Explainability-Centered Machine Learning for Reliable Solid-State Drive Failure Prediction

Siti Nur Aisyah
Department of Computer Science, Bandung Institute of Computing, Bandung, Indonesia
https://doi.org/10.37547/ibast/Volume06Issue09-01

Abstract.
Solid-state drives (SSDs) are increasingly deployed in computing infrastructures where storage reliability directly influences service continuity, data availability, and maintenance cost. Predicting SSD failure before catastrophic degradation therefore requires machine-learning models capable of identifying complex patterns in operational data while providing explanations that can be trusted by engineers and system administrators. This research develops an explainability-centered conceptual framework for reliable SSD failure prediction by integrating predictive machine learning with local and global interpretability mechanisms, particularly Local Interpretable Model-Agnostic Explanations (LIME) and Shapley-based explanations (SHAP). The methodological foundation is informed by the supplied literature on optimization, intelligent scheduling, knowledge-driven algorithms, reinforcement learning, and simulation–optimization. These studies demonstrate the importance of combining data-driven learning with domain knowledge, optimization objectives, dynamic decision mechanisms, and interpretable operational factors. The proposed framework organizes SSD telemetry into health indicators, constructs a failure-risk model, generates global and instance-level explanations, and incorporates explanation consistency into reliability assessment. Rather than reporting fabricated experimental measurements, the study presents analytical findings concerning the expected behavior, advantages, limitations, and deployment implications of an explainability-centered architecture. The framework emphasizes that high predictive accuracy alone is insufficient for safety-critical storage management; a reliable prediction should also identify the operational evidence responsible for its decision. The resulting architecture provides a structured basis for transparent preventive maintenance, risk prioritization, and human-centered SSD reliability management.

Keywords: Solid-State Drive, SSD Failure Prediction, Explainable Artificial Intelligence, LIME, SHAP, Machine Learning, Predictive Maintenance, Model Interpretability, Storage Reliability, Failure Risk

INTRODUCTION

The Solid-state drives have become fundamental components of contemporary computing systems because they provide high-speed data access, low latency, and efficient storage operations. Their increasing deployment in enterprise, cloud, embedded, and high-performance environments makes storage reliability a critical engineering concern. An SSD

failure can produce service interruption, data-access degradation, maintenance overhead, and potentially significant operational consequences. Consequently, early identification of abnormal drive behavior is an important predictive-maintenance problem.

Traditional reliability management often depends on threshold-based monitoring and manually defined indicators. Such mechanisms are useful when degradation follows relatively simple and observable patterns, but modern storage devices generate multidimensional operational information whose interactions may not be adequately represented through individual thresholds. A machine-learning model can potentially discover nonlinear relationships among health indicators and distinguish normal operating states from emerging failure conditions. However, the use of a highly complex predictive model introduces a second problem: why does the model believe that a particular SSD is approaching failure?

This question is central to reliable deployment. An unexplained prediction may be difficult for an engineer to validate, especially when maintenance actions involve cost, downtime, or replacement of apparently functional hardware. Explainable artificial intelligence (XAI) addresses this limitation by transforming model predictions into human-interpretable evidence. Kumar (2026) specifically positions LIME and SHAP as mechanisms for improving transparency in SSD failure prediction, providing an important direct foundation for the present explainability-centered perspective.

The broader methodological literature supplied for this study further demonstrates the value of combining intelligent models with structured domain information. Knowledge-driven optimization has been used to improve complex scheduling decisions (Chen et al., 2024; Tian et al., 2024), while reinforcement-learning approaches have been employed for dynamic scheduling environments (He & Li, 2024; Tang et al., 2024). Simulation–optimization has similarly been used to connect computational models with operational decision-making (Wang et al., 2025). Although these studies address manufacturing and scheduling rather than storage failure, they establish a relevant theoretical principle: intelligent prediction or decision systems become more useful when their outputs can be related to meaningful operational variables and decision objectives.

The objective of this research is therefore to formulate an explainability-centered machine-learning architecture for reliable SSD failure prediction. The framework focuses on four objectives: developing a structured SSD health representation, producing failure-risk predictions, generating interpretable explanations using complementary explanation techniques, and incorporating explanation reliability into the overall assessment of the prediction system. The study is conceptual and analytical rather than an empirical benchmark; consequently, it does not fabricate accuracy, F1-score, or failure-prediction results that cannot be established from the supplied evidence.

