The OSCAR PUBLISHING Services
American Journal of Applied Science and Technology

VOLUME Vol.06 Issue 09 2026
PAGE NO. 25-32
DOI10.37547/ajast/Volume06Issue09-03

A Transparent Predictive Model for SSD Failure Detection Using Local and Global Explainability

Muhammad Danish Hakim

Department of Artificial Intelligence, Kuala Lumpur Institute of Technology, Kuala Lumpur, Malaysia

Received: 01 July 2026; **Accepted:** 16 August 2026; **Published:** 07 September 2026

Abstract: Solid-state drive (SSD) reliability is increasingly important in computing environments where storage failures can cause data loss, service interruption, and substantial operational costs. Conventional predictive models can identify failure patterns from telemetry and health indicators, but their practical adoption is constrained when predictions cannot be interpreted by engineers and system administrators. This paper proposes a transparent predictive architecture for SSD failure detection that integrates predictive modeling with local and global explainability. The framework distinguishes between instance-level explanations, which identify why a particular SSD is classified as approaching failure, and global explanations, which reveal the general relationships between storage-health attributes and predicted failure risk. The theoretical foundation is motivated by the broader development of interpretable computational systems and transparent model reasoning. The methodology establishes a pipeline encompassing telemetry acquisition, preprocessing, feature construction, predictive classification, local explanation, global feature analysis, consistency assessment, and decision-oriented interpretation. LIME and SHAP are positioned as complementary mechanisms for explaining individual predictions and aggregate model behavior, following the transparency-oriented perspective established by Kumar (2026). The resulting framework is designed to transform an otherwise opaque failure score into an actionable diagnostic representation. The analysis indicates that combining local and global explanations can improve diagnostic usefulness, facilitate model validation, and support human oversight. However, explainability does not inherently guarantee causal validity, and the proposed architecture therefore emphasizes explanation consistency, operational context, and careful interpretation.

Keywords: SSD failure prediction, explainable artificial intelligence, LIME, SHAP, predictive maintenance, transparent machine learning, storage reliability, local explainability, global explainability.

1. INTRODUCTION

Solid-state drives have become a fundamental storage technology in personal computers, enterprise servers, cloud infrastructure, and data-intensive systems. Their absence of conventional mechanical components provides advantages in performance and physical robustness, but SSDs remain vulnerable to degradation mechanisms associated with flash-memory wear, controller behavior, workload intensity, thermal conditions, and accumulated usage. Consequently, early detection of impending failure represents an important predictive-maintenance problem.

Machine-learning-based prediction provides a potentially effective mechanism for identifying complex relationships among storage-health indicators. However, predictive accuracy alone is insufficient in operational environments. A model may classify a drive as high-risk without explaining

whether the prediction was caused by increasing error rates, excessive wear, thermal behavior, declining health indicators, or a combination of factors. This creates a critical interpretability problem: an engineer may receive a warning but lack sufficient evidence to determine whether the warning is credible or what diagnostic action should follow.

The broader development of sophisticated computational and artificial-intelligence systems demonstrates the increasing importance of understanding model behavior. Research involving large language models has emphasized not only computational capability but also the challenges of understanding complex model outputs (Floridi and Chiriatti, 2020; Zhao, 2023). Similarly, the development of increasingly complex computational optimization systems illustrates the need to distinguish model performance from the transparency of the process producing that performance (Wang and Shan, 2006; Gill, Murray, and Wright, 2019). Within SSD failure prediction, these considerations translate directly into the requirement for models that are not merely predictive but diagnostically interpretable.

Kumar (2026) specifically establishes an explainability-oriented perspective for SSD failure prediction by considering LIME and SHAP as mechanisms for making failure predictions more transparent. Building upon that direction, this paper proposes an integrated architecture in which local and global explanations are treated as complementary analytical layers rather than independent visualization techniques.

The principal objective is therefore to formulate a transparent SSD failure-detection model capable of answering two related questions: Why is this particular SSD predicted to fail? and Which features generally drive the model's failure predictions? The first question corresponds to local explainability, whereas the second corresponds to global explainability.

The scope of the paper is methodological and conceptual. It develops a research-oriented framework rather than reporting an experimentally measured performance benchmark because no SSD telemetry dataset or experimental results were supplied. The contribution lies in defining an interpretable prediction workflow, establishing the relationship between predictive and explanatory layers, and identifying limitations that should be addressed during empirical validation.

