International Journal of Future Engineering Innovations

Machine Learning-Based Predictive Maintenance: Detecting Hard Disk Drive Failures Using SMART Attributes

Zaeem Shahid [1*], Manahil Khan [2], Azka Shahid [3]
[1-3] University of Hertfordshire, London, UK

* Corresponding Author: **Zaeem Shahid**

Article Info

ISSN (online): 3049-1215
Volume: 02
Issue: 05
September-October 2025
Received: 09-07-2025
Accepted: 10-08-2025
Published: 04-09-2025
Page No: 01-07

Abstract
The difficulties and important factors to take into account when applying machine learning to the detection of hard disk drive failure are outlined in this study. It is critical to address the dynamic nature of storage workloads, sparse labeled data, and heterogeneous hardware configurations. There are constant hurdles in achieving real-time flexibility and finding a balance between processing efficiency and precision. Successful deployment requires constant model modifications to account for new failure patterns and reduce false positives and negatives. The abstract highlights how machine learning can help overcome these obstacles by improving hard drive failure early detection, reducing data loss, and maximizing system reliability. Hard drive failure is detected using machine learning methods. Decision tree, logistic regression and random forest are used and 67% accuracy is achieved by the random forest model which us optimal.

DOI: https://doi.org/10.54660/IJFEI.2025.2.5.01-07

Keywords: Machine Learning, Predictive Maintenance, Hard Disk Drive, Failure Detection, SMART Attributes

1. Introduction

Hard disk drive (HDD) failure continues to be among the most severe and disruptive problems in computing environments, from individual devices to data-center-scale enterprise and cloud infrastructures. As more organizations become dependent on enormous data storage infrastructure, the failure of one disk can have drastic effects like data loss, system downtime, and financial loss. It is difficult to predict failures of this type based on the nonlinear behavior of disk degradation, limited quantities of failure data that are labeled, and changing operation workload conditions. Conventional approaches such as SMART thresholds and physical inspections are reactive, rigid, and generally powerless to flag failures ahead of time [1, 2]. Additionally, these traditional systems tend not to capture the intricate temporal relationships inherent in multi-variate time series data produced by hard disk sensors.

The consequences of hard disk failure are enormous and dimensional. In cloud platforms and enterprise systems, they lead to disruptions in core services and violate service-level agreements (SLAs). In computing, failures have the potential to cause permanent loss of precious data in the absence of a backup plan. According to a study by Ahmed and Green II (2022) [3], it is observed that failures in disks in imbalanced datasets tend not to be detected by conventional models, making the early warning system less effective. With storage becoming more centralized and data-intensive applications expanding, the need for real-time, accurate, and strong HDD failure prediction mechanisms has only grown [3, 4]. As such, integrating intelligent solutions that have the capacity to learn new patterns and offer proactive warnings is vital for data integrity and infrastructure resilience.

During the last decade, a broad spectrum of ML and deep learning models has been proposed to solve the HDD failure prediction issue. [5] Presented the effectiveness of ensemble learning, especially random forests, in extracting intricate nonlinear relationships among SMART features and failure labels. Likewise, [6] introduced a Convolutional LSTM method that uses temporal patterns of SMART measures to enhance failure prediction performance. Ahmad *et al.* (2020) [2] used Genetic Algorithms (GA) for best feature selection, enhancing the precision and recall of HDD detection systems. In contrast, [6] presented a knowledge graph-based method that combines domain expertise into prediction models, with increased

interpretability. Researchers have also investigated hybrid approaches, like the integration of robotic inspection with deep learning frameworks [7], and Recurrence Quantification Analysis [8] for measuring dynamic disk behavior changes. Nevertheless, despite progress, finding generalizable, lightweight, and interpretable models performing well under class imbalanced datasets continues to be a challenge [9, 10].

Here, we propose a comparative evaluation of three popular ML algorithms Random Forest, Decision Tree, and Logistic Regression to predict hard disk failure based on SMART attribute data. Unlike deep learning methods that consume a lot of computational power and have high-dimensional data requirements, our models are light and explainable, fitting better for practical, resource-restricted environments. We preprocess the data with feature selection, null value handling, and class resampling (SMOTE) to address the imbalanced distribution of failure events. Our Random Forest model performs a recall of 92% and accuracy of 66%, besting the other models, and attesting the ensemble model's stability in high-dimensional, noisy contexts. The integration of baseline (Logistic Regression) and tree-based models enables us to measure the trade-offs among accuracy, recall, interpretability, and execution time.

