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

LLM-Explainable Consumer SSD Health Scoring with SMART Attributes and Reliability Statistics

David Chao

Computer Science, UCLA, CA, USA
david.chao0817@gmail.com

DOI: 10.69987/JACS.2026.60702

Keywords

SMART attributes; SSD health scoring; NVMe reliability; LinuxHW; explainable AI; language-model explanations; calibrated ensembles; MTBF; consumer storage.

Abstract

Reliable consumer SSD and NVMe assessment requires a model that distinguishes observed SMART evidence from population context and communicates risk without presenting a diagnostic certainty that the data cannot support. This paper presents LESHS, an explainable health-scoring framework built on the LinuxHW SMART Repository. The analysis processed 162,383 rows from the All_SSD.md and All_NVMe.md appendices, removed one exact duplicate, and retained 162,382 records: 81,660 SATA SSD samples and 80,722 NVMe samples. The repository-defined important-error flag was positive for 5.09% of retained records, with rates of 8.64% for SSD and 1.49% for NVMe. A strict non-leakage experiment withheld each record's Err and MTBF fields and predicted the important-error flag from drive type, power-on exposure, capacity, and training-set manufacturer, model, and capacity-bucket priors. On a stratified test set of 24,358 records, the calibrated LESHS model achieved ROC-AUC 0.8873, PR-AUC 0.3892, F1 0.4066, and Brier score 0.0390. A separate consistency analysis reconstructed the LinuxHW MTBF statistic with MAE 0.00255 years and R2 0.99999. Operational scoring then combined the calibrated prior with the observed important-error count and an explicit penalty for short observation windows. A constrained explanation schema translated the resulting evidence into concise statements about error presence, power-on duration, MTBF, and score band. The findings show that calibrated population priors and direct SMART evidence play complementary roles in consumer-drive health assessment.

1. Introduction

Solid-state drives have become the principal client-storage medium in laptops, desktops, small servers, and edge systems. Their practical appeal does not remove the need for a clear health assessment: users still need to decide whether a drive can remain in service, requires closer monitoring, should be backed up immediately, or should be replaced. SMART telemetry is the most accessible evidence for that decision, yet the fields exposed by ATA and NVMe devices vary by protocol, firmware, and vendor implementation [8]-[11]. Rule-based utilities can summarize individual counters [7], but they do not resolve the broader problem of combining heterogeneous evidence into a calibrated and understandable score.

Storage-reliability research has repeatedly shown that telemetry is informative but incomplete. Early large-population studies connected reallocated, pending, scan, and uncorrectable events with elevated failure risk while also demonstrating that no single counter provides a universal prediction

rule [2], [3]. Operational reports from Backblaze similarly identified nonzero SMART error counters as useful warning evidence across large drive populations [4]-[6]. Flash-specific field studies later documented the influence of device family, wear, firmware, workload, and system context on SSD reliability [48]-[51], while large-scale SMART prediction work showed that multivariate models can extract stronger signal than isolated thresholds [52].

The LinuxHW SMART Repository provides a complementary perspective because it aggregates community-contributed reports from desktop-class consumer drives rather than a centrally managed enterprise fleet [1]. Its SSD and NVMe appendices expose drive type, manufacturer, model, capacity, anonymized drive identifier, average power-on days, an aggregate important-error count, and an MTBF statistic. The public tables do not contain longitudinal failure timestamps or a complete vendor-normalized set of raw SMART attributes. They nevertheless support a well-defined task: estimating the probability that a held-out record contains repository-defined important-error evidence, and then summarizing a known record once its observed evidence is available.

The modeling problem has two distinct stages. The first is strict prediction, where each held-out record's own Err and MTBF fields must be excluded to prevent target leakage. The second is operational scoring, where the user already possesses the drive's SMART report and the observed error count should directly influence the result. Linear models, decision trees, random forests, extremely randomized trees, and gradient boosting provide complementary baselines for the first stage [12]-[16]. Because the positive class is rare, ROC-AUC alone is insufficient; PR-AUC, F1, and probability calibration are also required [18]-[20].

The communication problem is equally important. A numerical score is difficult to trust when it cannot be connected to the evidence that produced it. Explainable-AI research emphasizes fidelity, audience relevance, and explicit limitations [21]-[26]. Large language models can improve readability and adapt an explanation to a user's vocabulary [27]-[30], but unconstrained generation can also introduce unsupported claims. Recent evidence-grounding and self-checking work therefore favors source attribution, consistency checks, and bounded generation [31], [36]. LESHS follows that direction: the scoring pipeline remains numerical and deterministic, while the explanation interface receives a fixed evidence record and is not permitted to invent vendor-specific attributes or a future failure date.

This study contributes a complete consumer-drive assessment pipeline with three linked components. First, it establishes a non-leakage benchmark on 162,382 LinuxHW SSD and NVMe records. Second, it combines a calibrated population prior with observed SMART evidence and exposure-aware uncertainty in an operational health score. Third, it defines a grounded explanation schema whose statements can be traced to the record, the LinuxHW reliability formula, or the fitted model. The result is a practical health indicator that preserves the distinction between observed degradation evidence and uncertainty caused by limited observation.

2. Literature Review

2.1 SMART telemetry and flash-storage reliability

SMART-based reliability research began largely with hard-disk populations. Pinheiro et al. [2] and Schroeder and Gibson [3] showed that several error counters correlate with failures, but also found substantial overlap between failed and surviving drives. That result remains methodologically important: a nonzero error count is meaningful evidence, yet the absence of an error does not establish that a device is risk free. Backblaze analyses [4]-[6] extended the operational evidence base by publishing fleet-level observations and raw data, while smartmontools [8] made low-level telemetry broadly accessible to users and administrators.

SSD reliability introduces additional complexity because the media, controller, error-correction behavior, firmware, overprovisioning, and workload all interact. Meza et al. [51] examined flash-memory failures in a large field population, Schroeder et al. [48] reported several departures from commonly

assumed wear-out behavior, and Maneas et al. [49] demonstrated substantial variation across manufacturers, models, capacities, flash technologies, and firmware versions in enterprise storage systems. Xu et al. [50] further showed that SSD-related system failures cannot always be understood from device counters alone because operating-system and service interactions also contribute to incidents.

The standards and tools surrounding these studies provide definitions rather than a universal consumer-health score. NVMe specifies SMART and health-log information for nonvolatile memory devices [9], and JEDEC defines endurance requirements and workloads for SSD qualification [10], [11]. Acronis-style disk-health calculations illustrate how rules can convert selected counters into a user-facing percentage [7]. Such rules are useful when attribute semantics are stable, but heterogeneous consumer repositories require explicit handling of missing fields, vendor variation, observation duration, and the amount of evidence behind population-level priors.

LinuxHW offers a broad consumer-oriented snapshot rather than a controlled fleet [1]. Its aggregate Err field normalizes a set of important SMART indicators into a common count, making cross-device analysis feasible. At the same time, the aggregation removes attribute-level detail. This tradeoff motivates a framework that treats Err as observed evidence when scoring a known drive, while evaluating predictive models on features that remain available after Err and its derivative MTBF field are withheld.

