Manufacturing Rev. **11**, 11 (2024)

https://doi.org/10.1051/mfreview/2024008

Manufacturing Review
Available online at:
https://mfr.edp-open.org

RESEARCH ARTICLE OPEN ACCESS

Enhancing testing cell set efficiency: A machine learning approach on hard disk drive data

Maneerat Rakcheep[1], Metinan Laosakun[2], Sorada Khaengkarn[1], and Jiraphon Srisertpol[1,*]

[1] Mechatronics Engineering Program, School of Mechanical Engineering, Suranaree University of Technology, Nakhon Ratchasima Province, Thailand 30000
[2] Western Digital Storage Technologies (Thailand) Ltd., BangPa-In Industrial Estate, Ayutthaya Province, Thailand 13160

Received: 24 November 2023 / Accepted: 19 March 2024

Abstract. Hard Disk Drive (HDD) products undergo meticulous testing procedures to ensure their functionality prior to customer distribution. Nevertheless, anomalies can arise within the testing environment due to various factors, such as an increased number of media discs, leading to heightened current consumption by the spindle motor, and the frequent insertion and removal of HDDs during testing. These factors can induce malfunctions within the testing cell, which are identified by the tester's program. This study leverages diverse data measurements collected from tester HDDs within the testing cell to predict the status of the testing cell itself. Five distinct algorithms—Linear Discriminant Analysis (LDA), Ridge Classifier CV (RCCV), Extra-Tree Classifier (ETC), Random Forest Classifier (RFC), and Extreme Gradient Boosting (XGBoost)—were assessed. The research underscores that the proposed methodology, particularly utilizing XGBoost, achieves a notable prediction accuracy of 87.9% when applied to real datasets.

Keywords: Hard disk drive (HDD) / testing cell / data analysis / machine learning / classifier

1 Introduction

The current hard disk drive (HDD) production process is highly modernized and entirely automated. Various data generated by machines across the production line are collected to bolster Industrial 4.0 technology. Consequently, the factory has integrated artificial intelligence technology to aid in data analysis, particularly in fault detection, diagnostic procedures, classification, isolation, and fault tolerance control [1–4]. This integration involves compiling essential information for decision-making, conducting visual inspections to ensure work quality, identifying and categorizing machine damage, and strategically planning the production process to optimize efficiency and maximize output. Within the HDD industry, comprehensive functional testing of each individual HDD before market release is imperative to mitigate potential repercussions. Failures, often originating from malfunctions in the HDD testing equipment referred to as the 'HDD testing cell' or 'Cell,' can disrupt the testing process, necessitating retesting. Consequently, products failing to meet standards may undergo regarding, resulting in price reductions. This underscores the critical roles of production efficiency and effective testing procedures in ensuring product quality.

At Western Digital Storage Technologies (Thailand) Ltd., the test equipment engineering department primarily manages testing cell maintenance, predominantly focusing on corrective measures. This research specifically addresses scenarios where HDD testing encounters failures in supplying the required 12 volts (for the spindle motor) and 5 volts (for the PCB), or provides voltage below the specified level (falling below −5% of the voltages), indicated by the PF code 'HVL' (Hardware Voltage Low). Notably, in 2022, the HVL failure emerged as the most frequent malfunction, occurring 3,775 times over a six-month period from June to November, constituting 26% of all symptoms. Particularly in August alone, HVL failure peaked at 1,964 occurrences. The department's inability to anticipate HVL incidents due to a lack of readiness in procuring spare parts exacerbates this issue, providing the context for this research initiative. To minimize data redundancy, only the top 5 products (A, B, C, D, and E) with the largest datasets will be selected for modeling. The data will originate from two tester models, model X and model Y, both utilizing the same testing cell model 3. The primary objective is to develop a tool capable of categorizing testing cell performance levels into three

* e-mail: jiraphon@sut.ac.th

distinct categories: 'Strong' (Class 1), 'Average' (Class 2), and 'Weak' (Class 3). This categorization will involve a comprehensive evaluation of HVL risks, requiring the model to achieve at least 80% accuracy. This will be facilitated by applying both statistical insights [5] and machine learning techniques [6,7].

2 Literature reviews

Wang at al. [8], the authors present a study based on hardware failure reports collected over the past 4 years, encompassing hundreds of thousands of servers, where undertake a statistical examination of the dataset to discover patterns in failure characteristics across temporal, spatial, product line, and component dimensions. The study places particular emphasis on exploring correlations among diverse types of failures, including batch and recurring failures.

Customer behavior anomalies related to prepaid electricity pulse usage are examined by Lawi et al. [9]. The authors employ classification methods which are LDA and Logistic Regression (LR). Experimental results, conducted with varying amounts of data, demonstrate that the LR method achieves exceptional accuracy, precision, and recall values, all reaching 100% when compared with LDA. This robust performance is attributed to the method's ability to accurately predict irregularities, unaffected by the quantity of data used.

Pereira et al. [10], a performance comparison between an Artificial Neural Network (ANN) and LDA is conducted using the parameters from the Wisconsin Breast Cancer Database's Fine Needle Aspiration. Effective diagnosis is vital for improving the chances of a cure. This article analyzes the outcomes of these algorithms and explores potential enhancements in performance. The results indicate a correct diagnosis rate of 95.77% for LDA and 92.78% for ANN.

Classification event involves mapping real-time data scraped from Twitter to its corresponding generalized hashtag by Singh et al. [11]. The experiment's objective is to compare the prediction accuracy of the Ridge Classifier (RC) and LDA. The data comprises tweets from 6 different topics: soccer, USA, apple, cars, Hollywood, and philosophy, collected using the Tweepy library. The conclusion suggests that RC outperforms LDA because nonlinear methods can be overly powerful, making it challenging to prevent overfitting. It is possible that a nonlinear classifier could offer better generalization performance than the best linear model.

A prediction event involving wind speed forecasting is conducted, relying on the ETC by Grace and Priyadharshini [12]. Wind speed forecasting is a crucial aspect of wind farm management, especially for dealing with the nonlinear behavior of wind in time series data. The ETC approach is compared with the Bagging Classifier (BC) and Adaptive Boosting Classifier (ABC). Wind energy, although free, is highly unpredictable, making accurate wind speed and direction prediction challenging. Time series data is essential for precise processing with the ETC. The dataset comprises historical values of parameters such as wind speed, temperature, and atmospheric pressure. The results indicate an accuracy of 98.9% for ETC, 94.3% for BC, and 91.9% for ABC, respectively.

Dhananjay et al. [13] employed the ETC to classify cardiac abnormalities into three classes: Sinus Rhythm (SR), Sinus Tachycardia (ST) under physical stress, and Atrial Tachycardia (AT). Manually measuring these conditions, especially when they exhibit similar wave morphologies, is a challenging task. The clinical morphologies used for classification encompass the durations (ms) of P wave, PR Interval (PRI), QRS complex, T wave, QT Interval (QTI), PP Interval (PPI), and amplitudes (μV) of P, R, and T waves. The precision, recall, and f1-scores achieved by the developed ETC: SR – 0.99, 0.93, and 0.96, ST – 0.99, and AT – 0.95, 0.99, and 0.97, respectively. The advantage of the ETC over other classifiers is in its ensemble-based nature, derived from a Decision Trees (DT) classifier, which helps prevent overfitting.

In the study by Shen et al. [14] present an HDD failure prediction method utilizing the RFC algorithm. Their experiments were conducted on two real-world datasets from Baidu, including two drive families as S and T, and Backblaze, which belongs to family B. These datasets comprise both functional and failed drives and incorporate Self-Monitoring, Analysis, and Reporting Technology (SMART) data. The evaluation metrics of the prediction method yielded impressive results, achieving a Failure Detection Rate (FDR) of 97.67% with a relatively high False Alarm Rate (FAR) of 0.017% for family B. For family S, an FDR of 100% with an FAR of 1.764% was attained, and for family T, an FDR of 94.89% with an FAR of 0.44% was achieved.

Alshboul et al. [15] introduce a method for predicting construction costs in green building, where the aim is to estimate contract costs effectively. These green buildings incorporate innovative technologies to minimize their environmental and societal impacts during operation. The study employs machine learning techniques such as XGBoost, Deep Neural Network (DNN), and RFC for modeling. The objective is to provide stakeholders with an initial construction cost benchmark to enhance decision-making. XGBoost demonstrated the highest accuracy, achieving 0.96, followed by 0.91 for DNN, and 0.87 for RFC.

