Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Topology-Aware SSD Health Risk Prediction with SMART Signals, Location Features, and LLM-Based Failure Explanation

Ethan Ng
Computer Engineering, Monash University, Melbourne, VIC, Australia
eng_dev77@yahoo.com

DOI: 10.69987/JACS.2026.60604

Keywords

SSD failure prediction; SMART attributes; topology-aware machine learning; rack-level risk; imbalanced classification; interpretable AI; large language model explanations.

Abstract

Solid-state drive (SSD) failure prediction is commonly treated as a device-level classification problem based on Self-Monitoring, Analysis and Reporting Technology (SMART) counters. Production fleets, however, also exhibit model, application, and placement heterogeneity, and failures can cluster within nodes and racks. This study presents TopoHealth-ET, a topology-aware risk model that combines SMART indicators, deployment metadata, train-only topology priors, and an evidence-constrained language explanation layer. The empirical evaluation uses a deterministic schema-aligned benchmark that follows the documented Alibaba SSD table structures and contains 30,000 disks, 11 drive models, 200 racks, and a 1.89% failure rate. On the held-out test split, TopoHealth-ET achieved a ROC-AUC of 0.775, PR-AUC of 0.203, precision of 0.345, recall of 0.175, and F1 of 0.233. At a 1% maintenance review budget, the ranked worklist captured 18.42% of failures with 35.00% precision, compared with a fleet failure rate of 1.90%. The results indicate that topology context can materially improve rare-event maintenance triage, while structured evidence objects allow failure explanations to remain concise, auditable, and aligned with the classifier evidence.

1. Introduction

Flash-based SSDs are now central to cloud and enterprise storage because they offer high random-I/O throughput, low latency, and favorable power characteristics. Their reliability behavior is nevertheless governed by mechanisms that differ from those of magnetic disks. Program/erase cycling, retention loss, read disturb, controller behavior, firmware, workload mix, temperature, and manufacturing variation can all affect observed degradation and failure [10]–[13]. As a result, preventive maintenance depends on combining device telemetry with a realistic view of the deployment environment.

SMART attributes provide the most widely available device-health signals. Raw and normalized counters expose media errors, program and erase failures, wear indicators, temperature, power-on usage, written logical blocks, and interface errors. Field studies have shown that these signals contain useful precursors, but they are noisy, vendor-specific, and far from perfectly separable: many healthy drives exhibit abnormal counters, while some failed drives show only weak warning patterns [4]–[7]. SSD-specific studies further show that the relevance of individual attributes changes across models and wear stages [2].

A second source of information comes from topology. The Alibaba SSD studies cover SMART logs, failure tickets, physical locations, and application

assignments for a fleet of nearly one million SSDs across 11 drive models. Their analyses report non-negligible spatial and temporal correlations within nodes and racks, together with substantial heterogeneity by model and application [1], [2]. These observations motivate a prediction task that asks not only whether a disk has abnormal counters, but also whether those counters occur in a deployment context where related failures are more likely.

This paper formulates that task as topology-aware SSD health risk prediction. Each disk is represented by device-local SMART signals and by contextual variables including application, machine room, rack, node, slot, and drive model. The output is a continuous risk score, a budget-constrained maintenance worklist, and a short explanation grounded in the same evidence used for ranking. The model is designed for daily fleet triage, where the practical objective is to concentrate likely failures within a small review queue rather than to maximize uninformative overall accuracy.

The explanation layer follows an evidence-first design. Local drivers are extracted from wear, error, usage, temperature, and topology features, then rendered as a maintenance-oriented narrative. This arrangement is consistent with local explanation methods such as LIME and SHAP [22], [23], while recognizing that transformer and retrieval-augmented language models require explicit grounding to avoid unsupported causal statements [24]–[26]. The language layer does not alter the classifier score; it communicates the evidence that produced the ranking.

The study makes four contributions. First, it defines a reproducible data design aligned with the documented Alibaba SMART, failure-tag, and location schemas. Second, it introduces TopoHealth-ET, an ExtraTrees ensemble with train-only topology priors and deterministic late fusion. Third, it evaluates ranking, threshold, and review-budget behavior under severe class imbalance, including feature ablations and topology-stratified analysis. Fourth, it develops a constrained evidence-to-text interface that supports auditable LLM-based failure explanations without introducing a second decision mechanism.

2. Literature Review

2.1 Field Reliability and SSD Failure Prediction

Large-scale HDD studies established two durable lessons for predictive maintenance: vendor mean-time-to-failure figures do not fully describe field behavior, and individual SMART attributes rarely provide strong deterministic rules [4], [5]. Early machine-learning work therefore modeled failure as a multivariate pattern-recognition problem, using multiple-instance learning and statistical warning models to detect weak precursors [6], [7]. Related studies of latent sector errors and storage-stack corruption demonstrated that device faults interact with higher layers of the storage system [8], [9].

SSD reliability research adds flash-specific mechanisms. Large field studies identified strong effects from data written, wear, error correction, and device age [10], [11], while controlled characterization showed how retention loss and read disturb emerge from NAND behavior [12], [13]. Alibaba research extended this evidence to heterogeneous production fleets: correlated failures occur within nodes and racks [1], and robust feature selection must account for model-specific SMART availability and wear stages [2]. Large-scale disk-prediction systems also emphasize the need for pipelines that remain computationally feasible at fleet scale [3].

Recent SSD prediction methods increasingly combine multiple feature views and richer temporal structure. Chakraborttii and Litz focused on accuracy, adaptability, and interpretability across SSD models [41]. The MVTRF approach jointly modeled failure type, time, and explanatory decisions from long- and short-term monitoring views [42]. Gu, Wu, and He combined SMART attributes with device-level NAND wear characteristics [43], while the temporal-contextual attention network integrated recurrent and attention mechanisms to capture temporal patterns and inter-attribute dependencies [44]. These studies motivate richer representations, but they also leave room for a complementary question: how much can deployment topology improve a transparent single-snapshot risk-ranking pipeline?