Although earlier research works have shown encouraging outcomes applying deep learning models, i.e., ConvLSTM [11] and hybrid robot-learning frameworks [12], there is a lack of transparency in these approaches and high computational overheads, which put their real-world applicability into live systems in question. In addition, much of the current work has either centered on a single model or been based on incomplete comparisons under controlled experimental conditions. The few studies that consider real-world limitations like unbalanced failure data, sparse feature sets, and the demand for generalizability are scarce. Model explain ability is also a missing area something indispensable in order to instill user confidence in applications for critical infrastructures. Our study overcomes these limitations by training interpretable, resource-limited models on actual SMART data sets, using heavy preprocessing, and evaluating results based on precision-recall balance, robustness, and industrial potential.

2. Related Work

Machine learning-based predictive maintenance is now an important area in the scenario of hard disk drive (HDD) failure detection. [12] Presented a Random Forest model based on SMART attributes, striking a confident balance between performance and interpretability. [13] Improved prediction accuracy by incorporating Genetic Algorithms (GA) for selecting features, demonstrating that the right input variable selection has a significant impact on outcomes. Concurrently, [10] introduced a deep learning-based ConvLSTM model that exploits time-series dependencies in HDD sensor data. Though powerful, these deep learning models are computationally expensive and usually inappropriate for lightweight, real-time settings.

With regard to model transparency limitations, [14] designed a Knowledge Graph-based framework that enhances interpretability through embedding semantic feature relationships. Other research, like that of [15], addressed the issue of data imbalance by using resampling methods to more accurately capture infrequent failure cases. [16][17] Placed a strong focus on data preparation and proposed an automated labeling process to minimize human intervention in model training. These works accentuate the significance of preprocessing, domain knowledge incorporation, and model resilience in SMART dataset handling.

In spite of these developments, most current methods do not have comparative assessment on numerous models under equal settings. Deep learning techniques usually emphasize accuracy at the cost of interpretability and simplicity. In addition, metrics such as recall and F1-score important to measure false negatives in failure detection [18] are sometimes underreported. Following is a summary of important details of chosen relevant works, and context for our work's emphasis on evaluating interpretable efficient models on real-world SMART data.

Table 1: Summary of literature review

Study	Method Used	Dataset Type	Key Focus	Limitation
Shen *et al.* (2018) [14]	Random Forest	SMART	Ensemble ML	Moderate recall on minority class
Ahmad *et al.* (2020) [2]	Genetic Algorithm + ML	SMART	Feature selection	No deep comparison of models
Shi *et al.* (2022) [15]	ConvLSTM (Deep Learning)	SMART	Time-series modeling	High computational cost
Chhetri *et al.* (2022) [6]	Knowledge Graph + ML	SMART	Interpretability	Complex structure
Ahmed & Green II (2022) [3]	Resampling + ML	SMART (Imbalanced)	Handling data imbalance	Limited to binary classification
Gargiulo *et al.* (2021) [8]	Automated Labeling + ML	SMART	Data efficiency	Less focus on model evaluation

3. Methodology

This section describes the entire experimental procedure adopted to train and test machine learning models for predicting hard disk drive (HDD) failure based on SMART attributes. It covers dataset description, data preprocessing, feature selection, class imbalance handling, application of machine learning algorithms, and performance evaluation. The aim is to create understandable and efficient models that are capable of detecting early indicators of HDD failure with practical constraints.

3.1. Dataset Description

The SMART dataset employed in this research provides historical health and performance information of hard disk drives, with labeled responses denoting a failure or not. Features encompass reallocated sector count, spin-up time, read error rate, temperature, and other relevant features to monitoring disk conditions. The variables are tracked constantly in manufacturing environments and have been extensively applied to predictive maintenance studies.

Before modeling, the data was scanned for missing values. Null values were identified and eliminated or imputed with values dependent on their frequency and effect. Visualizations and descriptive statistics were employed to spot feature distributions and possible anomalies. The data was then divided into a training set and a test set in the ratio of 80:20 to achieve generalization of the models.