In the study by Karthigha and Akshaya [16], the focus is on predicting sepsis to enhance the likelihood of survival through early detection. The dataset used for this research is sourced from ICU patients across three distinct hospital systems. The primary objective is to develop and validate an XGBoost model for sepsis prediction. The performance of XGBoost is then compared with other machine learning models, including DT, Gradient Boosting Trees (GBT), and RFC. XGBoost algorithm demonstrates superior performance when compared to DT, GBT, and RFC, highlighting its effectiveness in sepsis prediction.

In the work of Irawati and Zakaria [17], a classification model for COVID-19 was developed based on cough sounds using the XGBoost algorithm. The dataset was obtained from Virufy, a global cough sound database for AI applications, and the Coswara project, an initiative focused on audio data for respiratory, cough, and speech analysis

for COVID-19 diagnostics. Mel Frequency Cepstral Coefficients (MFCC) were employed as the feature extraction method for machine learning. The XGBoost was utilized for classification, achieving an impressive accuracy rate of 86%. This suggests its potential as a pre-screening tool for COVID-19 in wider community contexts.

Li et al. [18], conducted XGBoost, Long Short-Term Memory (LSTM), and ensemble learning algorithms are employed to predict disk faults using SMART data from monitoring systems in computer HDDs and SSDs. These systems are designed to detect and report reliability indicators to anticipate hardware failures. The experiments demonstrate that their method can effectively predict disk faults within 42 days with an accuracy rate of 78%, meeting the standard for production availability.

Miller et al. [19] investigate the influence of age and workload on the Annualized Failure Rate (AFR) of HDDs, commonly used in Meta's applications. Their datacenter observations reveal that HDD AFR increases with both age and cumulative workload. Furthermore, they employ XGBoost, a decision tree-based machine learning algorithm, to analyze the correlation between SMART metrics and HDD health. They developed a classifier to distinguish healthy and unhealthy drives based on SMART metrics, finding that age and workload-related SMART parameters exhibit the highest correlation with drive health according to the trained machine learning model.

3 Material and methodology

3.1 Hard disk drive testing cell set

Post-production testing, or product testing, is a process aimed at evaluating the performance of a product prior to its delivery to customers. The primary objective of this process is to ensure product quality, with factors influencing quality including the efficiency of the production process. This encompasses the materials utilized, the production steps, such as forming and assembling parts, and another equally crucial factor, the performance of the machines used for product testing. This is significant as it relates to expenses, wasted resources, and energy consumption, all of which can impact product quality. In industries such as HDD manufacturing, product testing entails a 100% inspection and evaluation of products. This implies that every HDD must successfully pass a functional test before being released to the market. Therefore, if the performance of the testers or equipment is not controlled in accordance with standards, the repercussions of failed tests can be substantial.

The criteria for product testing in HDDs include instances where testing failures are not attributed to faults within the product itself, but rather to abnormalities in the tester or test equipment during operation. Such abnormalities may cause the machine to halt unexpectedly, even before completing the testing process. Consequently, these HDDs must undergo retesting. This issue directly impacts production speed, also known as Takt Time, as newly produced batches of HDDs must wait for previously tested batches to be retested before proceeding. Moreover, HDDs that fail initial testing due to such abnormalities may experience a downgrade (Regrade) in product quality, leading to price reductions. Additionally, considering the conditions of product orders, it has been observed that some customers refuse to accept regraded products, posing a new challenge and resulting in unsold inventory and the loss of revenue for the company. Thus, ensuring product quality goes beyond merely controlling production efficiency; it also involves efficiently managing the testing process, which significantly influences product quality.

Referring to statistics on the occurrence of HVL abnormalities mentioned earlier, in August alone, the number increased to nearly two thousand. This means that nearly two thousand testing cells require repairs and replacement of related parts under specific conditions. Despite the equipment being in use for over ten years, the shift from hundreds to thousands of occurrences in a short period has led to a shortage of spare parts. Consequently, there are fewer successfully repaired testing cells than those awaiting repair, primarily due to insufficient repair parts.

A testing cell model 3 is employed for inserting HDDs into an HDD tester to assess the functionality of the HDDs. Each tester module contains 144 testing cells arranged in 24 rows and 6 columns. The model cell being studied incorporates two slots, enabling the concurrent testing of two HDDs as shown in Figure 1. This apparatus is responsible for establishing the connection between the HDD and the tester, as well as receiving commands to direct the HDD's actions, including reading, writing, and adjusting temperature levels as specified. This testing cell is comprised of various components, including:

- Rigid-Flex PCB
- Motherboard
- Drive fan
- Electronics fan
- Heater

One crucial component within the testing cell apparatus is the Rigid-Flex PCB (RFPCB). When encountering the HVL malfunction, the RFPCB takes precedence as the primary component that undergoes repair or replacement. The RFPCB is a specialized circuit board that seamlessly integrates aspects of both rigid and flexible PCBs within a single design. This hybrid structure comprises rigid segments made of solid, inflexible materials and flexible sections crafted from flexible substrate materials. The transition between these segments, facilitated by PCB stiffeners, enhances rigidity and mechanical strength, preventing warping or bending, especially in larger or thinner PCBs.

The primary functions of a RFPCB find immense utility in electronic devices characterized by complex, compact designs, fitting seamlessly into devices where space is constrained, such as smartphones and wearables. Notably, the flexible sections of the PCB serve as shock absorbers, effectively minimizing signal interference and promoting superior signal integrity. This attribute is pivotal in preserving the performance of delicate electronic components. Given these advantages, RFPCBs are highly versatile and find application across various sectors, including aerospace, defense, medical devices, automotive

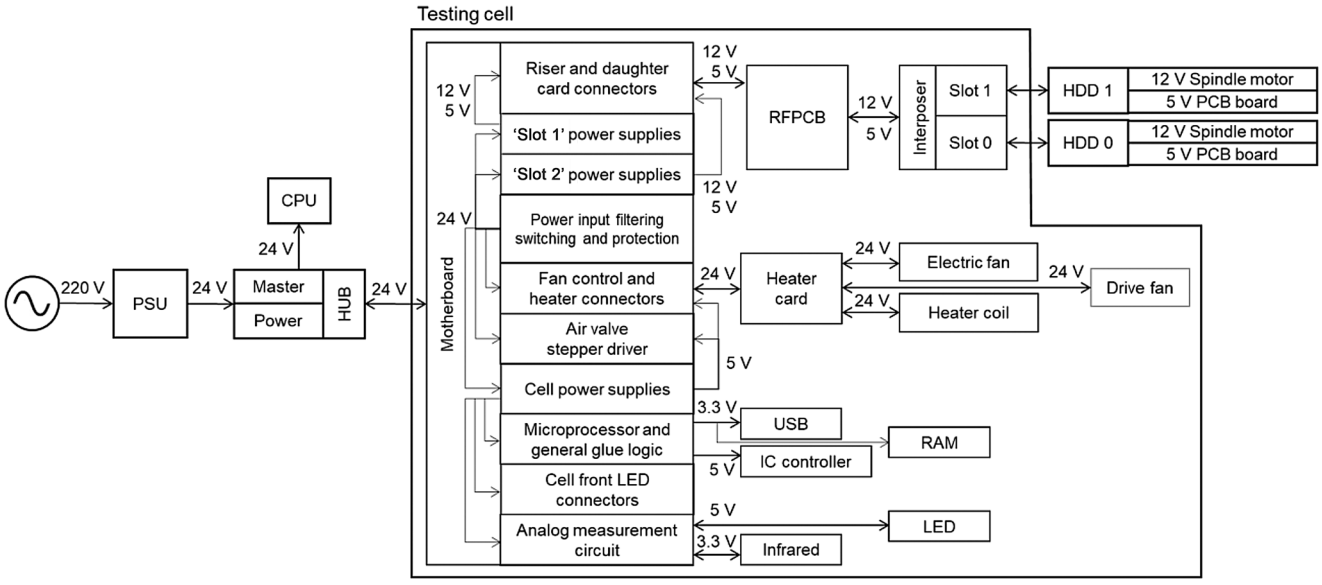

Fig. 1. Diagram illustrating the components. (From the voltage regulator to the internal details within the testing cell model 3).

electronics, consumer electronics, and industrial equipment. In these sectors, where space, reliability, and performance are paramount, RFPCBs prove to be indispensable.