2.2 Extreme Imbalance, Calibration, and Fusion

Failure prediction is an extreme-imbalance problem. Tree ensembles and gradient boosting offer strong nonlinear baselines [14]–[16], while resampling and cost-sensitive learning address the scarcity of positive cases [17], [18]. Evaluation must reflect this imbalance: ROC-AUC is useful for global ranking [19], but precision-recall analysis more directly expresses the quality of the flagged set when failures are rare [20]. Probability calibration is a separate concern because a well-ranked model can still produce scores that should not be read as literal probabilities [21].

Methodological work in other rare-event domains reinforces these principles. Cost-sensitive, positive-unlabeled, and one-class approaches have been compared for fraud detection [27], while bankruptcy studies examine resampling, focal loss, and explainable financial profiles under extreme imbalance [28]. Credit-risk workflows combine calibrated tiers, uncertainty-based rejection, and explanation to support bounded operational decisions [29], [30]. Late-fusion research likewise shows that combining heterogeneous signals is most useful when component confidence and fusion rules are explicit [31]. TopoHealth-ET adopts this operational perspective: the final score is evaluated both as a ranker and as a finite maintenance queue.

2.3 Topology and Deployment Context

Topology is not merely descriptive metadata. Alibaba measurements show that failures can be correlated at node and rack scope, and that model and application context affect observed behavior [1], [2]. Similar ideas appear in infrastructure monitoring: noisy-neighbor-aware VM degradation models treat co-location as part of the risk process [32], and cluster-capacity forecasting uses workload semantics and topology to characterize operational risk under heterogeneous demand [33]. These studies suggest that device health should be interpreted conditionally on where and how the device is deployed.

A practical topology representation must avoid leakage. Direct rack identifiers can capture stable placement effects, but label-derived rack statistics are valid only when estimated from historical training records. Smoothed empirical-Bayes priors offer a compact compromise: they preserve recurring group-level risk while shrinking small groups toward the fleet average. The present work uses such train-only priors for rack, node, model, application, and machine room, alongside raw physical identifiers.

2.4 Evidence-Grounded Operational Explanation

Predictive maintenance explanations serve a different purpose from causal diagnosis. LIME and SHAP provide local attributions for model behavior [22], [23], but an operator still needs a concise account of what signals were elevated and what action merits review. Transformer language models can generate fluent technical text [24], [25], and retrieval-augmented generation can bind text generation to external evidence [26]. In reliability-sensitive settings, the key design requirement is that every statement remain traceable to a feature or retrieved record.

Operational AI research has increasingly applied this principle to incident triage. Multi-source root-cause attribution and OpsLLM-style systems combine structured telemetry, logs, and grounded summaries for microservices and cloud operations [34], [35]. Evidence-grounded DevOps question answering and evidence-chain RAG emphasize source attribution, provenance, and hallucination control [36], [37]. Log-analysis studies add transformer-based anomaly representations, conformal alert control, and evidence-grounded incident ticket generation [38], [39], while incident visualization cards organize distributed-log evidence for human review [40]. TopoHealth-ET applies the same discipline at the disk level: the explanation layer receives a structured evidence object and is not permitted to infer an unobserved physical root cause.

3. Method

3.1 Data Design and Experimental Population

The data design follows the public Alibaba SSD structure. The location table provides application, drive model, rack, node, disk, and slot fields; the last-day SMART table provides model, disk identifier, date, and raw or normalized SMART attributes; and

the failure-tag table links failures and ticket-time SMART values to application and location fields [1], [2]. The local experiment uses a deterministic, schema-aligned population generated with seed 2023 so that the complete workflow can be evaluated under the documented field structure. Table 1 summarizes the source tables and the experimental population, while Figure 1 shows the processing sequence from joined records to ranking and explanation.

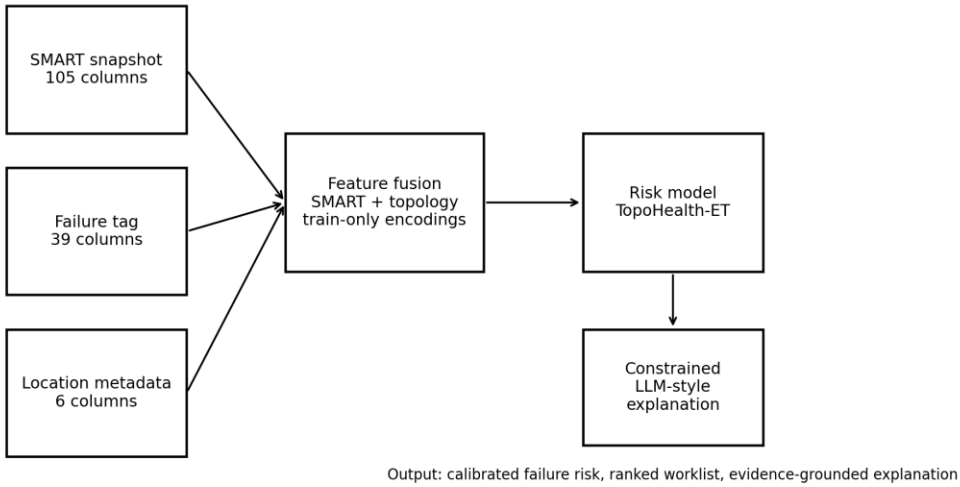

Figure 1. System architecture for topology-aware SSD health prediction and constrained explanation.

Table 1. Data tables and their roles in the topology-aware pipeline.

Data source	Columns	Principal fields	Role
Location table	6	app, model, rack_id, node_id, disk_id, slot_id	Topology and deployment metadata
SMART snapshot	105	model, disk_id, ds, raw and normalized SMART fields	Device-health snapshot
Failure-tag table	39	model, failure_time, failure, app, machine_room_id, rack_id, node_id, disk_id, SMART fields	Labels, tickets, and ticket-time health signals
Schema-aligned benchmark	—	30,000 disks, 11 models, 200 racks, seed 2023	Experimental population following the documented field semantics

The benchmark contains 30,000 disks and 567 failures. Failure probability is generated from a controlled combination of model baseline, application stress, rack effect, usage, wear, temperature, and SMART error counters. Failed disks tend to carry stronger wear and error evidence, but overlap is deliberately retained so that no single

feature separates the classes. The resulting 1.89% failure rate creates a realistic rare-event classification problem. Table 2 reports the split composition, and Table 3 reports the model distribution.