LITERATURE REVIEW

2.1 Intelligent Decision-Making and Complex Operational Systems

The supplied literature demonstrates that complex engineering systems frequently involve interacting variables, competing objectives, and operational constraints. Garey, Johnson, and Sethi (1976) established the computational complexity associated with flowshop and jobshop scheduling, illustrating why complex operational problems cannot always be addressed efficiently through simple deterministic procedures. In later work, Defersha and Movahed (2018) combined linear programming assistance with genetic algorithms for flexible job-shop scheduling with lot streaming, demonstrating the value of combining complementary computational mechanisms.

Bożek and Werner (2018) investigated flexible job-shop scheduling with lot streaming and sublot-size optimization, while Li (2020) addressed multi-objective optimization in lot-streaming hybrid flowshops with variable sublots. These studies collectively indicate that engineering decision systems often require models capable of handling nonlinear relationships, multiple variables, and competing objectives. Such characteristics are also relevant to SSD failure prediction, where a failure state may emerge from the interaction of several health indicators rather than from a single variable.

2.2 Knowledge-Driven and Learning-Based Optimization

Knowledge integration represents another important theme in the supplied literature. Chen et al. (2024) proposed a knowledge-driven many-objective algorithm for energy-efficient distributed heterogeneous hybrid flowshop scheduling. Similarly, Tian et al. (2024) examined knowledge-based lot-splitting optimization under energy-consumption considerations. These studies imply that purely computational optimization can benefit from incorporating domain knowledge into model construction and decision processes.

For SSD prediction, this principle suggests that machine-learning models should not be treated as isolated statistical mechanisms. SSD-specific knowledge can inform feature interpretation, failure-state definitions, and the plausibility of explanations. An explanation indicating that a particular operational variable strongly increases predicted risk should be assessed against known storage behavior rather than accepted solely because the model generated it.

2.3 Reinforcement Learning and Adaptive Decision Systems

Dynamic learning is another recurring theme. He and Li (2024) used deep reinforcement learning based on graph neural networks for flexible job-shop scheduling with lot streaming. Tang et al. (2024) similarly investigated deep reinforcement learning for integrated multi-objective lot-streaming hybrid flowshop scheduling with automated guided vehicles. Lu et al. (2024) combined meta-heuristics and Q-learning to address scheduling decisions.

These approaches are important conceptually because they demonstrate how learning systems can adapt to complex operational environments. For SSD reliability, an analogous perspective is useful: storage health is not necessarily static, and predictive models may need to account for temporal changes in workload, wear, error behavior, and other telemetry characteristics. Nevertheless, adaptive prediction also increases the importance of interpretability because changing models can produce changing explanations.

2.4 Simulation, Optimization, and Operational Integration

Wang et al. (2025) proposed an integrated simulation–optimization approach for flexible assembly job-shop scheduling with finite transportation resources. This work reinforces the importance of connecting computational predictions with operational decisions rather than evaluating algorithms solely as abstract mathematical systems.

The same principle applies to SSD failure prediction. A failure probability has practical value only when it supports an action such as additional monitoring, workload migration, backup prioritization, scheduled replacement, or maintenance intervention. Consequently, the proposed framework treats explanation as a bridge between prediction and operational response.

2.5 Research Gap

The major gap in the supplied literature is domain-specific. Most references focus on scheduling, production optimization, energy efficiency, and intelligent manufacturing. They provide methodological foundations for learning, optimization, knowledge integration, and operational decision-making, but they do not directly establish an empirical SSD-failure benchmark. The directly relevant work by Kumar (2026) addresses explainable SSD failure prediction using LIME and SHAP and therefore provides the closest conceptual basis for the proposed framework.

The research gap is consequently not simply the absence of another predictive model. It is the need for a reliability-oriented architecture in which prediction and explanation are treated as jointly important outputs. The present research extends this perspective by positioning explainability as a component of reliability assessment rather than as a visualization added after model development.

METHODOLOGY

3.1 Explainability-Centered Architecture

The proposed architecture consists of five interconnected layers: telemetry acquisition, feature engineering, failure-risk prediction, explanation generation, and reliability-oriented decision support.

At the first layer, operational SSD telemetry is collected over time. Potential variables may include device age, utilization, temperature, error-related indicators, write activity, read activity, wear-related attributes, and other health-monitoring parameters available from the storage environment. The methodology does not assume that every SSD exposes identical telemetry. Instead, the architecture is designed around a flexible feature representation so that different storage platforms can be accommodated.

