRE task time, workload, trust, or error rate.

RESULTS

Template Distribution and Parsing Consistency

The full archives show stronger scale and distributional diversity than the original 2,000-line slices. Hadoop contains 236 templates, OpenStack contains 48, and ZooKeeper contains 89. Table 3 lists the top ten templates per dataset, and Figure 2 visualizes the steep template-frequency skew that motivates visual summarization.

Table 4. Parser consistency against the Loghub-2.0 EventId reference.

Dataset	Method	PredictedClusters	Purity	NMI	ARI
Hadoop	Drain-lite normalization	794	1.000	0.971	0.992
OpenStack	Drain-lite normalization	123	1.000	0.945	0.736
Zookeeper	Drain-lite normalization	132	1.000	0.998	1.000

Received: Januari 2026; Revised: February 2026; Accepted: March 2026; Published: April 2026
*Corresponding author, jiayinie8@gmail.com

International Journal of Graphic Design (IJGD)
Volume 4 No. 1 April 2026
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Page 179-185
DOI: https://doi.org/10.51903/ijgd.v4i1.3703
Available online at: https://journal.stekom.ac.id/index.php/ijgd

Table 3. Top template distribution by dataset (top ten per system).

Dataset	Rank	Count	Share	Template
Hadoop	1	45882	0.2549	Progress of TaskAttempt <*> is : <*>
Hadoop	2	18140	0.1008	MapCompletionEvents request from <*>. startIndex <*> maxEvents <*>
Hadoop	3	5795	0.0322	Address change detected. Old: <*> New: <*>
Hadoop	4	5542	0.0308	(EQUATOR) <*> kvi <*>(<*>)
Hadoop	5	5339	0.0297	bufstart = <*>; bufend = <*>; bufvoid = <*>
Hadoop	6	5339	0.0297	Spilling map output
Hadoop	7	5339	0.0297	kvstart = <*>(<*>); kvend = <*>(<*>); length = <*>
Hadoop	8	5300	0.0294	Failed to renew lease for <*> for <*> seconds. Will retry shortly ...
Hadoop	9	5204	0.0289	Finished spill <*>
Hadoop	10	4843	0.0269	<*> TaskAttempt Transitioned from <*> to <*>
OpenStack	1	90475	0.4357	<*> "GET <*> <*>" status: <*> len: <*> time: <*>
OpenStack	2	12043	0.0580	image <*> at (<*>): checking
OpenStack	3	12043	0.0580	image <*> at (<*>): in use: on this node <*> local, <*> on other nodes sharing this instance storage
OpenStack	4	12035	0.0580	Active base files: <*>
OpenStack	5	6194	0.0298	<*> "POST <*> <*>" status: <*> len: <*> time: <*>
OpenStack	6	4132	0.0199	[instance: <*>] VM Resumed (Lifecycle Event)
OpenStack	7	4049	0.0195	[instance: <*>] During sync_power_state the instance has a pending task (spawning). Skip.
OpenStack	8	3059	0.0147	Unknown base file: <*>
OpenStack	9	3059	0.0147	Removable base files: <*>
OpenStack	10	2067	0.0100	[instance: <*>] VM Stopped (Lifecycle Event)
Zookeeper	1	10429	0.1404	Received connection request <*>
Zookeeper	2	10390	0.1399	Send worker leaving thread
Zookeeper	3	10390	0.1399	Interrupting SendWorker
Zookeeper	4	10388	0.1399	Connection broken for id <*>, my id = <*>, error =<*>
Zookeeper	5	10386	0.1398	Interrupted while waiting for message on queue
Zookeeper	6	2584	0.0348	Cannot open channel to <*> at election address <*>
Zookeeper	7	2020	0.0272	Accepted socket connection from <*>
Zookeeper	8	1861	0.0251	Connection request from old client <*>; will be dropped if server is in r-o mode
Zookeeper	9	1856	0.0250	Established session <*> with negotiated timeout <*> for client <*>
Zookeeper	10	1790	0.0241	Processed session termination for sessionid: <*>

Received: Januari 2026; Revised: February 2026; Accepted: March 2026; Published: April 2026
*Corresponding author, jiayinie8@gmail.com

International Journal of Graphic Design (IJGD)
Volume 4 No. 1 April 2026
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Page 179-185
DOI: https://doi.org/10.51903/ijgd.v4i1.3703
Available online at: https://journal.stekom.ac.id/index.php/ijgd

Table 4 shows that the normalized parser output remains highly consistent with the EventId reference on the full data. Drain-lite normalization achieved purity above 0.999 on all three systems. OpenStack has lower ARI than Hadoop and ZooKeeper because many Nova request templates are structurally similar even when their EventId labels differ.