2.2 Predictive modeling for storage and infrastructure health

Classical statistical learning provides a strong foundation for reliability scoring. Logistic regression offers an interpretable linear baseline, tree ensembles capture nonlinear interactions, gradient boosting provides an alternative additive ensemble, and ridge regression stabilizes correlated predictors [12]-[17]. Random forests [13] and extremely randomized trees [14] are especially suitable when exposure, capacity, technology type, and smoothed category priors interact in non-monotonic ways. Scikit-learn supplies consistent implementations and evaluation interfaces for these models [16].

Large-scale disk prediction confirms the value of combining multiple SMART signals. Lu et al. [52] trained failure-prediction models across a large data-center population and reported substantially stronger performance than would be expected from isolated thresholds. The LinuxHW setting differs because it lacks daily time series and confirmed failure events. Consequently, the present target is an important-error flag rather than future failure, and train-set population priors replace temporal trends that are unavailable in the public appendices.

Related AIOps research illustrates how heterogeneous operational evidence can support anomaly detection and diagnosis. Ambiguity-aware HDFS analysis combines anomaly detection with retrieval-grounded failure narratives and selective refusal [32]. Multi-source root-cause attribution integrates CPU and network evidence across microservices [33], while cloud-native RAG and incident-triage systems connect private operations documents, alerts, and explanatory summaries [34], [35]. These studies support the separation of a numerical detector from an explanatory layer, although their log and document inputs differ from the tabular SMART evidence considered here.

Recent infrastructure-health work also emphasizes robustness under weak labels and distribution shift. Noisy-neighbor-aware VM degradation modeling uses unsupervised residual fusion to isolate performance risk [44]. Self-supervised log transformers can learn anomaly structure without exhaustive labels [45], and conformal alert control can translate anomaly scores into risk-bounded incident tickets [46]. Evidence-constrained incident cards then organize anomaly scores, supporting logs, confidence, and recommended actions for human review [47]. LESHS adopts the same operational principle: a health score should expose the evidence and uncertainty that matter for action, not merely return a class label.

2.3 Probability calibration, uncertainty, and selective action

Imbalanced reliability tasks require metrics that reflect both ranking and decision quality. ROC-AUC measures pairwise ranking across thresholds [18], whereas PR-AUC focuses attention on positive-class retrieval and is more sensitive to class prevalence [19]. F1 summarizes a particular operating point, and the Brier score evaluates the accuracy of predicted probabilities. Platt scaling provides a simple monotonic calibration layer that can be fitted on validation data without changing the underlying ranking [20].

Cross-domain risk studies reinforce the importance of calibrated probabilities and explicit abstention or rejection. Credit-risk tiering has combined probability calibration with uncertainty-driven rejection [37], and model-risk-oriented default workflows have paired calibration with distribution-free uncertainty and feature explanations [38]. Explanation-enhanced gradient-boosting systems similarly show that predictive strength and communication quality should be evaluated as separate design goals [39]. Human-uncertainty distillation [40] and confidence-calibrated multimodal fusion [41] further demonstrate that a high raw score is not interchangeable with a well-calibrated probability.

Operational resource forecasting reaches the same conclusion from another domain. Calibrated uncertainty has been used to bound capacity forecasts in heterogeneous GPU clusters [42], and conformal demand envelopes have been used to control oversubscription risk [43]. These approaches are relevant to consumer-drive scoring because short observation histories create epistemic uncertainty even when no error has yet been observed. LESHS represents that condition with an explicit low-evidence penalty rather than interpreting a young, error-free record as strongly healthy.

2.4 Evidence-grounded explanations for operational decisions

Model explanations must be faithful to the computation and understandable to the intended audience. Local surrogate explanations [21], Shapley-based attribution [22], social-science accounts of explanation [23], the DARPA XAI program [24], the NIST AI Risk Management Framework [25], and model-card reporting [26] all emphasize traceability, transparency, and appropriate qualification. For a storage-health tool, those principles imply that the explanation should identify the observed error evidence, the amount of exposure, the relevant population context, and the limits of the target label.

General-purpose language models can produce fluent summaries and follow structured instructions [27]-[30]. Fluency alone, however, does not establish evidential correctness. A lightweight hallucination firewall can compare generated statements with available evidence and apply self-checking [31]. Evidence-chain RAG extends this idea with source attribution and provenance explanations [36], while financial RAG systems use numerical checks to reduce unsupported claims in corporate risk memos [53], [54]. These methods are directly relevant to health scoring because numerical fields such as Err, Days, MTBF, and the score band must not be altered during verbalization.

Grounded explanation has also become an interface-design problem. Explanation-enhanced risk models [39] show how readable narratives can accompany calibrated predictions, and evidence-constrained incident cards [47] organize confidence, supporting observations, and actions into a compact human-review format. LESHS therefore exposes a fixed evidence tuple rather than an unrestricted prompt. A language model may paraphrase the tuple in a deployment, but the evaluation uses deterministic sentence templates so that explanation fidelity can be inspected independently of generation style.

The literature leaves a specific gap at the intersection of consumer storage, probability calibration, and evidence-grounded explanation. Existing SSD field studies provide strong reliability insights but are usually based on controlled enterprise fleets [48]-[52]. Operational AIOps systems provide grounded narratives but target logs, alerts, or documents [32]-[36], [44]-[47]. LESHS connects these strands for the public LinuxHW consumer-drive tables, while

maintaining a clear boundary between important-error evidence and future physical failure.

3. Method

3.1 Data source and study target

The analysis used the LinuxHW SMART Repository appendices All_SSD.md and All_NVMe.md [1]. The SSD appendix contained 81,661 rows and the NVMe appendix contained 80,722 rows. The parser retained the seven public fields MFG, Model, Size, Drive ID, Days, Err, and MTBF, removed one exact duplicate, and produced a final table of 162,382 records. Drive ID was excluded from modeling because it is an anonymized identifier and repeated truncated values occur in a small subset of rows.

The main label was y = 1 when Err > 0 and y = 0 otherwise. This target represents repository-defined important SMART evidence; it is not a manufacturer-confirmed failure event. LinuxHW defines its MTBF statistic as power-on hours divided by one plus the important-error count and then converted to years. With exposure expressed in days, the equivalent formula is MTBF = Days / (365 x (1 + Err)). The consistency analysis compared this reconstructed value with the published MTBF column.

3.2 Preprocessing and population-prior features

Capacity strings ending in KB, MB, GB, TB, or PB were converted to gigabytes. Twenty-nine NVMe rows contained truncated capacity text consisting of a visible numeric prefix followed by ellipses. These rows were retained with a capacity-missing indicator and median capacity imputation. The affected share was 0.018% of the final dataset, and the procedure did not modify Days, Err, or MTBF. Power-on exposure was represented by Days and log(1 + Days).

