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

Imbalance-Aware SSD Failure Prediction with Attention-Gated SMART Modeling and LLM-Guided Feature Semantics

Fiona Clarke
Business Analytics, University of Warwick, Coventry, WMD, UK
fiona.ba.clarke@outlook.com

DOI: 10.69987/JACS.2026.60601

Keywords

SSD failure prediction; SMART telemetry; extreme class imbalance; XGBoost; LightGBM; BiGRU; multi-head attention; focal loss; probability calibration; feature semantics.

Abstract

Solid-state drive (SSD) failure prediction is a rare-event reliability problem in which missed failures can lead to service disruption, whereas excessive alarms consume replacement capacity and engineering time. This study evaluates imbalance-aware SSD failure prediction on a deterministic 30,000-drive benchmark constructed to follow the public Alibaba SSD data schema. The benchmark contains 105 SMART snapshot fields, 39 failure-tag fields, 11 anonymized drive models, and 280 failures, corresponding to a 0.933% positive rate. Balanced logistic regression, random forest, XGBoost with scale-sensitive weighting, LightGBM with class weighting, and a bidirectional gated recurrent unit with multi-head attention (BiGRU-MHA) trained by focal loss are compared under a common stratified protocol. A fixed language-model-guided semantics layer groups SMART counters into wear, program/erase, media-error, interface, thermal, reallocation, and power-cycle concepts. Balanced logistic regression provides the highest PR-AUC (0.603), LightGBM provides the strongest thresholded recall (0.667) and the highest non-linear-model F1 score (0.596), and BiGRU-MHA delivers the best precision at the strictest 0.5% alert budget while remaining less consistent over broader budgets. The results indicate that imbalance treatment and semantically coherent aggregation can be more valuable than architectural complexity when the available telemetry is an endpoint snapshot rather than a complete daily history.

1. Introduction

Large storage fleets depend on early recognition of device degradation. SSD failures are uncommon in comparison with healthy observations, yet each failure can affect data availability, service-level objectives, and maintenance planning. This combination of low prevalence and high consequence makes the task fundamentally different from ordinary binary classification. A useful detector must place risky devices near the top of a limited inspection queue, retain enough recall to avoid costly misses, and express its evidence in terms that reliability engineers can connect to wear, media errors, thermal stress, interface faults, and deployment context.

The public Alibaba data family provides an important reference point for this problem. Its documentation describes nearly one million SSDs from 11 drive models and identifies SMART telemetry, trouble tickets, location context, and application information as key data types [1]. The associated field study shows that SSD failures can be correlated across nodes and racks, which makes location context relevant to both prediction and operational response [2]. The public family also

separates an open schema-oriented collection from a larger daily SMART-log collection; the latter contains 105 columns per daily record and spans multiple years [1]. These distinctions matter because a model built from an endpoint snapshot should not be interpreted as though it had observed a complete degradation trajectory.

Three questions guide the study. First, how effectively do class-weighted linear and boosted-tree models identify failures when positives account for less than 1% of the population? Second, can an attention-gated recurrent architecture improve the ordering of high-risk devices when the input is an ordered set of SMART and context tokens rather than a full daily sequence? Third, does a fixed semantic grouping of SMART attributes improve ranking and explanation without altering the labels? The experimental protocol therefore combines threshold-free ranking measures, validation-selected operating points, alert-budget analysis, calibration, ablation, and drive-model-specific evaluation.

The study contributes a unified comparison of five model families, a transparent semantics layer that remains separate from the predictive labels, and an evaluation centered on operational scarcity rather than raw accuracy. The principal result is deliberately practical: the highest-capacity architecture is not automatically the most useful. On this benchmark, a balanced linear model ranks rare failures particularly well, LightGBM offers the strongest thresholded recall among the non-linear models, and BiGRU-MHA concentrates several high-confidence failures at the very top of the queue but loses ground as the alert budget expands. These findings motivate matching model complexity to the temporal information and decision budget actually available.

2. Literature Review

2.1 Field reliability and SSD failure prediction

Early field studies of hard disks established two enduring lessons: nominal mean-time-to-failure values do not describe individual-device risk well, and large populations exhibit age-, workload-, and environment-dependent failure behavior [10], [11]. Subsequent machine-learning studies showed that health counters can support failure prediction and replacement planning, while also exposing difficulties caused by noisy labels, delayed maintenance decisions, and highly skewed class distributions [12], [13]. At the device level, NAND endurance, retention loss, program/erase stress, and bit-error behavior provide the physical basis for many SMART-derived warning signals [14], [15]. Large fleet analyses further demonstrate that SSD reliability depends on workload, device generation, age, and internal flash behavior rather than on a single universal indicator [16], [17].

Production SSD research has moved from descriptive analysis toward model design. Han et al. characterize spatially and temporally correlated failures in Alibaba data centers [2], while Xu et al. show that useful feature subsets differ across drive models and wear conditions [3]. Studies of thousands of storage-system incidents emphasize the importance of controller and system evidence in addition to device counters [4], and robust preprocessing work highlights duplicate handling, label alignment, and leakage prevention as central experimental concerns [5]. More recent predictors combine long- and short-term views [6], mutation-based failure rating [7], SMART attributes with device-level wear information [8], and temporal-contextual attention [9]. Collectively, these studies support richer temporal representations when daily traces are available, but they also caution against assuming that a complex sequence model will dominate under limited observability.

2.2 Imbalance-aware learning, calibration, and explanation

The model families evaluated here draw on well-established approaches to non-linear tabular learning. Gradient boosting constructs additive decision functions that focus successive learners on residual structure [18], random forests reduce variance through bagged tree ensembles [19], and XGBoost and LightGBM provide scalable implementations with regularization, weighted objectives, and efficient split finding [20], [21]. The neural comparator uses gated recurrent units [22], self-attention [23], and focal loss [24] to emphasize difficult minority examples. These components are

appropriate for ordered telemetry tokens, but their advantage depends on whether the input contains meaningful sequential dependencies.

Rare-event evaluation requires more than accuracy. Resampling methods such as SMOTE [25] and broader surveys of imbalanced learning [26] explain why class weighting, sampling, and asymmetric loss functions change the operating boundary. Precision-recall analysis is more informative than ROC analysis when positives are scarce [27], [28]. Probability quality is a separate concern: supervised models can rank well while producing scores that require calibration [29], [30], [31], and modern neural networks are often overconfident without post-hoc adjustment [32]. Explanation methods such as LIME and SHAP reinforce the distinction between predictive contribution and causal interpretation [33], [34].