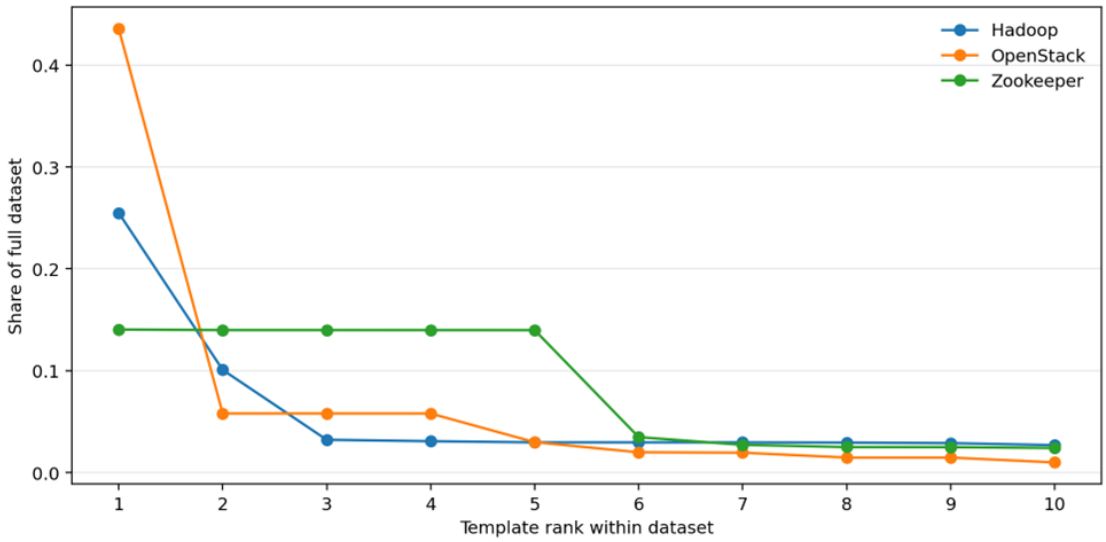

Figure 2. Top template frequencies across the full evaluated logs.

Window-Level Risk Ranking

Table 5 summarizes the fixed-window risk proxy. Hadoop produced 3,600 windows, OpenStack produced 4,153, and ZooKeeper produced 1,486. ZooKeeper has a much higher warning density, while Hadoop and OpenStack concentrate severe messages in smaller bursts. Figures 3 through 5 show the normalized ensemble timeline for each system and mark the highest-scoring windows.

Table 5. Window-level deterministic risk-prioritization proxy summary.

Dataset	Windows	High-risk proxy windows	Mean WARN per window	Mean ERROR/FATAL per window	Mean keyword hits	Mean risk relevance
Hadoop	3600	905	3.17	0.15	4.81	21.66
OpenStack	4153	1044	0.75	0.00	4.88	16.29
Zookeeper	1486	962	32.64	0.42	41.97	165.98

Table 6 compares the scoring models against the deterministic window-prioritization proxy. Because this proxy is built from visible severity, keyword, rare-template, and concentration signals, the results should be read as ranking alignment with the proxy rather than incident-response accuracy. The ensemble achieved Precision@5 of 1.000 on all three systems, while model-level AUPRC varied by dataset and scoring family.

Received: Januari 2026; Revised: February 2026; Accepted: March 2026; Published: April 2026
*Corresponding author, jiayinie8@gmail.com

Figure 3 shows the full-window ensemble risk timeline for Hadoop.

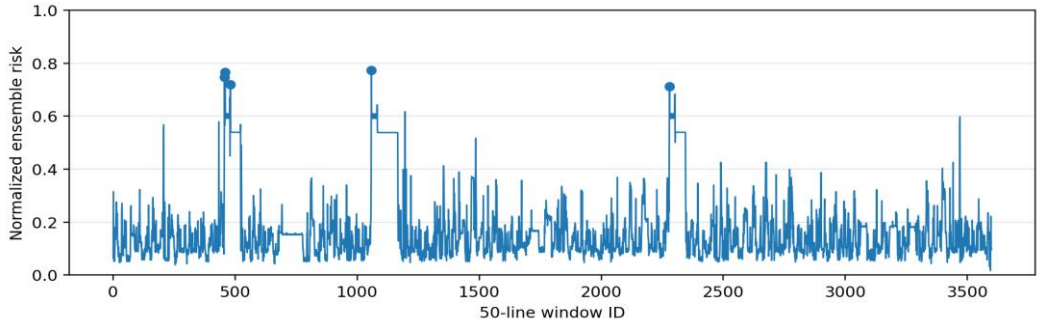

Figure 3. Hadoop anomaly timeline using the normalized ensemble risk score.

Figure 4 shows the full-window ensemble risk timeline for OpenStack.

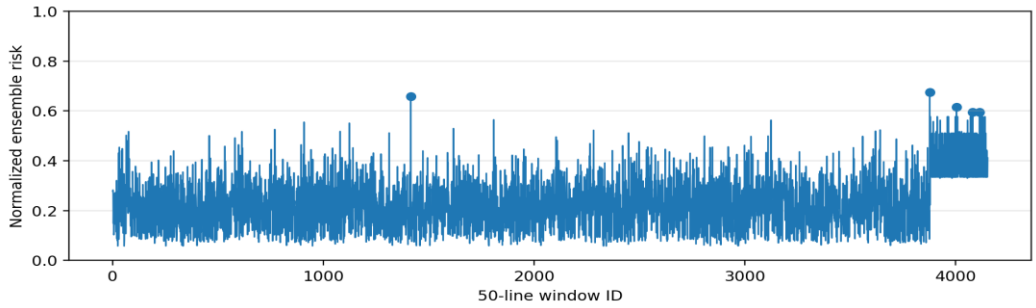

Figure 4. OpenStack anomaly timeline using the normalized ensemble risk score.