Population context was derived from the training partition only. For a category c, the smoothed risk prior was (positive_count_c + m x global_positive_rate) / (sample_count_c + m). The smoothing constant was 100 for manufacturer, 20 for model, and 100 for capacity bucket. The feature set also included log-transformed training counts for manufacturer, model, and capacity bucket so that the classifier could distinguish a prior supported by many records from an equally large prior supported by only a few records.

3.3 Strict non-leakage prediction protocol

Rows were divided into stratified training, validation, and test partitions containing 113,667, 24,357, and 24,358 records. The test set contained 1,239 positive records, corresponding to a positive rate of 5.087%. The held-out record's own Err and MTBF fields were excluded from every strict predictor because Err defines the target and MTBF is computed from Err and Days.

The evaluated predictors were POH-only logistic regression; drive-type, capacity, and POH logistic regression; a direct train-set prior score; all-feature logistic regression; a class-balanced decision tree; a class-balanced random forest; Extra Trees; and the LESHS strict calibrated model. The ensemble choices follow the established ability of random forests and extremely randomized trees to capture nonlinear feature interactions [13], [14]. The LESHS strict model applied Platt scaling to the Extra Trees probability using the validation partition [20]. Its decision threshold was selected on validation data to maximize F1 and was then held fixed for the test evaluation.

3.4 Operational LESHS health score

Operational scoring begins after the strict model produces a calibrated prior probability p_i. At this stage, the observed important-error count is reintroduced because a health score for a known drive should respond directly to its SMART evidence. The error component is e_i = 1 - exp(-0.85 x log(1 + Err_i)). A low-evidence component, u_i = 0.20 x exp(-Days_i / 90), lowers confidence for short observation windows without treating limited history as a SMART error.

The combined risk is risk_i = clip(1 - (1 - p_i)(1 - e_i) + u_i, 0, 1), and the health score is Score_i = 100 x (1 - risk_i). Scores below 40 are labeled critical, 40-59 degraded, 60-74 watch, 75-89 good, and 90-100 excellent. These thresholds are engineering bands for interpreting LinuxHW evidence rather than estimates of a physical failure deadline.

3.5 Grounded explanation schema

Each explanation receives a fixed record containing drive type, vendor and model, observed important-error count, power-on days, MTBF, calibrated prior, operational score, score band, and the most influential strict-model features. The permitted statements describe whether important-error evidence is present, whether the observation window is short or long, how the MTBF statistic relates to the observed fields, and what action band the score occupies. The schema does not authorize a predicted failure date, a causal diagnosis, or a vendor-specific attribute that is absent from the LinuxHW table.

The numerical score remains independent of the language layer. For the reported examples, deterministic sentence templates render the evidence record in natural language. This arrangement follows the fidelity and traceability principles in explainable-AI guidance [21]-[26] and the evidence-consistency mechanisms used in grounded generation [31], [36], [53], [54]. A deployed interface can allow a language model to paraphrase the same record while retaining field-level validation.

3.6 Evaluation and statistical reporting

All classifiers used the same training, validation, and test partitions. ROC-AUC, PR-AUC, F1, precision, recall, accuracy, Brier score, and confusion-matrix counts were computed on the held-out test set [18], [19]. The ablation study progressively introduced drive type, capacity, POH, manufacturer priors, model priors, and the complete feature set. Subgroup evaluation separated SSD from NVMe and capacities below 512 GB from capacities at or above 512 GB.

A separate MTBF consistency experiment used Days, log Days, Err, log Err, imputed capacity, and drive type. It compared the published formula with a POH-only baseline, ridge regression [17], and an Extra Trees regressor. The formula baseline tests whether parsing and unit conversion align with the repository definition; it is not treated as an independently learned reliability law. Figure 1 summarizes the data flow from the public appendices to strict prediction, operational scoring, and grounded explanation.

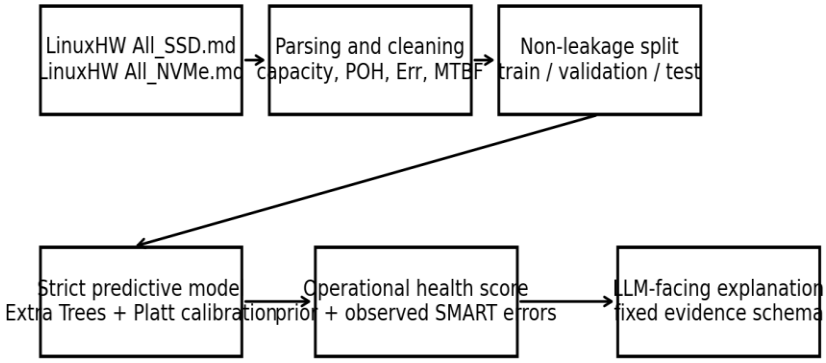

Figure 1. Data ingestion, strict prediction, operational scoring, and grounded explanation workflow.

Table 1. Cleaned LinuxHW SSD/NVMe dataset summary.

Drive type	Samples	Manufacturers	Models	Median days	Mean days	Median MTBF	Mean MTBF	Error-positive samples	Error rate	Missing capacity
NVMe	80722	177	2079	46	148.1528	0.12	0.3958	1201	0.0149	29
SSD	81660	334	2782	183	389.2298	0.41	0.9616	7058	0.0864	0

4. Results and Discussion

4.1 Dataset composition and descriptive reliability evidence

Table 1 shows a nearly balanced number of SSD and NVMe records but markedly different exposure profiles. SSD records have a median of 183 power-on days and an important-error rate of 8.64%, whereas NVMe records have a median of 46 days and an error rate of 1.49%. Across both technologies, 8,259 of 162,382 records are positive. Figure 2 visualizes this imbalance and shows why PR-AUC and threshold-dependent metrics are necessary alongside ROC-AUC.

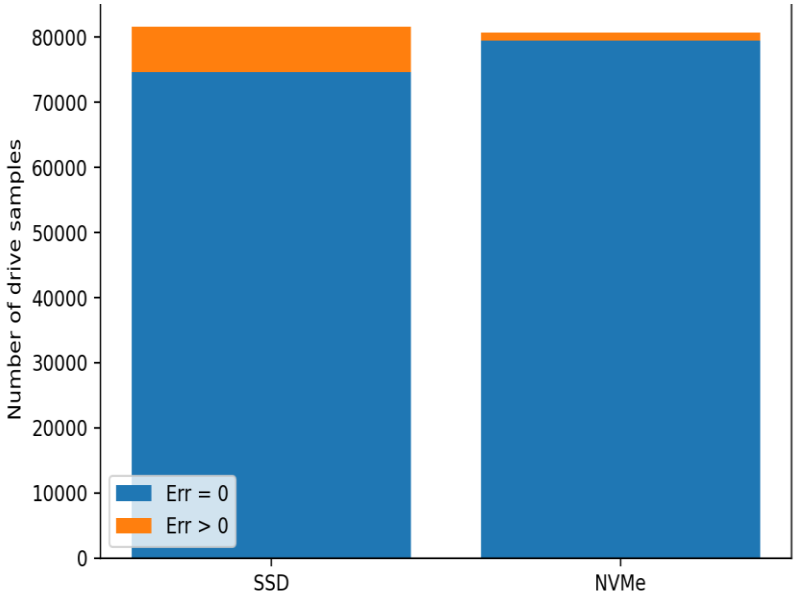

Figure 2. SSD and NVMe observations split by repository-defined important-error flag.