The second layer transforms raw telemetry into machine-learning features. This step is theoretically important because raw measurements may contain noise, different scales, missing observations, or strongly correlated indicators. Temporal aggregation can be used to represent

recent behavior, while trend-based variables can capture whether an indicator is stable, improving, or deteriorating.

The third layer generates a failure-risk estimate. Let the feature vector for drive ii at time tt be represented as:

Xi,t=[x1,x2,…,xn].X_{i,t}=[x_1,x_2,\ldots,x_n].

A predictive model f(·)f(\cdot) produces a failure-risk score:

p^i,t=f(Xi,t),\hat{p}_{i,t}=f(X_{i,t}),

where p^i,t\hat{p}_{i,t} represents the estimated probability or risk score associated with failure within a predefined prediction horizon.

The fourth layer produces explanations. LIME can construct a locally interpretable approximation around an individual prediction, allowing engineers to understand which features contributed most strongly to a particular decision. SHAP provides a theoretically grounded feature-attribution mechanism that can be used both locally and across a collection of predictions. Kumar (2026) identifies the complementary transparency value of LIME and SHAP for SSD failure prediction.

The fifth layer converts predictive and explanatory information into an operational reliability assessment. Instead of presenting only “high risk” or “low risk,” the system can report a risk category together with the dominant contributing factors. For example, a drive may be classified as high risk because a combination of increasing error indicators, elevated operating temperature, and abnormal workload behavior contributes strongly to the prediction. The explanation allows an engineer to investigate whether the decision is operationally plausible.

3.2 Feature Engineering and Temporal Representation

Failure prediction differs from ordinary classification because degradation is inherently temporal. A single telemetry measurement may not adequately represent device health. A temperature value that is acceptable in isolation may become more informative if it has increased consistently over several observation windows.

Accordingly, the proposed feature-generation process includes three conceptual categories: instantaneous indicators, historical statistics, and temporal trends. Instantaneous indicators describe the current state. Historical statistics characterize typical behavior. Trend variables represent the direction and rate of change.

A simplified temporal feature can be represented as:

Δxj(t)=xj(t)−xj(t−k),\Delta x_j(t)=x_j(t)-x_j(t-k),

where kk represents a previous observation point. More sophisticated features may use rolling means, variances, slopes, or deviations from device-specific baselines.

This device-specific representation is important because SSDs can differ in operating environments and workloads. A universal threshold may therefore produce false alarms when applied to heterogeneous devices. The framework instead favors contextual risk estimation.

3.3 Predictive Model Construction

The predictive model should be selected according to the structure of the available telemetry and the required balance between performance and interpretability. Tree-based ensemble models, linear models, or neural architectures can all theoretically be integrated into the proposed architecture. The important requirement is that the selected model can be connected to an explanation layer.

Training should use temporally appropriate validation to avoid information leakage. Randomly distributing observations from the same device across training and testing sets may produce overly optimistic estimates because temporally adjacent observations can be highly correlated. A more reliable evaluation design separates historical training information from later observations.

Class imbalance is another important issue. Actual failure events are generally expected to be less frequent than normal operating states. A model that predicts “healthy” for almost every observation could achieve deceptively high accuracy while failing at the central task. Therefore, evaluation should emphasize failure-sensitive measures such as recall, precision, F1-score, and area under an appropriate discrimination curve, while also considering false-alarm cost.

3.4 LIME-Based Local Explanation

LIME explains an individual prediction by approximating the complex model locally with a simpler interpretable model. For an SSD predicted as high risk, LIME can identify the local feature conditions associated with that classification.

Its primary advantage is operational specificity. An engineer examining a particular device does not necessarily need to know the global behavior of the entire predictive model. Instead, the engineer needs to know why this drive, at this point in time, was classified as high risk.

However, local approximation introduces sensitivity to the neighborhood used for explanation. Consequently, LIME explanations should not automatically be interpreted as immutable causal relationships. They represent local model behavior, not proof that a particular physical mechanism caused failure.

3.5 SHAP-Based Global and Local Explanation

SHAP assigns contribution values to input features relative to a model prediction. The resulting representation can be used locally to explain individual predictions and globally to identify variables that consistently influence model behavior.

