INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 7,854 Volume 13, issue 09 (2026)

A Explainability-Driven Approach to SSD Failure Detection Using LIME and SHAP Analysis

Minh T. Nguyen
Department of Artificial Intelligence, Hanoi Institute of Technology, Hanoi, Vietnam

Abstract: The Solid-state drives (SSDs) have become critical components of modern computing infrastructures because of their high performance, low latency, and increasing deployment in data-intensive systems. However, SSD degradation and failure can develop through complex interactions among workload characteristics, operational conditions, device behavior, and accumulated wear, making reliable failure detection a challenging predictive task. Conventional machine-learning-based failure prediction can provide accurate classifications while remaining difficult for engineers and system administrators to interpret. This creates an important requirement for explainability, particularly when predictive decisions influence preventive maintenance, storage migration, or system availability. This paper proposes an explainability-driven framework for SSD failure detection that integrates predictive modeling with Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). The methodology treats explainability not as a post-processing visualization but as an analytical layer connecting SSD telemetry, predictive outputs, feature contributions, and operational decisions. The framework incorporates principles derived from evolutionary optimization, intelligent search, recommender systems, large language model research, and computational materials discovery to establish a broader theoretical foundation for data-driven optimization and interpretable decision support. Particular emphasis is placed on feature-level attribution, local versus global explanations, explanation consistency, failure-risk prioritization, and the trade-off between predictive complexity and interpretability. The resulting framework provides a systematic basis for identifying influential SSD health indicators, validating model behavior, and translating failure predictions into actionable maintenance decisions. The study further identifies limitations associated with explanation stability, correlated telemetry variables, model dependence, and the absence of a standardized explanatory ground truth. The proposed approach contributes an interpretable architecture for trustworthy SSD failure detection and establishes directions for future empirical validation.

Keywords: SSD failure detection; Explainable Artificial Intelligence; LIME; SHAP; Predictive Maintenance; Machine Learning; Storage Reliability; Feature Attribution; Failure Prediction.

INTRODUCTION

The SSDs have transformed storage architectures by providing substantially different performance characteristics from traditional magnetic storage. Their deployment in personal computing, enterprise infrastructure, cloud systems, and data-intensive applications makes storage reliability an important component of overall system dependability. As SSDs operate under heterogeneous workloads, their observable health characteristics may change over time. A failure-detection system must therefore distinguish normal operational variation from patterns associated with degradation or impending failure.

Machine learning provides a promising mechanism for addressing this problem because predictive models can

integrate multiple telemetry variables and identify nonlinear relationships that may be difficult to encode using fixed threshold rules. Nevertheless, high predictive capability does not automatically produce an operationally trustworthy system. A model may classify an SSD as likely to fail without explaining which characteristics influenced that prediction. For maintenance engineers, such a result is incomplete because the prediction itself does not indicate whether the decision is associated with accumulated wear, abnormal workload behavior, temperature-related variation, error activity, or another observable factor.

This interpretability problem motivates the integration of Explainable Artificial Intelligence (XAI) into SSD failure detection. The central objective is not merely to determine whether an SSD is likely to fail, but to provide a defensible explanation of why a particular prediction was generated. LIME and SHAP provide complementary approaches to this problem by estimating feature contributions around individual predictions and decomposing model outputs into feature-level contributions, respectively. The explainability perspective is particularly relevant to failure prediction because maintenance decisions frequently require evidence rather than a binary risk label (Kumar, 2026).

The broader methodological foundation for this framework can be connected to research on optimization and intelligent search. Evolutionary algorithms demonstrate how complex solution spaces can be explored through systematic evaluation and iterative improvement (Coello et al., 2007; Deb, 1999). Multiobjective optimization further establishes the importance of balancing competing objectives rather than optimizing a single measure in isolation (Branke et al., 2008). These principles are applicable to SSD failure detection because predictive accuracy, false-alarm reduction, computational efficiency, and interpretability may represent competing objectives.

Research on recommender systems similarly demonstrates the importance of transforming complex computational outputs into useful decisions. Recommender-system research has evolved from conventional recommendation mechanisms toward architectures incorporating large language models and more sophisticated computational reasoning (Bobadilla et al., 2013; Zhao, 2024). Although recommendation and SSD failure prediction are different tasks, both require models to convert heterogeneous information into decisions that users can act upon.

