Combining Supervised and Unsupervised Learning on SMART Data for Hard Drive Failure Assessment

Luka Petek
University of Maribor, Faculty of Electrical Engineering and Computer Science
Maribor, Slovenia
luka.petek1@student.um.si

Vili Podgorelec
University of Maribor, Faculty of Electrical Engineering and Computer Science
Maribor, Slovenia
vili.podgorelec@um.si

Abstract

Unexpected disk failure is one of the most common causes of downtime in storage systems and data loss. However, the built-in SMART warnings on modern drives catch only a small fraction of actual failures. Manufacturer-embedded SMART thresholds detect only 3–10% of failures at acceptably low false-alarm rates. Machine learning can be used to improve failure assessment. Classical methods (SVM or Naive Bayes) perform reasonably well on SMART data, but most of them rely on a single model and do not combine different learning strategies. Four models spanning supervised and unsupervised learning are combined: a random forest classifier, an unsupervised autoencoder-based anomaly detector, a classifier built on the autoencoder bottleneck, and an HDBSCAN clustering model. Each model captures a different kind of failure signal and has its own interpretational value. All 4 model scores are combined into a single Aggregated Health Index (AHI). On unseen evaluation data from the first quarter of 2026, AHI achieves a ROC-AUC of 0.923 and a PR-AUC of 0.936, and the final AHI can be decomposed into the individual contributions of each model, providing an explanation of the assessment.

Keywords

hard drive failure assessment, SMART data, hybrid model, autoencoder, anomaly detection, HDBSCAN clustering, interpretable machine learning

1 Introduction

Hard disk drives are among the most frequently replaced components in storage systems, and their unexpected failure is a leading cause of data loss, service unavailability, and increased operational cost [1]. To provide early warning, drive manufacturers embed Self-Monitoring, Analysis, and Reporting Technology (SMART) into virtually every modern drive. However, the threshold-based failure flags built into SMART firmware are deliberately conservative: manufacturers report failure detection rates of only 3–10% at an acceptable false-alarm rate of roughly 0.1% per year [2], leaving the majority of failures undetected until it is too late.

Machine learning methods are a viable approach to improving these basic failure signals. Various machine learning methods and models have been recently used to assess hard drive condition based on historical operational data, specifically SMART (Self-Monitoring Analysis and Reporting Technology) attributes [3]. However, each machine learning method has a distinct blind spot. No single model covers all failure modes: gradual degradation may go unnoticed by a supervised classifier trained on binary labels, while explicit failure signals may be missed by an unsupervised anomaly detector that never saw them. Murray et al. [2] established a foundational comparison between SVMs and clustering against manufacturer thresholds but did not integrate the paradigms. Ahmed and Green [4] addressed class imbalance with a cost-aware LSTM but remained within a single supervised paradigm. Xu et al. [5] predicted a continuous health degree with noise-based training in a semi-supervised setting, while Zhou et al. [6] used reconstruction-based anomaly detection to identify deterioration modes – both, however, committed to one learning strategy. This is where a hybrid approach that combines multiple paradigms into a single, interpretable output stands out.

The main contributions of this paper are:

- Four models spanning supervised and unsupervised learning – a random forest classifier, an unsupervised autoencoder anomaly detector, a classifier built on the autoencoder bottleneck, and an HDBSCAN clustering model – are combined into a single Aggregated Health Index (AHI).
- All four per-component scores remain visible alongside AHI, ensuring the final output is interpretable and traceable to each model's contribution.
- Evaluation results and the trade-offs of using this hybrid as a small, fully offline pipeline for data-centre and edge deployment are reported.

The remainder of this paper is organised as follows. Section 2 reviews related work on SMART-based failure prediction. Section 3 describes the dataset, preprocessing, the four component models, and the AHI aggregation formula. Section 4 reports evaluation results. Section 5 discusses deployment use cases, trade-offs, and limitations, while Section 6 concludes the paper.