This dual role makes SHAP particularly appropriate for reliability analysis. If a specific variable repeatedly dominates failure predictions across many devices, engineers can investigate its

operational relevance. Conversely, if explanations change dramatically between similar observations, this may indicate model instability or feature interaction.

The framework therefore proposes two levels of SHAP analysis. The first is instance-level attribution for maintenance decisions. The second is population-level analysis for model auditing. Kumar (2026) provides the direct SSD-focused foundation for using LIME and SHAP as complementary interpretability mechanisms.

3.6 Explanation Reliability

A central methodological contribution is the distinction between prediction confidence and explanation reliability. A model may produce a confident prediction while its explanation is unstable. Therefore, reliability assessment should examine whether similar observations produce reasonably consistent explanatory patterns.

An explanation-stability indicator can conceptually be represented as:

Es=1−D(Et,Et+δ),E_s=1-D(E_t,E_{t+\delta}),

where EtE_t and Et+δE_{t+\delta} represent explanation vectors at nearby time points and DD is a normalized distance function. Higher EsE_s indicates greater similarity between explanations.

This measure should not be interpreted as a universal statistical standard; rather, it is a conceptual mechanism for integrating interpretability into system evaluation. The objective is to identify situations where a prediction is strong but its rationale is unstable.

3.7 Decision-Support Mechanism

The final stage maps risk and explanation into maintenance priorities. A hypothetical operational policy could classify drives into monitoring, investigation, and intervention categories. A medium-risk drive with stable explanations might remain under enhanced monitoring, while a high-risk drive with consistent evidence from multiple health indicators could receive higher replacement priority.

This decision-oriented approach follows the broader methodological lesson from optimization and scheduling studies: intelligent computation is most valuable when its output can be translated into a meaningful operational decision (Defersha & Movahed, 2018; Wang et al., 2025).

RESULTS

The analytical findings indicate that an explainability-centered architecture provides a more complete reliability perspective than a prediction-only design. The first finding is that SSD failure prediction should be treated as a multidimensional decision problem. The methodological evidence from complex scheduling research demonstrates that engineering outcomes can depend on interactions among numerous variables rather than a single

deterministic factor (Garey et al., 1976; Li, 2020). Transferred to SSD reliability, this supports the use of multivariate machine-learning representations rather than isolated threshold rules.

The second finding concerns the complementary roles of local and global explanation. LIME is particularly suitable for answering the operational question of why an individual SSD received a particular prediction, whereas SHAP can support both individual attribution and broader model auditing. This complementary structure is consistent with the explainability-oriented SSD perspective presented by Kumar (2026). The resulting architecture therefore avoids treating interpretability as a single visualization technique.

The third finding is that domain knowledge should be integrated into explanation assessment. Knowledge-driven optimization research shows that domain information can improve the meaningfulness of computational decisions (Chen et al., 2024; Tian et al., 2024). In the SSD context, a model explanation should consequently be examined for engineering plausibility. An unexpected feature dominating predictions may indicate a previously unknown risk relationship, but it may also indicate data leakage, measurement artifacts, or model bias.

The fourth finding is that temporal consistency is essential. A failure-risk model should respond to meaningful degradation patterns rather than random fluctuations. This suggests that explanations should be monitored over time. A drive whose risk increases while the same operational indicators repeatedly contribute to the prediction represents a stronger maintenance signal than a drive whose risk changes without stable explanatory evidence.

The fifth finding is that prediction performance and interpretability represent complementary evaluation dimensions. A highly accurate black-box model may still be operationally difficult to validate. Conversely, an easily interpretable model with inadequate failure sensitivity may not provide sufficient protection. The appropriate objective is therefore a balanced architecture that simultaneously considers discrimination, calibration, explanation quality, stability, and operational usefulness.

Finally, the literature synthesis reveals a clear research opportunity. The supplied optimization studies provide strong methodological analogies concerning adaptive learning, domain knowledge, multi-objective decision-making, and operational integration, but direct empirical evidence for explainable SSD failure prediction remains comparatively limited within the provided reference set. The proposed framework addresses this gap conceptually without claiming experimental improvements unsupported by the available evidence.

DISCUSSION

The central implication of the proposed framework is that explainability should be incorporated into SSD reliability engineering from the beginning of model development rather than added after predictive modeling is completed. This changes the objective from simply estimating whether a drive will fail to constructing a prediction that can be inspected, challenged, and acted upon.