The objective of this study is therefore to formulate an explainability-driven architecture for SSD failure detection in which predictive modeling and explanation generation operate as interconnected components. The study focuses on the theoretical design of such a framework, its functional stages, interpretability mechanisms, expected findings, limitations, and practical implications.

LITERATURE REVIEW

2.1 Optimization as a Foundation for Data-Driven Failure Detection

Evolutionary computation provides a useful theoretical foundation for complex optimization problems. Coello et al. (2007) describe evolutionary algorithms as general mechanisms for exploring solution spaces, while Deb (1999) emphasizes their application to multi-criterion engineering optimization. Branke et al. (2008) further establish that multiobjective optimization is particularly valuable when multiple competing criteria must be considered simultaneously.

For SSD failure detection, this perspective is relevant because predictive-system development rarely involves only one objective. A model should ideally achieve high detection capability while minimizing false alarms, maintaining manageable computational cost, and producing explanations that are sufficiently stable for operational use. Treating model development as a multiobjective problem therefore provides a stronger

theoretical foundation than optimizing classification accuracy alone.

Gupta et al. (2019) demonstrate the relevance of high-dimensional search and inverse machine-learning perspectives to optimization. Their work is conceptually useful for SSD telemetry because storage devices can produce numerous potentially correlated measurements. A high-dimensional feature space may contain redundant or weakly informative variables, increasing the difficulty of constructing robust prediction models.

The principal implication is that SSD failure detection should incorporate feature selection and model optimization as explicit methodological stages. Explainability can subsequently be used to determine whether the selected features produce technically meaningful predictions.

2.2 Intelligent Computational Systems and Model Development

Recent literature demonstrates increasing interaction between machine learning, large language models, and optimization. Guo (2023) investigates the use of large language models for evolutionary prompt optimization, while Wu et al. (2024) examine evolutionary computation in the large-language-model era. Meyerson (2024) investigates language-model crossover and variation through few-shot prompting, and Xu (2022) considers genetic prompt search for efficient few-shot learning.

Nasir et al. (2024) extend this direction through LLMatic, which investigates neural architecture search using large language models and quality-diversity optimization. Bradley et al. (2024) similarly explore the OpenELM library in relation to language models and evolutionary algorithms. These studies collectively illustrate a broader research direction in which automated search can support the construction of complex computational models.

Although these works do not specifically address SSD reliability, they provide a methodological analogy: complex predictive systems can benefit from systematic search rather than arbitrary model configuration. In an SSD failure framework, similar principles could be used to optimize model hyperparameters, feature subsets, decision thresholds, or multiobjective criteria.

The work of Li et al. (2023) on large language models for supply-chain optimization provides another relevant conceptual analogy. Supply-chain optimization involves heterogeneous variables, constraints, uncertainty, and decisions with operational consequences. SSD reliability prediction similarly requires transforming heterogeneous telemetry into actionable decisions. The common methodological principle is the integration of computational prediction with operational decision support.

2.3 Data-Driven Scientific and Recommendation Applications

Oganov et al. (2019) demonstrate how computational prediction and structure discovery can contribute to materials discovery. Buehler (2024) further illustrates the application of language-based computational strategies to mechanics and materials modeling. Bran et al. (2023) demonstrate how computational models can be augmented with specialized chemistry tools. These studies collectively show that predictive systems become more valuable when their computational outputs are connected to domain-specific reasoning.

The relevance to SSD failure detection lies in the need to connect machine-learning predictions with physical or operational interpretation. An explanation that merely identifies a feature mathematically is less useful than one that allows an engineer to associate that feature with an interpretable aspect of device behavior.

Recommender-system literature offers another important perspective. Bobadilla et al. (2013) survey the

development of recommender systems, while Koren et al. (2021) examine collaborative filtering as a central recommendation paradigm. Zhao (2024) discusses recommender systems in the era of large language models. These studies emphasize the transformation of complex information into personalized or decision-oriented outputs.

The SSD problem can similarly be viewed as a decision-support problem: a model observes device-level information, calculates failure risk, identifies influential variables, and communicates the evidence needed for an operational response.

2.4 Research Gap

The supplied literature is predominantly concerned with optimization, evolutionary computation, language models, materials modeling, and recommendation. None of these references establishes a complete explainability-driven SSD failure detection framework. This reveals an important methodological gap.