3.2 Methodology

This section provides an in-depth overview of the dataset used in the study and outlines the analytical process utilizing Jupyter software to construct and implement machine learning algorithms (specifically LDA, RCCV, ETC, RFC, and XGBoost). Jupyter, an open-source software, was chosen for its extensive suite of learning algorithms encompassing both base classifiers and ensemble generation. These algorithms cover a spectrum of techniques, including supervised learning and unsupervised methods such as classification, association, clustering, and data visualization. The section further explores the intelligent classifiers employed in this research, shedding light on their selection and application. The architectural intricacies of the proposed model are elucidated and visually represented in Figure 2 for comprehensive understanding.

The process illustrated begins with receiving the testing cell displaying the HVL error code. However, as previously noted, the repair record is utilized as a screening criterion. Consequently, even if the error code manifests, the test equipment is only considered if the RFPCB has been repaired or replaced. In this phase, data is exported from the tester, encompassing 548 cells from HVL testing cells that meet the repair verification criteria. Subsequently, the raw data files within each device are amalgamated, extraneous information is removed like NaN, 'None', and 'none', and the dataset is sorted by datetime. Further refinement involves five conditions aimed at reducing algorithmic complexity and confusion in the model process:

Condition 1: Data is selected exclusively for product A, B, C, D, and E due to their extensive testing and rich dataset. This results in 21 cells being removed, leaving 527 cells.

Condition 2: The dataset is narrowed down to data from the hot temperature operation (at 60 °C), where HVL anomalies are mostly visible. Consequently, 145 cells are removed, resulting in 382 cells.

Condition 3: The dataset is filtered to include testing cells that are only in the plug-in status for each slot, providing test pairs where two HDDs are tested simultaneously.

Condition 4: Data sets that are more than six months old are selected. This six-month threshold was chosen due to the predominant availability of data averaging around 6 months, leaving 345 cells and 37 cells removed.

Before proceeding to the final condition, it is essential to divide the dataset into three groups: overall parameters, slot 0 parameters, and slot 1 parameters, for separate consideration.

Our focus is on slot 0 and slot 1, even though both HDDs start up simultaneously, the test details vary in terms of timing.

Condition 5: The dataset at this stage exclusively focuses on the duration when the testing cell is in an electric current supplying status. Our problem is specifically centered around the operational periods of the device.

Next, continuous parameters are selected, and through statistical tools. The dataset is then separated, allocating 45 cells out of total 345 cells for unseen data and 300 cells for model training using an 80–20% split for training and testing. Five distinct classification algorithms are employed for comparative analysis, aiming to select the optimal algorithm. The selection process involves hyperparameter tuning and prevention of model overfitting.

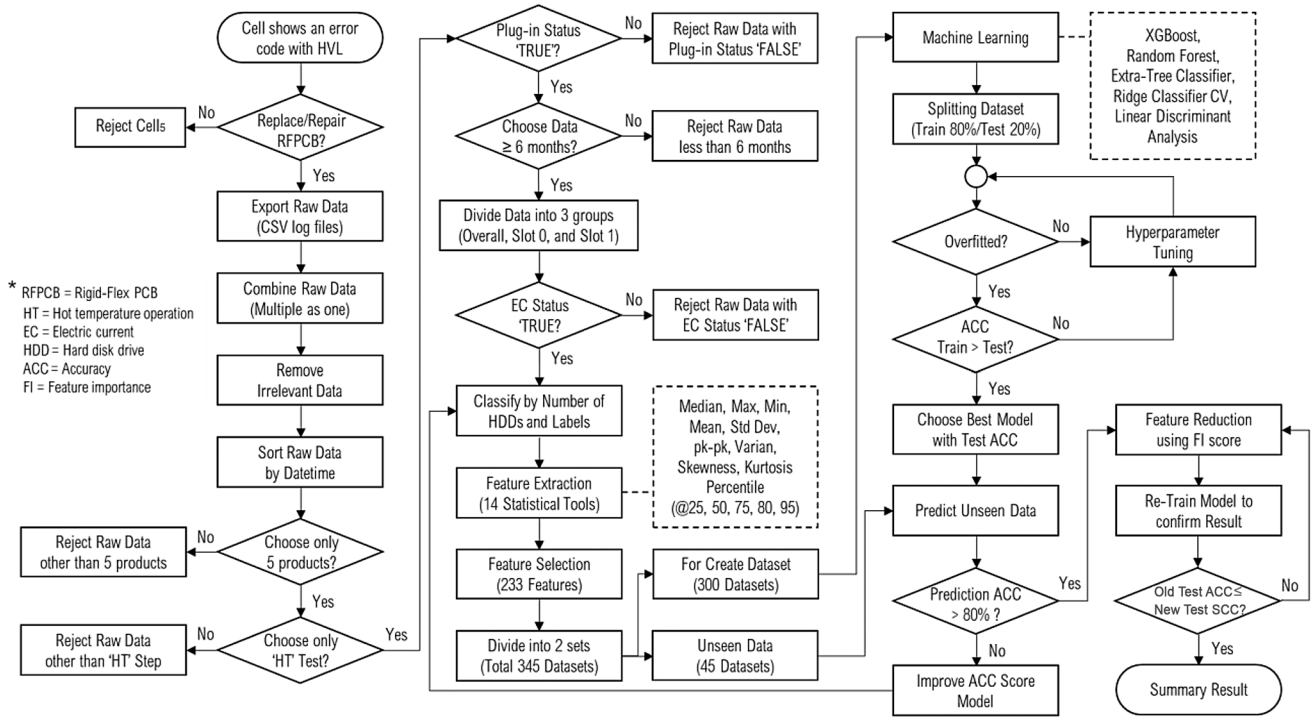

Fig. 2. Flowchart of the developed algorithm.

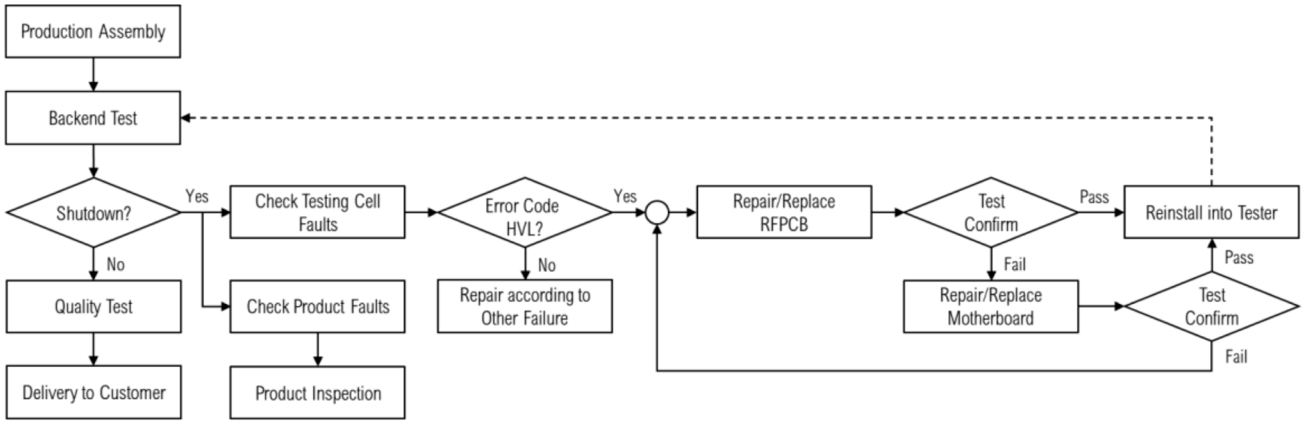

Fig. 3. The sequence of steps involved in testing cell maintenance following an HVL failure.

Further feature reduction is performed using Feature Importance (FI), reducing the selected feature set to retain only relevant features in order (where the sum of all feature scores equals 1). The algorithm model is retrained and utilized to predict unseen datasets by determining the appropriate importance score through evaluating the model's performance.