Figure 5 shows the full-window ensemble risk timeline for ZooKeeper.

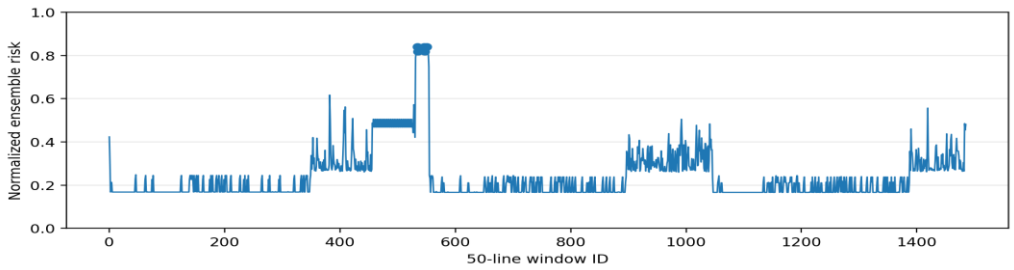

Figure 5. ZooKeeper anomaly timeline using the normalized ensemble risk score.

International Journal of Graphic Design (IJGD)
Volume 4 No. 1 April 2026
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Page 179-185
DOI: https://doi.org/10.51903/ijgd.v4i1.3703
Available online at: https://journal.stekom.ac.id/index.php/ijgd

Table 6. Anomaly and risk model comparison under the deterministic window-prioritization proxy.

Dataset	Model	Precision@5	AUPRC	ROC-AUC	Top window IDs
Hadoop	Template-frequency z-score	1.000	0.783	0.859	459, 3470, 3469, 1196, 1486
Hadoop	Isolation Forest	1.000	0.895	0.959	459, 1082, 3470, 2302, 3469
Hadoop	Local Outlier Factor	1.000	0.390	0.635	1058, 2280, 456, 480, 522
Hadoop	TF-IDF centroid distance	0.000	0.458	0.703	2188, 818, 816, 2187, 2186
Hadoop	Ensemble	1.000	0.829	0.902	1058, 459, 456, 480, 2280
OpenStack	Template-frequency z-score	1.000	0.966	0.984	3877, 1414, 1808, 2835, 2448
OpenStack	Isolation Forest	1.000	0.462	0.653	1414, 3877, 4081, 4004, 4114
OpenStack	Local Outlier Factor	0.600	0.295	0.547	1617, 2697, 809, 275, 949
OpenStack	TF-IDF centroid distance	0.000	0.249	0.527	4102, 3955, 4093, 4050, 4083
OpenStack	Ensemble	1.000	0.604	0.820	3877, 1414, 4004, 4081, 4114
Zookeeper	Template-frequency z-score	1.000	0.838	0.772	534, 546, 550, 531, 541
Zookeeper	Isolation Forest	1.000	0.587	0.093	531, 407, 534, 546, 550
Zookeeper	Local Outlier Factor	0.200	0.658	0.324	554, 1045, 1388, 350, 897
Zookeeper	TF-IDF centroid distance	1.000	0.542	0.208	547, 535, 551, 545, 541
Zookeeper	Ensemble	1.000	0.506	0.071	546, 550, 534, 547, 535

Table 7. Top anomalous windows selected by the hybrid ensemble.

Dataset	WindowId	StartLine	EndLine	TopService	TopNode	WARN	ERROR_ FATAL	Risk Relevance	Ensemble
Hadoop	1058	52901	52950	hdfs	msra-sa-41	47	2	200.100	0.774
Hadoop	459	22951	23000	mapreduce	msra-sa-41	13	5	176.600	0.765
Hadoop	456	22801	22850	hdfs	LeaseRenewer:msrabi@msra-sa-41:9000	50	0	198.600	0.747
Hadoop	480	24001	24050	ipc	msra-sa-41	39	1	166.000	0.720
Hadoop	2280	114001	114050	ipc	msra-sa-41	44	3	190.600	0.712
OpenStack	3877	193851	193900	nova-compute	nova-compute-host	1	2	43.400	0.676
OpenStack	1414	70701	70750	nova-compute	10.11.10.1	2	2	31.100	0.658
OpenStack	4004	200201	200250	nova-compute	nova-compute-host	0	0	27.800	0.615
OpenStack	4081	204051	204100	nova-compute	nova-compute-host	0	0	26.600	0.595
OpenStack	4114	205701	205750	nova-compute	nova-compute-host	0	0	26.800	0.595
Zookeeper	546	27301	27350	LearnerHandler	LearnerHandler-/10.10.34.11	24	26	272.200	0.840
Zookeeper	534	26701	26750	LearnerHandler	LearnerHandler-/10.10.34.12	24	26	272.200	0.840
Zookeeper	550	27501	27550	LearnerHandler	LearnerHandler-/10.10.34.11	24	26	272.200	0.840
Zookeeper	547	27351	27400	LearnerHandler	10.10.34.11	26	24	263.200	0.817
Zookeeper	535	26751	26800	LearnerHandler	10.10.34.12	26	24	263.200	0.817