Capacity also separates distinct exposure and error patterns. Table 2 reports the sample count, important-error rate, and median MTBF for each drive-type and capacity bucket. SSDs below 128 GB have the highest SSD error rate in the table, 11.86%, while SSDs at or above 4 TB have the lowest, 2.50%. NVMe error rates remain lower across all nonmissing capacity buckets, but their shorter observation histories limit direct reliability comparisons between technologies.

Table 2. Capacity-bucket distribution, error rate, and median MTBF.

Drive type	Capacity bucket	Samples	Error rate	Median MTBF
NVMe	1-2TB	24640	0.0169	0.1
NVMe	128-255GB	4024	0.0231	0.16
NVMe	2-4TB	8619	0.0132	0.09
NVMe	256-511GB	18614	0.018	0.15
NVMe	512-1023GB	22463	0.0095	0.12
NVMe	<128GB	668	0.021	0.205
NVMe	>=4TB	1665	0.009	0.14
NVMe	missing	29	0	0.05
SSD	1-2TB	9759	0.0576	0.41
SSD	128-255GB	24748	0.0922	0.41
SSD	2-4TB	2623	0.0427	0.37
SSD	256-511GB	23257	0.0776	0.42
SSD	512-1023GB	6355	0.0946	0.28
SSD	<128GB	14159	0.1186	0.48
SSD	>=4TB	759	0.025	0.53

The evidence strata in Table 3 provide a descriptive view of observed Err and MTBF rather than a replacement for the LESHS score. Low-evidence records dominate the NVMe population: 74.69% of NVMe rows are in the C stratum, compared with 46.22% of SSD rows. Critical evidence appears in 7.52% of SSD rows and 1.38% of NVMe rows. These strata are distinct from the calibrated operational score bands reported later in Table 8.

Table 3. Descriptive operational evidence strata derived from Err and MTBF.

Drive type	Evidence stratum	Samples	Share
NVMe	A: stable evidence	2963	0.0367
NVMe	B: normal	16267	0.2015
NVMe	C: low evidence	60291	0.7469
NVMe	D: degraded	88	0.0011
NVMe	E: critical	1113	0.0138
SSD	A: stable evidence	12139	0.1487
SSD	B: normal	24722	0.3027
SSD	C: low evidence	37741	0.4622
SSD	D: degraded	917	0.0112
SSD	E: critical	6141	0.0752

Figure 3 displays the strongly right-skewed MTBF distribution. The concentration near zero reflects both positive Err counts and short power-on histories, while the long tail represents records with extended exposure and little or no important-error evidence. The difference in technology age remains visible: NVMe contributes more mass near the origin, whereas SSD records extend farther into the distribution.

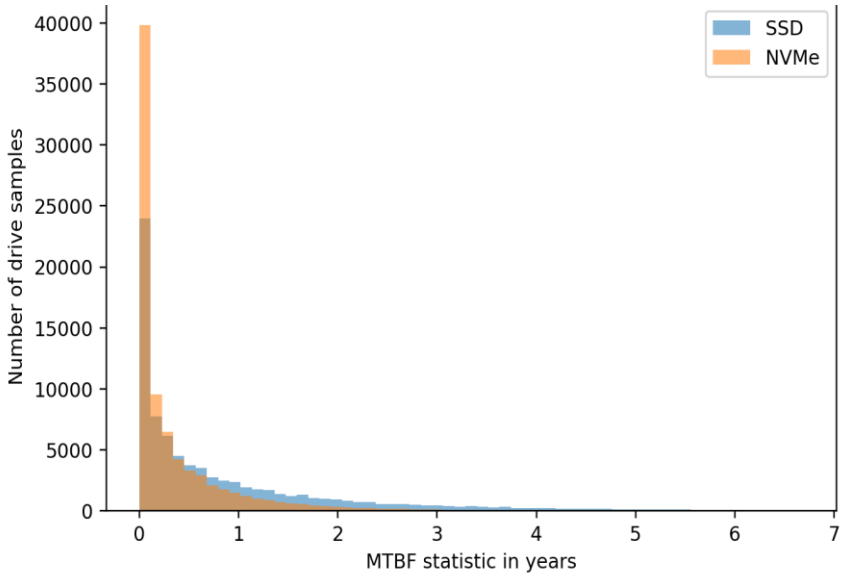

Figure 3. Empirical LinuxHW MTBF distribution for SSD and NVMe records. 4.2 Strict model performance

Table 4 compares all strict predictors on the untouched test set. POH-only logistic regression achieves ROC-AUC 0.6613 and PR-AUC 0.0949. Adding drive type and capacity raises ROC-AUC to 0.7475. The direct train-set prior score reaches ROC-AUC 0.8593 and PR-AUC 0.3131, demonstrating that manufacturer and model history carries substantial signal even before nonlinear interactions are introduced. Random forest produces the highest PR-AUC, 0.3950, while the calibrated LESHS model reaches the highest reported ROC-AUC, 0.8873, and the lowest Brier score, 0.0390.

Table 4. Held-out model comparison for strict important-error prediction.

Model	Validation threshold	ROC-AUC	PR-AUC	F1	Precision	Recall	Accuracy	Brier	TP	FP	TN	FN
POH-only logistic	0.59	0.6613	0.0949	0.1454	0.0884	0.4092	0.7554	0.2328	507	5227	17892	732
Kind+capacity+POH logistic	0.65	0.7475	0.1244	0.1887	0.1156	0.5125	0.7758	0.2101	635	4857	18262	604
Train-set prior score	0.0888	0.8593	0.3131	0.3604	0.2867	0.4851	0.9124	0.0431	601	1495	21624	638
Logistic regression	0.84	0.8722	0.3266	0.3568	0.3301	0.3882	0.9288	0.1286	481	976	22143	758

Model	Validation threshold	ROC-AUC	PR-AUC	F1	Precision	Recall	Accuracy	Brier	TP	FP	TN	FN
Decision tree	0.8814	0.8431	0.3545	0.3831	0.4252	0.3487	0.9429	0.1279	432	584	22535	807
Random forest	0.83	0.8743	0.395	0.4096	0.3992	0.4205	0.9383	0.1081	521	784	22335	718
Extra Trees	0.83	0.8873	0.3892	0.3859	0.4202	0.3567	0.9422	0.1169	442	610	22509	797
LESHS strict calibrated model	0.25	0.8873	0.3892	0.4066	0.3658	0.4576	0.9321	0.039	567	983	22136	672

The validation-selected LESHS threshold of 0.25 produces 567 true positives, 983 false positives, 22,136 true negatives, and 672 false negatives. Its F1 of 0.4066 reflects a watchlist-oriented operating point that favors greater recall than the uncalibrated Extra Trees threshold. Calibration does not change the ROC or PR ranking because Platt scaling is monotonic, but it substantially improves probability interpretation. Figure 4 confirms that the largest gain comes from adding population priors; the ensemble models then refine the ranking through nonlinear interactions.