3.2.1 Database analysis

The HDD test, also known as the "Backend test" is conducted on each product family, which includes legacy-X (L-X) and legacy-Y (L-Y). While L-X represents the client drive category, it is a common internal/external storage drive used in general computers on the market or portable storage devices, L-Y is categorized as an enterprise drive, it is a cloud storage used in large organizations. The backend test for both L-X and L-Y comprises several stages in which HDDs undergo writing and reading operations. These stages include helium filling, ambient temperature, hot temperature, post-hot temperature, and configuration testing.

Inside the tester, there is a program installed to check for malfunctions in the device and to shut down when the program detects a fault. Each fault is indicated by an error code for maintenance purposes. The maintenance method for HVL fault symptoms is shown in Figure 3. The repair method for HVL fault symptoms is valuable for selecting

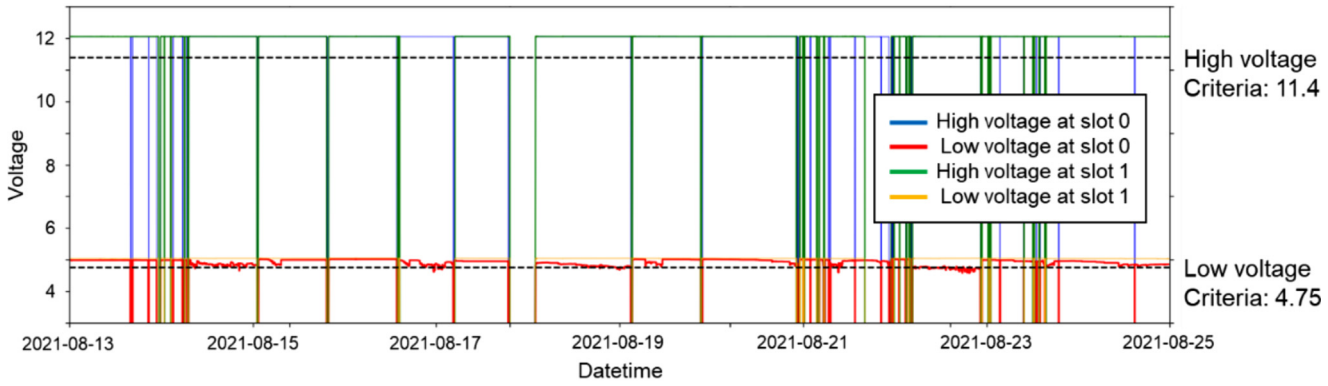

Fig. 4. Plot of the voltage at slot 0&1 of the testing cell model 3 from tester model Y.

Table 1. The weighted sum for each product family.

Specification	Product family Product A	Product B	Product C	Product D	Product E
Platters/RPM	2/5,400	2/5,400	1/5,040	1/7,200	1/7,200
Average testing time (hr)/2HDDs	55.249	30.109	30.142	11.573	10.566
Number of HDDs	656	788	636	362	1,832
Assigned weight	2.302	1.255	1.256	0.482	0.440
Number of HDD weighted	1,510	989	799	175	807

raw data to create a machine learning model. In situations where the testing cell stopped working, but no abnormalities were detected upon bench test inspection, data export will commence from the previous repair record instead of the direct search from the tester by error code.

In reference to HVL, the key parameters closely linked with the disorder are high voltage and low voltage at slot 0, as well as high voltage and low voltage at slot 1. These variables function as criteria for detecting abnormalities. As illustrated in Figure 4, a notable portion of abnormal symptoms is evident in slot 0. This occurrence is primarily attributed to the device's internal mechanism exerting more force against slot 0 during the HDD's insertion and removal, as compared to slot 1.

3.2.2 Dataset

The dataset comprises 345 testing cells, with 45 testing cells reserved for predictive unseen data. The parameters are gathered from diverse components within the testing cell, as detailed in Figure 1. For machine learning model development, we utilize a dataset spanning at least six months, focusing specifically on cells that were shut down due to HVL (bad cells). The Comma-Separated Values (CSV) log file's structure organizes the dataset, emphasizing HVL-related parameters such as testing cell current, fan speed, temperature, voltage, etc. This dataset encompasses five product families (A, B, C, D, and E), which represent the largest volume of datasets and are currently in production and testing. It is important to note that the tests are specifically conducted under hot temperature operation conditions.

3.2.3 Fault types

In this research, we use the number of HDDs to classify performance level. However, the dataset contains varying numbers of HDDs for each product family of each testing cell. Hence, we adopt a weighted sum calculation, considering the specifications of each product family based on the testing time, represented as T_i, where T is the testing time average in hours and i is the product family divided by 24 h (the maximum testing time). Next, we multiply by N_i the number of HDDs for each product family. Let W_i be the weights attached to variable values X_i respectively. The overall weighted sum for each product type is calculated as follows:

$$\Sigma X_w = \Sigma\left(\frac{\overline{T}_i}{24} \times N_i\right) = \Sigma W_i X_i. \quad (1)$$

In this process, we achieve an overall weighted sum. Following this, we use equation (1) to calculate the weighted sum of 4,279 HDDs, each corresponding to a specific product, various weights. The results are presented in Table 1, where we showcase the outcomes. We possess the testing time for each product and aim to determine the weighted value for every product, as illustrated in the

Table 2. Testing cell performance levels considered in this work.

Alarm level	Level type	Level conditions	Dataset		Class number
			Training	Unseen	
0	Strong	$N \geq 6$	165	32	Class 1
1	Average	$N = 6$	234	39	Class 2
2	Weak	$N = 2$	300	45	Class 3

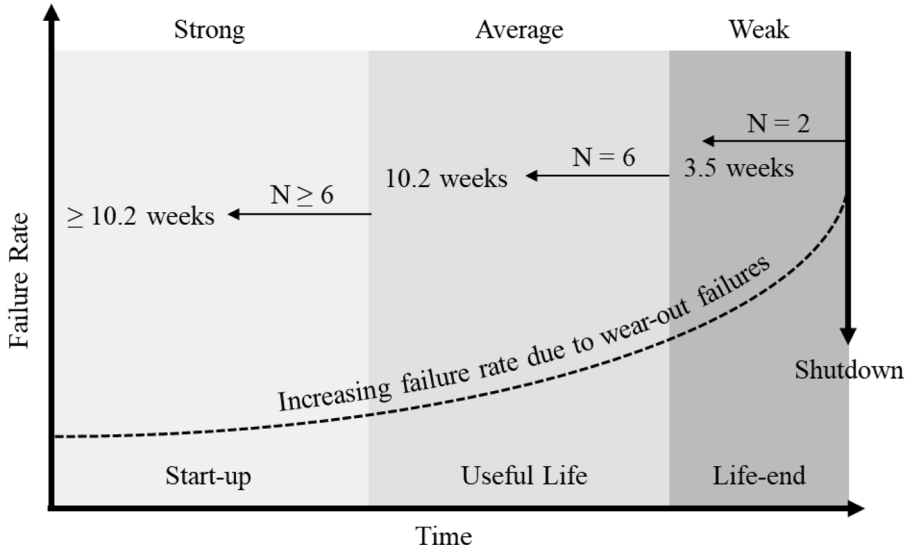

Fig. 5. Cell performance classification based on number of HDDs over time series.

'HDDs weighted' section. For example, let's consider the initial value of '1,510' in the HDDs weighted row for product A. We multiply the testing time (55.249 h) by the maximum testing time (24 hrs) to obtain the 'Assigned weight' of 2.302 for product A. Then, we multiply this result by the number of product A (656 HDDs). This dynamic approach allows for convenient adjustments to the weights.

Lastly, to calculate the average number of HDDs per week, we simply divide the total of 4,279 HDDs weighted by the number of testing cells for model training (300 cells) and the number of weeks (24 weeks; 6 months), resulting in an average of 0.59 HDDs per week.

The testing cell system experiences various types of faults, including overheating, or controller malfunctions. These issues are relatively easy to detect, occurring when the controller breaks down or the monitoring unit lags. Our study focuses on predicting and classifying three performance levels (presented in Tab. 2) that are challenging to determine solely based on voltage parameters at slot 0 and slot 1. This complexity arises because performance levels may not be directly reflected in the voltage of the testing cell.