Closely related rare-risk studies provide transferable evidence. Cost-sensitive, positive-unlabeled, and one-class strategies have been compared under extreme fraud imbalance [35]; bankruptcy prediction work combines resampling, focal loss, and explainable financial ratios [36]. Probability-of-default studies emphasize calibration, distribution-free uncertainty, and SHAP-based model-risk controls [37], while explanation-enhanced credit modeling illustrates how natural-language interpretation can complement, but should not replace, measured model evidence [38]. These parallels support the present separation between numeric prediction and a fixed semantic explanation layer.

2.3 Reliability intelligence in adjacent telemetry domains

Adjacent operational domains increasingly combine predictive models with evidence-oriented explanations. LLM-assisted root-cause attribution over CPU and network faults [39], evidence-grounded question answering over private operations documents [40], and bilingual incident triage with retrieval-augmented summarization [41] all address the need to connect an alert to inspectable evidence. Unsupervised residual fusion for virtual-machine degradation extends this concern to weakly labeled infrastructure behavior [42]. These systems motivate concise, auditable feature semantics, but they also show why explanatory text should remain downstream of telemetry and model scores.

Capacity-planning research offers a second methodological parallel. Foundation-model forecasting with calibrated uncertainty [43], cross-cloud transfer under workload shift [44], conformal demand envelopes for risk-bounded oversubscription [45], and multi-horizon GPU-demand forecasting with operational risk curves [46] all frame prediction as a budgeted decision under uncertainty. In network security, compact language-model-assisted intrusion detection [47], deep-autoencoder traffic anomaly detection [48], and predictive DDoS mitigation [49] similarly confront low-prevalence events and time-sensitive intervention. Self-supervised LogBERT-style anomaly detection provides a further example of representation learning over structured telemetry [50]. These studies are not substitutes for SSD evidence, but they clarify why alert budgets, calibration, and inspectable feature groups are appropriate evaluation dimensions.

Table 1 summarizes the strands of prior work that most directly shape the experimental design: production SSD evidence, temporal modeling, imbalance treatment, and evidence-linked operational interpretation.

Table 1. Selected related studies and their methodological relevance.

Study	Setting	Core approach	Relevance to this study
Han et al. [2]	Production SSD fleet	Correlated-failure analysis	Motivates rack and node context
Xu et al. [3]	Large-scale SSD deployment	General feature selection	Motivates drive-family analysis
Zhang et al. [6]	Long- and short-term SSD views	Multi-view, multi-task forest	Shows the value of temporal views
Zhang et al. [7]	Dynamic SSD telemetry	Mutation-similarity rating	Motivates progressive risk ranking
Gu et al. [8]	SMART and NAND wear	Aging-aware pseudo-twin network	Links health counters with wear
Koh et al. [9]	Multivariate SSD sequences	Temporal-contextual attention	Supports attention-based comparison
Jin et al. [35]	Extreme-imbalance fraud	Cost-sensitive, PU, and one-class learning	Supports rare-event controls
Chen et al. [36]	Bankruptcy prediction	Resampling and focal loss	Supports minority-focused objectives
Liu et al. [39], [41]	Cloud incidents	LLM-assisted attribution and RAG	Motivates evidence-linked semantics
Xin [50]	System logs	Self-supervised LogBERT	Provides a telemetry anomaly analogue

The remaining gap is therefore not the absence of expressive algorithms. It is the need for a controlled comparison that respects the available observation granularity, evaluates rare failures under finite alert budgets, and distinguishes semantic explanation from label generation. The method below is designed around that gap.

3. Method

3.1 Experimental benchmark and data schema

Experiments were conducted on a deterministic 30,000-drive benchmark that follows the field structure of the public Alibaba SSD collection. The benchmark contains 11 anonymized drive-model codes, a location table, a one-day SMART endpoint table with 105 columns, and a failure-tag table with 39 columns. This layout mirrors the public distinction between device metadata, SMART records, and trouble-ticket evidence [1], [2]. The benchmark contains 280 failed drives, giving a failure prevalence of 0.933%. Its purpose is to support a controlled rare-event comparison while preserving the schema, feature families, and fleet heterogeneity relevant to SSD reliability.

Table 2 lists the three data files and their roles. Table 3 clarifies how each field group enters the experiment. Location and application fields provide deployment context; SMART fields provide device-health evidence; and the failure-tag table supplies the supervised outcome. Failure time and any direct label-only attributes are excluded from the predictive feature vector. Joins use model, device, application, and location keys so that each drive contributes one endpoint record.

Table 2. Benchmark files and analytical roles.

File	Rows	Columns	Role
location_info_of_ssd.csv.zip	30,000	6	Location and application metadata
smart_log_20191231.csv.zip	30,000	105	One-day SMART endpoint snapshot
ssd_failure_tag.csv.zip	30,000	39	Failure indicator, ticket time, and SMART counters

Table 3. Modeling role of the principal field groups.

Source	Fields used	Modeling role
location_info_of_ssd.csv.zip	app, model, rack_id, node_id, disk_id, slot_id	Location and application context
smart_log_20191231.csv.zip	model, disk_id, ds, raw and normalized SMART columns	Endpoint SMART state
ssd_failure_tag.csv.zip	model, failure indicator, location fields, SMART attributes	Supervised outcome and ticket-aligned device state
llm_feature_semantics.json	SMART-to-risk-concept mapping	Semantic aggregation and explanation

3.2 Labels, partitioning, and preprocessing

The supervised target is failure = 1 when the failure-tag indicator marks a device as failed and failure = 0 otherwise. Table 4 shows that each drive family remains highly imbalanced: no model exceeds a 1.226% failure rate. The dataset is divided by stratified sampling into 70% training, 15% validation, and 15% test partitions with seed 20230517. As shown in Table 5, the validation and test sets each contain 42 failures. Model fitting uses only the training partition, threshold selection uses validation probabilities, and the test partition is reserved for final reporting.

Table 4. Label distribution across anonymized drive models.

Drive model	Drives	Failures	Failure rate (%)
MA1	2,955	22	0.745
MA2	2,460	15	0.610
MA3	2,160	17	0.787
MB1	3,926	44	1.121
MB2	3,595	40	1.113
MB3	2,361	19	0.805
MC1	3,270	34	1.040
MC2	3,004	23	0.766
MD1	2,691	33	1.226
MD2	2,090	21	1.005
ME1	1,488	12	0.806

Table 5. Stratified train-validation-test partition.