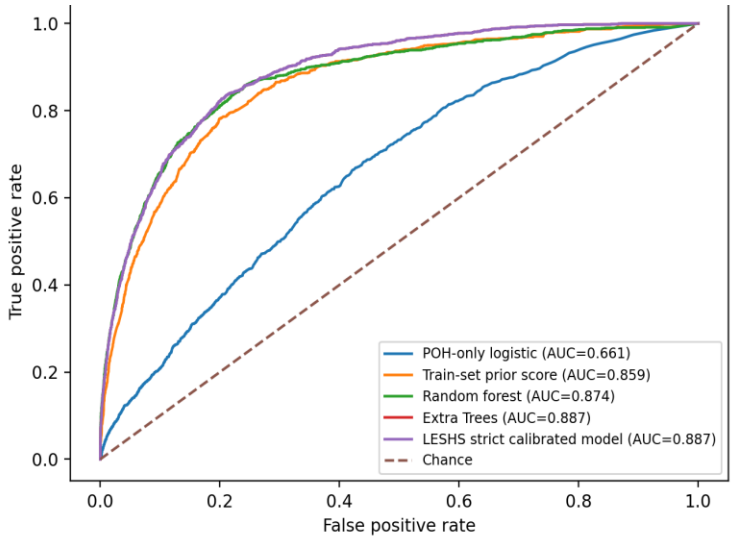

Figure 4. ROC curves for strict held-out important-error prediction.

4.3 Ablation and feature importance

The ablation results in Table 5 isolate the contribution of each feature group. Drive type alone mainly separates the higher-positive-rate SSD population from NVMe and therefore has high recall but low precision. Adding capacity and POH increases ROC-AUC from 0.6829 to 0.7693. Manufacturer priors raise PR-AUC to 0.2025, and model priors produce the largest incremental gain, raising PR-AUC to 0.3579. The complete strict feature set reaches PR-AUC 0.3843 in the ablation run.

Table 5. Ablation study for strict LESHS feature groups.

Model	ROC-AUC	PR-AUC	F1	Precision	Recall	Brier
Type only	0.6829	0.0808	0.1565	0.0862	0.8467	0.2199
Type + capacity	0.7478	0.1327	0.1963	0.1166	0.619	0.209
Type + capacity + POH	0.7693	0.1477	0.2154	0.1352	0.5303	0.2015
+ manufacturer prior	0.8206	0.2025	0.2802	0.2324	0.3527	0.1764
+ model prior	0.8841	0.3579	0.3903	0.3127	0.519	0.1274
Complete LESHS strict features	0.8838	0.3843	0.3834	0.4021	0.3664	0.1218

Figure 5 is consistent with the ablation study. The model-level risk prior is the strongest feature, followed by drive type, log power-on days, model training count, and manufacturer risk prior. Capacity-bucket context contributes additional signal, while the capacity-missing indicator has negligible importance because only 29 records require imputation. The ordering indicates that LESHS relies on both expected risk and the amount of evidence behind that expectation.

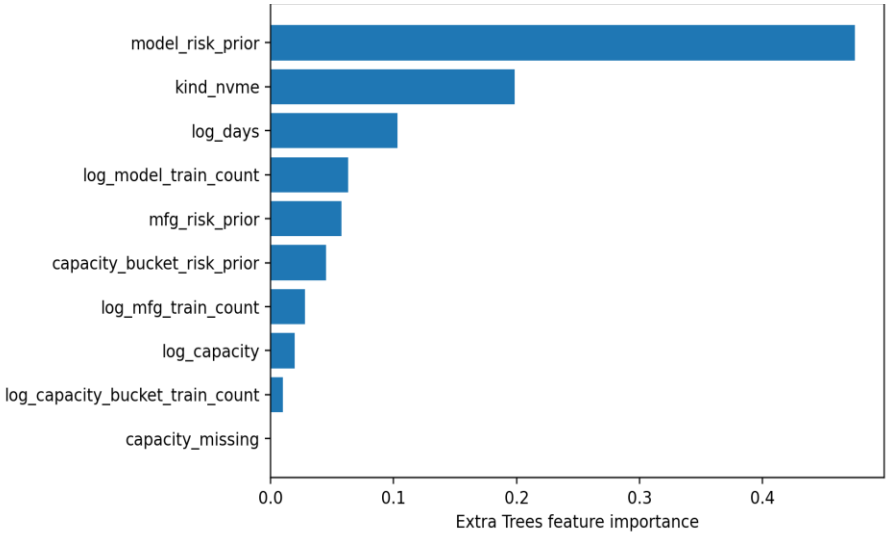

Figure 5. Feature importance for the Extra Trees strict model.

4.4 Subgroup behavior and calibration

Table 6 shows that performance differs with class prevalence and technology. SSD records achieve PR-AUC 0.4230 at a positive rate of 8.62%. NVMe records retain a high ROC-AUC of 0.8827 but have PR-AUC 0.1781 because only 1.56% are positive. Capacity below 512 GB reaches PR-AUC 0.4324, compared with 0.2511 for larger devices. These differences illustrate why subgroup PR-AUC must be interpreted relative to the subgroup base rate rather than compared as an absolute reliability ranking.

Table 6. Subgroup performance of the calibrated LESHS strict model.

Subgroup	N	Positive rate	ROC-AUC	PR-AUC	F1	Precision	Recall
SSD	12166	0.0862	0.8462	0.423	0.4287	0.3688	0.5119
NVMe	12192	0.0156	0.8827	0.1781	0.2113	0.3191	0.1579

Subgroup	N	Positive rate	ROC-AUC	PR-AUC	F1	Precision	Recall
Capacity < 512 GB	12783	0.0738	0.8693	0.4324	0.4285	0.3767	0.4968
Capacity >= 512 GB	11575	0.0255	0.8864	0.2511	0.3267	0.3213	0.3322

Figure 6 plots the calibration curve for the strict LESHS probability. The observed error fraction remains close to the diagonal across the populated probability bins. The curve is based on a highly imbalanced dataset, so the sparsely populated upper range should not be interpreted as evidence of precision at probabilities for which few records exist. The Brier score in Table 4 provides the aggregate calibration comparison.