Note. N = Number of HDDs. The conditioning of the number of HDDs to each level is determined through experimentation, involving adjustments and result comparisons until achieving the appropriate distribution of product numbers for each performance level.

We determine the specified number of HDDs for each performance level through numerical adjustments in classifying data sets for model training and result comparisons to obtain the appropriate numerical values. Considering our dataset spans a minimum of 6 months (24 weeks) and the average weight of about 0.59 HDDs per week, it is evident that we tested approximately 14 HDDs over a 24-week period. Referring to Table 2, we account for a total of 8 HDDs from the weak and average levels. Consequently, the strong level should have a minimum of 6 or more HDDs.

Testing cell performance classification is determined by the number of HDDs tested over a time series. As shutdown day approaches, the performance gradually weakens. Moving further from this point, the testing cell performance is categorized as average, and at a significant distance, it reaches the classification of strong performance, as in Figure 5. The number of HDD conditions in each

Table 3. Dataset parameters.

Parameters	Units
High voltage at slot 0	volt
Low voltage at slot 0	volt
High voltage at slot 1	volt
Low voltage at slot 1	volt
Riser temperature at slot 0	°C
Riser temperature at slot 1	°C
Temperature at slot 0	°C
Temperature at slot 1	°C
Drive fan voltage	volt
Drive fan demand	–
Drive fan speed	rpm
Electronic fan demand	–
Electronic fan speed	rpm
Cell voltage	volt
Cell current	amp
%CPU utilization	–
Motherboard temperature	°C
LDO voltage	volt
Cell power suppliers (3.3v side)	volt
Cell power suppliers (5v side)	volt

performance level is converted to time using the rule of three in arithmetic, derived from an average of 0.59 HDDs per week. For instance, in the weak level $N = 2$, this corresponds to approximately 3.5 weeks. On the other hand, for the average level where $N = 6$ and the strong level where $N \geq 6$, they correspond to approximately 10.2 weeks for the average and for the strong level. This transformation allows us to link the product conditions to a meaningful time frame, facilitating a better understanding of the performance levels over time.

3.2.4 Feature extraction

We are breaking down the feature extraction problem into two steps: feature construction and feature selection to choose the relevant ones. The raw data contains continuous data from 20 parameters and are detailed in Table 3, which we transform into the desired format using 14 distinct statistical methods, including mean, median, standard deviation, minimum, maximum, variance, peak-to-peak, skewness, kurtosis, 25th, 50th, 75th, 80th, and 95th percentiles [20]. The initial processing results in the creation of an extensive set of 280 features. Following this, we apply One-way Analysis of Variance (ANOVA) [21], utilizing the performance level variable and a statistical variable. One-way ANOVA is a statistical technique used to compare the means of three or more groups to determine if there are statistically significant differences between them. It's a hypothesis testing method that helps you assess whether the group means are equal or if at least one group significantly differs from the others. Based on this, we formulate the following hypotheses:

Null Hypothesis (H⊠): There is no significant difference in the statistical data among the different performance groups.

Alternative Hypothesis (H⊠): There is a significant difference in the statistical data among the different performance groups.

By considering a p-value less than 0.05 to reject the null hypothesis and retaining the features meeting this criterion, this step reduces the features to a remaining set of 224 from the initial 280 features.

We analyzed the box plot alongside the p-values to understand the differences between groups and evaluate their significance. The box plot, providing a visual representation of the data, greatly aids in interpreting the statistical results. Figure 6 showcases 10 features selected from a total of 280 features, illustrating the distribution of values for each feature class. This enables a direct comparison with the p-values presented in Table 4.

3.2.5 Linear discriminant analysis

Linear Discriminant Analysis (LDA) serves the purpose of identifying a linear combination of features that effectively differentiates classes within a dataset. This technique involves projecting data onto a lower-dimensional space, thereby enhancing the separation between these classes. This enhancement is achieved through the determination of linear discriminants that maximize the ratio of between-class variance to within-class variance. In essence, LDA identifies directions in the feature space that optimally distinguish various data classes. It assumes a gaussian distribution for the data and assumes equal covariance matrices for distinct classes. Furthermore, LDA expects the data to exhibit linear separability, implying that a linear decision boundary can proficiently classify the various classes.

3.2.6 Ridge classifier CV

Ridge Classifier CV (RCCV) is a linear classifier and an extension of ridge regression, tailored for classification tasks. It employs a 'one-vs-rest' strategy for multi-class scenarios and features 'built-in-cross-validation' to automatically optimize the regularization parameter during training. The process involves training distinct ridge classifiers on each fold, with cross-validation procedure executed by RCCV entails dividing the training data into multiple folds to determine the most suitable alpha value. The ultimate model is then trained on the complete training data using the chosen alpha. Subsequently, the label data is transformed into the range of (–1, 1), and a regression method is employed to address the problem. The highest prediction value is accepted as the target class. RCCV streamlines the hyperparameter tuning process by automating the selection of the optimal regularization parameter. This feature proves particularly beneficial when

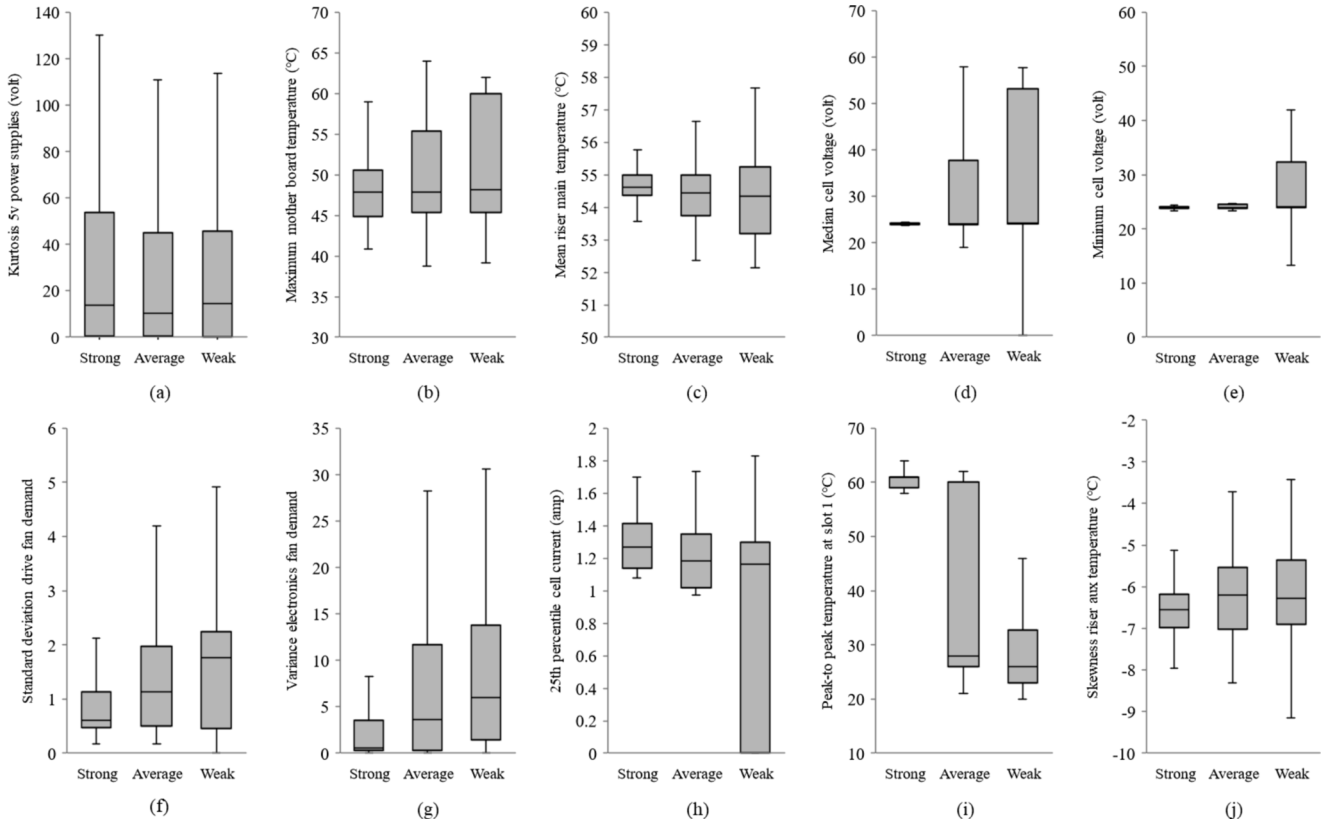

Fig. 6. Box plot for strong, average, and weak identification for 10 features.