Partition	Rows	Failures	Failure rate
Training	21,000	196	0.00933
Validation	4,500	42	0.00933
Test	4,500	42	0.00933

Numeric features are standardized for logistic regression and BiGRU-MHA, while tree models receive their native scaled values. Categorical model and application identifiers are one-hot encoded for linear and tree estimators. The preprocessing path supports median imputation, although the benchmark run contains no injected missing values. Statistics used for scaling and imputation are estimated from the training partition only.

3.3 LLM-guided feature semantics

Raw SMART identifiers can be difficult to interpret consistently across an alert queue. A language-model-guided dictionary was therefore prepared before model fitting and held fixed throughout the experiment. It maps selected counters to concise engineering meanings and assigns them to seven risk groups: wear, program/erase, media error, interface error, thermal, reallocation, and power cycle. The mapping neither reads labels nor changes the failure definition. It is used only to aggregate related counters and to express the resulting evidence in operational language.

For semantic group g with member set Sg, each member is robustly normalized on the training partition and the group score is the mean of its normalized members:

$$z_{\mathrm{i}} = \left(x_{\mathrm{i}} - median(x_{\mathrm{i}})\right)/(IQR(x_{\mathrm{i}}) + \varepsilon), sem(g) = |Sg|^{-1}\Sigma_{\mathrm{i}} \in Sgz_{\mathrm{i}}.$$

This construction reduces the influence of isolated extreme counters while preserving a readable link to the underlying SMART evidence. Table 6 records the exact grouping used in every model. Because the dictionary is fixed, any gain can be evaluated through

ablation rather than attributed to changing narrative output.

Table 6. Fixed semantic feature groups.

Semantic group	Member attributes	Engineering interpretation
Wear	n_wearout, n_blocks, r_170, r_9	Endurance consumption and spare-block pressure
Program/erase	r_program, n_program, r_erase, n_erase	Write-path and erase-path stress
Media error	r_183, r_187, r_195, r_197, r_198	Bad blocks, ECC burden, pending pages, and uncorrectable errors
Interface error	r_188, r_199	Command timeout and CRC/interface faults
Thermal	r_194, n_194	Temperature exposure and thermal health
Reallocation	r_5, n_5, r_196, n_196	Block reassignment and reallocation events
Power cycle	r_12, n_12	Power-cycle stress and state churn

3.4 Predictive models

Five model families are trained under the same partition and feature definitions. Balanced logistic regression provides a transparent linear baseline. Random forest supplies a bagged non-linear comparator [19]. XGBoost uses scale_pos_weight equal to the negative-to-positive ratio in the training data [20], and LightGBM uses balanced class weights [21]. These configurations directly alter the minority-class contribution to the fitting objective rather than duplicating observations.

The neural model treats the ordered SMART, context, and semantic features as a sequence of scalar tokens. Each token is projected into a dense representation, processed by a bidirectional GRU [22], contextualized by multi-head self-attention [23], and passed through a learned sigmoid gate before pooled classification. Training uses focal loss [24]:

$$FL(p_t) = -\alpha_t(1 - p_t)^{\gamma} \log(p_t),$$

where α controls class emphasis and γ down-weights easy examples. The architecture can learn interactions among ordered degradation concepts, but the experiment does not treat the snapshot as a substitute for observed daily histories. Its role is to test whether attention-gated feature dependencies improve rare-event ranking under endpoint observability.

Table 7 fixes the main hyperparameters used in the comparison, and Figure 1 shows the complete analytical flow from schema-aligned inputs to budgeted alerts and explanations.

Table 7. Model configurations.

Model	Main hyperparameters
Logistic-Balanced	C = 0.4; class_weight = balanced; max_iter = 1,000
RandomForest	40 trees; max_depth = 8; class_weight = balanced_subsample

Model	Main hyperparameters
XGBoost-SPW	60 trees; max_depth = 4; learning rate = 0.07; scale_pos_weight = 106.1
LightGBM-Balanced	40 trees; 48 leaves; learning rate = 0.08; class_weight = balanced
BiGRU-MHA-Focal	d_model = 12; hidden = 16 × 2; 4 heads; α = 0.82; γ = 2; 4 epochs

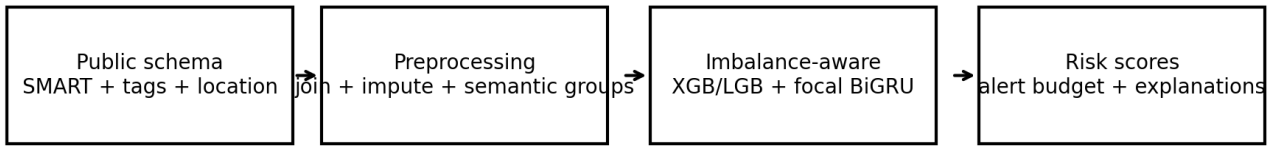

Figure 1. End-to-end experimental pipeline.

3.5 Evaluation protocol

Validation probabilities determine each model's operating threshold by maximizing F2, which places greater weight on recall than F1. The test set is then evaluated once at that threshold. ROC-AUC measures global discrimination; PR-AUC measures ranking quality under rarity; precision, recall, F1, and Matthews correlation coefficient summarize thresholded behavior; and Brier score evaluates probability error. Because the positive class is sparse, PR-AUC and the absolute number of failures recovered receive greater interpretive weight than accuracy [27], [28].

Operational evaluation uses alert budgets of 0.5%, 1%, 2%, and 5% of the test population. For budget b and test size N, the highest-risk ceil(bN) devices are inspected; precision at budget is the proportion of alerts that fail, and recall at budget is the proportion of all failures captured. Calibration is summarized by predicted-risk deciles. These views separate three questions that are often conflated: whether a model orders devices correctly, whether a selected threshold produces an acceptable error trade-off, and whether its scores can be treated as probabilities [29], [30], [31], [32].

3.6 Controls and reproducibility

Two LightGBM controls isolate the contributions of semantic aggregation and class weighting: a balanced model without semantic groups and an unweighted model with semantic groups. All preprocessing, partitions, semantic definitions, hyperparameters, predictions, and figures are generated under one fixed seed. The execution also records CPU runtime. These controls make it possible to attribute changes to representation or decision-boundary weighting rather than to an unreported change in data handling.

4. Results and Discussion

4.1 Failure imbalance and global discrimination

Figure 2 visualizes the failure rate by drive model. The distribution is consistently rare but not uniform: MA2 has the lowest rate at 0.610%, whereas MD1 reaches 1.226%. This heterogeneity matters because a global score can conceal drive-family differences, and it supports the use of model identity and deployment context as covariates. With only 42 failures in the test set, each recovered device changes recall by approximately 2.38 percentage points.