Table 2. Stratified disk-level split used in the experiment.

Split	Disks	Failures	Failure rate (%)	Racks	Models
Train	18,000	340	1.89	200	11
Validation	6,000	113	1.88	200	11
Test	6,000	114	1.90	200	11

Table 3. Drive-model distribution and failure rate in the experimental population.

Model	Disks	Failures	Failure rate (%)	Racks
MA1	2,413	46	1.91	200
MA2	2,622	67	2.56	200
MA3	2,091	21	1.00	200
MA4	2,337	28	1.20	200
MB1	3,061	44	1.44	200
MB2	2,956	27	0.91	200
MB3	2,602	84	3.23	200
MC1	3,214	73	2.27	200
MC2	2,994	35	1.17	200
MC3	2,826	107	3.79	200
MC4	2,884	35	1.21	200

3.2 Feature Engineering

Feature engineering is organized into device-local, contextual, and derived groups. SMART features include raw and normalized media-error counters, program and erase errors, wear-out proxies, temperature, power-on usage, read/write activity, interface errors, and sums such as program_erase_sum and read_crc_sum. Because attribute availability differs across models, optional counters are handled conservatively and the model does not assume that every vendor exposes the same fields [2], [41]–[44].

Context features include model and application, physical identifiers for machine room, rack, node, and slot, and simple density measures such as logarithmic rack disk count and node slot count. These fields allow the learner to represent workload and placement heterogeneity without treating topology as an unstructured text label. Table 4 summarizes the feature groups and their roles.

Table 4. Feature groups used by TopoHealth-ET.

Feature group	Fields and transformations	Purpose
SMART counters	Raw and normalized media errors, program/erase counts, wear-out, temperature, usage, CRC, and derived sums	Device-level health and wear evidence [10]–[13], [43]
Categorical context	model, app	Model and workload heterogeneity [1], [2], [42]
Physical topology	machine_room_id, rack_id, node_id, slot_id, rack_disk_count_log, node_slot_count	Deployment location and co-placement effects [1], [32]
Train-only topology priors	rack_id_te, node_id_te, model_te, app_te, machine_room_id_te	Smoothed historical risk without validation or test leakage
Explanation evidence	Top local drivers from SMART and topology features	Grounded maintenance narratives [22]–[26], [34]–[40]

3.3 Train-Only Topology Priors

Topology target encodings are estimated only from the training split. For each categorical group g, the observed training failure rate is shrunk toward the global training prior. Let n_g be the number of training disks in the group, r_g its empirical failure rate, r_0 the global training failure rate, and m a smoothing constant. The encoded risk is

$$TE(g) = (n_g r_g + m r_0) / (n_g + m) \quad (1)$$

The fitted maps are then applied to validation and test records. Unseen groups receive r_0. This construction is essential because rack or node averages computed over the full dataset would transmit evaluation labels into the features. The same rule is used for rack, node, model, application, and machine room.

3.4 TopoHealth-ET and Baselines

TopoHealth-ET uses an ExtraTrees classifier to model nonlinear interactions between SMART and topology variables. Its tree score is combined with a compact health index h built from normalized wear-out, program/erase error intensity, read/CRC error intensity, temperature margin, usage, and topology prior. Component weights are fixed before evaluation, and the final score is

$$s_{final} = 0.35 s_{ET} + 0.65 h \quad (2)$$

The late-fusion form retains nonlinear modeling while keeping a direct operational path from named evidence groups to the final ranking. It also limits the influence of any single noisy SMART counter. Logistic regression, random forests, ExtraTrees, and isolation forest provide supervised and unsupervised baselines. Table 5 lists the exact configurations.

Table 5. Model configurations used in the comparison.

Model	Feature set	Configuration
Logistic-SMART	SMART	Logistic regression; balanced class weights; liblinear; max_iter = 500
RandomForest-SMART	SMART	20 trees; max_depth = 9; min_samples_leaf = 10; balanced_subsample

Model	Feature set	Configuration
ExtraTrees-SMART	SMART	30 trees; max_depth = 10; min_samples_leaf = 8; balanced
Logistic-Topo	SMART + topology	Logistic regression with fused features and balanced class weights
RandomForest-Topo	SMART + topology	25 trees; max_depth = 10; min_samples_leaf = 10; balanced_subsample
IsolationForest-SMART	SMART	25 trees trained on normal training disks; anomaly score converted to risk
TopoHealth-ET	SMART + topology	60-tree ExtraTrees plus deterministic SMART/topology late-fusion index

3.5 Training and Evaluation Protocol

Disks are split 60%/20%/20% into training, validation, and test subsets with stratification by the failure label. This design represents ongoing triage within a known fleet: racks may appear in all splits, but each disk appears only once. Numeric preprocessing, one-hot encoding, imputation, topology maps, and model fitting are learned from the training subset. The validation subset is used to choose the F1-maximizing decision threshold, which is then frozen before test evaluation.

The primary ranking metrics are ROC-AUC and PR-AUC. PR-AUC receives greater interpretive weight because the positive class is rare [19], [20]. Threshold metrics include precision, recall, F1, specificity, false-alarm rate, and the confusion matrix. To connect prediction to maintenance capacity, the analysis also evaluates the top 0.5%, 1%, 2%, 5%, and 10% of ranked disks. These operating points answer how many failures are captured when only a fixed share of the fleet can be reviewed.

All random components use seed 2023. The models share the same split and preprocessing artifacts, and every supervised transformation is restricted to training records. Scores are interpreted primarily as priorities; an operational deployment would recalibrate them on recent fleet data before treating them as failure probabilities [21], [29], [30].