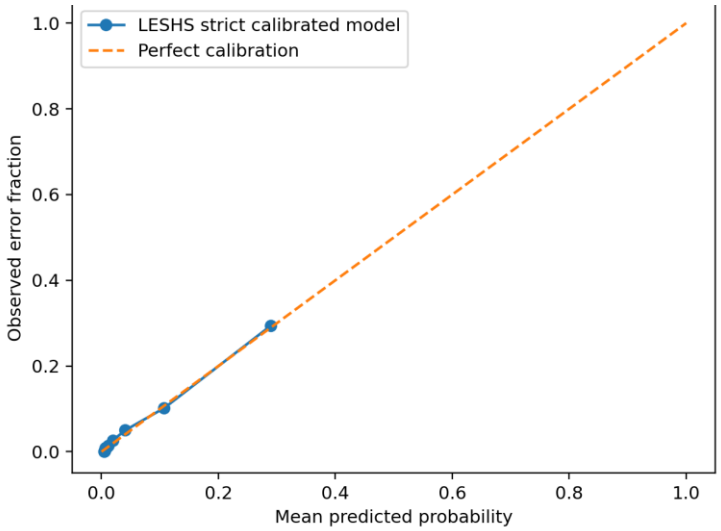

Figure 6. Calibration curve for the LESHS strict calibrated model.

4.5 MTBF consistency

Table 7 verifies the relationship between the parsed fields and the public MTBF statistic. The LinuxHW formula Days/(365 x (1 + Err)) attains MAE 0.0026 years, RMSE 0.0032 years, and R2 equal to 1.0000 at the displayed precision. The remaining discrepancy is consistent with rounding in the public column. Extra Trees also fits the relationship closely, but the formula is the correct interpretation because MTBF is deterministically defined from Days and Err. The weaker ridge result further demonstrates that treating MTBF as an independent prediction target would obscure the underlying formula.

Table 7. MTBF consistency and regression results.

Model	MAE (years)	RMSE (years)	R2
POH/365 only	0.0618	0.4411	0.8531
Published LinuxHW formula Days/(365*(1+Err))	0.0026	0.0032	1
Ridge regression	0.1065	0.3418	0.9118
Extra Trees regressor	0.0025	0.0197	0.9997

4.6 Operational scores, vendor context, and explanations

Operational scoring combines the calibrated strict prior with the record's observed Err value and the short-history penalty. Table 8 shows that most NVMe test records fall in the good or excellent bands because the calibrated prior is low and Err is zero, while 0.94% remain in watch because the available exposure is limited. SSD records have larger critical, degraded, and watch shares, consistent with their older age distribution and higher observed important-error rate.

Table 8. Held-out operational health-score band distribution.

Drive type	Score band	Test samples	Share within type
NVMe	critical	161	0.0132
NVMe	degraded	31	0.0025
NVMe	excellent	4668	0.3829
NVMe	good	7217	0.5919
NVMe	watch	115	0.0094
SSD	critical	995	0.0818
SSD	degraded	325	0.0267
SSD	excellent	5109	0.4199
SSD	good	4460	0.3666
SSD	watch	1277	0.105

Figure 7 shows the full held-out score distribution. The NVMe distribution is concentrated between approximately 78 and 100, whereas the SSD distribution extends across the full score range. The small cluster near zero corresponds to records with substantial observed important-error evidence. The distribution is intentionally not interpreted as a future-failure curve; it is a summary of current LinuxHW evidence and population context.

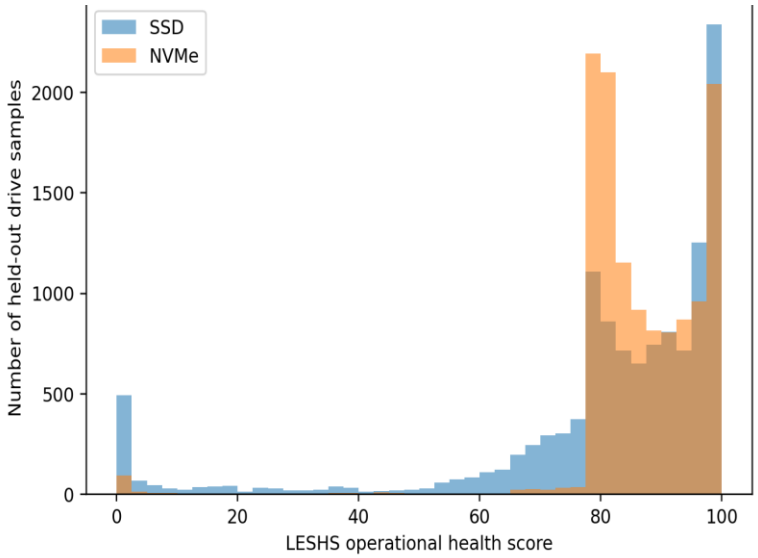

Figure 7. Held-out operational LESHS score distribution by drive type.

Table 9 provides descriptive manufacturer-level error rates for groups with at least 100 records. The table should not be read as a universal vendor ranking. Repository composition, model mix, device age, workload, and the likelihood of contributing a report after a problem can all influence the observed rates. Manufacturer and model priors improve prediction within the dataset, but the operational explanation continues to display the record's own error evidence and exposure rather than treating the prior as a verdict.

Table 9. Descriptive manufacturer error rates among groups with at least 100 records.

Drive type	Manufacturer	Samples	Error rate	Median MTBF
NVMe	XPG	506	0.0751	0.2
NVMe	Lenovo	147	0.0748	0.3
NVMe	Lite-On	188	0.0745	0.125
NVMe	KingSpec	214	0.0654	0.04
NVMe	ADATA	1880	0.059	0.14
NVMe	HP	256	0.0586	0.4
NVMe	SPCC	620	0.05	0.22
NVMe	Netac	216	0.0463	0.08
NVMe	Team	353	0.0397	0.14
NVMe	Hikvision	106	0.0377	0.105
SSD	SK hynix	1270	0.274	0.37
SSD	Intel	2513	0.2587	0.71
SSD	Corsair	442	0.2489	1.005
SSD	Micron	1415	0.2085	0.25
SSD	OCZ	1100	0.2009	1.2
SSD	Mushkin	137	0.146	0.27
SSD	Lite-On	891	0.1414	0.14
SSD	SanDisk	6462	0.1261	0.41
SSD	ADATA	2697	0.1175	0.2
SSD	Netac	565	0.115	0.05

Table 10 illustrates four explanation patterns. Long-history, error-free records receive statements that identify both the absence of important-error evidence and the available exposure. Young, error-free records explicitly disclose the limited history. Records with nonzero Err values cite the observed count and MTBF statistic rather than substituting a vague warning. The examples therefore preserve a direct correspondence between each sentence and a field used by the operational score.

Table 10. Grounded LESHS explanation examples from held-out records.