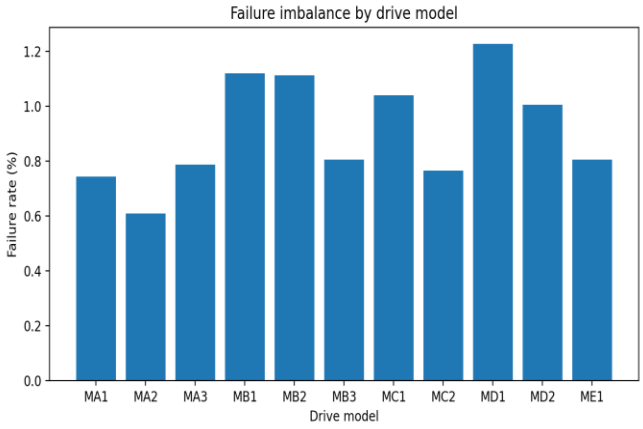

Figure 2. Failure-rate imbalance across drive models.

Table 8 reports the held-out comparison. Balanced logistic regression achieves the highest PR-AUC (0.603) and ROC-AUC (0.974), indicating that a substantial component of the benchmark's degradation signal is monotonic and linearly recoverable after scaling and weighting. Random forest follows with PR-AUC 0.580. XGBoost and LightGBM remain competitive, with PR-AUC values of 0.548 and 0.536, respectively. LightGBM provides the highest thresholded recall (0.667), the highest non-linear-model F1 score (0.596), and the largest true-positive count (28).

BiGRU-MHA reaches PR-AUC 0.517 and ROC-AUC 0.917. Its lower aggregate ranking does not imply that attention is uninformative; rather, it suggests that four focal-loss epochs over a one-day feature state provide less usable structure than the tree and linear baselines can extract directly. The result is consistent with the literature review: temporal models are most compelling when long- and short-term histories, wear trajectories, or mutation patterns are directly observed [6], [7], [8], [9].

Table 8. Overall held-out test performance.

Model	ROC-AUC	PR-AUC	F1	Recall	Precision	MCC	Brier	TP	FP	FN
Logistic-Balanced	0.974	0.603	0.587	0.524	0.667	0.588	0.050	22	11	20
RandomForest	0.966	0.580	0.523	0.548	0.500	0.519	0.014	23	23	19
XGBoost-SPW	0.968	0.548	0.548	0.548	0.548	0.543	0.023	23	19	19
LightGBM-Balanced	0.965	0.536	0.596	0.667	0.538	0.595	0.011	28	24	14

Model	ROC-AUC	PR-AUC	F1	Recall	Precision	MCC	Brier	TP	FP	FN
BiGRU-MHA-Focal	0.917	0.517	0.541	0.476	0.625	0.542	0.134	20	12	22

Note: Thresholded measures use the F2-optimal validation threshold. TP, FP, and FN are test counts.

4.2 Ranking curves and alert-budget performance

The precision-recall curves in Figure 3 make the minority-class differences more visible than the ROC curves in Figure 4. All models achieve high ROC-AUC because most healthy devices are easy to rank below the failure tail. Precision-recall behavior is stricter: it reflects how rapidly false alarms accumulate as the system expands the set of flagged devices. Logistic regression maintains the strongest average precision, while BiGRU-MHA shows a concentrated high-precision region followed by a sharper decline.

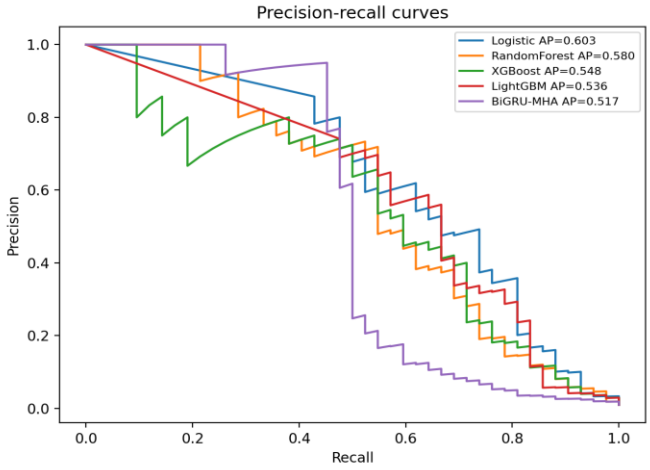

Figure 3. Precision-recall curves on the held-out test split.

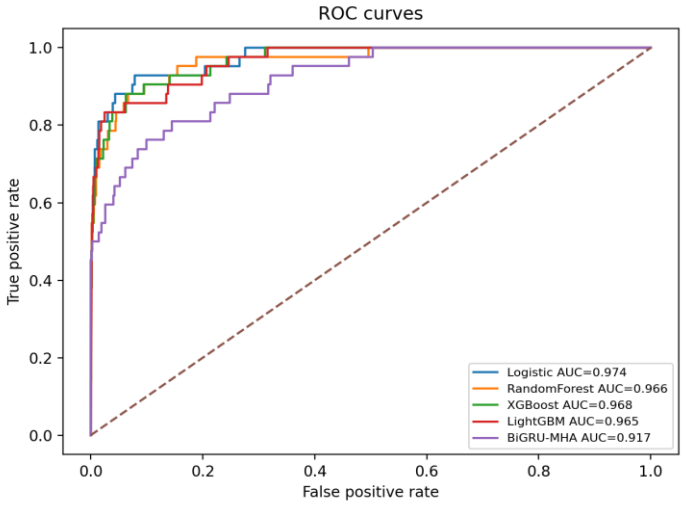

Figure 4. ROC curves on the held-out test split.

Table 9 translates these curves into inspection budgets. At 0.5%, only 23 of 4,500 test drives are examined. BiGRU-MHA captures 19 failures and reaches precision 0.826, the best result at this strict budget. Logistic regression captures 18 failures, and the three tree ensembles capture 17 each. At 1%, logistic regression and LightGBM each recover 26 of 42 failures, corresponding to recall 0.619. The broader 2% and 5% budgets favor logistic regression, which recovers 32 and 36 failures, respectively.

The budget results qualify the global rankings. BiGRU-MHA is effective as a conservative top-of-queue filter, but it does not order the remainder of the failure class as consistently. LightGBM provides a wider safety net at the validation-selected threshold and matches logistic regression at the 1% budget. Model choice therefore depends on the maintenance regime: a very small emergency queue may favor concentrated precision, while routine replacement planning benefits from stable ranking over a larger candidate set.