3.6 Evidence-Constrained Explanation Layer

For each high-risk disk, the predictor produces an evidence object containing the disk identifier, model, rack, risk score, and available local drivers. Candidate drivers include high wear-out, program/erase errors, read/CRC errors, elevated rack prior, high usage, and a narrow temperature margin. The explanation layer converts this object into a short sentence with three rules: every stated driver must be present in the object; no unmeasured physical cause may be asserted; and the output must distinguish a risk indicator from a confirmed failure mechanism.

The study uses deterministic evidence-to-text rendering so that explanations are stable across runs. The same structured object can be passed to an LLM with a constrained prompt and optional retrieval over maintenance records, following the provenance and grounding practices described in [34]–[40]. Prediction and prose generation remain separate: changing the wording cannot change the risk score or the maintenance rank.

4. Results and Discussion

4.1 Population and Topology Heterogeneity

The experimental population includes 30,000 disks, 567 failures, 11 drive models, and 200 racks. Model prevalence and failure rate are visibly heterogeneous in Figure 2. MC3 has the highest model-level rate at 3.79%, whereas MB2 has the lowest at 0.91%. Such variation creates a setting in which the same SMART pattern can have different implications across model and application context.

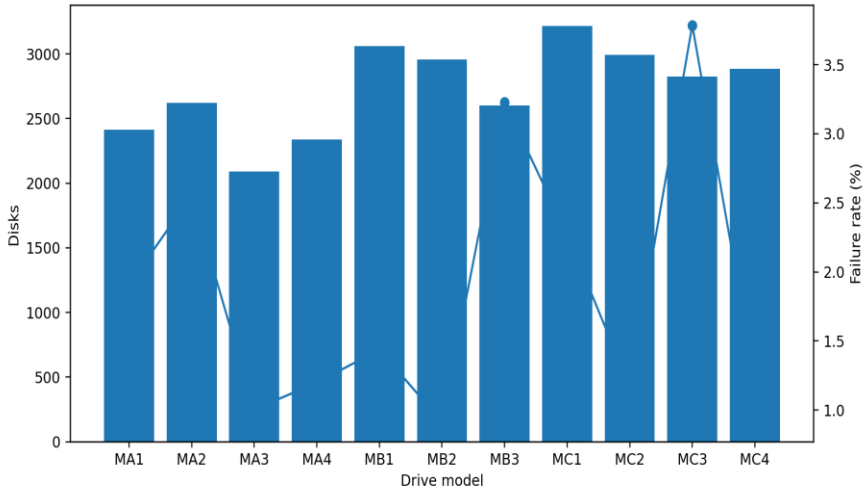

Figure 2. Drive population and observed failure rate by model.

Figure 3 maps observed failure rate across machine rooms and rack indices. Elevated regions form a nonuniform risk surface rather than a flat fleet-wide baseline. The heatmap does not by itself establish causality, but it shows why rack and machine-room history can add information to device-local counters. The train-only priors in Equation (1) convert this pattern into bounded features while shrinking sparse groups toward the global rate.

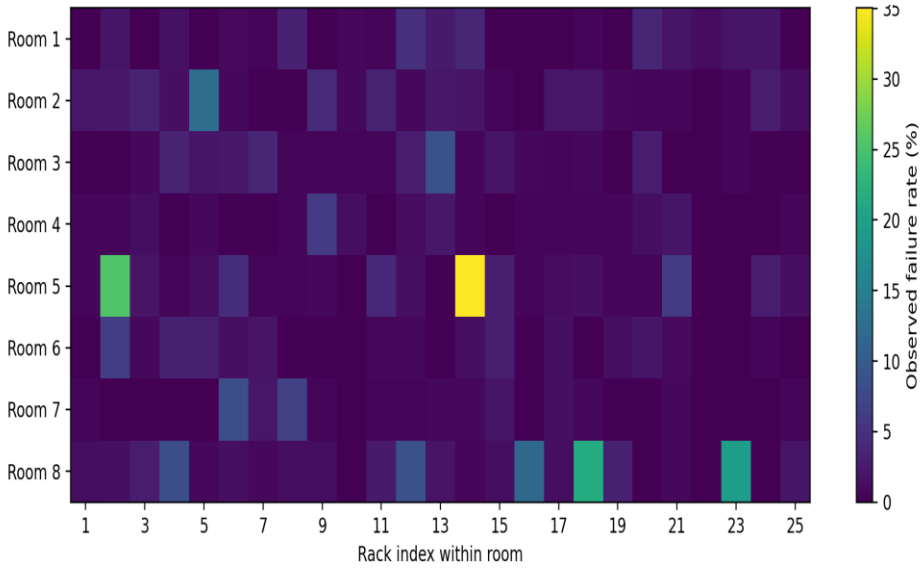

Figure 3. Topology-aware risk surface across machine rooms and rack indices.

4.2 Main Model Comparison

Table 6 reports the full held-out test comparison. TopoHealth-ET is the strongest ranker, with ROC-AUC 0.775 and PR-AUC 0.203. The best SMART-only supervised baseline by ROC-AUC is Logistic-SMART at 0.646, while its PR-AUC is 0.044. Relative to the 1.90% test failure rate, the TopoHealth-ET PR-AUC indicates a substantial concentration of positives near the top of the ranking.

Table 6. Main comparison on the held-out test split.