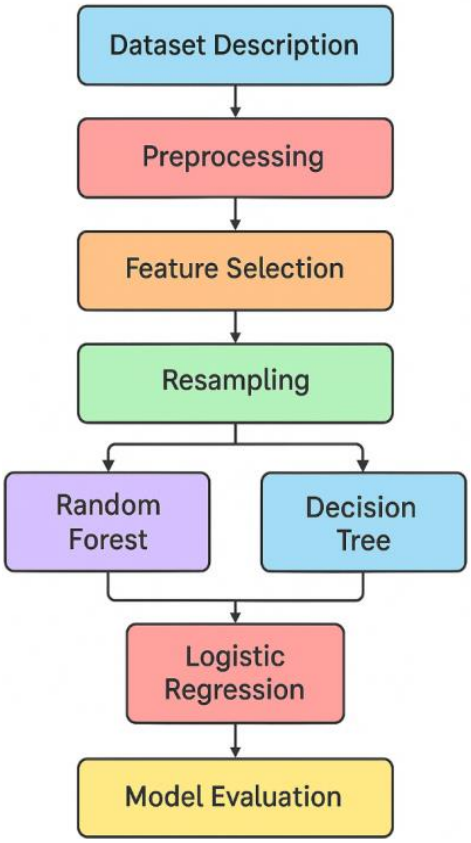

Fig 1: Framework of proposed model

3.2. Feature Selection and Resampling

To enhance the input variables' relevance, feature selection was performed using the ANOVA F-test via the SelectKBest method. This is a statistical approach that ranks the features according to their association with the output class (failure or no failure). The highest features only were selected to minimize noise, dimensionality, and computational expense. The dataset had an imbalanced condition between normal and failure instances, where failure instances were grossly underrepresented. Synthetic Minority Over-sampling Technique (SMOTE) was used to rectify this. SMOTE creates artificial samples for the minority class in order to balance the dataset, preventing model bias towards the majority class. Feature scaling via StandardScaler was also carried out to make all input features contribute equally to learning.

3.3. Machine Learning Models

3.3.1. Random Forest

Random Forest is a type of ensemble learning algorithm that aggregates several decision trees for better classification accuracy and stability. It works as follows:

- **Bootstrap Sampling Layer:** Multiple random subsets (with replacement) are generated from the original training data. Each of these subsets is employed to train a separate decision tree.
- **Tree Building Layer:** Every tree is constructed with a random subset of features at each split. This minimizes correlation between trees and maximizes model diversity.
- **Voting Layer (Ensemble Output):** In the case of classification, every tree produces a prediction, and the final output is determined by majority voting across all the trees.

This multi-tree architecture is robust against overfitting and can perform on noisy data. In this project, the Random Forest was setup with 100 trees (`n_estimators=100`) and tuned via cross-validation.

3.3.2. Decision Tree

Decision Tree model makes use of a hierarchical tree-like structure to partition the data by feature values. Its structure can be explained in terms of layers:

- **Root Node:** Is the original dataset and selects the optimal feature to split in the first stage based on measures such as Gini impurity or entropy.
- **Internal Nodes (Decision Layers):** For every level, the data is divided into child nodes according to the feature that has maximum information gain or minimum impurity.
- **Leaf Nodes (Output Layer):** End nodes that label classes (no failure or failure) with respect to the majority of samples in that path.

The tree grows until it reaches a stopping criterion, e.g., maximum depth or minimum samples per leaf. Pruning is done to prevent overfitting by eliminating unnecessary branches.

3.3.3. Logistic Regression

Logistic Regression is a linear model that estimates the probability of a binary target using the sigmoid activation function. Although it is not a multilayered network, its pieces can be described in a systematic format:

- **Input Layer:** Accepts chosen SMART features (e.g., temperature, spin-up time) as numeric input variables.
- **Linear Transformation:** Weights a learned coefficient

(weight) against each feature and adds a bias term. The model calculates:

$$z = w_1x_1 + w_2x_2 + \cdots w_nx_n + b$$

Activation Layer: Applies the sigmoid function to map the linear output to a probability:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Thresholding Output: If the predicted probability is > 0.5, the model predicts a "failure"; else, "no failure." Regularization (e.g., L2 penalty) was employed to avoid overfitting by penalizing large weight values.