Kumar (2026) directly establishes the relevance of LIME and SHAP to SSD failure prediction and transparency. However, a broader framework is required to connect predictive modeling, feature engineering, explanation generation, explanation validation, and maintenance decisions into one architecture. The present study addresses this gap by positioning explainability as an integral component of SSD failure detection rather than an optional visualization layer.

METHODOLOGY

3.1 Proposed Framework

The proposed methodology consists of six interconnected stages: SSD telemetry acquisition, data preprocessing, feature engineering, predictive modeling, explainability analysis, and decision-oriented interpretation.

The conceptual pipeline can be expressed as:

SSD Telemetry → Data Preprocessing → Feature Engineering → Failure Prediction Model → LIME/SHAP Explanation → Risk Interpretation and Maintenance Decision

The framework is designed around the principle that the final output should contain both a failure-risk estimate and evidence explaining that estimate.

3.2 SSD Telemetry Acquisition

The first stage involves collecting operational indicators capable of describing SSD health. Depending on the available monitoring environment, these may include error-related measurements, workload statistics, temperature indicators, accumulated operating information, wear-related characteristics, and performance-related variables.

The objective is not to assume that every telemetry variable is equally informative. Instead, the dataset should be treated as a multidimensional representation of SSD operational state. Gupta et al. (2019) provide a useful theoretical basis for recognizing the challenges associated with high-dimensional search spaces.

A hypothetical dataset could contain thousands of SSD observations, where each observation represents a device state at a particular monitoring interval. The target variable may be defined as a binary failure indicator

or as multiple risk categories such as normal, warning, and high-risk.

3.3 Data Preprocessing

Raw telemetry may contain missing values, inconsistent measurement scales, duplicate observations, and highly skewed distributions. Preprocessing should therefore include missing-value treatment, normalization where appropriate, outlier analysis, and temporal consistency checking.

Particular attention should be given to class imbalance. In practical failure prediction, failed devices are generally less frequent than healthy devices. A model trained without considering this imbalance may achieve apparently strong accuracy while performing poorly on the minority failure class.

Therefore, evaluation should prioritize metrics such as precision, recall, F1-score, sensitivity, specificity, and area-under-curve measures rather than accuracy alone.

3.4 Feature Engineering and Selection

Feature engineering transforms raw telemetry into predictive variables. Features may represent instantaneous device conditions, accumulated wear, temporal trends, rates of change, or interactions among operational variables.

Feature selection should be treated as both a predictive and interpretability problem. A model containing excessive redundant variables can produce explanations that are difficult to interpret. Conversely, removing informative variables can reduce predictive performance.

The optimization perspective of Coello et al. (2007), Deb (1999), and Branke et al. (2008) suggests that feature selection can be formulated as a multiobjective problem balancing predictive performance and model complexity.

For example, one candidate feature subset may provide superior recall but require 40 telemetry variables, whereas another may achieve slightly lower recall with only 12 variables. The second model may be operationally preferable if its explanations are substantially clearer and easier to validate.

3.5 Predictive Model Construction

The prediction stage may employ an interpretable baseline model alongside a nonlinear ensemble or other high-capacity machine-learning model. The baseline provides a reference against which the performance and complexity of advanced models can be evaluated.

The proposed framework does not prescribe one universal classifier because the appropriate model depends on dataset characteristics, sample size, feature distributions, temporal structure, and deployment constraints.

The primary output of the predictive model is a failure probability:

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

where $F=1F=1$ represents a failure condition and XX denotes the observed SSD feature vector.

A decision threshold can then convert the probability into an operational classification. However, the threshold should not be selected solely according to generic statistical performance. Its value should reflect the relative

costs of missed failures and false alarms.

3.6 LIME-Based Local Explanation

LIME is used to generate local explanations for individual predictions. The fundamental idea is to approximate a complex model around a particular observation using a simpler interpretable representation.

For SSD failure detection, this means that when a model predicts high failure probability for a particular device, LIME can identify the local variables that most strongly contributed to that prediction.

For example, a hypothetical explanation could indicate that elevated error activity and an abnormal performance trend increased the predicted failure probability, while stable operating characteristics reduced it. Such an explanation is more actionable than a risk score alone because it identifies the local evidence underlying the decision.

The value of this approach is its case-specific nature. A maintenance engineer can inspect the explanation for an individual SSD rather than relying exclusively on global feature rankings (Kumar, 2026).