Case	Drive type	Vendor	Model	Days	Err	MTBF	Score	Explanation
No-error long evidence	SSD	SK hynix	HFS256G32TND-N210A	502	0	1.38	80.03	No repository-defined important SMART errors were present. The record provides 502 power-on days of evidence, and the LinuxHW MTBF statistic is 1.38 years.
No-error long evidence	SSD	ADATA	SU900	535	0	1.47	80.08	No repository-defined important SMART errors were present. The record provides 535 power-on days of evidence, and the LinuxHW MTBF statistic is 1.47 years.
No-error low evidence	SSD	SanDisk	pSSD	0	0	0	35.65	No repository-defined important SMART errors were present, but the record contains no power-on history. The LinuxHW MTBF statistic is 0.00 years.
No-error low evidence	SSD	WDC	WD Green 2.5	3	0	0.01	36.32	No repository-defined important SMART errors were present, but only 3 power-on days are available. The LinuxHW MTBF statistic is 0.01 years.
Degraded with errors	SSD	Crucial	CT480M500SSD1	3500	16	0.56	5.03	The record contains 16 repository-defined important SMART errors and 3,500 power-on days of evidence. The LinuxHW MTBF statistic is 0.56 years.
Degraded with errors	SSD	Crucial	CT512M550SSD3	3235	16	0.52	5.55	The record contains 16 repository-defined important SMART errors and 3,235 power-on days of evidence. The LinuxHW MTBF statistic is 0.52 years.
Critical with errors	SSD	SanDisk	SSD U110	145	1008	0	0	The record contains 1,008 repository-defined important SMART errors and 145 power-on days of evidence. The LinuxHW MTBF statistic is 0.00 years.
Critical with errors	SSD	Samsung	SSD 870 EVO	160	116	0	0	The record contains 116 repository-defined important SMART errors and 160 power-on days of evidence. The LinuxHW MTBF statistic is 0.00 years.

4.7 Practical interpretation

The experiments support a two-stage interpretation of consumer SMART evidence. Population features are useful when a record's own error field is withheld, as shown by the gain from smoothed model and manufacturer priors. Once a user is assessing a known drive, however, the observed important-error count should dominate the operational response. LESHS preserves this distinction by keeping Err and MTBF out of the strict benchmark and reintroducing Err only after the calibrated prior has been estimated.

Calibration changes the meaning of the strict output from an arbitrary ensemble score to an estimated probability scale that can be combined with observed evidence. The low-evidence term addresses a separate concern: an error-free record with only a few days of exposure carries less reliability evidence than an error-free record observed for several years. This design prevents the score from collapsing all zero-error records into a single category.

The explanation layer is deliberately narrower than a general storage assistant. It answers what evidence is present, how much observation is available, how the repository statistic behaves, and which operational band results. It does not infer firmware defects, warranty status, data-loss probability, or a failure date. This constraint makes the explanation useful for action while keeping its claims aligned with the public fields.

5. Limitations

The LinuxHW SSD and NVMe appendices do not contain physical failure timestamps, replacement records, warranty outcomes, complete workload histories, temperature time series, write amplification, total bytes written for every device, or every vendor-specific raw SMART attribute. The main target is therefore the repository-defined important-error flag rather than future device failure. LESHS summarizes current evidence; it does not estimate when a drive will fail.

The repository is community-contributed and reflects Linux users who ran hw-probe and shared their results. This population differs from a controlled enterprise fleet and may overrepresent devices examined after a problem. NVMe records also have shorter power-on histories than SSD records. The low-evidence penalty acknowledges that difference but cannot replace missing longitudinal observation.

The aggregate Err count improves comparability across heterogeneous devices but removes attribute-level distinctions. Two records with Err = 1 may reflect different underlying counters and different operational implications. The explanation layer therefore uses the phrase repository-defined important SMART error unless a verified attribute-level record is available.

Manufacturer and model priors are learned from the LinuxHW training partition and should not be interpreted as universal product-quality rankings. Rare categories are smoothed toward the global training rate, and the model receives category counts to represent the strength of the prior. External validation on an independent consumer population would be required before using those priors as a general market comparison.

The score bands are fixed engineering thresholds rather than thresholds optimized against failure cost, replacement policy, or service-level objectives. Longitudinal labels would allow future work to tune them for backup urgency, expected loss, or maintenance scheduling. The present bands are intended to make the observed LinuxHW evidence easier to interpret.

The language component is evaluated through deterministic rendering of a fixed evidence schema. This isolates explanation fidelity from stochastic generation, but it does not measure user comprehension, preferred wording, or robustness across general conversational prompts. A deployment that permits free-form paraphrasing should retain field validation, source attribution, and a refusal rule for unsupported questions.

6. Conclusion

LESHS provides an explainable consumer SSD and NVMe health-scoring framework for the LinuxHW

SMART Repository. The study retained 162,382 nonduplicate records, established a strict non-leakage benchmark, validated the repository MTBF formula, and combined a calibrated population prior with observed important-error evidence and exposure-aware uncertainty. On 24,358 held-out records, the calibrated strict model achieved ROC-AUC 0.8873, PR-AUC 0.3892, F1 0.4066, and Brier score 0.0390.

The results show that population context and direct SMART evidence answer different questions. Manufacturer, model, capacity, technology type, and power-on exposure improve prediction when Err and MTBF are hidden. For a known drive, the observed important-error count and the amount of power-on history provide the most direct operational evidence. The grounded explanation schema connects these components to readable statements without extending the claim beyond the available data.

The resulting score is best understood as an evidence-based monitoring aid. It can distinguish long-observed error-free records from young records with little history, identify records with substantial important-error evidence, and disclose the population context behind the calibrated prior. Future longitudinal data can extend the same architecture toward failure-time prediction and cost-sensitive maintenance decisions while preserving the separation between scoring, calibration, and explanation.

References

[1] Linux Hardware, "SMART Repository: HDD/SSD Desktop-Class Reliability Test," GitHub repository, 2024.

[2] 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, pp. 17-28.

[3] 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, pp. 1-16.

[4] Backblaze, "Analyzing hard drive S.M.A.R.T. stats," Backblaze Blog, 2014.

[5] Backblaze, "What SMART stats tell us about hard drives," Backblaze Blog, 2016.

[6] Backblaze, "Hard drive test data," Backblaze, 2024.

[7] Acronis, "Acronis Drive Monitor: Disk health calculation," Acronis Knowledge Base, 2010.

[8] smartmontools, "smartctl manual page and ATA SMART report documentation," smartmontools Documentation, 2024.

[9] NVM Express, "NVM Express Base Specification, Revision 2.0e," NVM Express, 2024.

[10] JEDEC, "Solid-State Drive (SSD) Requirements and Endurance Test Method, JESD218B," JEDEC Solid State Technology Association, 2016.

[11] JEDEC, "Solid-State Drive Endurance Workloads, JESD219B," JEDEC Solid State Technology Association, 2016.

[12] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. New York, NY, USA: Springer, 2009.

[13] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[14] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine Learning, vol. 63, no. 1, pp. 3-42, 2006.

[15] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001.

[16] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[17] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," Technometrics, vol. 12, no. 1, pp. 55-67, 1970.

[18] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," Pattern Recognition, vol. 30, no. 7, pp. 1145-1159, 1997.