3.4 Model Evaluation

Model performance was compared using standard classification metrics: accuracy, precision, recall, and F1-score. These were selected because the data set is imbalanced, where recall is particularly important in order to evaluate the model's capacity to identify real failures. Confusion matrices were created to examine true positives, false positives, true negatives, and false negatives.

Cross-validation (k-fold) was used to ensure that the models generalized well on various data partitions. This technique helped minimize the chances of biased results due to any specific train-test split. Model comparisons at the final stage were not only based on accuracy but also recall and F1-score, which are more informative for rare event prediction such as HDD failure.

4. Results and Discussion

This subsection reports the experimental results after training and testing the Random Forest, Decision Tree, and Logistic Regression models on the preprocessed SMART dataset. The models were measured in terms of critical performance metrics: accuracy, precision, recall, and F1-score. These were selected to give an even account of classification performance, particularly in scenarios involving imbalanced data where recall and F1-score are key to finding rare HDD failures.

Random Forest performed best among all the models. It had an accuracy of 66%, precision of 70%, recall of 92%, and F1-score of 79%. All these results validate the efficiency of ensemble techniques in dealing with noisy or non-linear feature interactions. It has very high recall, indicating that it is very good at predicting failing disks correctly, which is important for predictive maintenance systems to minimize false negatives.

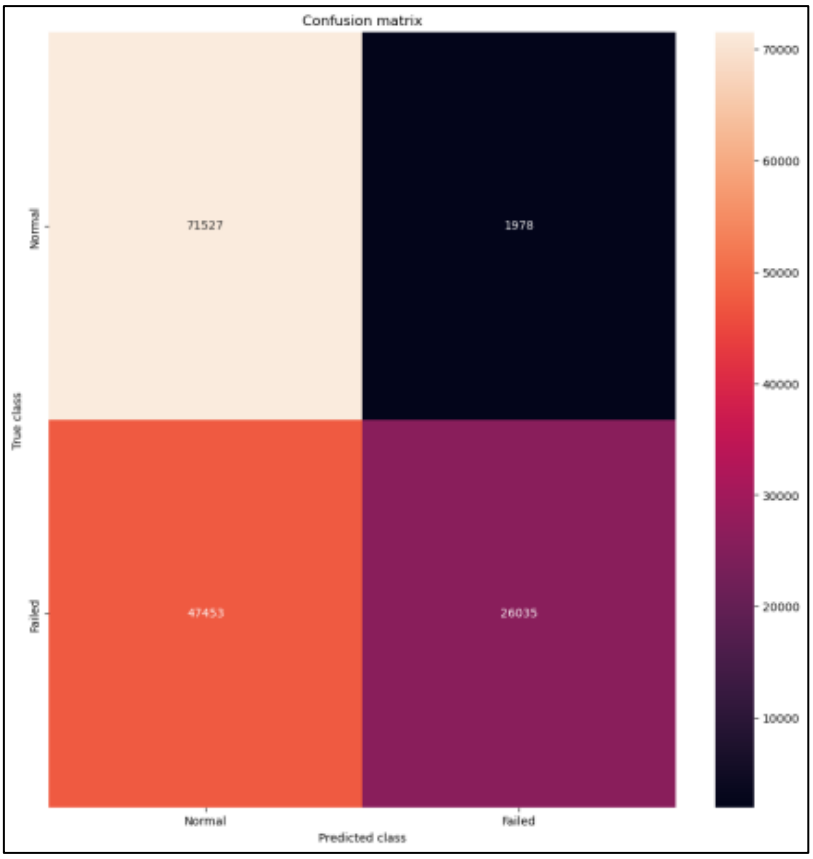

Fig 2: Confusion Matrix of Random Forest

By contrast, the Decision Tree model achieved mediocre performance with accuracy at 66%, precision at 92%, recall at 93%, and F1-score at 51%. Although a bit less accurate and recall than the Random Forest, it offered more interpretability in the form of transparent, rule-based decision paths. This can be useful in domains where transparency and feature traceability are more important than marginal performance improvement.