2 Related Work

Machine learning (ML) applied to raw SMART attribute values, rather than the manufacturer's own threshold flags, has been shown to improve detection rates [2, 1]. Yet ML-based disk failure prediction remains fragmented. Supervised classifiers, such as the SVMs used by Murray et al. [2], achieve high precision when enough labelled failures are available but struggle with the extreme class imbalance typical of SMART datasets (often exceeding 10,000:1 [4]). Unsupervised anomaly detectors, such as autoencoders trained on healthy drives only, do not require failure labels but cannot exploit known failure patterns when they exist. Clustering approaches group drives by operating regime and provide population-level risk context, but do not output a direct failure probability. Semi-supervised methods, such as the reconstruction-based approach of Zhou et al. [6] and the health-degree prediction of Xu et al. [5], attempt to bridge this gap but still rely on a single learning strategy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.97

3 Data and Methodology

Dataset. The study uses Backblaze's 2025 hard drive SMART dataset [7], which covers daily records from the prepared 2024-10-01–2025-09-30 period. The fleet is mostly HDDs [8], and failures are rare compared to healthy records (high class imbalance), which was one of the main challenges.

Two dataset variants. Two training datasets were prepared:

- *Healthy-only sample* (for autoencoder training): up to 1,000 healthy rows per daily CSV were sampled, giving approximately 362,463 healthy rows, split by serial number into 289,982 for training and 72,481 for validation and threshold selection. No failure rows were used.
- *Balanced 50:50 set* (for supervised models and clustering): all 4,414 failure rows plus 4,414 healthy rows sampled at random, giving 8,828 records total. Used for training the random forest and bottleneck classifier, and for learning HDBSCAN clusters.

Prediction target. Each record is a single daily SMART snapshot. Positive examples are failure-day records — the daily report on which a drive was marked as failed. Healthy instances are snapshots from drives that did not fail on that day. The system does not track a drive over multiple days and does not use a defined lead time (e.g. 1 or 7 days before failure). A healthy label on one day therefore does not imply long-term health; it only means the drive had not failed by that date.

Preprocessing. Capacity bytes are converted to gigabytes, an SSD indicator is derived from the model name, missing counters are filled with 0, medians are used for a few mechanical counters, and three engineered features are added: `any_critical_error`, `total_error_count`, and `error_per_gb`. Neural models use MinMax scaling; the scikit-learn model uses the same processed features without the NN scaler.

Features. A set of 19 features is used:

- `capacity_gigabytes`, SSD indicator
- SMART raw counters: 1, 3, 4, 5, 7, 9, 12, 187, 188, 191, 192, 193, 197, 198
- Engineered: `any_critical_error`, `total_error_count`, `error_per_gb`

Models. The full implementation is available at [9]. All four models start from the same 19 processed features but differ in their input representation and training data:

(1) *Random forest* (scikit-learn): input is the 19 features plus manufacturer one-hot encoding. Trained on the balanced 8,828-row dataset, which is grouped by serial numbers, with train/test splitting and GroupKFold cross-validation. Outputs a failure probability in $[0, 1]$.
(2) *Anomaly autoencoder* (TensorFlow): input is the 19 features after MinMax scaling, passed through a symmetric encoder–decoder with a 12-dimensional bottleneck. Trained only on 289,982 healthy rows; 72,481 healthy rows used for validation and threshold selection (99th percentile of reconstruction error). Failure rows were never used for training. Outputs a 0–1 anomaly score.
(3) *Bottleneck classifier* (TensorFlow, two-stage): a separate autoencoder with a symmetric encoder–decoder and an 8-dimensional bottleneck is trained on the same healthy training rows, grouped by serial numbers. Its 8-dimensional encoder is then frozen. A small dense classifier with hidden layers of 16 and 8 units and a single sigmoid output is trained on the 8-dimensional bottleneck features of the balanced 8,828-row dataset, split 70/15/15 (train/val/test). Output is a failure probability in $[0, 1]$.
(4) *HDBSCAN clustering*: input is the same 8-dimensional bottleneck from the classifier autoencoder. Fitted on the serial-grouped training partition; held-out serials are used for internal evaluation. Each cluster is assigned a risk score equal to its failure rate in the training data.

The bottleneck classifier and HDBSCAN share the same 8-dimensional encoder, so their outputs are not independent. The anomaly autoencoder is a separate model with its own 12-dimensional bottleneck.

Aggregation. To obtain a single score, an Aggregated Health Index (AHI) is computed as a weighted root-mean-square of the four component scores, then clipped for stability. Fixed manual weights were used (RF 0.30, bottleneck classifier 0.40, anomaly 0.20, clustering 0.10), and the final AHI can be explained from per-component contributions.