[19] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in Proc. 23rd Int. Conf. Machine Learning (ICML), 2006, pp. 233-240.

[20] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in Advances in Large Margin Classifiers, 1999, pp. 61-74.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of

any classifier," in Proc. ACM SIGKDD, 2016, pp. 1135-1144.
[22] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proc. NeurIPS, 2017, pp. 4765-4774.
[23] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," Artificial Intelligence, vol. 267, pp. 1-38, 2019.
[24] D. Gunning and D. Aha, "DARPA's Explainable Artificial Intelligence (XAI) program," AI Magazine, vol. 40, no. 2, pp. 44-58, 2019.
[25] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, 2023.
[26] M. Mitchell et al., "Model cards for model reporting," in Proc. FAT*, 2019, pp. 220-229.
[27] T. B. Brown et al., "Language models are few-shot learners," in Proc. NeurIPS, 2020, pp. 1877-1901.
[28] L. Ouyang et al., "Training language models to follow instructions with human feedback," in Proc. NeurIPS, 2022, pp. 27730-27744.
[29] OpenAI, "GPT-4 technical report," arXiv:2303.08774, 2024.
[30] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in Proc. NeurIPS, 2022, pp. 24824-24837.
[31] C. Li, W. Su, and E. Zhang, "Lightweight Hallucination Firewall for Enterprise LLM Applications: Evidence Consistency, Self-Checking, and Small-Model Detection on TruthfulQA," Journal of Advanced Computing Systems, vol. 3, no. 1, pp. 49-65, Jan. 2023, doi: 10.69987/JACS.2023.30104.
[32] J. Nie and D. Zheng, "Ambiguity-Aware HDFS Log Anomaly Detection with Retrieval-Augmented Failure Narratives and Selective Refusal," Journal of Advanced Computing Systems, vol. 3, no. 1, pp. 66-80, Jan. 2023, doi: 10.69987/JACS.2023.30105.
[33] G. Liu, S. He, and I. Liu, "LLM-Augmented Multi-Source Root Cause Attribution for CPU and Network Faults in Microservices," Journal of Advanced Computing Systems, vol. 3, no. 6, pp. 39-57, Jun. 2023, doi: 10.69987/JACS.2023.30604.
[34] B. Zhang, H. Rao, and D. Zhao, "Evidence-Grounded RAG for Cloud-Native DevOps: Hallucination-Resistant AIOps Question Answering over Private Operations Documents," Journal of Advanced Computing Systems, vol. 4, no. 3, pp. 109-125, Mar. 2024, doi: 10.69987/JACS.2024.40308.
[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," Journal of Advanced Computing Systems, vol. 4, no. 4, pp. 97-111, Apr. 2024, doi: 10.69987/JACS.2024.40408.
[36] C. Li, J. Bai, and S. Wang, "Evidence-Chain Reliable RAG: Word-Level Hallucination Detection, Source Attribution, and Provenance Explanation for LLM Applications," Journal of Advanced Computing Systems, vol. 4, no. 2, pp. 76-92, Feb. 2024, doi: 10.69987/JACS.2024.40207.
[37] 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," Journal of Advanced Computing Systems, vol. 3, no. 4, pp. 31-47, Apr. 2023, doi: 10.69987/JACS.2023.30403.
[38] 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," Journal of Advanced Computing Systems, vol. 4, no. 6, pp. 74-85, Jun. 2024, doi: 10.69987/JACS.2024.40606.
[39] H. Zhou and S. Zhao, "LLM-Explanation-Enhanced Retail Credit Default Prediction with Gradient Boosting on the UCI Default of Credit Card Clients Dataset," Journal of Advanced Computing Systems, vol. 4, no. 5, pp. 102-118, May 2024, doi: 10.69987/JACS.2024.40508.
[40] Z. S. Zhong, R. Ma, and H. Zhao, "Human-Uncertainty Distillation for Calibrated Vision Models on CIFAR-10H," Journal of Advanced Computing Systems, vol. 3, no. 2, pp. 77-89, Feb. 2023, doi: 10.69987/JACS.2023.30206.
[41] Q. Xin, "Uncertainty-Aware Late Fusion for 3D Perception (Confidence Calibration + Fusion Rule Learning)," Journal of Technology Informatics and Engineering, vol. 4, no. 1, pp. 215-238, Apr. 2025, doi: 10.51903/jtie.v4i1.485.
[42] S. He, H. Tu, and I. Liu, "Safe PD Capacity Forecasting with Time-Series Foundation

Models and Calibrated Uncertainty for Heterogeneous GPU Clusters," Journal of Advanced Computing Systems, vol. 3, no. 4, pp. 48-66, Apr. 2023, doi: 10.69987/JACS.2023.30404.

[43] S. Chen, S. He, and E. Sun, "Risk-Bounded GPU Resource Oversubscription via Conformal Demand Envelopes in Production AI Clusters," Journal of Advanced Computing Systems, vol. 4, no. 5, pp. 119-134, May 2024, doi: 10.69987/JACS.2024.40509.

[44] J. Nie and D. Zheng, "Noisy-Neighbor-Aware VM Degradation Risk Modeling with Unsupervised Residual Fusion," Journal of Advanced Computing Systems, vol. 4, no. 4, pp. 112-123, Apr. 2024, doi: 10.69987/JACS.2024.40409.

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

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

[47] 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," International Journal of Graphic Design, vol. 4, no. 1, pp. 179-185, Apr. 2026, doi: 10.51903/ijgd.v4i1.3703.

[48] 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), 2016, pp. 67-80.

[49] 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.

[50] E. Xu, M. Zheng, F. Qin, Y. Xu, and J. Wu, "Lessons and Actions: What We Learned from 10K SSD-Related Storage System Failures," in Proc. USENIX Annual Technical Conf. (USENIX ATC), 2019, pp. 961-976.

[51] J. Meza, Q. Wu, S. Kumar, and O. Mutlu, "A Large-Scale Study of Flash Memory Failures in the Field," in Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems, 2015, pp. 177-190, doi: 10.1145/2796314.2745848.

[52] S. Lu, B. Luo, T. Patel, Y. Yao, D. Tiwari, and W. Shi, "Making Disk Failure Predictions SMARTer!" in Proc. 18th USENIX Conf. File and Storage Technologies (FAST), 2020, pp. 151-167.

[53] Q. Wu, J. Bai, and X. Zhou, "Evidence-Grounded Financial RAG: Reducing Numerical Hallucination in LLM-Generated Corporate Risk Memos," Journal of Advanced Computing Systems, vol. 3, no. 3, pp. 65-84, Mar. 2023, doi: 10.69987/JACS.2023.30306.

[54] K. Zhang, S. Meng, and E. Zhou, "Evidence-Grounded Trading Desk Risk Memos over SEC Filings: Retrieval-Augmented Generation with XBRL Numeric Verification," Journal of Advanced Computing Systems, vol. 3, no. 2, pp. 60-76, Feb. 2023, doi: 10.69987/JACS.2023.30205.