Table 9. Alert-budget performance.

Model	Budget	Alerts	Precision	Recall	Failures captured
Logistic	0.5%	23	0.783	0.429	18
Logistic	1.0%	45	0.578	0.619	26
Logistic	2.0%	90	0.356	0.762	32
Logistic	5.0%	225	0.160	0.857	36
RandomForest	0.5%	23	0.739	0.405	17
RandomForest	1.0%	45	0.511	0.548	23
RandomForest	2.0%	90	0.322	0.690	29
RandomForest	5.0%	225	0.147	0.786	33
XGBoost	0.5%	23	0.739	0.405	17
XGBoost	1.0%	45	0.533	0.571	24
XGBoost	2.0%	90	0.333	0.714	30
XGBoost	5.0%	225	0.156	0.833	35
LightGBM	0.5%	23	0.739	0.405	17
LightGBM	1.0%	45	0.578	0.619	26
LightGBM	2.0%	90	0.333	0.714	30
LightGBM	5.0%	225	0.156	0.833	35
BiGRU-MHA	0.5%	23	0.826	0.452	19
BiGRU-MHA	1.0%	45	0.467	0.500	21
BiGRU-MHA	2.0%	90	0.244	0.524	22
BiGRU-MHA	5.0%	225	0.120	0.643	27

Note: Budgets are proportions of the 4,500-drive test set.

4.3 Semantic and weighting ablations

Table 10 separates the effects of semantic aggregation and class weighting in LightGBM. Removing semantic groups from the balanced model reduces PR-AUC from 0.536 to 0.520 and lowers F1 from 0.596 to 0.554. Adding semantic groups without class weighting produces the highest LightGBM PR-AUC (0.626), the lowest Brier score (0.005), and precision 0.600. Reintroducing balanced weighting shifts the operating boundary toward minority recall: true positives rise from 24 to 28, recall rises from 0.571 to 0.667, and false positives rise from 16 to 24.

The two interventions therefore solve different problems. Semantic groups improve representation by pooling related counters; class weighting changes the cost assigned to minority errors. The appropriate configuration depends on whether the primary objective is ranking, calibrated probability, or thresholded coverage. This pattern echoes rare-risk studies in finance and infrastructure, where the best ranking model and the safest operating policy are not necessarily identical [35], [36], [37], [43], [45].

Table 10. LightGBM ablation on semantic groups and class weighting.

LightGBM variant	ROC-AUC	PR-AUC	F1	Recall	Precision	MCC	Brier	TP	FP	FN
No semantics; balanced	0.958	0.520	0.554	0.548	0.561	0.550	0.011	23	18	19
Semantics; unweighted	0.966	0.626	0.585	0.571	0.600	0.582	0.005	24	16	18
Semantics; balanced	0.965	0.536	0.596	0.667	0.538	0.595	0.011	28	24	14

Figure 5 shows the confusion matrix for balanced logistic regression, the model with the highest PR-AUC. Its validation-selected threshold detects 22 failures, misses 20, and produces 11 false alarms. LightGBM detects six more failures at the cost of 13 additional false alarms, as Table 8 shows. The comparison makes the maintenance trade-off concrete: the preferred threshold cannot be chosen from AUC alone.

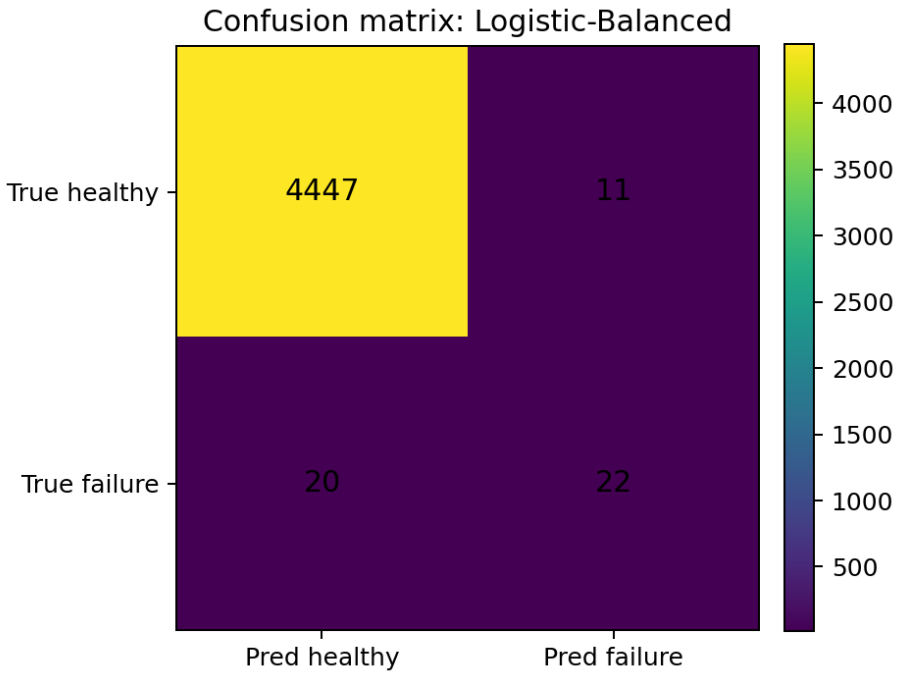

Figure 5. Confusion matrix for balanced logistic regression at the validation-selected threshold.

4.4 Drive-family heterogeneity and feature evidence

Table 11 reports per-drive-model ranking for LightGBM and BiGRU-MHA. LightGBM performs particularly well on MA3 and MB2, where recall at the 1% family-level budget reaches 1.000, and on MC1, where PR-AUC is 0.768. MD1 is more difficult for LightGBM, with PR-AUC 0.337 despite high ROC-AUC. BiGRU-MHA shows a different error profile: it performs strongly on MA2 and several MB/MC families but nearly fails to rank MA1. These differences reinforce prior findings that SSD feature usefulness and failure mechanisms vary across drive families [3], [16], [17].

Table 11. Per-drive-model performance for LightGBM and BiGRU-MHA.