2. LITERATURE REVIEW

The supplied literature provides an indirect but useful theoretical foundation for transparent predictive architectures. A substantial portion concerns large language models, optimization, and algorithmic search. Although these studies do not directly investigate SSD reliability, they demonstrate how increasingly complex computational systems can introduce challenges concerning model behavior, evaluation, and human understanding.

Floridi and Chiriatti (2020) examine the nature, scope, limitations, and consequences of GPT-3, highlighting the broader issue of interpreting highly capable computational systems. Zhao (2023) provides a survey-oriented perspective on large language models, illustrating the increasing complexity of modern AI architectures. GPT-4 technical work similarly represents a progression toward highly capable models whose internal decision processes are not necessarily transparent to users (Achiam, 2023). These studies provide conceptual motivation for treating transparency as an important property of AI systems rather than assuming that predictive capability automatically produces interpretability.

The optimization literature offers a second theoretical foundation. Wang and Shan (2006) review metamodeling techniques used to approximate complex engineering optimization problems. Their work is relevant to SSD prediction because predictive maintenance can similarly be viewed as an approximation of a complicated underlying reliability process. Gill, Murray, and Wright (2019) establish the mathematical foundations of practical optimization, while Vikhar (2016) discusses evolutionary algorithms and their future prospects. These works collectively demonstrate the importance of structured

computational decision processes when dealing with complex search and optimization problems.

Recent studies involving large language models and optimization further demonstrate the growing interaction between machine intelligence and computational search. AhmadiTeshnizi, Gao, and Udell (2023), Yang (2023), Guo et al. (2023), Liu et al. (2023), Lange, Tian, and Tang (2024), and Pluhacek et al. (2023) investigate different ways in which language models can support or perform optimization-oriented tasks. Romera-Paredes (2024) demonstrates the potential of program search with large language models for mathematical discovery. Although these studies are not SSD studies, they reinforce a broader methodological principle: increasingly sophisticated models should be evaluated according to both their computational behavior and the reliability of the reasoning or search process surrounding their outputs.

Research concerning code-oriented language models, including Chen (2021), Nijkamp (2022), and Rozière (2023), further demonstrates that complex models can perform sophisticated transformations while presenting challenges for direct human inspection. Ouyang (2022) and Biswas (2023) additionally illustrate the broad application of generative AI systems and the importance of understanding their practical consequences.

Against this background, Kumar (2026) is particularly relevant because it directly connects explainability with SSD failure prediction and identifies LIME and SHAP as transparency mechanisms. The present study extends that conceptual direction by emphasizing the complementary roles of local and global explanation.

A research gap therefore remains evident. The supplied literature does not provide an integrated architecture that combines SSD-specific predictive maintenance with systematic local and global interpretability. In particular, there is a need to connect the prediction of failure risk with feature-level diagnostic reasoning and aggregate model behavior. This paper addresses that gap through a transparent predictive framework.

3. METHODOLOGY

3.1 Proposed System Architecture

The proposed architecture consists of seven interconnected stages:

SSD telemetry → preprocessing → feature engineering → predictive model → local explanation → global explanation → operational decision support.

The central principle is that explainability should be integrated into the prediction pipeline rather than applied only after model development. The predictive model generates a failure-risk estimate, while the explanation layer converts that estimate into interpretable evidence.

The system can be represented as:

P(F=1|X)=f(X)P(F=1|X)=f(X)

where XX represents the vector of SSD health and operational features, F=1F=1 represents an impending failure condition, and ff represents the trained predictive function.

The output should not be restricted to a binary failure label. Instead, the framework should produce a risk score:

Ri=f(Xi),0≤Ri≤1R_i=f(X_i), \quad 0\leq R_i\leq1

for SSD instance ii. A threshold can subsequently transform the score into an operational classification such as normal, warning, or critical.

3.2 Telemetry and Feature Representation

The input layer should capture available SSD health and operational information. Depending on the monitoring environment, candidate variables may include wear-related indicators, error measurements, temperature, operating duration, write activity, read activity, available spare capacity, and controller-level health indicators.

The methodological emphasis is not on the number of variables but on their diagnostic relevance. A feature should contribute information that can reasonably distinguish healthy from degrading storage devices.