4 Results

As this is a hybrid approach, each of the 4 components is evaluated independently first, and their combined performance is then assessed using the AHI. For a direct comparison, all components and AHI are evaluated on exactly the same strict-unseen records from the first quarter of 2026. The main metrics are summarised in Table 1. The evaluation uses equal numbers of failure and healthy records, so this balance does not represent the real fleet.

Table 1: All components and AHI evaluated on the same strict-unseen first quarter of 2026 records (1,018 failure and 1,018 healthy records).

Model	ROC-AUC	PR-AUC	Recall	FPR	Precision	F1
Random Forest	0.922	0.938	0.812	0.067	0.924	0.865
Bottleneck classifier	0.906	0.912	0.772	0.086	0.899	0.831
Anomaly detector	0.737	0.742	0.522	0.064	0.891	0.658
HDBSCAN cluster risk	0.853	0.816	0.820	0.227	0.783	0.801
AHI	0.923	0.936	0.783	0.091	0.896	0.835

4.1 Random Forest Classifier

A random forest classifier was trained on a balanced 50:50 dataset using serial-number-grouped splitting and GroupKFold cross-validation. On the common evaluation set from the first quarter of 2026, it reaches ROC-AUC 0.922, PR-AUC 0.938, recall 0.812, precision 0.924, and F1 0.865. The model catches most failing drives while keeping false alarms relatively low. The random forest results are solid and it has the highest F1 and precision among the listed components, while AHI has a slightly higher ROC-AUC.

4.2 Autoencoder Anomaly Detector

The anomaly detection model was trained only on healthy rows and evaluated on the common evaluation set from the first quarter of 2026. Reconstruction error reasonably separates failures from healthy records, with a ROC-AUC of 0.737 and PR-AUC of 0.742. However, with a failure recall of 0.522, precision of 0.891, and F1 of 0.658, the model still misses some actual failures. Its main value is that it does not require any failure labels during training, and it catches a different kind of problem than supervised classifiers do, which could potentially contribute to the overall performance of the combined model.

4.3 Bottleneck Classifier

The bottleneck classifier is a strong individual model and consists of two parts. First, the autoencoder compresses the 19 input features into an 8-dimensional bottleneck — a latent data representation — and then a small dense classifier is trained on those 8 dimensions using a balanced 50:50 dataset. On the common evaluation set from the first quarter of 2026, it reaches ROC-AUC 0.906, PR-AUC 0.912, recall 0.772, precision 0.899, and F1 0.831. Its internal grouped test metadata reports ROC-AUC 0.928 and PR-AUC 0.940, but the common first quarter of 2026 result is the primary comparison because every component is evaluated on the same records.

4.4 HDBSCAN Clustering

The clustering model also works on the same 8-dimensional bottleneck data representation from the autoencoder. HDBSCAN [10, 11] is applied directly in that 8-dimensional data space; the number of clusters is determined automatically from the data density. In the grouped training partition, 21 clusters and one outlier group emerged, with 20.7% of points classified as noise. Each cluster is given a risk score equal to its failure rate in the training data; the outlier group had a 66.4% failure rate. When evaluating a new disk, it is placed into a cluster based on the local density structure using HDBSCAN's `approximate_predict` procedure. On the common evaluation set from the first quarter of 2026, cluster risk reaches ROC-AUC 0.853, PR-AUC 0.816, recall 0.820, precision 0.783, and F1 0.801.

4.5 Aggregated Health Index

The Aggregated Health Index (AHI) is simply a weighted root-mean-square (RMS) formula that combines all 4 model scores into one final value representing the current risk state of the disk:

$$\mathrm{AHI} = \sqrt{\frac{w_K \cdot K^2 + w_R \cdot R^2 + w_A \cdot A^2 + w_C \cdot C^2}{\sum w}} \times 100 \quad (1)$$

where K is the random forest failure probability ($w_K = 0.30$), R is the bottleneck classifier failure probability ($w_R = 0.40$), A is the autoencoder anomaly score normalized to $[0, 1]$ ($w_A = 0.20$), and C is the empirical failure rate of the assigned cluster ($w_C = 0.10$). All scores are within the range $[0, 1]$, weights sum to 1, and the result is multiplied by 100 to express AHI on a 0–100 index. The final value is clipped to $[3,\ 97]$ to avoid extreme boundary values. AHI is a risk index, not a calibrated probability: higher values indicate a more failure-like current SMART state, but the RMS aggregation means the output is not calibrated against real-world failure rates. Values below 45 mean *Healthy*, 45–65 is *Warning*, and above 65 is *Critical*.