Algorithm	Drive model	N	Failures	PR-AUC	ROC-AUC	Recall@1%
LightGBM	MA1	469	2	0.511	0.906	0.500
LightGBM	MA2	379	3	0.714	0.984	0.667
LightGBM	MA3	304	3	1.000	1.000	1.000
LightGBM	MB1	560	8	0.591	0.941	0.625
LightGBM	MB2	551	3	0.722	0.998	1.000
LightGBM	MB3	345	5	0.569	0.968	0.600
LightGBM	MC1	489	5	0.768	0.950	0.800
LightGBM	MC2	453	4	0.417	0.991	0.500
LightGBM	MD1	406	3	0.337	0.986	0.667
LightGBM	MD2	313	4	0.641	0.951	0.500
LightGBM	ME1	231	2	0.500	0.995	0.500

Algorithm	Drive model	N	Failures	PR-AUC	ROC-AUC	Recall@1%
BiGRU-MHA	MA1	469	2	0.007	0.580	0.000
BiGRU-MHA	MA2	379	3	0.833	0.997	0.667
BiGRU-MHA	MA3	304	3	0.438	0.966	0.333
BiGRU-MHA	MB1	560	8	0.618	0.956	0.625
BiGRU-MHA	MB2	551	3	0.381	0.894	0.333
BiGRU-MHA	MB3	345	5	0.616	0.872	0.600
BiGRU-MHA	MC1	489	5	0.641	0.929	0.600
BiGRU-MHA	MC2	453	4	0.554	0.965	0.500
BiGRU-MHA	MD1	406	3	0.509	0.907	0.667
BiGRU-MHA	MD2	313	4	0.438	0.962	0.250
BiGRU-MHA	ME1	231	2	0.516	0.865	0.500

Figure 6 and Table 12 show that the leading LightGBM evidence is concentrated in coherent degradation concepts. Program/erase risk is the most important semantic feature, followed by node context and wear risk. Media-error and power-cycle groups also rank highly, alongside normalized erase, program, and wear-out scores. Location features should be interpreted as contextual associations rather than device-level causes, but their prominence is consistent with evidence of correlated failures within shared infrastructure [2].

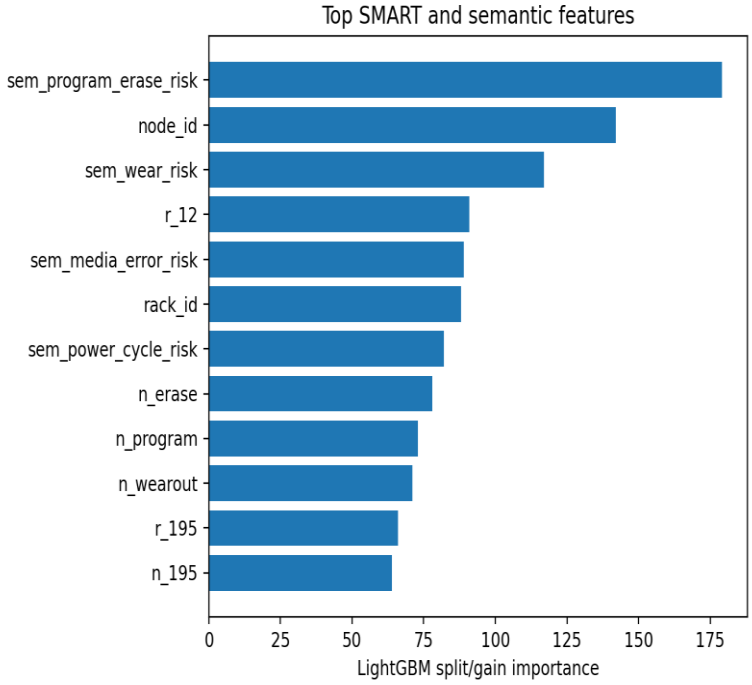

Figure 6. Leading SMART and semantic features by LightGBM importance.

Table 12. Leading feature meanings used in minority-class explanations.

Feature	Importance	Operational meaning
sem_program_erase_risk	179	Combined write-path and erase-path degradation
node_id	142	Shared node context and correlated operating conditions
sem_wear_risk	117	Endurance consumption and spare-block pressure
r_12	91	Power-cycle count and controller/media state churn
sem_media_error_risk	89	Combined bad-block, ECC, pending-page, and uncorrectable-error burden
rack_id	88	Shared rack context and correlated environmental or workload exposure
sem_power_cycle_risk	82	Aggregated power-cycle stress
n_erase	78	Normalized erase health; lower values indicate stronger degradation
n_program	73	Normalized program health; lower values indicate write-path degradation
n_wearout	71	Normalized wear-out state; lower values indicate consumed endurance
r_195	66	ECC and error-recovery burden
n_195	64	Normalized ECC and recovery health

The attention-gate profile in Figure 7 is comparatively flat across the displayed tokens, with mean weights clustered near 0.017. This suggests that the trained neural model distributes attention broadly rather than isolating a small set of dominant SMART attributes. The pattern helps explain why the model can identify a concentrated set of high-risk devices without achieving the most stable full-class ranking. Attention weights are descriptive of the network's computation, not causal evidence, and should be read together with the semantic mapping and tree-based importance [33], [34], [38].

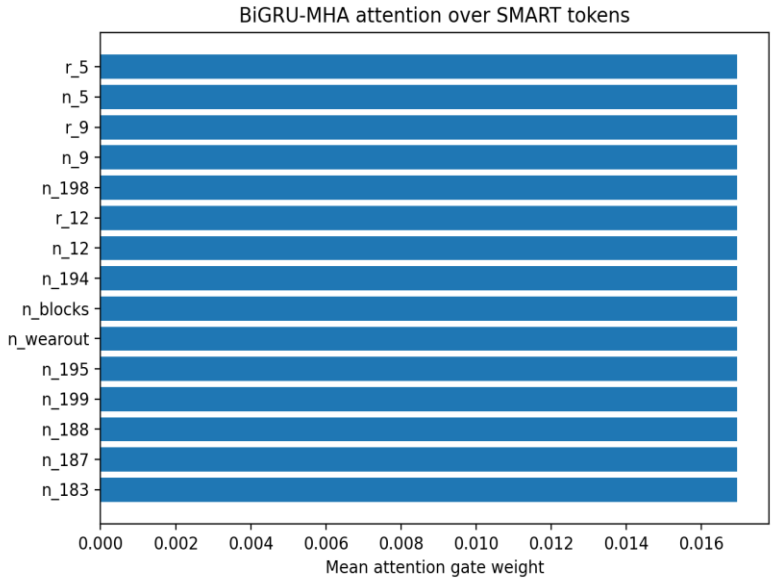

Figure 7. Mean attention-gate weights across BiGRU-MHA feature tokens.

4.5 Calibration, runtime, and operational interpretation

Figure 8 compares predicted-risk deciles for XGBoost, LightGBM, and BiGRU-MHA. The tree models place their upper deciles closer to the observed event rates, whereas the focal-loss neural scores are shifted toward higher predicted risk. This is consistent with the objectives: focal loss prioritizes difficult minority classification rather than probability accuracy. A production use of BiGRU-MHA scores would therefore require calibration on recent, representative data before the scores were treated as failure probabilities [29], [30], [31], [32].