Model	Features	ROC-AUC	PR-AUC	Precision	Recall	F1	Specificity	FAR	TP	FP	TN	FN
TopoHealth-ET	SMART+Topo	0.775	0.203	0.345	0.175	0.233	0.994	0.006	20	38	5,848	94
RandomForest-Topo	SMART+Topo	0.742	0.094	0.152	0.351	0.212	0.962	0.038	40	224	5,662	74
Logistic-SMART	SMART	0.646	0.044	0.060	0.167	0.089	0.950	0.050	19	296	5,590	95
Logistic-Topo	SMART+Topo	0.635	0.055	0.074	0.219	0.110	0.947	0.053	25	314	5,572	89
IsolationForest-SMART	SMART	0.627	0.045	0.097	0.061	0.075	0.989	0.011	7	65	5,821	107
ExtraTrees-SMART	SMART	0.618	0.048	0.036	0.289	0.064	0.849	0.151	33	888	4,998	81
RandomForest-SMART	SMART	0.599	0.030	0.037	0.167	0.061	0.916	0.084	19	492	5,394	95

At the validation-selected threshold, TopoHealth-ET identifies 20 true failures and 38 false positives, leaving 5,848 true negatives and 94 false negatives. The corresponding false-alarm rate is 0.646%. RandomForest-Topo captures 40 failures but produces 224 false positives, giving a 3.806% false-alarm rate. The comparison illustrates an operational trade-off: TopoHealth-ET accepts lower threshold recall in exchange for a much smaller and higher-yield review queue.

The SMART-only tree models show why context and score design matter. ExtraTrees-SMART reaches recall 0.289 but flags 888 healthy disks, reducing precision to 0.036. Logistic-SMART offers a more restrained threshold but still produces 296 false positives. Topology does not guarantee improvement when appended mechanically: Logistic-Topo has only a small PR-AUC gain, whereas the tree interactions and late-fusion index in TopoHealth-ET convert context into a more useful high-risk tail.

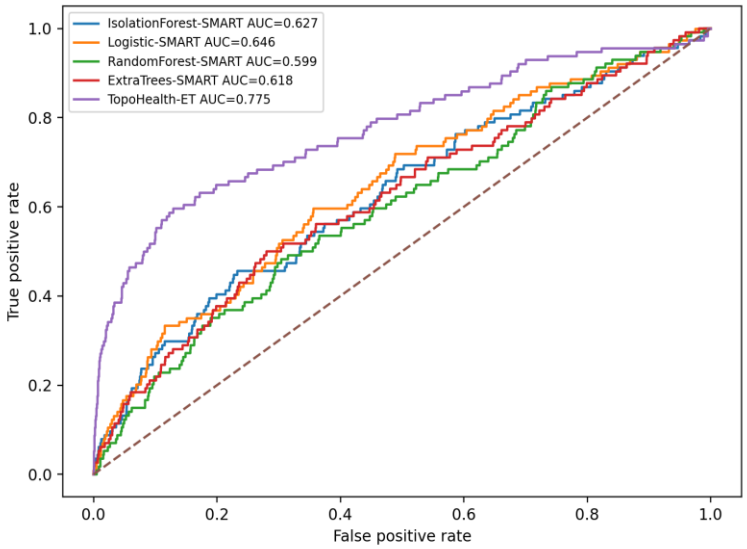

Figure 4. ROC curves for principal baselines and TopoHealth-ET on the held-out test split.

The ROC curves in Figure 4 show the global ranking advantage of TopoHealth-ET, and the precision-recall curves in Figure 5 make the rare-event advantage clearer. The left portion of the PR curve corresponds to the small review budgets used in practice. TopoHealth-ET maintains substantially higher precision in this region than the SMART-only alterna tives.

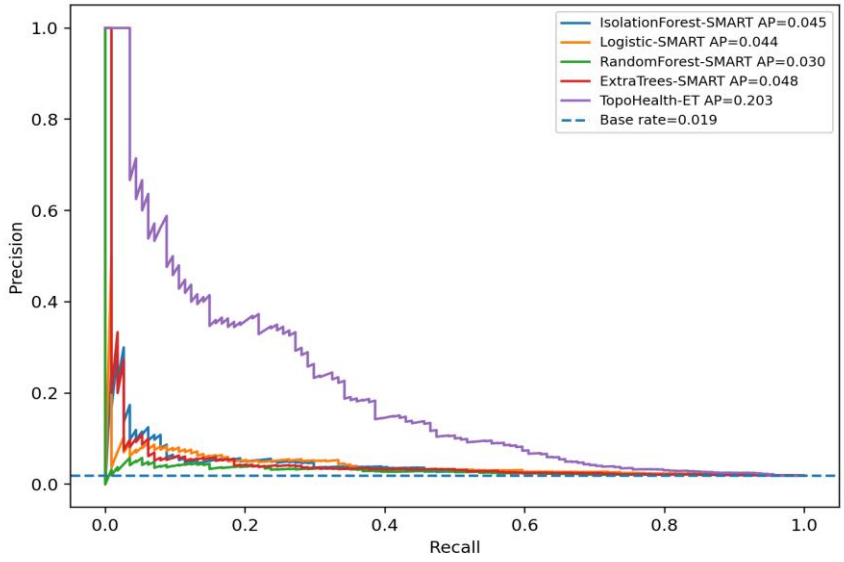

Figure 5. Precision-recall curves for principal baselines and TopoHealth-ET.

4.3 Feature-Group Ablation

Table 7 isolates feature availability with a common logistic learner. SMART alone yields ROC-AUC 0.641. Adding application and model raises ROC-AUC to 0.687, and adding room/rack identifiers raises it to 0.692. These gains show that model and placement contain predictive information even before nonlinear interactions are introduced.

Table 7. Ablation over SMART and topology feature groups.

Ablation	ROC-AUC	PR-AUC	Precision	Recall	F1
SMART only	0.641	0.039	0.050	0.140	0.074
SMART + app/model	0.687	0.048	0.073	0.184	0.105
SMART + room/rack IDs	0.692	0.051	0.068	0.140	0.092
SMART + topology encodings	0.636	0.053	0.040	0.281	0.069
SMART + full topology	0.630	0.054	0.072	0.175	0.103

The ablation also shows that topology encodings are not universally beneficial under a linear decision surface. Smoothing compresses group differences, and correlated identifiers can reduce linear separability even while improving the high-risk tail. Figure 6 visualizes this non-monotonic pattern. The full TopoHealth-ET result should therefore be interpreted as a joint effect of feature availability, nonlinear interactions, and deterministic fusion rather than as evidence for a single dominant field.