Received: Januari 2026; Revised: February 2026; Accepted: March 2026; Published: April 2026
*Corresponding author, jiayinie8@gmail.com

International Journal of Graphic Design (IJGD)
Volume 4 No. 1 April 2026
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Page 179-185
DOI: https://doi.org/10.51903/ijgd.v4i1.3703
Available online at: https://journal.stekom.ac.id/index.php/ijgd

Figure 6 aggregates the same ensemble scores by dataset and dominant service. The heatmap shows that Hadoop risk concentrates around HDFS, IPC, and MapReduce-related windows; OpenStack concentrates around Nova compute and API behavior; and ZooKeeper concentrates around LearnerHandler and quorum-connection windows.

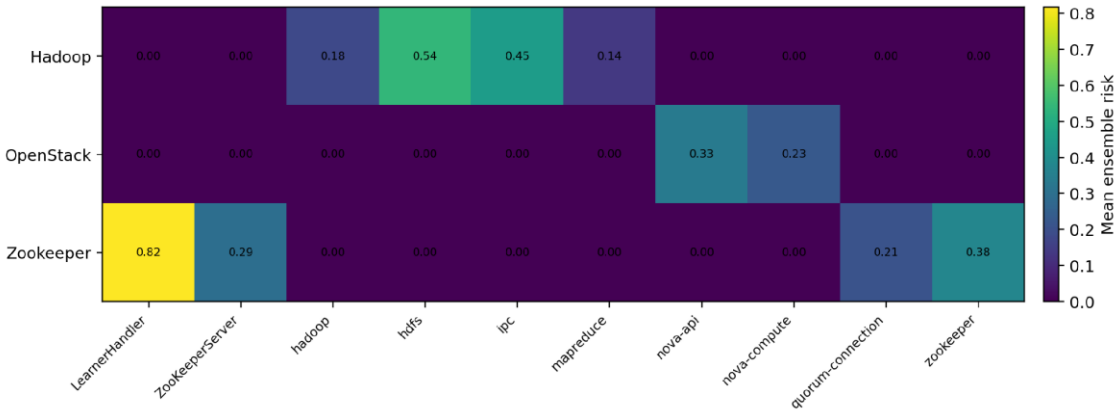

Figure 6. Service-risk heatmap from mean normalized ensemble risk.

Generated Cards

Figure 7 shows the revised visual card wireframe. The design separates salience from evidence: severity and timeline provide fast visual orientation, while evidence badges and source line ranges provide inspectable support for the suspected root cause.

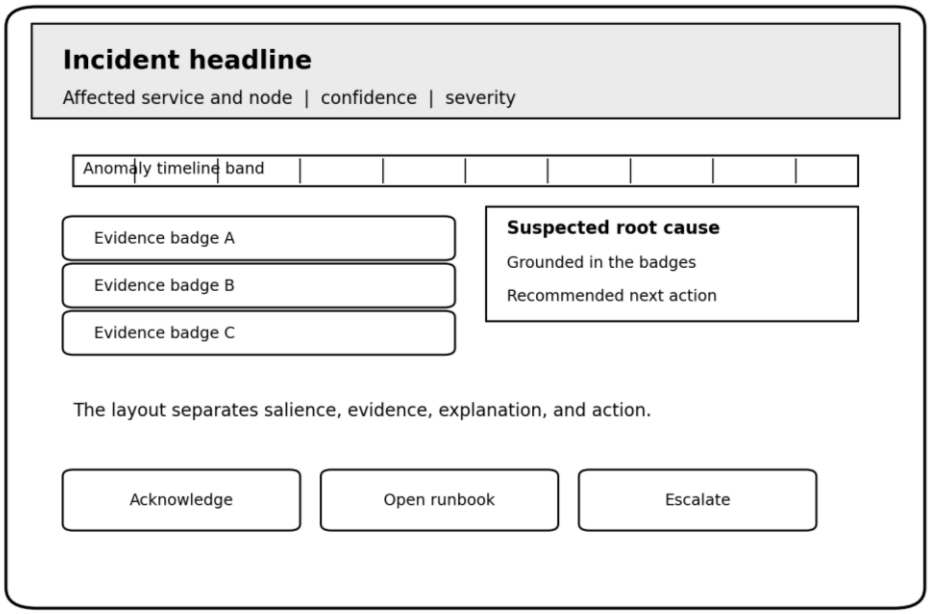

Figure 7. Wireframe of the proposed visual incident card.

Received: Januari 2026; Revised: February 2026; Accepted: March 2026; Published: April 2026
*Corresponding author, jiayinie8@gmail.com

International Journal of Graphic Design (IJGD)
Volume 4 No. 1 April 2026
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Page 179-185
DOI: https://doi.org/10.51903/ijgd.v4i1.3703
Available online at: https://journal.stekom.ac.id/index.php/ijgd

Table 8. Generated incident cards for the top three windows in each dataset.