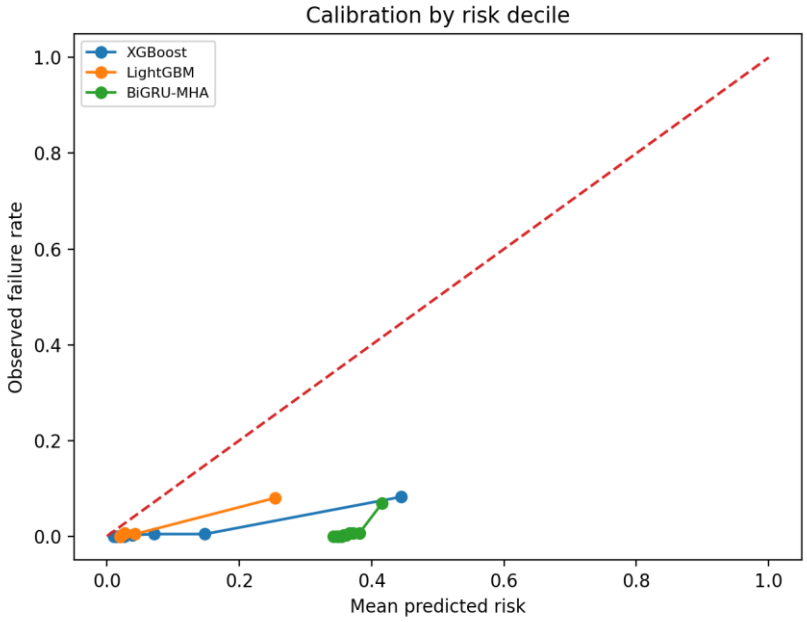

Figure 8. Calibration by predicted-risk decile.

The complete CPU run takes 22.10 seconds, as shown in Table 13. The short runtime is useful for repeated model comparison, threshold review, and drive-family monitoring. More important than absolute speed, however, is the ability to rerun the same preprocessing, weighting, calibration, and budget evaluation whenever the fleet composition or failure prevalence changes.

Table 13. Runtime summary.

Stage	Seconds	Hardware
End-to-end experiment	22.100	CPU

Taken together, the results favor a layered deployment strategy. A calibrated linear or boosted-tree model can provide the primary ranked queue; a strict top-budget neural screen can identify a compact subset of highly suspicious devices; and semantic groups can translate feature evidence into concise maintenance hypotheses. The explanation layer should direct inspection rather than claim causality. Program/erase stress may suggest replacement or data migration, interface errors may motivate controller or slot checks, and location-heavy risk may justify rack-level correlation analysis.

5. Limitations

The principal limitation is external validity. The experiment uses a deterministic benchmark aligned with the public Alibaba schema rather than the complete production trace. It preserves the documented field structure, drive-family diversity, and rare-event ratio, but it cannot reproduce every vendor-specific distribution, firmware behavior, or correlated event pattern present in a live fleet. The numerical results should therefore be interpreted as a controlled comparative benchmark and verified on operational data before deployment.

Temporal observability is also limited. The SMART input is an endpoint snapshot, so BiGRU-MHA models ordered feature interactions rather than observed day-by-day degradation. Stronger claims about temporal forecasting require daily histories such as those described for the larger Alibaba SMART collection [1] and evaluated in multi-view, mutation-based, wear-aware, or temporal-contextual SSD studies [6], [7], [8], [9].

The test set contains 42 failures. This is appropriate for a rare-event scenario, but a difference of only a few recovered devices can materially change recall, F1, and PR-AUC. Multi-seed evaluation, bootstrap confidence intervals, and longer observation windows would provide more stable uncertainty estimates. Drive-family estimates are especially sensitive because several families contain only two to five test failures.

Feature semantics require hardware review. SMART identifiers and normalized meanings can vary by vendor and controller generation. The fixed dictionary improves readability and supports consistent aggregation, but it does not establish a physical cause. Finally, the evaluation is offline: it does not model spare inventory, replacement lead time, maintenance windows, correlated rack risk, or the economic cost of false alarms and missed failures. Those constraints should be integrated before an alert score is converted into an automated maintenance action.

6. Conclusion

This study presents an imbalance-aware comparison of linear, tree-based, and attention-gated neural models for SSD failure screening with SMART telemetry and fixed feature semantics. The 30,000-drive benchmark contains 280 failures across 11 drive models, creating a 0.933% positive rate that makes ranking, alert budgets, and calibration more informative than accuracy.

Balanced logistic regression achieves the highest PR-AUC, LightGBM delivers the strongest thresholded recall and non-linear-model F1, and BiGRU-MHA provides the best precision at the strictest alert budget. Semantic aggregation improves the unweighted LightGBM ranking, whereas class weighting expands minority coverage at the cost of additional false alarms. These results show that

representation and operating policy must be evaluated separately.

The broader finding is that model complexity should follow the available evidence. Attention-gated recurrent modeling can be useful for concentrated top-risk screening, but a one-day snapshot does not offer the temporal information needed to guarantee an advantage over strong tabular baselines. A practical next step is to apply the same protocol to complete daily SMART histories, add uncertainty intervals across time and drive families, and couple calibrated risk with explicit maintenance-cost and inventory constraints.

References

[1] Alibaba, "dcbrain: Data center operation datasets," GitHub repository, 2020. [Online]. Available: Alibaba-edu/dcbrain.

[2] 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.

[3] 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.

[4] 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. (ATC), 2019, pp. 961–976.

[5] S. Han, J. Wu, E. Xu, C. He, P. P. C. Lee, Y. Qiang, Q. Zheng, T. Huang, Z. Huang, and R. Li, "Robust data preprocessing for machine-learning-based disk failure prediction in cloud production environments," arXiv:1912.09722, 2019.

[6] 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," in Proc. 21st USENIX Conf. File and Storage Technologies (FAST), 2023, pp. 409–424.

[7] Y. Zhang et al., "MSFRD: Mutation similarity based SSD failure rating and diagnosis for complex and volatile production environments," in Proc. USENIX Annual Technical Conf. (ATC), 2024, pp. 869–884.

[8] Y. Gu, C. Wu, and X. He, "Exploit both SMART attributes and NAND flash wear characteristics for SSD failure prediction," in Proc. USENIX Annual Technical Conf. (ATC), 2024, pp. 1101–1117.