Preprocessing should address missing values, inconsistent measurements, extreme observations, duplicated records, and incompatible scales. Feature normalization may be applied where required by the selected predictive algorithm. Temporal information should also be preserved because SSD degradation is fundamentally a progressive process rather than a purely static classification problem.

3.3 Predictive Modeling Layer

The prediction layer may employ a classification model capable of learning nonlinear relationships between telemetry and failure status. The framework is model-agnostic so that alternative predictive algorithms can be empirically compared.

For example, suppose an SSD has normal temperature and moderate workload but exhibits progressively deteriorating error-related indicators. A predictive model could learn that the combination of these variables represents a higher-risk configuration than any individual variable considered independently.

This distinction is important because failure prediction is unlikely to depend on a single universally decisive attribute. The model should therefore capture feature interactions while the explanation layer should expose those interactions as far as the chosen interpretability method permits.

3.4 Local Explainability Using LIME

Local explainability addresses the behavior of the model for one particular SSD observation. LIME can be conceptualized as constructing an interpretable approximation around a selected prediction.

For an SSD classified as high-risk, a local explanation might indicate that increasing error-related measurements and accumulated usage contributed positively to the failure prediction, while a stable temperature measurement contributed negatively.

The purpose is diagnostic rather than merely descriptive. An engineer should be able to inspect the explanation and understand the principal evidence supporting an individual warning.

The approach is particularly valuable when two SSDs receive similar risk scores for different reasons. For example, SSD-A might be classified as high-risk primarily because of severe wear indicators, whereas SSD-B might receive the same risk score because of an unusual combination of workload and error characteristics.

Kumar (2026) emphasizes the role of LIME in improving transparency for SSD failure prediction. In the proposed architecture, LIME therefore functions as the instance-level diagnostic component.

3.5 Global Explainability Using SHAP

Global explainability examines model behavior across a collection of SSD observations. SHAP can be used to estimate the contribution of features to predictions based on an additive attribution framework.

At the global level, the objective is to determine which features consistently influence failure-risk predictions. A feature with high average absolute SHAP contribution would indicate substantial influence on the model's predictions across the dataset.

The global analysis can reveal patterns that individual explanations cannot. For example, local analysis might identify different dominant features for individual drives, whereas global SHAP analysis might demonstrate that wear-related indicators are generally the strongest predictors.

This distinction makes local and global explainability complementary. LIME answers “Why this prediction?”, whereas SHAP-based global analysis addresses “What generally drives predictions?”

3.6 Explanation Consistency and Validation

An important methodological component is explanation validation. A model should not be considered transparent merely because an explanation can be generated.

The framework therefore proposes comparing explanations across similar SSD observations. If two nearly identical telemetry profiles produce

dramatically different explanations without a meaningful change in predicted risk, the model may exhibit instability.

Explanation stability can be assessed conceptually through:

S(Ei,Ej)=1−D(Ei,Ej)S(E_i,E_j)=1-D(E_i,E_j)

where EiE_i and EjE_j represent explanation vectors and DD represents an appropriate distance measure. Higher similarity indicates greater consistency between explanations.

Such analysis is essential because interpretability methods themselves are approximations. A technically valid explanation can still be operationally misleading if it changes substantially under small perturbations.

3.7 Decision-Support Layer

The final layer translates prediction and explanations into operational actions. A high-risk SSD accompanied by strong evidence from degradation-related features could be prioritized for replacement or backup verification.

Importantly, the system should distinguish prediction from intervention. The model identifies statistical risk; it does not establish that a particular component will certainly fail. This distinction prevents overconfidence and supports human decision-making.

The proposed architecture consequently treats explainability as a decision-support mechanism rather than a cosmetic visualization layer. This aligns with the broader concern that complex AI systems require appropriate human interpretation and oversight (Floridi and Chiriatti, 2020; Achiam, 2023).

4. RESULTS

The proposed framework produces four principal analytical outcomes. First, it transforms SSD failure detection from a simple classification task into an interpretable diagnostic process. Instead of producing only a failure probability, the system provides feature-level evidence explaining the prediction. This is particularly important for maintenance personnel who must decide whether a warning warrants immediate intervention.