IncidentId	Dataset	WindowId	Incident headline	Affected service	Affected node	Log evidence badge	Suspected root cause	Confidence level	Visual severity color hierarchy
HAD-1058	Hadoop	1058	Hadoop hdfs risk: MapReduce execution or ResourceManager contact failure path	hdfs	msra-sa-41	23× Address change detected. Old: <*> New: <*> | 20× Failed to renew lease for <*> for <*> seconds. Will re...	MapReduce execution or ResourceManager contact failure path	0.95	Red /critical
HAD-0459	Hadoop	459	Hadoop mapreduce risk: MapReduce execution or ResourceManager contact failure...	mapreduce	msra-sa-41	8× <*> TaskAttempt Transitioned from <*> to <*> | 7× Address change detected. Old: <*> New: <*> | 4× Diagno...	MapReduce execution or ResourceManager contact failure path	0.90	Red /critical
HAD-0456	Hadoop	456	Hadoop hdfs risk: MapReduce execution or ResourceManager contact failure path	hdfs	LeaseRenewer: msrabi@msra-sa-41:9000	23× Address change detected. Old: <*> New: <*> | 23× Failed to renew lease for <*> for <*> seconds. Will re...	MapReduce execution or ResourceManager contact failure path	0.95	Amber /elevated
OPE-3877	OpenStack	3877	OpenStack nova-compute risk: Nova instance lifecycle synchronization delay	nova-compute	nova-compute-host	17× <*> "GET <*> <*>" status: <*> len: <*> time: <*> | 5× Active base files: <*> | 5× image <*> at (<*>): c...	Nova instance lifecycle synchronization delay	0.89	Red /critical
OPE-1414	OpenStack	1414	OpenStack nova-compute risk: Nova API request concentration	nova-compute	10.11.10.1	28× <*> "GET <*> <*>" status: <*> len: <*> time: <*> | 2× [instance: <*>] VM Stopped (Lifecycle Event) | 1×...	Nova API request concentration	0.89	Red /critical
OPE-4004	OpenStack	4004	OpenStack nova-compute risk: OpenStack service-event concentration	nova-compute	nova-compute-host	14× Active base files: <*> | 13× image <*> at (<*>): checking | 13× image <*> at (<*>): in use: on this nod...	OpenStack service-event concentration	0.95	Amber /elevated
ZOO-0546	Zookeeper	546	Zookeeper LearnerHandler risk: ZooKeeper coordination-event concentration	LearnerHandler	LearnerHandler -/10.10.34.11	26× Unexpected exception causing shutdown while sock still open | 24× ******* GOODBYE <*> ********	ZooKeeper coordination-event concentration	0.92	Red / critical
ZOO-0534	Zookeeper	534	Zookeeper LearnerHandler risk: ZooKeeper coordination-event concentration	LearnerHandler	LearnerHandler -/10.10.34.12	26× Unexpected exception causing shutdown while sock still open | 24× ******* GOODBYE <*> ********	ZooKeeper coordination-event concentration	0.92	Red / critical
ZOO-0550	Zookeeper	550	Zookeeper LearnerHandler risk: ZooKeeper coordination-event concentration	LearnerHandler	LearnerHandler -/10.10.34.11	26× Unexpected exception causing shutdown while sock still open | 24× ******* GOODBYE <*> ********	ZooKeeper coordination-event concentration	0.92	Red / critical

Received: Januari 2026; Revised: February 2026; Accepted: March 2026; Published: April 2026
*Corresponding author, jiayinie8@gmail.com

International Journal of Graphic Design (IJGD)
Volume 4 No. 1 April 2026
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Page 179-185
DOI: https://doi.org/10.51903/ijgd.v4i1.3703
Available online at: https://journal.stekom.ac.id/index.php/ijgd

Table 8 lists the generated cards for the top three windows in each dataset. The table reports compact card fields rather than full raw logs; each evidence badge is derived from the measured EventTemplate distribution inside the selected window.

External Label Sanity Check On HDFS_100k

Table 9 reports the HDFS_100k sanity check. The 104,815 structured HDFS log lines produced 7,940 block sessions after BlockId grouping, including 313 anomalous sessions. The ensemble achieved ROC-AUC 0.730, AUPRC 0.492, Precision@50 0.980, and Precision@100 0.990. This result does not validate the visual card with human operators, but it does show that the evidence-based ranking is not evaluated only against a self-defined window proxy.

Table 9. External label-anchored sanity check on HDFS_100k block-session anomaly labels.