Table 4. The p-value results of one-way ANOVA.

No.	Feature name	p-value
(a)	Kurtosis cell power suppliers (5v side)	0.178
(b)	Maximum motherboard temperature	0.099
(c)	Mean riser temperature at slot 1	0.003
(d)	Median cell voltage	0.384
(d)	Median cell voltage	0.384
(e)	Minimum cell voltage	0.429
(f)	Standard deviation drive fan demand	0.000
(g)	Variance electronics fan demand	0.000
(h)	25th percentile cell current	0.000
(i)	Peak-to-peak temperature at slot 1	0.443
(j)	Skewness riser temperature at slot 0	0.001

aiming to avert overfitting in a linear classification scenario, without the need for manual exploration of the ideal alpha value.

3.2.7 Extra-trees classifier

The scikit-learn Python machine learning package includes support for the Extra Trees Classifier (ETC). This classifier is available in the latest version of the library. ETC, an extension of the random forest algorithm and adds an additional level of randomness to the tree building process. It operates as an estimator by fitting randomized decision trees on diverse sub-samples of the dataset. The method of averaging is employed to enhance accuracy and manage data overfitting. ETC distinguishes itself from traditional decision trees in its construction process. Unlike the conventional approach of identifying the optimal split to separate node samples into two groups, ETC draws random splits for a predefined number of features (max features) that are selected randomly. Among these random splits, the one yielding the best result is chosen.

3.2.8 Random forest classifier

Random Forests Classifier (RFC) is an extension of decision trees, falling under the category of ensemble methods. Ensemble methods aim to achieve high accuracy by creating multiple classifiers and allowing each one to independently make predictions. When a classifier reaches a decision, the most common decision or an average decision from all classifiers can be utilized is known as voting. Each classifier in an ensemble specializes in a distinct perspective of the data and can be of various types. For example, you can combine classifiers like decision trees, logistic regression, and neural networks. Alternatively, classifiers may be of the same type but trained on different sections or subsets of the training data. An RFC comprises an ensemble of decision trees, where each tree is trained on a random subset of attributes, contributing to the collective decision-making process.

3.2.9 Extreme gradient boosting classifier

The Extreme Gradient Boosting (XGBoost) classifier is an algorithmic implementation of gradient boosted trees designed.

This methodology involves sequentially constructing trees, with each subsequent tree aiming to rectify errors made by the preceding one. This approach has demonstrated remarkable effectiveness that aims to predict a target variable by aggregating estimates from a collection of simpler and weaker models. XGBoost stands out due to its adeptness in handling diverse data types, relationships, distributions, robust predictive capabilities and operates almost ten times faster than alternative gradient boosting techniques. It is adept in solving a variety of problems, including regression, binary and multiclass classification, as well as ranking, which effectively counteract overfitting and enhance overall machine learning performance. The algorithm offers a plethora of finely adjustable hyperparameters, contributing to its success in machine learning competitions.

3.3 Hyperparameter tuning

The dataset is initially divided into 'features' and 'labels'. All models undergo evaluation using an 80-20% split, involving 699 datasets from 300 cells. If an algorithm demonstrates training accuracy that approaches 100% or lower when compared with the testing accuracy, it indicates an overfit model. To effectively prevent overfitting while learning and classifying, the approach outlined below is employed.

Algorithm 1. Overfitting prevention

```
Input: Dataset, Hyperparameter, Curr_test_acc = 0
Output: Trained model M
for i = 1, 2, 3, …, N do
  M ← get machine learning model
  accuracy(Train_acc, Test_acc) ← update score
  if Train_acc > Test_acc then
    if Train_acc < 100 then
      if Test_acc > Curr_test_acc then
        Curr_train_acc = Train_acc
        Curr_test_acc = Test_acc
      end if
    end if
  else
    goto for
  end if
return M
```

In model implementation, the precise selection and tuning of hyperparameters are crucial as they profoundly impact the model's behaviour and performance compared to using default values. In this study, trial and error [22] were utilized to choose effective hyperparameter values. For effective comparison of individual models, each model is implemented with meticulously tuned parameters. The specific hyperparameters values that were tuned, and the corresponding default values are detailed in Table 5.

3.4 Performance metrics

We employ a single performance evaluation metric for each supervised classification, which is the Confusion Matrix (CM). A CM is a table frequently utilized to illustrate the performance of a classification model on a set of test data with known actual values. It is formulated as shown in equation (2).

$$\mathrm{CM} = \begin{bmatrix} \mathrm{TP} & \mathrm{FP} \\ \mathrm{FN} & \mathrm{TN} \end{bmatrix}. \tag{2}$$

Here are the four quadrants in a confusion matrix:

True Positive (TP): an outcome where the model correctly predicts the positive class.

True Negative (TN): an outcome where the model correctly predicts the negative class.

False Positive (FP): an outcome where the model incorrectly predicts the positive class.

False Negative (FN): an outcome where the model incorrectly predicts the negative class.

Along with accuracy, we utilize performance metrics such as precision, recall, and f1-score to evaluate the model, which can be calculated by the equations (3), (4), and (5).

$$\text{precision} = \frac{|TP|}{|TP| + |FP|}, \tag{3}$$

$$\text{recall} = \frac{|TP|}{|TP| + |FN|}, \tag{4}$$

$$\begin{aligned} \text{f1} - \text{score} &= \frac{2|TP|}{2|TP| + |FP| + |FN|} \\ &= \frac{2 \times precision \times recall}{precision + recall} \end{aligned} \tag{5}$$

A Receiver Operating Characteristic (ROC) [23–25] is a graphical representation and evaluation metric used in the field of binary classification (a classification task with two possible outcomes, typically labeled as positive and negative). It depicts the True Positive Rate (TPR) against the False Positive Rate (FPR). TPR, also known as sensitivity, quantifies the ratio of detected anomalies to the total anomalies within the signal. The FPR signifies the rate of false alarms generated by the algorithm, calculated by the equations (6) and (7). The Area Under (AUC) the ROC Curve is a single scalar metric that summarizes the overall performance of a classification model. AUC-ROC values range from 0 to 1, with higher values indicating better performance. A value of 0.5 represents a random classifier, while a value of 1 indicates a perfect classifier. AUC-ROC is commonly used for model evaluation.

$$\text{TPR} = \frac{|TP|}{|TP| + |FN|}, \tag{6}$$

Table 5. Comparison of hyperparameters of various algorithm by default and tuned values.

Algorithm	Hyperparameter	Tuned value[a]	Default value[b]
LDA	n_components	0	None
	tol	0.0001	0.0001
	random_state[c]	815	1
RCCV	alphas	array([1.0])	array([0.1, 1.0, 10.0])
	cv	2	None
	normalize	'deprecated'	'deprecated'
	random_state[c]	416	1
ETC	criterion	'gini'	'gini'
	max_depth	11	None
	min_samples_leaf	1	1
	min_samples_split	2	2
	n_estimators	8	100
	random_state	42	None
	random_state[c]	11	1
RFC	criterion	'gini'	'gini'
	max_depth	16	None
	min_samples_leaf	1	1
	min_samples_split	2	2
	n_jobs	-1	None
	n_estimators	27	100
	random_state	11	None
	random_state[c]	755	1
XGBoost	max_depth	2	None
	n_estimators	45	100
	objective	'multi:softprob'	'binary:logistic'
	random_state	1	None
	random_state[c]	8	1

[a] The hyperparameters that resulted in the best accuracy scores.
[b] The default hyperparameters in the super learner package.
[c] The randomize parameter controls the random splitting of a dataset into training and testing sets.

$$\text{FPR} = \frac{|FP|}{|TN| + |FP|} \tag{7}$$

Accuracy, is calculated as follows:

$$\text{Accuracy} = \frac{|TP| + |TN|}{|TP| + |FP| + |FN| + |TN|} \tag{8}$$

The CM allows visualizing correctly classified and misclassified samples.Fig. 7. ROC response of each developed algorithm.