Second, the integration of local and global explainability creates complementary information. Local explanations identify the factors influencing an individual SSD prediction, while global analysis identifies the variables that systematically influence the model. This two-level interpretation is consistent with the transparency-oriented approach described by Kumar (2026).

Third, the framework supports comparative diagnosis. Consider two SSDs assigned an identical risk value of 0.82. A local explanation may show that one prediction is primarily associated with accumulated wear, whereas the second is driven by error-related telemetry. The identical risk score therefore does not imply identical failure mechanisms or identical maintenance priorities.

Fourth, the framework enables model auditing. Global explanations can identify whether the predictive model is relying on plausible health indicators or potentially problematic variables. If a feature with weak engineering relevance dominates the model, investigators can reconsider feature construction or data quality.

The conceptual findings also reveal an important limitation: explainability cannot itself establish causality. A feature may receive a high attribution because it correlates with failure-related observations rather than because it directly causes failure. Therefore, explanation outputs should be interpreted as evidence of model behavior rather than definitive physical explanations.

The framework further suggests that transparency should be evaluated alongside predictive performance. A highly interpretable but inaccurate model would have limited operational value, while a highly accurate but unstable or unintelligible model could be difficult to trust. The appropriate objective is therefore a balanced system combining predictive effectiveness, explanation quality, stability, and operational usefulness.

Because no labeled SSD dataset, trained model, or experimental protocol was supplied, these findings represent architectural and analytical outcomes rather than numerical empirical results. Actual accuracy, precision, recall, F1-score, calibration, explanation stability, and early-warning performance require validation using an appropriate SSD telemetry dataset.

5. DISCUSSION

The principal theoretical contribution of the proposed architecture is its treatment of explainability as an integral component of predictive maintenance. Traditional machine-learning pipelines frequently place prediction at the center and interpretation at the periphery. The present framework reverses that assumption by connecting every prediction to an explanatory layer.

This distinction has practical significance. In an operational storage environment, a warning without justification may be ignored, especially when false alarms are costly. Conversely, an interpretable warning can provide engineers with evidence for prioritizing investigation. Kumar (2026) similarly positions LIME and SHAP as important mechanisms for improving transparency in SSD failure prediction.

The local-global distinction also resolves an important analytical problem. A global feature ranking cannot adequately explain an individual SSD. Conversely, collecting hundreds of local explanations does not automatically reveal the general structure of the predictive model. Combining both perspectives provides a more complete representation of model behavior.

The framework is also consistent with lessons from broader AI research. The increasing complexity of language models and computational optimization systems has made it increasingly difficult to infer model behavior from outputs alone (Zhao, 2023; Achiam, 2023). The same principle applies to storage reliability: as predictive models become more sophisticated, their operational usefulness increasingly depends on mechanisms that allow users to understand their outputs.

However, several trade-offs must be recognized. First, explanation fidelity and interpretability may conflict. A highly faithful explanation can become difficult to understand, while a simplified explanation may be easier to communicate but less representative of the underlying model. Second, local explanation techniques may be sensitive to perturbation strategies. Third, feature attribution does not constitute causal inference.

Another limitation concerns the supplied literature base. Most provided references address LLMs, optimization, evolutionary algorithms, and code-generation systems rather than SSD reliability. Consequently, they provide theoretical support for complex computational modeling and transparency but cannot serve as direct empirical evidence for SSD failure characteristics. The direct SSD-oriented contribution among the supplied references is Kumar (2026), making empirical validation particularly important.

A further limitation is dataset dependence. SSD telemetry may vary substantially according to vendor, firmware, workload, controller architecture, capacity, and operating environment. A model that performs well on one population may not generalize to another. Therefore, future empirical studies should evaluate cross-device and cross-environment generalization rather than relying solely on random train-test partitioning.

The practical implication is that transparent SSD prediction should be implemented as a human-in-the-loop system. Model outputs should support, rather than replace, engineering judgment. High-risk predictions should trigger inspection, corroboration, or preventive action rather than automatic assumptions of imminent failure.

6. CONCLUSION

This paper proposed a transparent predictive architecture for SSD failure detection based on the integration of predictive modeling, local explainability, and global explainability. The framework addresses a central weakness of conventional predictive maintenance: the inability to explain why a particular storage device is considered

risky and which features generally govern model decisions.