Dataset HDFS_100k					
Model	**Template-frequency z-score**	**Isolation Forest**	**Local Outlier Factor**	**Template-count centroid distance**	**Ensemble**
Sessions/blocks	7940	7940	7940	7940	7940
Anomaly sessions	313	313	313	313	313
Anomaly rate	0.039	0.039	0.039	0.039	0.039
ROC-AUC	0.543	0.724	0.540	0.730	0.730
AUPRC	0.315	0.450	0.107	0.474	0.492
Precision@50	0.940	1.000	0.560	0.980	0.980
Precision@100	0.960	1.000	0.310	0.990	0.990
Precision@500	0.200	0.294	0.068	0.314	0.300
F1 at top-10% threshold	0.125	0.174	0.076	0.182	0.182

UI Proxy Metrics

Table 10 reports the deterministic UI proxy comparison, and Figure 8 visualizes the field-completion, evidence-localization, and decision-readiness components. The card condition scores higher because all nine required fields are visible, evidence is localized into badges, and action controls are present. These are interface-structure measures, not observed SRE task outcomes.

Table 10. Deterministic UI proxy comparison across three interface conditions.

Interface	**Field completion rate**	**Evidence localization score**	**Decision-readiness score**	**Interaction-cost proxy**	**Decision-error opportunity proxy**
Raw log list	0.222	0.108	0.146	67.500	0.858
Plain-text summary	0.556	0.414	0.501	12.600	0.542
Visual incident card	1.000	0.873	0.956	3.000	0.123

Table 11 shows the card-component ablation used to calculate the decision-readiness proxy. Evidence badges and suspected root-cause statements create the largest gains because they

Received: Januari 2026; Revised: February 2026; Accepted: March 2026; Published: April 2026
*Corresponding author, jiayinie8@gmail.com

connect a visual warning to inspectable log support. Figure 9 summarizes how evidence categories from the generated cards flow into root-cause categories.

Table 11. Card-component ablation for the deterministic decision-readiness proxy.

Card configuration	Decision-readiness score
Headline only	0.45
+ severity hierarchy	0.54
+ timeline band	0.62
+ evidence badges	0.76
+ suspected root-cause statement	0.88
+ recommended action	0.96
+ decision buttons	1.00

DISCUSSION

The revised results support a narrower and more defensible design argument. The card does not prove that SREs will make better decisions in live incidents. It shows that full-scale public logs can be transformed into structured, evidence-preserving visual objects, and that the risk ranking is aligned with both a deterministic window-prioritization proxy and an external HDFS anomaly-label check. Raw log lists preserve all evidence but require the operator to discover structure manually.

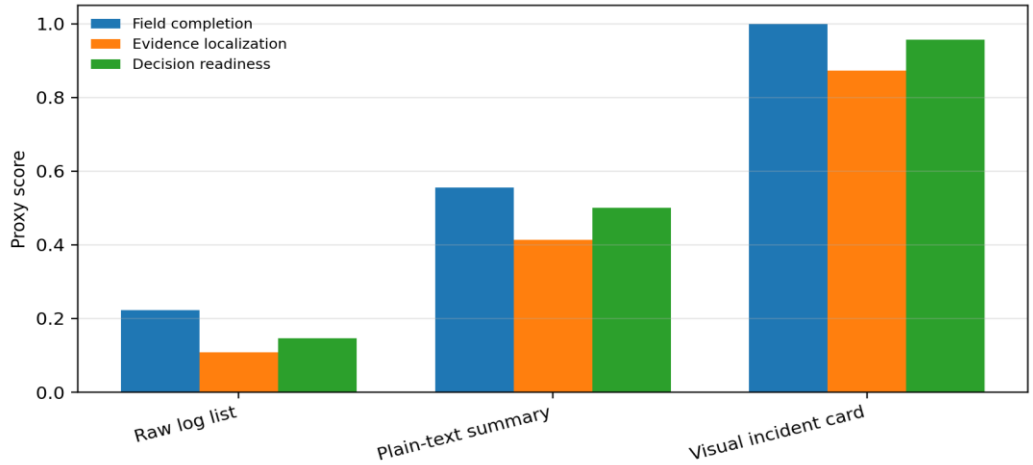

Figure 8. UI comparison under deterministic interface proxy metrics.

Plain summaries reduce volume but can detach claims from the underlying log lines. Incident cards keep both advantages by compressing the window while preserving service, node, timeline, evidence, confidence, and action fields. This is especially important for ZooKeeper, where repeated quorum and session messages can overwhelm a list view, and for OpenStack, where many benign API requests coexist with a smaller number of lifecycle warnings. The parser results also show why the card should display evidence badges rather than only a narrative.

Even high-purity normalization can merge or split messages differently from the EventId reference. In an operator interface, those differences matter because a wrong grouping can create a misleading explanation. Evidence badges act as a guardrail: they reveal the templates that produced the card, allowing the operator to reject or confirm the system's interpretation. The anomaly results indicate that risk scoring must be aligned with the operational task.