However, local explanations can be sensitive to sampling and perturbation choices. Consequently, an explanation should not automatically be interpreted as a causal statement. It represents an approximation of model behavior around a specific observation.

3.7 SHAP-Based Feature Attribution

SHAP provides a complementary mechanism for estimating feature contributions based on Shapley-value principles. Rather than focusing exclusively on one local approximation, SHAP can be used to analyze individual predictions and aggregate feature contributions across a dataset.

For SSD failure detection, SHAP can answer two distinct questions. First, which variables contributed to a particular failure prediction? Second, which variables tend to be influential across the population of monitored SSDs?

This distinction is important because a feature can be globally important while contributing weakly to a specific prediction. Conversely, a variable with moderate global importance may become decisive for an individual device.

A hypothetical SHAP analysis might reveal that a particular wear-related indicator is generally influential across the population, while an error-related variable dominates the explanation for one specific SSD. Such differentiation enables maintenance decisions to remain device-specific while still benefiting from population-level knowledge.

3.8 LIME–SHAP Integration

The proposed methodology uses LIME and SHAP as complementary rather than competing explainability methods.

LIME emphasizes local surrogate interpretation, whereas SHAP provides an additive feature-attribution perspective that can support both local and aggregated analysis. Comparing their explanations can provide an additional consistency check.

If both methods identify similar influential variables for a high-risk SSD, confidence in the model explanation may increase. If they produce substantially different explanations, the disagreement should be investigated rather than hidden.

This principle is especially important because explainability itself can contain uncertainty. Kumar (2026) emphasizes the role of LIME and SHAP in improving transparency for SSD failure prediction; the present framework extends this idea by treating agreement and disagreement between explanation methods as an analytical signal.

3.9 Explanation Validation

Explanation validation is a critical methodological component. An explanation should be assessed for stability, consistency, domain plausibility, and usefulness.

Stability asks whether similar observations produce reasonably similar explanations. Consistency examines whether LIME and SHAP identify comparable influential variables. Domain plausibility considers whether identified variables are operationally meaningful. Usefulness evaluates whether the explanation can support a maintenance decision.

This stage prevents XAI from becoming merely a graphical presentation layer.

3.10 Decision-Oriented Risk Interpretation

The final stage translates prediction and explanation into operational categories. A possible framework is:

Low risk: no immediate intervention.

Moderate risk: increased monitoring and diagnostic review.

High risk: preventive migration, replacement planning, or accelerated inspection.

The exact thresholds should be calibrated using operational costs and historical failure behavior. Multiobjective optimization provides a useful conceptual foundation because risk thresholds must balance competing consequences rather than maximize a single statistical metric (Branke et al., 2008).

RESULTS

The analytical findings of the proposed framework indicate that explainability can fundamentally alter the interpretation of SSD failure detection. A conventional predictive model produces a classification or probability, whereas the proposed architecture additionally provides feature-level evidence supporting the prediction. This transforms failure detection from a purely predictive task into an interpretable decision-support process.

The first major finding is that local explanation is particularly valuable for individual SSD investigation. Two devices may receive similar failure probabilities but for substantially different reasons. One may be influenced primarily by wear-related variables, while another may receive its risk classification because of abnormal error behavior. LIME provides a mechanism for exposing these local differences. This is operationally significant because maintenance actions should respond to the evidence associated with the individual device rather than merely its numerical risk category.

The second finding concerns the value of global feature attribution. SHAP can aggregate feature contributions across multiple observations, enabling analysts to identify variables that repeatedly influence failure predictions. This can support telemetry prioritization, monitoring-system design, and feature reduction. Features that consistently demonstrate weak contribution may be candidates for removal, although such decisions should be validated carefully because low model contribution does not necessarily imply physical irrelevance.

The third finding is that explanation agreement can serve as an additional reliability signal. When LIME and SHAP identify comparable influential variables for the same prediction, the resulting interpretation is stronger than an explanation obtained through a single method. Conversely, disagreement can reveal observations near complex decision boundaries, correlated variables, or model behaviors requiring further examination. Thus, disagreement between explanation methods should be interpreted diagnostically rather than considered a methodological failure.

The fourth finding is that predictive performance and explainability should be treated as competing but jointly optimized objectives. A highly complex model may improve predictive capability while increasing interpretive difficulty. Evolutionary and multiobjective optimization research provides a theoretical basis for addressing such trade-offs (Coello et al., 2007; Deb, 1999). The optimal SSD detection architecture should therefore seek a balance between failure-detection sensitivity, false-alarm control, computational efficiency, and explanation quality.