4 Experiments and results

Table 6 presents a comprehensive comparison of performance metrics across five different algorithmic structures utilizing the same dataset and software. Models employing default values exhibit lower performance metrics compared

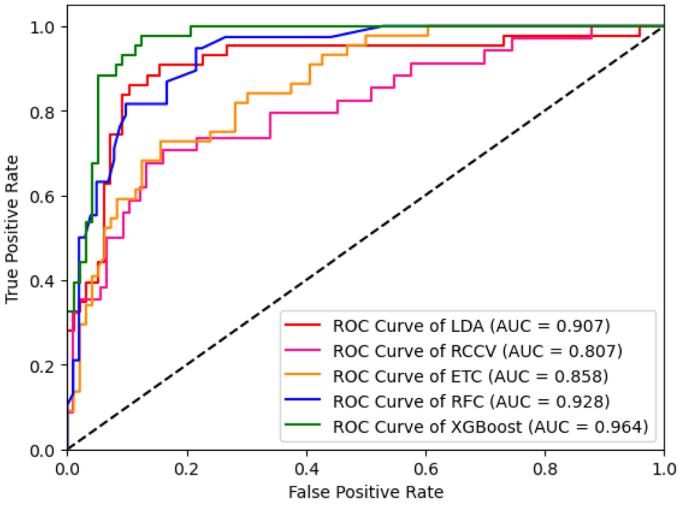

Fig. 7. ROC response of each developed algorithm.

Table 6. The proposed model has the best performance for all evaluation metrics.

	LDA		RCCV		ETC		RFC		XGBoost	
Metric	Tuned	Default	Tuned	Default	Tuned	Defaul	Tuned	Default	Tuned	Default
Precision	0.882	0.831	0.841	0.801	0.860	0.831	0.931	0.842	0.936	0.852
Recall	0.879	0.829	0.843	0.80	0.857	0.829	0.921	0.829	0.936	0.843
F1-score	0.880	0.829	0.838	0.799	0.856	0.829	0.923	0.830	0.936	0.843
Train ACC	0.927	0.946	0.884	0.895	0.986	0.10	0.998	0.10	0.993	0.10
Test ACC	0.879	0.829	0.843	0.80	0.857	0.829	0.921	0.829	0.936	0.843

to those with adjusted hyperparameters, and models without conditions to prevent overfitting. Notably, every model achieves a test accuracy of no more than 85%. RCCV has the lowest test accuracy at 80%, followed by LDA, ETC, and RFC at 82.9%, while XGBoost exhibits the highest accuracy at 84.3%. However, it's worth noting that ETC, RFC, and XGBoost, all three models, display overfitting with training accuracy equal to 100%, indicating a substantial gap between training and testing accuracy. Models with adjusted hyperparameters, designed to mitigate overfitting, consistently outperform default models across all metrics. RCCV exhibits the lowest accuracy at 84.3%, while XGBoost achieves the highest at 93.6%. LDA, ETC, and RFC demonstrate accuracies of 85.7%, 87.9%, and 92.1%, respectively.

Note.–ACC = Accuracy. 'Train accuracy' and 'Test accuracy' are commonly used metrics to evaluate the performance of a machine learning model. These metrics measure how well the model can make predictions on the training data (data used for training the model) and on unseen data (test data).

In addition to performance metrics, the AUC-ROC responses in Figure 7 are considered. RCCV produces the lowest AUC-ROC value of 0.807, while XGBoost excels with an AUC-ROC of 0.964. Other models consistently have AUC-ROC values higher than 0.85, aligning with the model order of test accuracy values. Consequently, the top two ranked models, RFC and XGBoost, are selected for testing on unseen data.

The CM of these effective classifiers are displayed in Figure 8, where performance levels 0 to 2 denote strong, average, and weak performance, respectively. RFC achieves a maximum correctly predicted recall or sensitivity of 91.1% for level 2 and a minimum of 71.9% for level 0, resulting in a model accuracy of 81% as shown in Figure 8a. In Figure 8b, XGBoost demonstrates a recall of 93.3% for level 2 but faces challenges in level 0 classification (with a minimum of 75%) due to overlap with level 1. This leads to XGBoost accuracy of 86.2%. Considering various performance indicators of all models, we will select the best-performing model for further analysis in the next step.

From this point onward, the XGBoost optimal model is employed to prune the feature set [26] via FI analysis [27, 28] to assess the contribution of the feature set consisting of 224 features is fed into XGBoost, achieving the highest prediction accuracy on unseen data at 86.2%. Considering the large number of features, Figure 9 displays FI scores greater than 0 for 92 features, providing a visual representation.

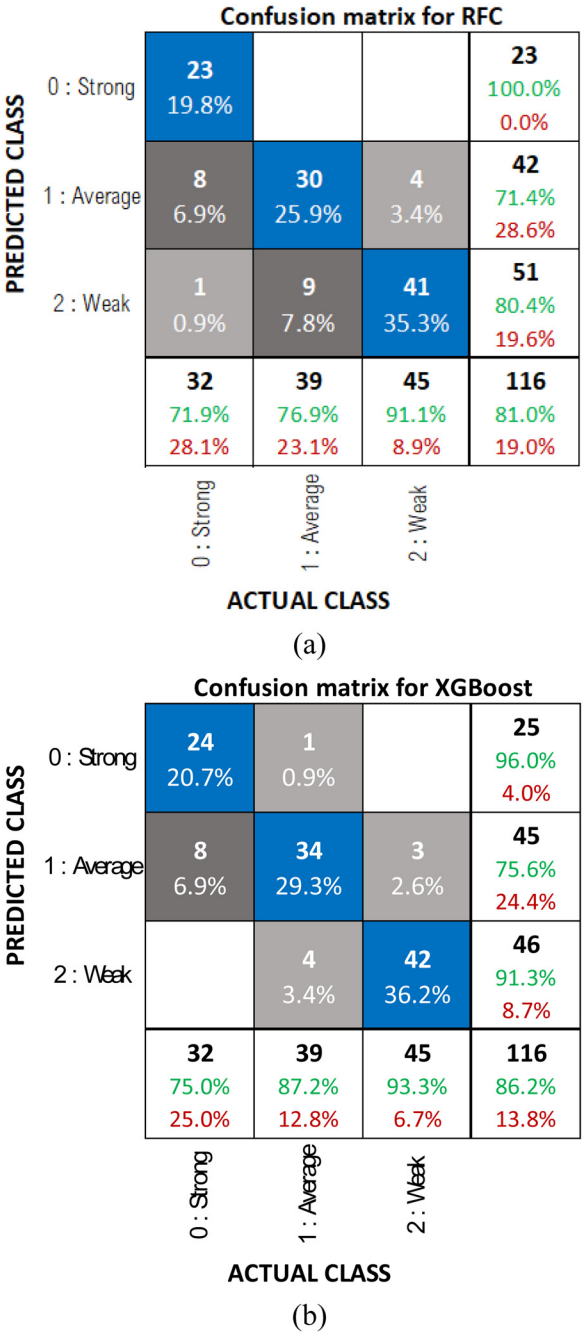

Fig. 8. Output CM of the scoring classifier using unseen data. (a) RFC algorithm, (b) XGBoost algorithm.