[9] 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. 154449–154464, 2024, doi: 10.1109/ACCESS.2024.3482368.

[10] 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.

[11] 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.

[12] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, "Machine learning methods for predicting failures in hard drives: A multiple-instance application," J. Mach. Learn. Res., vol. 6, pp. 783–816, 2005.

[13] M. M. Botezatu, I. Giurgiu, J. Bogojeska, and D. Wiesmann, "Predicting disk replacement towards reliable data centers," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 39–48.

[14] L. M. Grupp, J. D. Davis, and S. Swanson, "The bleak future of NAND flash memory," in Proc. 10th USENIX Conf. File and Storage Technologies (FAST), 2012.

[15] N. Mielke, T. Marquart, N. Wu, J. Kessenich, H. Belgal, E. Schares, F. Trivedi, E. Goodness, and L. R. Nevill, "Bit error rate in NAND flash memories," in Proc. IEEE Int. Reliability Physics Symp. (IRPS), 2008, pp. 9–19.

[16] S. Maneas, K. Mahdaviani, T. Emami, and B. Schroeder, "A study of SSD reliability in large scale enterprise storage deployments," in Proc. 18th USENIX Conf. File and Storage Technologies (FAST), 2020, pp. 137–149.

[17] J. Meza, Q. Wu, S. Kumar, and O. Mutlu, "A large-scale study of flash memory failures in the field," in Proc. ACM SIGMETRICS, 2015, pp. 177–190, doi: 10.1145/2745844.2745848.

[18] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Ann. Statist., vol. 29, no. 5, pp. 1189–1232, 2001.
[19] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
[20] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[21] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in Adv. Neural Inf. Process. Syst., 2017, pp. 3146–3154.
[22] K. Cho, B. van Merriënboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in Proc. EMNLP, 2014, pp. 1724–1734.
[23] A. Vaswani et al., "Attention is all you need," in Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.
[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 2980–2988.
[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.
[26] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, 2009.
[27] 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.
[28] 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, e0118432, 2015.
[29] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in Proc. 22nd Int. Conf. Machine Learning (ICML), 2005, pp. 625–632.
[30] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2002, pp. 694–699.
[31] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in Advances in Large Margin Classifiers. Cambridge, MA, USA: MIT Press, 1999.
[32] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in Proc. 34th Int. Conf. Machine Learning (ICML), 2017, pp. 1321–1330.
[33] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[34] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Adv. Neural Inf. Process. Syst., 2017, pp. 4765–4774.
[35] J. Jin, T. Huang, and S. Lu, "Cost-sensitive learning, simulated PU learning, and one-class autoencoding for extreme-imbalance credit card fraud detection," J. Adv. Comput. Syst., vol. 4, no. 6, pp. 64–73, Jun. 2024, doi: 10.69987/JACS.2024.40605.
[36] 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," J. Adv. Comput. Syst., vol. 4, no. 4, pp. 80–96, Apr. 2024, doi: 10.69987/JACS.2024.40407.
[37] 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," J. Adv. Comput. Syst., vol. 4, no. 6, pp. 74–85, Jun. 2024, doi: 10.69987/JACS.2024.40606.
[38] H. Zhou and S. Zhao, "LLM-explanation-enhanced retail credit default prediction with gradient boosting on the UCI default of credit card clients dataset," J. Adv. Comput. Syst., vol. 4, no. 5, pp. 102–118, May 2024, doi: 10.69987/JACS.2024.40508.
[39] G. Liu, S. He, and I. Liu, "LLM-augmented multi-source root cause attribution for CPU and network faults in microservices," J. Adv. Comput. Syst., vol. 3, no. 6, pp. 39–57, Jun. 2023, doi: 10.69987/JACS.2023.30604.

[40] B. Zhang, H. Rao, and D. Zhao, "Evidence-grounded RAG for cloud-native DevOps: Hallucination-resistant AIOps question answering over private operations documents," J. Adv. Comput. Syst., vol. 4, no. 3, pp. 109–125, Mar. 2024, doi: 10.69987/JACS.2024.40308.

[41] 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," J. Adv. Comput. Syst., vol. 4, no. 4, pp. 97–111, Apr. 2024, doi: 10.69987/JACS.2024.40408.

[42] J. Nie and D. Zheng, "Noisy-neighbor-aware VM degradation risk modeling with unsupervised residual fusion," J. Adv. Comput. Syst., vol. 4, no. 4, pp. 112–123, Apr. 2024, doi: 10.69987/JACS.2024.40409.

[43] S. He, H. Tu, and I. Liu, "Safe PD capacity forecasting with time-series foundation models and calibrated uncertainty for heterogeneous GPU clusters," J. Adv. Comput. Syst., vol. 3, no. 4, pp. 48–66, Apr. 2023, doi: 10.69987/JACS.2023.30404.

[44] S. He, X. Chang, and E. Sun, "Cross-cloud transfer learning for AI training capacity forecasting under workload and topology distribution shift," J. Adv. Comput. Syst., vol. 4, no. 1, pp. 100–120, Jan. 2024, doi: 10.69987/JACS.2024.40108.

[45] S. Chen, S. He, and E. Sun, "Risk-bounded GPU resource oversubscription via conformal demand envelopes in production AI clusters," J. Adv. Comput. Syst., vol. 4, no. 5, pp. 119–134, May 2024, doi: 10.69987/JACS.2024.40509.

[46] S. Zhao, J. Bai, and D. Roberson, "Multi-horizon GPU demand forecasting with workload semantics and operational risk curves: An empirical study on Alibaba Clusterdata GPU trace," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 544–571, Dec. 2025, doi: 10.51903/jtie.v4i3.498.

[47] S. Lu and D. Zhou, "TinyLLM-assisted intrusion detection for real-time IoT networks," J. Adv. Comput. Syst., vol. 4, no. 8, pp. 72–87, Aug. 2024, doi: 10.69987/JACS.2024.40809.

[48] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT traffic classification and anomaly detection method based on deep autoencoders," in Proc. 6th Int. Conf. Computing and Data Science (CDS), 2024.

[49] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive optimization of DDoS attack mitigation in distributed systems using machine learning," in Proc. 6th Int. Conf. Computing and Data Science (CDS), 2024, pp. 89–94.

[50] Q. Xin, "Self-supervised log anomaly detection with LogBERT-style transformers: Full empirical evaluation on a reproducible SynHDFS benchmark," J. Electr. Eng. Comput. Sci., vol. 11, no. 1, pp. 23–35, May 2026, doi: 10.54732/jeecs.v11i1.3.