Finally, the framework indicates that explainability does not eliminate uncertainty. LIME and SHAP describe model behavior rather than proving physical causation. Consequently, an explanation that identifies a telemetry variable as influential should not automatically be interpreted as evidence that the variable physically caused the SSD failure. The distinction between statistical attribution and causal diagnosis is essential for responsible deployment.

Because the supplied references do not contain a common SSD benchmark dataset or experimental measurements, these findings are methodological rather than claims of measured predictive performance.

DISCUSSION

The proposed explainability-driven architecture contributes to SSD failure detection by integrating prediction and interpretation within a unified analytical process. Its central theoretical implication is that predictive reliability should not be evaluated exclusively through classification metrics. In safety- and availability-sensitive environments, the ability to explain why a model produces a high-risk decision can be equally important.

The framework also demonstrates a useful connection between explainable prediction and optimization. Evolutionary computation literature emphasizes systematic exploration of complex search spaces (Coello et al., 2007; Deb, 1999), while multiobjective optimization demonstrates how conflicting objectives can be balanced (Branke et al., 2008). SSD failure detection naturally exhibits such conflicts. Maximizing sensitivity may increase false alarms; reducing the number of monitored variables may simplify explanations but remove useful information; and selecting a highly expressive model may improve nonlinear pattern recognition while reducing transparency.

The proposed LIME–SHAP combination addresses another important trade-off. LIME is useful for understanding an individual decision, while SHAP can provide a broader attribution perspective. Their combined use can therefore support both maintenance-level and system-level analysis. However, neither

method should be treated as inherently authoritative. Explanations are generated from the predictive model and its learned relationships. If the underlying model is biased, poorly calibrated, or trained on unrepresentative data, an apparently clear explanation may still support an incorrect decision.

Correlated telemetry variables create an additional limitation. When several SSD indicators carry overlapping information, attribution methods may distribute importance among them in ways that are mathematically defensible but operationally ambiguous. An engineer could therefore observe several highly ranked features without knowing which underlying physical mechanism is most important. Explanation analysis should consequently be supplemented by domain validation.

The literature on recommender systems also provides an important conceptual lesson. Recommender systems are valuable not simply because they calculate scores but because those scores are transformed into useful decisions (Bobadilla et al., 2013; Koren et al., 2021). Similarly, SSD failure prediction should be evaluated according to whether its outputs support better maintenance decisions. The transition from prediction to action is therefore a central component of system effectiveness.

Research on large language models and optimization further indicates that modern computational systems increasingly integrate heterogeneous reasoning and search mechanisms (Guo, 2023; Wu et al., 2024). Nevertheless, introducing more sophisticated computational components should not automatically be interpreted as improving reliability. For SSD monitoring, model complexity should be justified by measurable improvements in detection capability, robustness, or decision quality.

A further practical implication concerns trust. Explainability can help engineers identify whether a model is relying on plausible telemetry or on potentially spurious correlations. If a failure prediction is repeatedly driven by an unexpected variable, analysts can investigate the data pipeline or model before deployment. In this sense, explainability can support model auditing in addition to end-user interpretation (Kumar, 2026).

Several limitations remain. The proposed framework is conceptual and requires empirical validation on longitudinal SSD datasets. Explanation stability should be quantitatively evaluated across repeated observations and model configurations. The relationship between explanation quality and actual maintenance outcomes also requires investigation. Finally, future work should investigate temporal explainability, where changes in feature contributions over time may provide additional information about degradation trajectories.

CONCLUSION

This paper proposed an explainability-driven framework for SSD failure detection based on machine-learning prediction combined with LIME and SHAP analysis. The framework addresses a central limitation of conventional predictive maintenance systems: a failure probability alone does not provide sufficient evidence for an operational decision.

The proposed architecture integrates telemetry acquisition, preprocessing, feature engineering, predictive modeling, local and global explanation, explanation validation, and risk-oriented decision support. LIME provides case-specific explanations, while SHAP supports feature-attribution analysis across individual observations and larger populations. Their combined use provides a mechanism for investigating both why an individual SSD is classified as high risk and which characteristics are generally influential across the monitored population.