The distinction is theoretically important. Machine-learning systems identify statistical associations, whereas engineers are generally interested in operational mechanisms. An explanation does not automatically establish causality. If SHAP or LIME assigns a high contribution to a particular telemetry feature, the result means that the feature influenced the model's decision under the relevant explanation mechanism; it does not prove that the feature physically caused the SSD failure. This distinction is essential for preventing inappropriate maintenance decisions.

The proposed dual-explanation approach also creates an important trade-off. LIME provides highly localized explanations but can be sensitive to how the local neighborhood is constructed. SHAP offers a more systematic attribution framework but may become computationally demanding for complex models or large datasets. The two methods should therefore be treated as complementary rather than interchangeable. Agreement between them can provide stronger interpretive evidence, while disagreement should trigger model auditing rather than arbitrary selection of one explanation.

The broader literature supports this integrated perspective. Knowledge-driven optimization demonstrates the value of embedding domain information into computational decision processes (Chen et al., 2024; Tian et al., 2024). Reinforcement-learning research demonstrates the ability of learning systems to operate in complex and changing environments (He & Li, 2024; Tang et al., 2024). Simulation–optimization demonstrates how computational outputs can be connected to operational constraints (Wang et al., 2025). Together, these findings suggest that SSD failure prediction should ultimately evolve from a static classifier into a decision-support system.

Another implication concerns false alarms. Preventive maintenance systems can become ineffective when alerts are too frequent. A model that identifies nearly every unusual telemetry event as impending failure may increase maintenance cost and reduce user confidence. Explainability can help distinguish meaningful alerts from suspicious predictions because engineers can inspect the evidence supporting each decision. Nevertheless, explanations do not eliminate false positives and should not be used as a substitute for statistical validation.

Data quality remains a major limitation. SSD telemetry may contain missing observations, firmware-dependent measurements, heterogeneous device populations, and workload-specific patterns. A model trained on one hardware family may not generalize to another. Explanation methods can expose these limitations by revealing unexpected dependence on device-specific or environment-specific variables, but they cannot solve distribution shift automatically.

Another limitation is the absence of an empirical SSD dataset in the supplied evidence. Accordingly, the present findings are methodological rather than quantitative. Future experimental work should evaluate the framework on longitudinal SSD datasets containing both healthy and failed devices. Such evaluation should compare predictive accuracy, failure recall, false-alarm rates, calibration, explanation stability, and cross-device generalization. The direct explainability perspective of Kumar (2026) provides an important starting point for such validation.

Overall, the proposed architecture supports a shift from accurate prediction alone toward reliable, inspectable prediction. This is particularly relevant when model outputs trigger hardware replacement, workload migration, or other consequential actions. The practical value of explainability therefore lies not merely in making artificial intelligence understandable, but in making its operational decisions auditable.

CONCLUSION

This research presented an explainability-centered machine-learning architecture for reliable SSD failure prediction. The framework integrates telemetry processing, temporal feature engineering, failure-risk prediction, LIME-based local interpretation, SHAP-based global and local attribution, explanation-stability assessment, and operational decision support.

The literature synthesis demonstrates that complex engineering problems benefit from knowledge integration, adaptive learning, optimization, and operationally meaningful decision mechanisms. These principles are evident throughout the supplied scheduling and intelligent-optimization literature (Defersha & Movahed, 2018; Chen et al., 2024; Lu et al., 2024; Wang et al., 2025). Their methodological relevance provides a foundation for designing storage-reliability systems in which machine learning is connected to practical engineering decisions.

The principal contribution is the positioning of explainability as a reliability dimension rather than merely a presentation layer. A prediction becomes more operationally useful when engineers can identify the factors contributing to the predicted risk, compare explanations across time, and assess whether those explanations are plausible. LIME and SHAP provide complementary mechanisms for achieving this objective, consistent with the SSD-focused explainability perspective of Kumar (2026).

The study deliberately avoids unsupported empirical claims because the provided references do not contain a common SSD failure dataset or experimental benchmark. Future research should therefore implement the proposed architecture using longitudinal SSD telemetry and evaluate predictive discrimination, calibration, explanation consistency, cross-device robustness, and maintenance effectiveness. Particular attention should be given to temporal validation and distribution shift because these factors can substantially influence real-world reliability.

Ultimately, explainable SSD failure prediction offers a pathway toward storage-management systems that are not only predictive but also transparent, auditable, and actionable. Such systems can support earlier intervention while preserving the human oversight required for reliable infrastructure management.