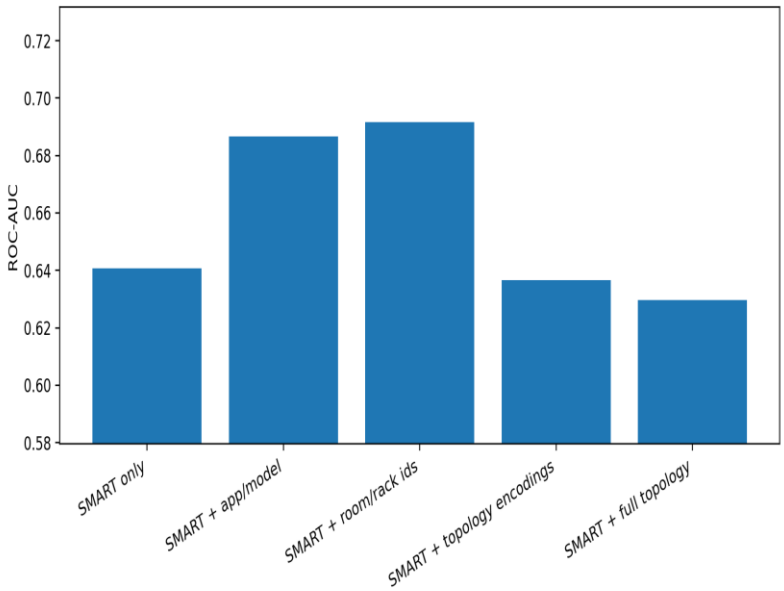

Figure 6. Ablation trend for topology-aware feature variants.

4.4 Budgeted Maintenance Triage

Budgeted ranking provides the clearest operational interpretation. Table 8 reports outcomes when only the highest-scoring share of the test fleet is reviewed. At a 0.5% budget, 30 disks are inspected and 13 failures are captured, for 43.33% precision. At a 1% budget, 60 disks contain 21 failures, for 35.00% precision and 18.42% recall. A random set of 60 disks would contain about 1.14 failures at the observed base rate, so the ranked list provides a large enrichment in review yield.

Table 8. Maintenance operating points for the TopoHealth-ET ranking.

Review budget (%)	Flagged disks	Precision	Recall	Failures caught
0.5	30	0.433	0.114	13
1.0	60	0.350	0.184	21
2.0	120	0.275	0.289	33
5.0	300	0.147	0.386	44
10.0	600	0.098	0.518	59

Increasing the review budget raises recall but reduces precision. At 5%, the system catches 44 failures with 14.67% precision; at 10%, it catches 59 failures with 9.83% precision. Both remain well above the 1.90% base rate. This curve allows a maintenance team to select an operating point from staffing and spare-inventory constraints rather than from an arbitrary probability cutoff.

4.5 Topology-Stratified Behavior and Feature Importance

Table 9 divides the test set into quartiles by rack risk. The highest-risk quartile has a 5.67% failure rate, compared with no observed failures in the lowest quartile. Within the high-risk quartile, TopoHealth-ET reaches ROC-AUC 0.768 and PR-AUC 0.285. The model is most useful where topology actually concentrates failures; the undefined Q1 metrics correctly reflect the absence of positive cases rather than a numerical failure of the method.

Table 9. Topology-stratified performance of TopoHealth-ET.

Rack-risk quartile	Disks	Failure rate (%)	ROC-AUC	PR-AUC
Q1 low	1,500	0.00	n/a	n/a
Q2	1,500	0.80	0.514	0.009
Q3	1,500	1.13	0.666	0.101
Q4 high	1,500	5.67	0.768	0.285

Grouped permutation importance in Table 10 and Figure 7 reinforces the same interpretation. Topology target encodings produce the largest positive AUC drop when permuted, followed by model/application context, usage/age, and wear/program/erase features. Read/CRC and physical-identifier groups show negative drops because groups are correlated and the runtime-oriented analysis uses a single deterministic permutation per group; in that setting, permutation values should be read as directional diagnostics rather than independent causal effects.

Table 10. Grouped permutation importance for the TopoHealth-ET tree component.

Feature group	Mean AUC drop	Repeat SD
Topology target encodings	0.107	0.000
Model/application	0.084	0.000
Usage/age	0.068	0.000
Wear/program/erase	0.040	0.000
Temperature	0.006	0.000
Read/CRC/error	-0.021	0.000
Physical identifiers	-0.041	0.000

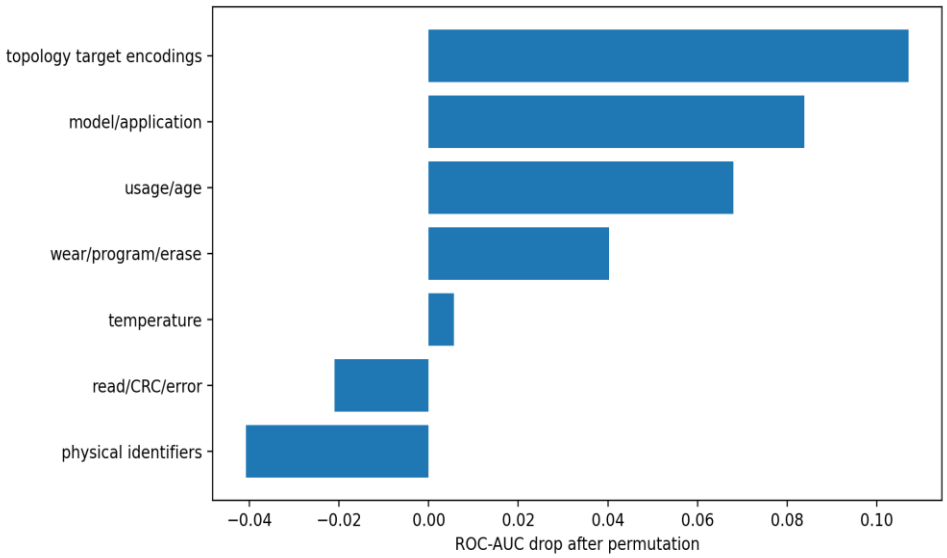

Figure 7. Grouped feature importance for the TopoHealth-ET tree component.