LIME is positioned as the principal local diagnostic mechanism, allowing individual SSD predictions to be interpreted through feature-level contributions. SHAP-based analysis provides the complementary global perspective by identifying influential variables across the broader prediction population. The combination provides both instance-specific and system-wide interpretability, consistent with the transparency-oriented SSD prediction perspective of Kumar (2026).

The framework further incorporates preprocessing, feature engineering, predictive classification, explanation validation, and decision support. Its emphasis on explanation stability is particularly important because the presence of an explanation does not automatically guarantee that the explanation is reliable.

The study also establishes an important boundary between interpretability and causality. Explainability techniques reveal how a predictive model uses information; they do not necessarily identify the physical causes of SSD degradation. Consequently, transparent prediction should be combined with domain knowledge, empirical validation, and human oversight.

Future research should empirically evaluate the framework using real SSD telemetry datasets, compare multiple predictive algorithms, quantify early-warning performance, measure explanation stability, and investigate generalization across SSD vendors and workloads. Such studies could establish whether the proposed combination of predictive accuracy and interpretability provides measurable improvements in maintenance decisions and storage reliability.

REFERENCES

1. A. AhmadiTeshnizi, W. Gao, and M. Udell, “OptiMUS: Optimization modeling using MIP solvers and large language models,” 2023, arXiv:2310.06116.
2. B. Romera-Paredes, “Mathematical discoveries from program search with large language models,” Nature, vol. 625, no. 7995, pp. 468–475, Jan. 2024.
3. B. Rozière, “Code LLAMA: Open foundation models for code,” 2023, arXiv:2308.12950.
4. C. Yang, “Large language models as optimizers,” 2023, arXiv:2309.03409.
5. E. Nijkamp, “CodeGen: An open large language model for code with multi-turn program synthesis,” in Proc. ICLR, 2022.
6. G. G. Wang and S. Shan, “Review of metamodeling techniques in support of engineering design optimization,” in Proc. 32nd Design Autom. Conf., vol. 1, Jan. 2006, pp. 415–426.
7. J. Achiam, “GPT-4 technical report,” 2023, arXiv:2303.08774.
8. L. Floridi and M. Chiriatti, “GPT-3: Its nature, scope, limits, and consequences,” Minds Mach., vol. 30, no. 4, pp. 681–694, Dec. 2020.
9. L. Ouyang, “Training language models to follow instructions with human feedback,” in Proc. NIPS, 2022, pp. 27730–27744.
10. M. Chen, “Evaluating large language models trained on code,” 2021, arXiv:2107.03374.
11. M. Pluhacek, A. Kazikova, T. Kadavy, A. Viktorin, and R. Senkerik, “Leveraging large language models for the generation of novel metaheuristic optimization algorithms,” in Proc. Companion Conf. Genetic Evol. Comput., Jul. 2023, pp. 1812–1820.
12. P.-F. Guo, Y.-H. Chen, Y.-D. Tsai, and S.-D. Lin, “Towards optimizing with large language models,” 2023, arXiv:2310.05204.
13. P. A. Vikhar, “Evolutionary algorithms: A critical review and its future prospects,” in Proc. Int. Conf. Global Trends Signal Process., Inf. Comput. Commun. (ICGTSPICC), Dec. 2016, pp. 261–265.

14. P. E. Gill, W. Murray, and M. H. Wright, Practical Optimization. Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 2019.

15. R. Li, “StarCoder: May the source be with you!,” 2023, arXiv:2305.06161.

16. R. T. Lange, Y. Tian, and Y. Tang, “Large language models as evolution strategies,” 2024, arXiv:2402.18381.

17. S. Liu, C. Chen, X. Qu, K. Tang, and Y.-S. Ong, “Large language models as evolutionary optimizers,” 2023, arXiv:2310.19046.

18. S. S. Biswas, “Role of chat GPT in public health,” Ann. Biomed. Eng., vol. 51, no. 5, pp. 868–869, May 2023.

19. W. X. Zhao, “A survey of large language models,” 2023, arXiv:2303.18223v13.

20. Kumar, S. K. (2026). Explainable AI for SSD Failure Prediction: Using LIME and SHAP for Transparency. Journal of Engineering Research and Sciences, 5(4), 1–16. https://doi.org/10.55708/js0504001