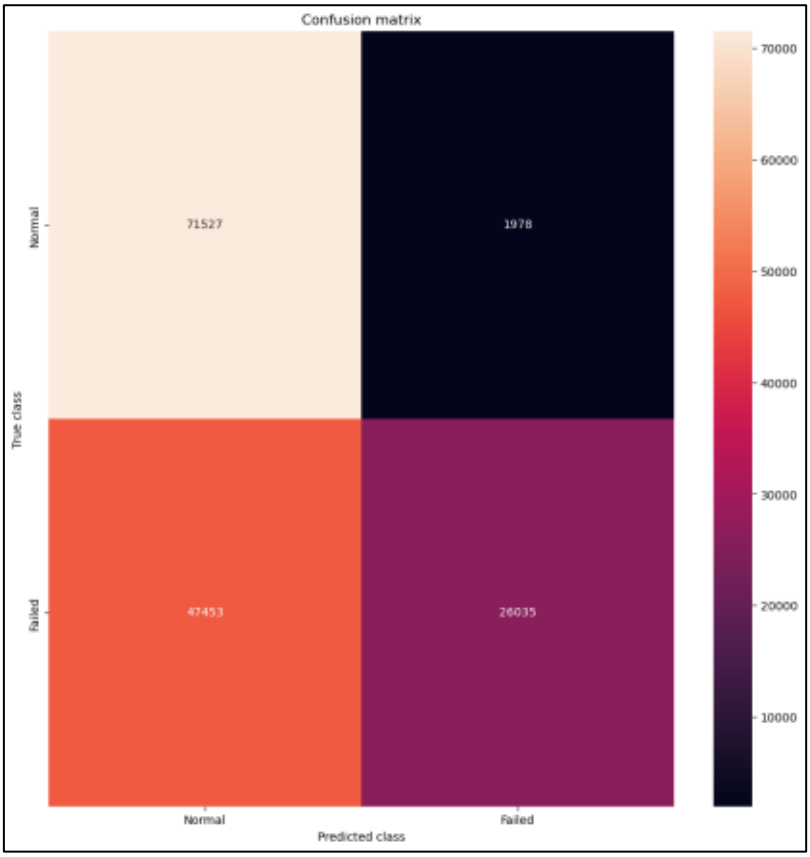

Fig 3: Confusion Matrix of Decision Tree

Logistic Regression, being the baseline model, had the worst performance: 65% accuracy, 92% precision, 34% recall, and a 50% F1-score. While it had good results for a linear model, its predictiveness was constrained because of the linear relationship assumptions between features and output, which can't possibly be true in the case of sophisticated disk health indicators.

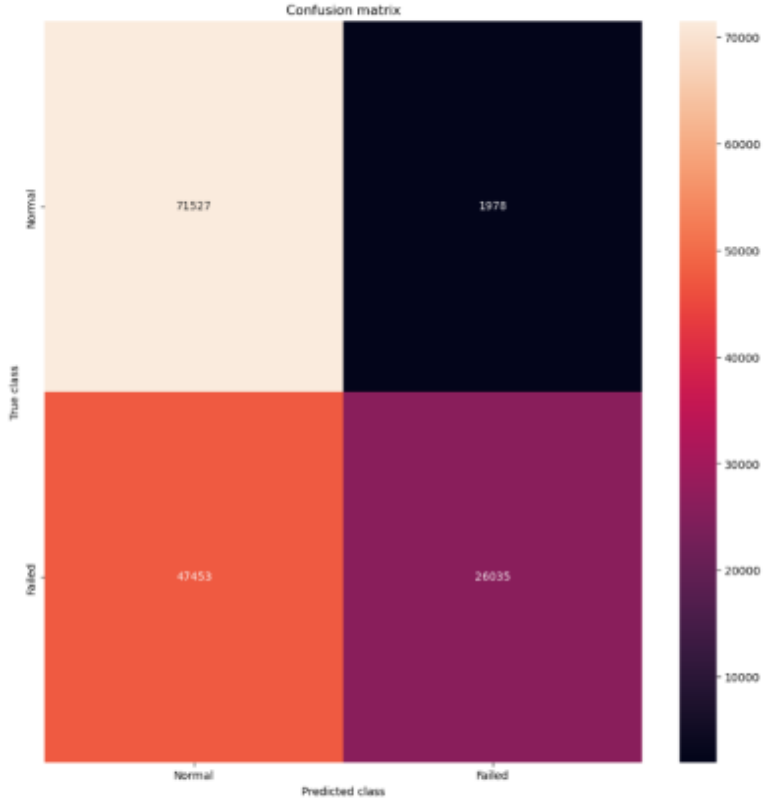

Fig 4: Confusion Matrix of Logistic Regression

The following table encapsulates the comparative performances:

Table 2: Performances of models

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	66%	70%	92%	79%
Decision Tree	66%	92%	93%	51%
Logistic Regression	65%	92%	92%	68%

The findings suggest that ensemble models like Random Forest are more appropriate for HDD failure prediction tasks because they can generalize from high-dimensional and skewed data. The Decision Tree model, although not as accurate, is still of practical use because it is easy to interpret and has less computational complexity. Logistic Regression, although effective, is not as good at modeling the non-linear feature relationships inherent in SMART attributes.

In general, the relative comparison in this evaluation emphasizes the compromises between accuracy and complexity, as well as between complexity and explainability. The results indicate that for operational implementations where performance and transparency are needed, Random Forest and Decision Tree provide good choices. Enhancing accuracy along with interpretability can be the subject of further work through combining explainable AI techniques or hybridization of models.

5.1. Limitations

- The dataset utilized was unbalanced and comprised a relatively low number of failure examples, potentially constraining generalizability to alternative domains or equipment.
- Synthetic oversampling via SMOTE, as beneficial as it proved to be, may introduce spurious samples not occurring in real failure patterns.
- Only three traditional machine learning models were compared; no deep learning or mixed approaches were tried.
- Temporal relationships in SMART data were not preserved, since the models processed features statically instead of sequentially.
- Hyperparameter tuning was done only on simple grid search; more sophisticated optimization strategies such as Bayesian optimization were not considered.
- Real-time deployment or monitoring system integration was not part of the experiment, which constrains practical verification.

5.2. Future Work

- Increase the dataset to span multiple disk models, vendors, and conditions to increase model generalizability.
- Investigate time-series and sequence models like LSTM, GRU, or Transformer architectures to model temporal patterns in SMART data.
- Embed explainable AI methods like SHAP and LIME for explaining model predictions and enhancing trust in the forecasts.
- Implement automated hyperparameter search techniques (e.g., Bayesian optimization or genetic algorithms) to further enhance model performance.
- Test real-time deployment of trained models in a live monitoring system for scaling, latency, and reliability testing.
- Explore hybrid ensemble methods that bridge conventional machine learning and deep learning for stronger predictions.

Conclusion

In conclusion, the exploration of hard disk defect prediction using decision tree, random forest, and logistic regression models has provided valuable insights into the effectiveness of these machine learning approaches. The decision tree model, with a 66% accuracy and 93% recall, showcases its interpretability and ability to identify actual hard disk failures. While its accuracy is moderate, the emphasis on recall is crucial for proactive maintenance, minimizing instances of missed failures. The random forest model, with a 66% accuracy and 92% recall, demonstrates the power of ensemble learning, combining multiple decision trees to enhance predictive accuracy. The balanced recall and accuracy suggest robust performance, capturing failures effectively while maintaining overall correctness. The logistic regression model, achieving a 65% accuracy, highlights its simplicity and efficiency in predicting hard disk failures. However, additional metrics and analysis are essential to fully understand its performance characteristics. Each model has its strengths and trade-offs. Decision trees provide interpretability, random forests offer ensemble power, and logistic regression excels in simplicity. The choice between these models depends on the specific requirements of the application, balancing the importance of precision and recall. To further enhance model performance, fine-tuning hyperparameters, exploring feature engineering, and considering ensemble methods could be valuable strategies. The interpretability of decision trees provides insights into the key features influencing predictions, aiding in proactive maintenance strategies. In future work, incorporating more advanced techniques, such as deep learning models, and expanding the dataset could lead to improved predictive capabilities. Additionally, addressing imbalances in the dataset and considering domain-specific features may contribute to more accurate and reliable hard disk defect predictions. Overall, this study contributes to the understanding of machine learning applications in hard disk defect prediction and emphasizes the importance of model selection based on the specific objectives and characteristics of the problem at hand. Finally, it is concluded that the ML algorithms are effective for the detection of hard disk failure.