REFERENCES

1. (Airbus, Blagnac, France). 2024 Global Market Forecast. 2024, Accessed: Mar. 4, 2025. [Online]. Available: https://www.airbus.com/en/products-services/commercial-aircraft/global-market-forecast

2. B. J. V. Da Silva, R. Morabito, D. S. Yamashita, and H. H. Yanasse, "Production scheduling of assembly fixtures in the aeronautical industry," Comput. Ind. Eng., vol. 67, pp. 195–203, Jan. 2014.

3. S. Chen, X. Wang, Y. Wang, and X. Gu, "A knowledge-driven many-objective algorithm for energy-efficient distributed heterogeneous hybrid flowshop scheduling with lot-streaming," Swarm Evol. Comput., vol. 91, Dec. 2024, Art. no. 101771.

4. F. M. Defersha and S. B. Movahed, "Linear programming assisted (not embedded) genetic algorithm for flexible jobshop scheduling with lot streaming," Comput. Ind. Eng., vol. 117, pp. 319–335, Mar. 2018.

5. M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," Math. Oper. Res., vol. 1, no. 2, pp. 117–129, May 1976.

6. J. He and J. Li, "Deep reinforcement learning based on graph neural network for flexible job shop scheduling problem with lot streaming," in Proc. 20th Int. Conf. Adv. Intell. Comput. Technol. Appl., 2024, pp. 85–95.

7. J.-q. Li, "Efficient multi-objective algorithm for the lot-streaming hybrid flowshop with variable sub-lots," Swarm Evol. Comput., vol. 52, Feb. 2020, Art. no. 100600.

8. L. Li, "Research on discrete intelligent workshop lot-streaming scheduling with variable sublots under engineer to order," Comput. Ind. Eng., vol. 165, Mar. 2022, Art. no. 107928.

9. R. Li, L. Wang, W. Gong, and F. Ming, "An evolutionary multitasking memetic algorithm for multi-objective distributed heterogeneous welding flow shop scheduling," IEEE Trans. Evol. Comput., early access, Apr. 25, 2024, doi: 10.1109/TEVC.2024.3393620.

10. B. Lu, K. Gao, Y. Ren, D. Li, and A. Slowik, "Combining meta-heuristics and Q-learning for scheduling lot-streaming hybrid flow shops with consistent sublots," Swarm Evol. Comput., vol. 91, Dec. 2024, Art. no. 101731.

11. A. Bożek and F. Werner, "Flexible job shop scheduling with lot streaming and sublot size optimisation," Int. J. Prod. Res., vol. 56, no. 19, pp. 6391–6411, Oct. 2018.

12. W. Shao, Z. Shao, and D. Pi, "Modelling and optimization of distributed heterogeneous hybrid flow shop lot-streaming scheduling problem," Expert Syst. Appl., vol. 214, Mar. 2023, Art. no. 119151.

13. H. Tang, J. Huang, C. Ren, Y. Shao, and J. Lu, "Integrated scheduling of multi-objective lot-streaming hybrid flowshop with AGV based on deep reinforcement learning," Int. J. Prod. Res., vol. 63, no. 4, pp. 1275–1303, 2024.

14. G. Tian, W. Wang, H. Zhang, X. Zhou, C. Zhang, and Z. Li, "Multi-objective optimization of energy-efficient remanufacturing system scheduling problem with lot-streaming production mode," Expert Syst. Appl., vol. 237, Mar. 2024, Art. no. 121309.

15. Z. Tian, X. Jiang, G. Tian, Z. Li, and W. Liu, "Knowledge-based lot-splitting optimization method for flexible job shops considering energy consumption," IEEE Trans. Autom. Sci. Eng., vol. 21, no. 3, pp. 4864–4875, Jul. 2024.

16. H. Wang, T. Peng, X. Li, J. He, W. Liu, and R. Tang, "An integrated simulation–optimization method for flexible assembly job shop scheduling with lot streaming and finite transport resources," Comput. Ind. Eng., vol. 200, Feb. 2025, Art. no. 110790.

17. Y. Zhu, Q. Tang, L. Zhang, M. He, and J. Kapenda, "Improved multi-objective artificial bee colony algorithm for parallel machine lot-streaming scheduling problem with limited and unequal sub-lots," Comput. Ind. Eng., vol. 183, Sep. 2023, Art. no. 109428.

18. 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