4.6 Evidence-Grounded Failure Explanations

Table 11 presents the evidence objects supplied to the explanation layer. Four of the five highest-risk examples are true failures; the fifth is a false positive whose wear, error, and rack signals resemble the failed cases. The explanation remains valid for that disk because it describes elevated evidence rather than asserting that failure occurred.

Table 11. Example evidence objects used by the constrained explanation layer.

Disk	Model	Rack	True failure	Risk score	Top drivers
11392	MC4	193	1	0.747	High wear-out; program/erase errors; read/CRC errors; elevated rack risk

Disk	Model	Rack	True failure	Risk score	Top drivers
9939	MC4	114	1	0.696	High wear-out; program/erase errors; read/CRC errors; elevated rack risk
13260	MB3	102	1	0.695	High wear-out; program/erase errors; elevated rack risk; narrow temperature margin
18784	MB1	102	1	0.692	High wear-out; program/erase errors; read/CRC errors; elevated rack risk
26048	MC2	102	0	0.692	High wear-out; program/erase errors; read/CRC errors; elevated rack risk

A corresponding narrative can read: “Disk 11392 is ranked high because it combines elevated wear-out, program/erase errors, read/CRC errors, and an above-baseline rack prior.” The wording is intentionally evidential. It does not claim that the rack caused the degradation or that a specific NAND mechanism has been confirmed. This distinction is central to trustworthy maintenance communication [22], [23], [34]–[40].

4.7 Operational Interpretation

The strongest result is the change in worklist yield rather than a single AUC value. At a 1% budget, the model turns a 1.90% fleet prevalence into 35.00% precision and captures 21 of 114 test failures. This is useful even though absolute recall remains limited: the model is a triage layer for earlier inspection, workload migration, firmware review, or replacement planning, not a substitute for redundancy, backups, or ordinary health monitoring.

False positives deserve operational review rather than rhetorical dismissal. A disk can exhibit strong degradation evidence without failing during the label window, and maintenance action can censor an impending event. Nevertheless, every such case is counted as a false positive in the reported metrics. This conservative treatment keeps the model comparison aligned with the observed labels while acknowledging that real failure tickets may not capture every soft-degradation episode.

The score should be used as a priority index unless recent deployment data support calibration. The experiment demonstrates ranking and queue enrichment, but a data center that needs expected failure probabilities should apply post-deployment calibration and monitor drift by model, rack, and application [21], [29]–[31].

5. Limitations

The experiment uses a deterministic benchmark aligned with the public Alibaba schemas rather than a full reconstruction of the production fleet. This design preserves column meanings, class imbalance, model diversity, and topology structure, but it cannot reproduce every temporal, firmware, vendor, workload, and maintenance process present in a real data center. The numerical findings therefore establish behavior under the stated benchmark and should be validated on current fleet records before deployment.

The main split holds out disks while allowing racks to appear in all subsets. That setting matches ongoing triage in a known fleet, but it is less conservative than holding out entire racks or machine rooms. Cold-start evaluation is especially important for new facilities, new drive models, and rapidly changing applications, where historical topology priors will shrink toward the global baseline. Leave-rack-out, leave-room-out, and time-ordered tests are natural extensions.

The study uses a last-day snapshot rather than a multi-day sequence. Snapshot models are inexpensive and compatible with the documented tables, but they cannot directly capture acceleration, repeated bursts, or the temporal ordering of SMART changes. Methods that combine long- and short-term views, NAND wear trajectories, or temporal attention may improve early warning when dense histories are available [42]–[44].

Topology priors are predictive summaries, not causal estimates. Elevated rack risk may reflect workload, cooling, firmware, maintenance practice, shared power, or correlated deployment age. The model can identify where risk is concentrated, but root cause requires additional measurements and intervention-aware analysis. The constrained explanation layer therefore reports observed drivers and avoids causal language.

Finally, the model favors transparency and modest runtime over exhaustive optimization. Larger boosted ensembles, survival models, temporal transformers, or graph neural networks may improve accuracy, and repeated permutation analysis would provide more stable importance estimates. Any extension should retain the same safeguards: train-only supervised transformations, rare-event metrics, review-budget reporting, and traceable explanation evidence.

6. Conclusion

This study developed TopoHealth-ET, a topology-aware SSD health-risk pipeline that joins SMART counters with application, model, machine-room, rack, node, and slot context. The model combines nonlinear ExtraTrees interactions with a deterministic health index and converts high-risk predictions into constrained, evidence-grounded maintenance explanations.

On the 30,000-disk schema-aligned benchmark, TopoHealth-ET achieved ROC-AUC 0.775, PR-AUC 0.203, precision 0.345, recall 0.175, F1 0.233, and a 0.646% false-alarm rate. At a 1% review budget, it captured 18.42% of failures with 35.00% precision. The results show that deployment context can materially improve the yield of a constrained maintenance queue, especially in racks where risk is concentrated.

The broader implication is that SSD health should be modeled as a joint device-and-environment problem. SMART counters describe what is happening inside the drive; topology and application features describe the context in which those counters occur. Combining the two supports more effective ranking, clearer operational triage, and explanations that remain anchored to observable evidence.

References

[1] S. Han, P. P. C. Lee, F. Xu, Y. Liu, C. He, and J. Liu, “An In-Depth Study of Correlated Failures in Production SSD-Based Data Centers,” in Proc. 19th USENIX Conf. File and Storage Technologies (FAST), pp. 417–429, 2021.

[2] F. Xu, S. Han, P. P. C. Lee, Y. Liu, C. He, and J. Liu, “General Feature Selection for Failure Prediction in Large-Scale SSD Deployment,” in Proc. 51st IEEE/IFIP Int. Conf. Dependable Systems and Networks (DSN), 2021.