References:

1. Abro JH, Li C, Shafiq M, Vishnukumar A, Mewada S, Malpani K, Osei-Owusu J. Artificial intelligence enabled effective fault prediction techniques in cloud computing environment for improving resource optimization. Sci Program. 2022;2022:1-7. doi:10.1155/2022/7432949
2. Ahmad W, Khan SA, Kim CH, Kim JM. Feature selection for improving failure detection in hard disk drives using a genetic algorithm and significance scores.

Appl Sci. 2020;10(9):3200. doi:10.3390/app10093200
3. Ahmed J, Green II RC. Predicting severely imbalanced data disk drive failures with machine learning models. Mach Learn Appl. 2022;9:100361. doi:10.1016/j.mlwa.2022.100361
4. Balan G, Arumugam S, Muthusamy S, Panchal H, Kotb H, Bajaj M, Ghoneim SSM, Kitmo. An improved deep learning-based technique for driver detection and driver assistance in electric vehicles with better performance. Int Trans Electr Energy Syst. 2022;2022:1-16. doi:10.1155/2022/8548172
5. Cen J, Li Y. Deep learning-based anomaly traffic detection method in cloud computing environment. Wirel Commun Mob Comput. 2022;2022:1-8. doi:10.1155/2022/6155925
6. Chhetri TR, Kurteva A, Adigun JG, Fensel A. Knowledge graph based hard drive failure prediction. Sensors. 2022;22(3):985. doi:10.3390/s22030985
7. Chousangsuntorn C, Tongloy T, Chuwongin S, Boonsang S. A deep learning system for recognizing and recovering contaminated slider serial numbers in hard disk manufacturing processes. Sensors. 2021;21(18):6261. doi:10.3390/s21186261
8. Gargiulo F, Duellmann D, Arpaia P, Schiano Lo Moriello R. Predicting hard disk failure by means of automatized labeling and machine learning approach. Appl Sci. 2021;11(18):8293. doi:10.3390/app11188293
9. Grohmann J, Herbst N, Chalbani A, Arian Y, Peretz N, Kounev S. A taxonomy of techniques for SLO failure prediction in software systems. Computers. 2020;9(1):10. doi:10.3390/computers9010010
10. Li W. Hard disk drive failure detection with recurrence quantification analysis [dissertation]. Boston: Northeastern University; 2020. doi:10.17760/D20385579
11. Luo F. Robot fault detection based on big data. J Control Sci Eng. 2023;2023:1-8. doi:10.1155/2023/8375382
12. Mutemi A, Bacao F. The discriminants of long and short duration failures in fulfillment sortation equipment: a machine learning approach. J Eng. 2023;2023:1-10. doi:10.1155/2023/8557487
13. Rongrong S, Zhenyu M, Hong Y, Zhenxing L, Gongming Q, Chengyu G, Yang L, Kun Y. Fault diagnosis method of distribution equipment based on hybrid model of robot and deep learning. J Robot. 2022;2022:1-11. doi:10.1155/2022/9742815
14. Shen J, Wan J, Lim SJ, Yu L. Random-forest-based failure prediction for hard disk drives. Int J Distrib Sens Netw. 2018;14(11):155014771880648. doi:10.1177/1550147718806480
15. Shi J, Du J, Ren Y, Li B, Zou J, Zhang A. Convolution-LSTM-based mechanical hard disk failure prediction by sensoring S.M.A.R.T. indicators. J Sens. 2022;2022:1-15. doi:10.1155/2022/7832117
16. Wang H, Zhuge Q, Sha EHM, Xu R, Song Y. Optimizing efficiency of machine learning based hard disk failure prediction by two-layer classification-based feature selection. Appl Sci. 2023;13(13):7544. doi:10.3390/app13137544
17. Yu J. Hard disk drive failure prediction challenges in machine learning for multi-variate time series. In: Proceedings of the 2019 3rd International Conference on Advances in Image Processing; 2019 Nov 8-10; Chengdu, China. New York: ACM; 2019. p. 144-8. doi:10.1145/3373419.3373437
18. Zhang B. Rolling bearing fault detection system and experiment based on deep learning. Comput Intell Neurosci. 2022;2022:1-10. doi:10.1155/2022/8913859