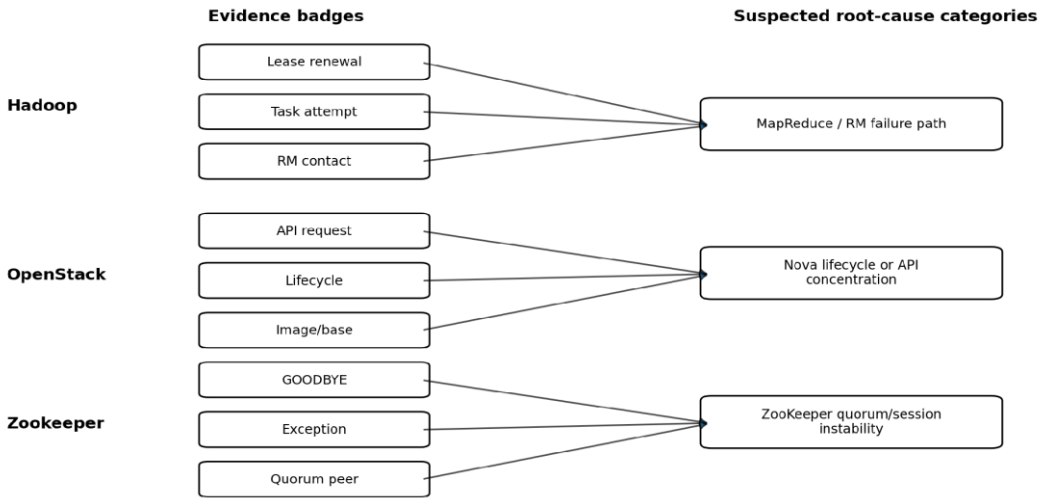

Figure 9. Evidence-to-root-cause flow for generated incident cards.

Isolation Forest and Local Outlier Factor are useful for structural novelty, but the window-prioritization proxy is based on severity, keywords, rare templates, and concentration. The HDFS_100k sanity check adds an external label reference, but it remains a log-analysis benchmark rather than a live incident-response study. The UI claims are therefore limited to structure, traceability, and reproducible design proxies. The generation layer is intentionally conservative. The paper does not evaluate hallucination rate, prompt instability, or root-cause factuality for a live LLM. Instead, it fixes the schema, root-cause library, evidence badges, and confidence calculation so that every card can be traced back to the logs. A future LLM version should be tested against the same evidence constraints and should report hallucination, factuality, calibration, and user-trust measures.

Limitation

The first limitation concerns human evaluation. The decision-readiness score, interaction-cost proxy, and decision-error opportunity proxy are deterministic interface metrics. They are not human-subject outcomes and should not be interpreted as measured reductions in SRE task time, cognitive load, trust, or decision error. A controlled SRE study remains necessary.

The second limitation concerns the generation layer. The reported experiment uses deterministic, rule-locked card generation. This design improves reproducibility and prevents unsupported root-cause text, but it does not measure a live proprietary or open-weight LLM. The framework should therefore be described as evidence-constrained and LLM-compatible rather than as a validated LLM incident-response system.

The third limitation concerns the window-prioritization proxy. For Hadoop, OpenStack, and ZooKeeper, the high-risk target is derived from observable log evidence rather than from confirmed production incident labels. The HDFS_100k label check partly addresses this weakness, but it is a different HDFS block-session task and cannot prove that the same cards improve live triage decisions. The fourth limitation concerns organizational generalization.

The card fields are designed for common SRE triage work, but organizations differ in runbook naming, escalation policy, severity taxonomy, ownership metadata, and automation rules. A production deployment would require site-specific service maps, owner metadata, and runbook integration. The fifth limitation concerns multimodal evidence. The current framework uses logs and templates. Production SRE work often combines logs with metrics, traces, alerts, tickets, and deployment history. Extending the card schema to those data types is a necessary next step.

CONCLUSION

This paper presented a reproducible UI/UX framework for transforming distributed cloud logs into evidence-constrained incident visualization cards. The revised evaluation used 461,898 full Loghub-2.0 lines from Hadoop, OpenStack, and ZooKeeper, generated parsing and window-risk metrics, produced incident cards for top-ranked windows, and added an HDFS_100k external-label sanity check over 7,940 block sessions. Drain-lite normalization achieved purity at or above 0.999 on the three full Loghub-2.0 systems. The hybrid risk ensemble achieved Precision@5 of 1.000 under the deterministic window-prioritization proxy and achieved ROC-AUC 0.730 with AUPRC 0.492 on HDFS_100k external anomaly labels.

The main finding is that incident understanding can be made more structured when model outputs are reorganized as visual evidence objects rather than delivered only as isolated scores or unstructured paragraphs. The proposed card schema keeps the operator's decision context visible: what happened, where it happened, why it is suspected, how confident the system is, what evidence supports it, and what action is available. The claim is intentionally bounded: the study provides full-data and label-anchored design-proxy evidence for an incident-card framework, not validated proof of improved SRE performance in human incident-response testing.

Author's CRediT

J.N. led the implementation of the log-processing pipeline, experimental evaluation, incident-card generation framework, and preparation of the initial manuscript. G.L. contributed to the conceptualization of the study, methodology design, validation of the evidence-constrained reasoning framework, project supervision, and manuscript review. C.L. contributed to data analysis, case-study evaluation, and interpretation of operational findings. T.Z. contributed to the design of the incident-visualization interfaces, user-centered visualization strategies, interaction-design analysis, and manuscript editing. All authors reviewed and approved the final version of the manuscript. J.N. and G.L. contributed equally to this work and share first authorship. G.L. served as the corresponding author.