The theoretical contribution is the integration of explainability with optimization-oriented thinking. Predictive accuracy, interpretability, computational efficiency, and false-alarm control should be treated as jointly

relevant objectives rather than isolated performance criteria. This perspective is consistent with the broader principles established in evolutionary and multiobjective optimization research (Coello et al., 2007; Branke et al., 2008).

The study also emphasizes that XAI explanations represent model behavior rather than guaranteed causal mechanisms. Therefore, explanation outputs require domain validation and should be interpreted as evidence supporting a prediction, not as definitive physical diagnoses. This distinction is particularly important in SSD reliability applications where inappropriate maintenance decisions may impose operational costs.

Future empirical research should validate the proposed architecture using longitudinal SSD telemetry, compare multiple predictive algorithms, quantify LIME and SHAP stability, investigate temporal changes in feature importance, and evaluate whether explainable predictions improve maintenance decisions. Such work could establish a stronger empirical foundation for trustworthy, transparent, and operationally useful SSD failure detection systems.

REFERENCES

1. J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowl. Based Syst., vol. 46, pp. 109–132, Jul. 2013.

2. H. Bradley, H. Fan, T. Galanos, R. Zhou, D. Scott, and J. Lehman, "The OpenELM library: Leveraging progress in language models for novel evolutionary algorithms," in Genetic Programming Theory and Practice XX. Singapore : Springer, 2024, pp. 177–201.

3. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White, and P. Schwaller, "ChemCrow: Augmenting large-language models with chemistry tools," 2023, arXiv:2304.05376.

4. J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Multiobjective Optimization: Interactive and Evolutionary Approaches. Berlin, Germany : Springer, 2008.

5. M. J. Buehler, "MechGPT, a language-based strategy for mechanics and materials modeling that connects knowledge across scales, disciplines, and modalities," Appl. Mech. Rev., vol. 76, no. 2, 2024, Art. no. 21001.

6. "Chatgpt 3.5. " OpenAI. Accessed: Nov. 12, 2024. [Online]. Available: https://platform.openai.com/docs/api-reference

7. C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems. New York, NY, USA : Springer, 2007.

8. K. Deb, "Evolutionary algorithms for multi-criterion optimization in engineering design," Evol. Algorithms Eng. Comput. Sci., vol. 2, pp. 135–161, May 1999.

9. A. Gupta, Y.-S. Ong, M. Shakeri, X. Chi, and A. Z. NengSheng, "The blessing of dimensionality in many-objective search: An inverse machine learning insight," in Proc. IEEE Int. Conf. Big Data (Big Data), 2019, pp. 3896–3902.

10. Q. Guo, "Connecting large language models with evolutionary algorithms yields powerful prompt optimizers," 2023, arXiv:2309.08532.

11. Y. Koren, S. Rendle, and R. Bell, "Advances in collaborative filtering," in Recommender Systems Handbook. Boston, MA, USA : Springer, 2021, pp. 91–142.

12. B. Li, K. Mellou, B. Zhang, J. Pathuri, and I. Menache, "Large language models for supply chain optimization," 2023, arXiv:2307.03875.

13. E. Meyerson, "Language model crossover: Variation through few-shot prompting," ACM Trans. Evol. Learn. Optim., vol. 4, pp. 1–40, Nov. 2024, [Online]. Available: https://doi.org/10.1145/3694791

14. M. U. Nasir, S. Earle, J. Togelius, S. James, and C. Cleghorn, "LLMatic: neural architecture search via large language models and quality diversity optimization," in Proc. Genet. Evol. Comput. Conf., 2024, pp. 1110–1118.

15. A. R. Oganov, C. J. Pickard, Q. Zhu, and R. J. Needs, "Structure prediction drives materials discovery," Nat. Rev. Mater., vol. 4, no. 5, pp. 331–348, 2019.

16. X. Wu, S.-h. Wu, J. Wu, L. Feng, and K. C. Tan, "Evolutionary computation in the era of large language model: Survey and roadmap," 2024, arXiv:2401.10034.

17. H. Xu, "GPS: Genetic prompt search for efficient few-shot learning," 2022, arXiv:2210.17041.

18. W. X. Zhao, "A survey of large language models," 2025, arXiv:2303.18223.

19. Z. Zhao, "Recommender systems in the era of large language models (LLMs)," IEEE Trans. Knowl. Data Eng., vol. 36, no. 11, pp. 6889–6907, Nov. 2024.

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