One property of the RMS formula is that a single strong signal from a low-weight component may not be enough to cross a threshold on its own. For example, if the anomaly score is 1 and all other scores are 0, AHI is $\sqrt{0.20} \times 100 \approx 44.7$, which remains in the Healthy range. This is a design trade-off: RMS amplifies large individual signals, but the weight still scales the contribution, so a single low-weight component cannot dominate the final score by itself.

AHI was evaluated on a strict-unseen evaluation set from the first quarter of 2026 [12] containing 1,018 failure-day records and 1,018 healthy records. Figure 1 shows the AHI scores for both groups.

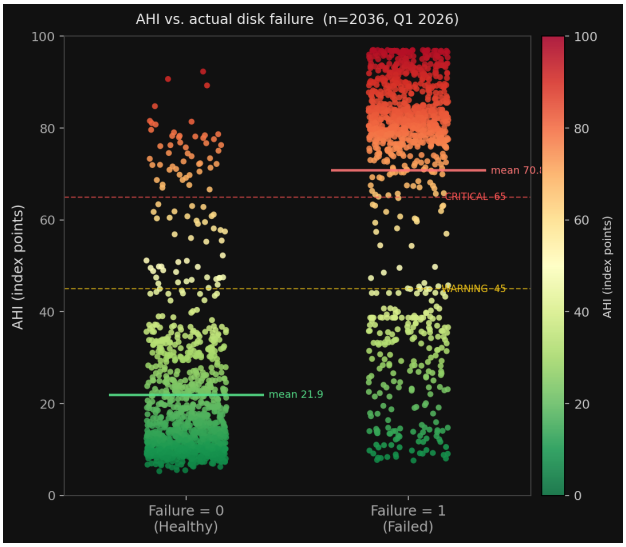


Figure 1: AHI scores for the strict-unseen evaluation from the first quarter of 2026. Each point is one daily SMART record; the vertical axis shows AHI index points.

AHI reaches ROC-AUC 0.923, PR-AUC 0.936, recall 0.783, precision 0.896, and F1 0.835 at the fixed 45-point threshold. Because the evaluation set is balanced by design, its precision must not be interpreted as precision at the natural fleet failure prevalence.

5 Discussion

The results show that no single model is the best at everything. The random forest has the highest precision and F1, while AHI has the highest ROC-AUC and PR-AUC. The autoencoder-based anomaly detector catches a different kind of failure – one it can find even without seeing any failure examples during training. The clustering model adds something different on top of that: each cluster has a human-readable description of its dominant SMART pattern, so both the AHI score and the type of disk behaviour causing it can be examined. The main value of the hybrid approach is not the final AHI alone, but also the ability to look at each of the four component scores individually and critically assess what each model is saying about the drive.

The evaluation is balanced and uses failure-day positives, so it measures current-state rather than future failure prediction. It does not by itself show that AHI is always better than the individual models. The specific causes of false negatives remain unknown. Also, because this evaluation set is balanced, its precision is not the same as in a real fleet where failures are much less common. For example, if failures represent 0.1%, 0.5%, or 1% of records, the estimated precision would be approximately 0.85%, 4.1%, and 8.0%, respectively.

Two further limitations are worth noting. First, all training data comes from a single storage provider (Backblaze), so results may not generalise directly to other hardware vendors or real-life scenarios. Second, the system outputs the current health state of a drive rather than an estimated probability or time of a future failure — remaining life estimation is left for future work.

The implementation also includes a React dashboard that shows AHI, the four component scores, disk information, and analysis history. Setup instructions, a dashboard screenshot, and API usage are available in the GitHub README [9]. The pipeline

works locally and does not need a remote service. A small CPU test on the development computer showed the model files used to evaluate one disk were about 7.1 MB in total. Evaluating one disk took about 0.20 s. The program used about 0.40 GB of memory after loading the models and about 0.95 GB during normal evaluation. Testing on real NAS or edge hardware is left for future work.