REFERENCES

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93-104. https://doi.org/10.1145/342009.335388

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys,* 41(3), Article 15. https://doi.org/10.1145/1541880.1541882

Chen, Y., & Li, M. (2025). From hand-drawn sketches to interactive web prototypes: A reproducible vision-language approach with structural and visual consistency evaluation. *Journal of Technology Informatics and Engineering*, 4(2), 364–384. https://doi.org/10.51903/jtie.v4i2.490

Chen, Y., Xie, H., Ma, M., Kang, Y., Gao, X., Shi, L., Cao, Y., Gao, X.-C., Fan, H., Wen, M., Zeng, J., Ghosh, S., Zhang, X., Lin, Q., Rajmohan, S., & Zhang, D. (2024). Automatic root cause analysis via large language models for cloud incidents. *Proceedings of the Nineteenth European Conference on Computer Systems*, 674-688. https://doi.org/10.1145/3627703.3629553

Cui, T., Ma, S., Chen, Z., Xiao, T., Tao, S., Liu, Y., Zhang, S., Lin, D., Liu, C., Cai, Y., Meng, W., & Pei, D. (2024). LogEval: A comprehensive benchmark suite for large language models in log analysis. *arXiv*. https://arxiv.org/abs/2407.01896

Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1285-1298. https://doi.org/10.1145/3133956.3134015

Fariha, A., Gharavian, V., Makrehchi, M., Rahnamayan, S., Alwidian, S., & Azim, A. (2024). Log anomaly detection by leveraging LLM-based parsing and embedding with attention mechanism. *2024 IEEE Canadian Conference on Electrical and Computer Engineering*, 859-863. https://doi.org/10.1109/CCECE59415.2024.10667308

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX: Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 139-183). North-Holland. https://doi.org/10.1016/S0166-4115(08)62386-9

He, M., Tong, J., Duan, C., Cai, H., Li, Y., & Huang, G. (2024). LLMeLog: An approach for anomaly detection based on LLM-enriched log events. *2024 IEEE 35th International Symposium on Software Reliability Engineering*, 132-143. https://doi.org/10.1109/ISSRE62328.2024.00023

He, P., Zhu, J., He, S., Li, J., & Lyu, M. R. (2016). An evaluation study on log parsing and its use in log mining. *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 654-661. https://doi.org/10.1109/DSN.2016.66

He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An online log parsing approach with fixed depth tree. *2017 IEEE International Conference on Web Services*, 33-40. https://doi.org/10.1109/ICWS.2017.13

Jiang, Z., Liu, J., Huang, J., Li, Y., Huo, Y., Gu, J., Chen, Z., Zhu, J., & Lyu, M. R. (2024). A large-scale evaluation for log parsing techniques: How far are we? *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*. https://doi.org/10.1145/3650212.3652123

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *2008 Eighth IEEE International Conference on Data Mining*, 413-422. https://doi.org/10.1109/ICDM.2008.17

Makanju, A., Zincir-Heywood, A. N., & Milios, E. E. (2009). Clustering event logs using iterative partitioning. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1255-1264. https://doi.org/10.1145/1557019.1557154

Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., Zhang, R., Tao, S., Sun, P., & Zhou, R. (2019). LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 4739-4745. https://doi.org/10.24963/ijcai.2019/658

Munzner, T. (2014). Visualization analysis and design. CRC Press.

Nielsen, J. (1994). Usability inspection methods. John Wiley & Sons.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135-1144. https://doi.org/10.1145/2939672.2939778

Samanta, S., Chatterjee, O., Mohapatra, P., de Magalhaes, A., Gupta, H., & Rahane, A. (2024). Efficient incident summarization in ITOps: Leveraging entity-based grouping. Proceedings of the 2024 International Workshop on Software Engineering for AI in the Enterprise. IBM Research.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings of the IEEE Symposium on Visual Languages*, 336-343. https://doi.org/10.1109/VL.1996.545307

Xu, W., Huang, L., Fox, A., Patterson, D., & Jordan, M. I. (2009). Detecting large-scale system problems by mining console logs. *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles*, 117-132. https://doi.org/10.1145/1629575.1629587

Yushan Chen, & Evelyn Chan. (2023). Multimodal UI Representation Learning: Ablation of Screenshot, Wireframe, and View-Hierarchy Proxies on an Uploaded 168-Screen Dataset. *Journal of Advanced Computing Systems* , 3(1), 1-15. https://doi.org/10.69987/JACS.2023.30101

Zhu, J., He, S., He, P., Liu, J., & Lyu, M. R. (2023). Loghub: A large collection of system log datasets for AI-driven log analytics. *2023 IEEE 34th International Symposium on Software Reliability Engineering.*

Zhu, J., He, S., Liu, J., He, P., Xie, Q., Zheng, Z., & Lyu, M. R. (2019). Tools and benchmarks for automated log parsing. *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings*, 121-124. https://doi.org/10.1109/ICSE-Companion.2019.00051