[3] C. He, M. Feng, P. P. C. Lee, P. Wang, S. Han, and Y. Liu, "Large-Scale Disk Failure Prediction," in Proc. PAKDD Workshops, AI Ops, 2020.

[4] B. Schroeder and G. A. Gibson, "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?" in Proc. 5th USENIX Conf. File and Storage Technologies (FAST), 2007.

[5] E. Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure Trends in a Large Disk Drive Population," in Proc. 5th USENIX Conf. File and Storage Technologies (FAST), 2007.

[6] J. F. Murray, G. F. Hughes, K. Kreutz-Delgado, and D. Schuurmans, "Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application," J. Mach. Learn. Res., vol. 6, pp. 783–816, 2005.

[7] G. F. Hughes, J. F. Murray, K. Kreutz-Delgado, and C. Elkan, "Improved Disk-Drive Failure Warnings," IEEE Trans. Reliability, vol. 51, no. 3, pp. 350–357, 2002.

[8] L. N. Bairavasundaram, G. R. Goodson, S. Pasupathy, and J. Schindler, "An Analysis of Latent Sector Errors in Disk Drives," in Proc. ACM SIGMETRICS, 2007.

[9] L. N. Bairavasundaram, G. R. Goodson, B. Schroeder, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "An Analysis of Data Corruption in the Storage Stack," in Proc. 6th USENIX Conf. File and Storage Technologies (FAST), 2008.

[10] J. Meza, Q. Wu, S. Kumar, and O. Mutlu, "A Large-Scale Study of Flash Memory Failures in the Field," ACM SIGMETRICS Perform. Eval. Rev., vol. 43, no. 1, pp. 177–190, 2015.

[11] B. Schroeder, R. Lagisetty, and A. Merchant, "Flash Reliability in Production: The Expected and the Unexpected," in Proc. 14th USENIX Conf. File and Storage Technologies (FAST), pp. 67–80, 2016.

[12] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai, S. Ghose, and O. Mutlu, "Data Retention in MLC NAND Flash Memory: Characterization, Optimization, and Recovery," in Proc. IEEE Int. Symp. High Performance Computer Architecture (HPCA), 2015.

[13] Y. Cai, Y. Luo, S. Ghose, E. F. Haratsch, K. Mai, and O. Mutlu, "Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery," in Proc. IEEE/IFIP Int. Conf. Dependable Systems and Networks (DSN), 2015.

[14] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[15] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

[16] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. ACM SIGKDD, pp. 785–794, 2016.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," J. Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[18] H. He and E. A. Garcia, "Learning from Imbalanced Data," IEEE Trans. Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[19] T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.

[20] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot when Evaluating Binary Classifiers on Imbalanced Datasets," PLOS ONE, vol. 10, no. 3, 2015.

[21] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in Proc. Int. Conf. Machine Learning (ICML), 2017.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier," in Proc. ACM SIGKDD, pp. 1135–1144, 2016.

[23] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Proc. Neural Information Processing Systems (NeurIPS), 2017.

[24] A. Vaswani et al., "Attention Is All You Need," in Proc. Neural Information Processing Systems (NeurIPS), 2017.

[25] T. B. Brown et al., "Language Models Are Few-Shot Learners," in Proc. Neural Information Processing Systems (NeurIPS), 2020.

[26] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. Neural Information Processing Systems (NeurIPS), 2020.

[27] J. Jin, T. Huang, and S. Lu, "Cost-Sensitive Learning, Simulated PU Learning, and One-Class

Autoencoding for Extreme-Imbalance Credit Card Fraud Detection," JACS, vol. 4, no. 6, pp. 64–73, 2024, doi: 10.69987/JACS.2024.40605.

[28] Y. Chen, Y. Zhang, and M. Sherman, "Going Concern and Bankruptcy Prediction under Extreme Class Imbalance: Cost-Sensitive Learning, Resampling, and Focal Loss with Explainable Financial-Ratio Portraits," JACS, vol. 4, no. 4, pp. 80–96, 2024, doi: 10.69987/JACS.2024.40407.

[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," JACS, vol. 3, no. 4, pp. 31–47, 2023, doi: 10.69987/JACS.2023.30403.

[30] J. 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," JACS, vol. 4, no. 6, pp. 74–85, 2024, doi: 10.69987/JACS.2024.40606.

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

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

[33] 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. Technology Informatics and Engineering, vol. 4, no. 3, pp. 544–571, 2025, doi: 10.51903/jtie.v4i3.498.

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

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

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

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

[38] Q. Xin, "Self-Supervised Log Anomaly Detection with LogBERT-Style Transformers: Full Empirical Evaluation on a Reproducible SynHDFS Benchmark," J. Electrical Engineering and Computer Science, vol. 11, no. 1, pp. 23–35, 2026, doi: 10.54732/jeecs.v11i1.3.

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

[40] 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. Graphic Design, vol. 4, no. 1, pp. 179–185, 2026, doi: 10.51903/ijgd.v4i1.3703.

[41] 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 2025 6th International Conference on Electronic Communication and Artificial Intelligence (ICECAI), Chengdu, China, 2025, doi: 10.1109/ICECAI66283.2025.11171441.

[42] Y. Zhang, W. Hao, B. Niu, K. Liu, S. Wang, N. Liu, X. He, Y. Gwon, and C. Koh, "Multi-view Feature-based SSD Failure Prediction: What, When, and Why," in Proc. 21st USENIX Conf. File and Storage Technologies (FAST), pp. 409–424, 2023.

[43] Y. Gu, C. Wu, and X. He, "Exploit both SMART Attributes and NAND Flash Wear Characteristics to Effectively Forecast SSD-Based Storage Failures in Clusters," in Proc. USENIX Annual Technical Conf. (ATC), pp. 1101–1117, 2024.

[44] C. Koh, J. Kang, T. Kim, and S. W. Han, "Temporal-Contextual Attention Network for Solid-State Drive Failure Prediction in Data Centers," IEEE Access, vol. 12, pp. 154455–154466, 2024, doi: 10.1109/ACCESS.2024.3482368.