6 Conclusion and Future Work

This paper presents and evaluates a hybrid system that combines four models into a single Aggregated Health Index (AHI): a random forest classifier, an autoencoder-based anomaly detector, a bottleneck classifier, and an HDBSCAN clustering model. Each model captures a different kind of failure signal, and the weighted RMS formula combines these signals into one current-state index. On the same 2,036 unseen records from the first quarter of 2026, AHI achieved ROC-AUC 0.923, PR-AUC 0.936, precision 0.896, recall 0.783, and F1 0.835 at the fixed 45-point threshold. These results support current-state assessment, but they do not show how likely a drive is to fail in the future, how early a failure could be detected, or that AHI is always better than each individual model.

For future work, it would be good to tune the weights based on data rather than setting them manually, perform a full SMART feature analysis for each cluster to improve the cluster descriptions, and extend the system to NVMe drives, which have a different set of SMART attributes than traditional HDDs. Evaluation on a naturally imbalanced dataset that reflects real fleet failure rates is also left for future work.

References

[1] Sidi Lu, Bing Luo, Tirthak Patel, Yongtao Yao, Devesh Tiwari, and Weisong Shi. 2020. Making disk failure predictions smarter! In *Proceedings of the 18th USENIX Conference on File and Storage Technologies (FAST)*. USENIX Association, 359–372. https://www.usenix.org/conference/fast20/presentation/lu.

[2] Joseph F. Murray, Gordon F. Hughes, and Kenneth Kreutz-Delgado. 2005. Machine learning methods for predicting failures in hard drives: a multiple-instance application. *Journal of Machine Learning Research*, 6, 5, 783–816. https://www.jmlr.org/papers/v6/murray05a.html.

[3] Elizabeth Atekoja. 2024. Prediction of hard drive failure using machine learning. *Global Journal of Computer Science and Technology: D Neural Artificial Intelligence*, 24, 1.

[4] Jishan Ahmed and Robert C. Green II. 2024. Cost aware lstm model for predicting hard disk drive failures based on extremely imbalanced s.m.a.r.t. sensors data. *Engineering Applications of Artificial Intelligence*, 127, Part B, 107339. doi:10.1016/j.engappai.2023.107339.

[5] Chuanmao Xu, Guangyan Wang, Xie Liu, Deke Guo, and Tian-Yi Liu. 2018. Layerwise perturbation-based adversarial training for hard drive health degree prediction. *arXiv preprint arXiv:1809.04188*. https://arxiv.org/abs/1809.04188.

[6] Hao Zhou, Zhiheng Niu, Gang Wang, Xiaoguang Liu, Dongshi Liu, Bingnan Kang, Zheng Hu, and Yong Zhang. 2024. Proactive drive failure prediction for cloud storage system through semi-supervised learning. *IEEE Transactions on Dependable and Secure Computing*, 21, 4, 1528–1543. doi:10.1109/TDSC.2023.3286093.

[7] Backblaze Inc. 2025. Hard drive test data — open dataset. https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data. Daily SMART snapshots from Backblaze's production storage fleet. (2025).

[8] Andy Klein, Stephanie Doyle, and David Johnson. 2026. Backblaze drive stats for 2025. https://www.backblaze.com/blog/backblaze-drive-stats-for-2025/. Annualized failure rates and lifetime statistics for 344,196 drives. (2026).

[9] Luka Petek and Vili Podgorelec. 2026. DiskGuard: hybrid hard drive failure assessment using SMART data. https://github.com/Luka-Petek/DiskGuard. (2026).

[10] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining (PAKDD 2013)* (Lecture Notes in Computer Science). Vol. 7819. Springer, 160–172. doi:10.1007/978-3-642-37456-2_14.

[11] Leland McInnes, John Healy, and Steve Astels. 2017. Hdbscan: hierarchical density based clustering. *The Journal of Open Source Software*, 2, 11, 205. doi:10.21105/joss.00205.

[12] Backblaze Inc. 2026. Hard drive test data — first quarter 2026. https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data. Daily SMART snapshots from 1 January to 31 March 2026. (2026).