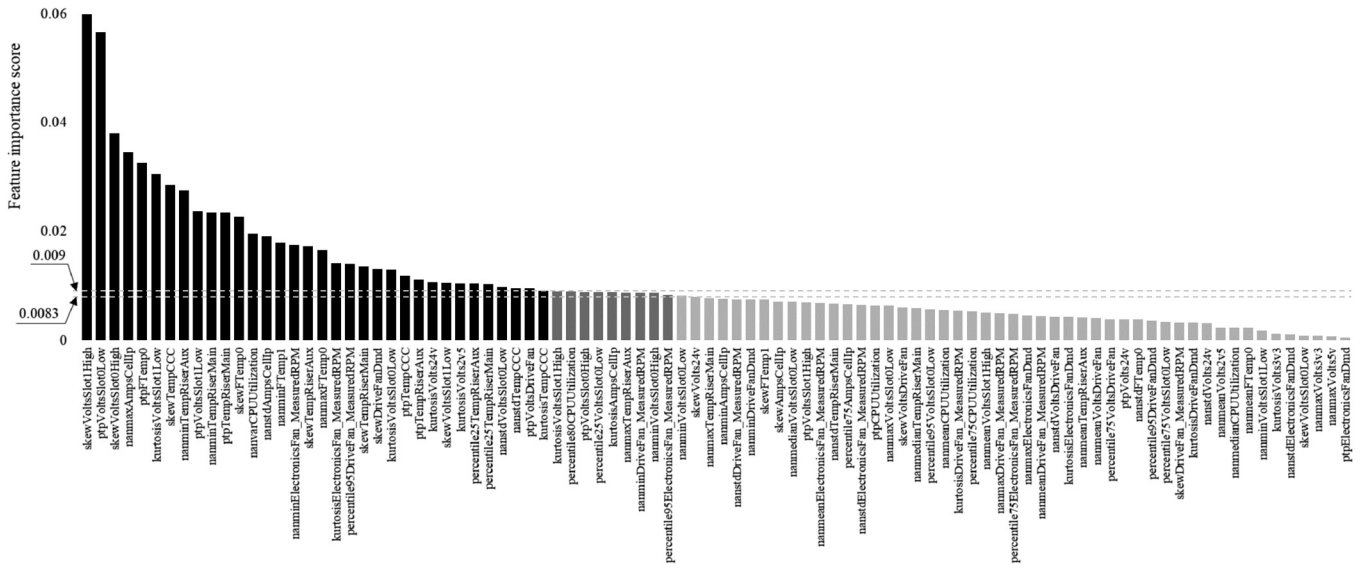

Fig. 9. Ranking of feature importance in 224 dimensions using XGBoost. (Top 92 features with importance score > 0).

Table 7. Performance metrics of XGBoost after feature importance reduction.

Ex. No.	No. of feature	FI-score	Precision	Recall	F1-score	Accuracy		
						Train	Test	Prediction
1	224	–	0.936	0.936	0.936	0.993	0.936	0.862
2	92	0	0.944	0.943	0.936	0.989	0.943	0.845
3	43	0.0083	0.929	0.929	0.936	0.982	0.929	0.871
4	34	0.009	0.926	0.921	0.936	0.971	0.921	0.879

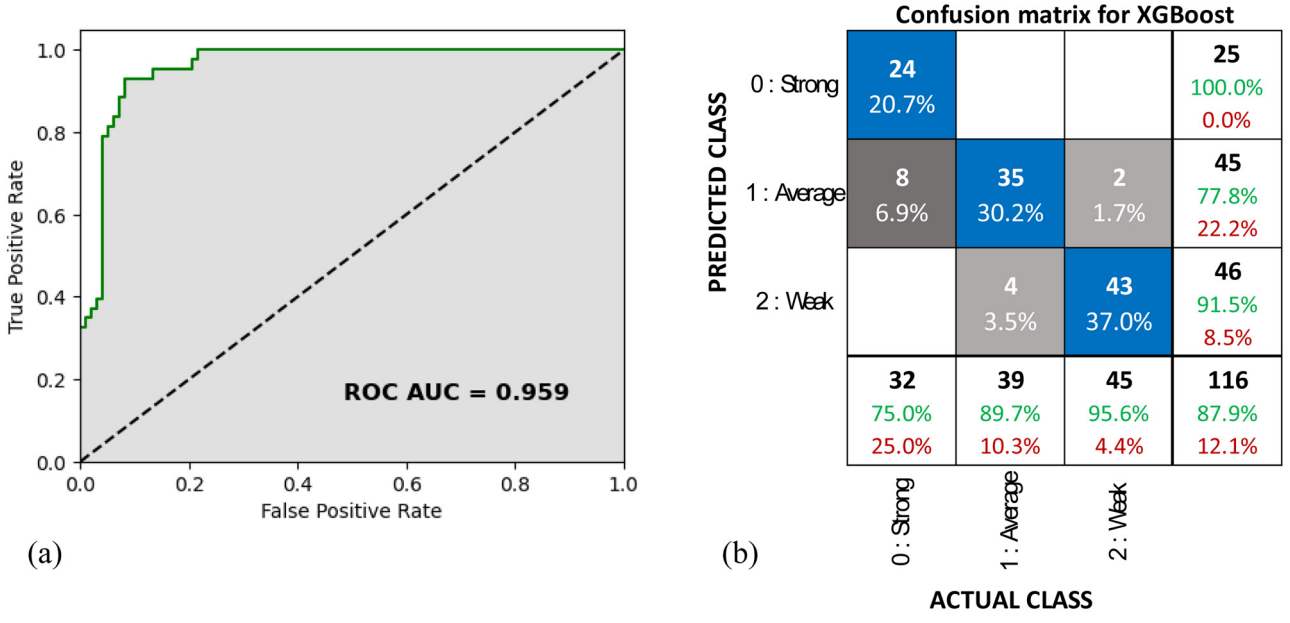

Fig. 10. XGBoost performance after reduction to 34 features. (a) ROC curve, (b) CM.

Table 8. Testing frequency per performance level for each product before shutdown.

Level	Level conditions	Number of tests (times)				
		Product A	Product B	Product C	Product D	Product E
0	$N \geq 6$	≥32	≥57	≥57	≥148	≥162
1	$N = 6$	11-31	20-56	20-56	50-147	55-161
2	$N = 2$	≤10	≤19	≤19	≤49	≤54

Following this, we proceed to further reduce features based on FI scores and present experimental results of performance metrics and prediction accuracy on unseen data. In addition to the 224 features (Example No. 1) shown in Table 7, Example No. 2, reduced to 92 features by setting FI = 0, sees a decrease in prediction accuracy from the original to 84.5%. In Example No. 3, reduced to 43 features with FI = 0.0083, the accuracy in prediction increases to 87.1%. Finally, in Example No. 4, with FI = 0.009 and 34 features, the prediction accuracy increases from the original to 87.9%. This demonstrates that Example No. 4 has a significant impact on the accuracy of the model. Through FI analysis, we can effectively reduce features while enhancing model accuracy by selecting appropriate FI scores.

Figure 10 illustrates the performance classification results by selecting the model with the best performance (Example No. 4). The model now provides recall accuracy of 95.6% for level 2, 89.7% for level 1, and 75% for level 0. However, there is still some confusion between level 1 and level 2. When compared to the original, recall scores for level 2 and level 1 increased, while those for level 0 still showed similar results, as shown in Figure 10a, although the AUC-ROC decreased to 0.959, slightly different from the original, as shown in Figure 10b.

Let's evaluate the predictive capabilities beyond just determining the performance level of the testing cell. Referring to the 'Fault types' section, we know that, on average, 0.59 HDDs are tested per week. Additionally, we have data on the number of hours each product has been tested. Utilizing this information, we can calculate how many test-cycles each performance level can undergo before the device is damaged.

When the prediction results for the testing cell indicate level 2, it suggests that the testing cell can be utilized for just under 3.5 weeks before being shut down due to a fault. At level 1, the testing cell can continue to function for an additional no more than 10.2 weeks, and at level 0, it can function for no less than 10.2 weeks before ceasing operation. A closer examination reveals that a testing cell at level 2 exclusively testing product A can conduct no more than 10 additional tests. Conversely, if the device exclusively tests product E, it can perform no more than 54 tests before facing potential damage. This is due to product A having an average testing time of 55.249 hours compared to product E of 10.566 hours, allowing the device to conduct more tests on product E than on product A. In conclusion, this indicates that the testing equipment can conduct more test cycles for product E than for product A before potential damage occurs. This description pertains solely to the results of products A and E. Detailed results regarding the number of tests for products B, C, and D for each performance level can be found in Table 8.

5 Conclusion

This article focuses on a robust data analysis approach that categorizes testing cell performance into three distinct classes: strong, average, and weak. This classification enables the prediction of outcomes based on this categorized performance. The process begins with a meticulous feature extraction phase, resulting in 224 relevant features derived from raw data collected under HVL fault conditions. These features are carefully selected through the application of 14 statistical methods on 20 parameters.

To determine the performance level, the number of HDDs tested over a 6-month period, weighted by product type, is utilized. The classification task is carried out using five distinct algorithms: LDA, RCCV, ETC, RFC, and XGBoost. Additionally, a comparison is made with an algorithm utilizing default values for hyperparameters. The results showcase XGBoost with tuned hyperparameters as the optimal performer, achieving an impressive prediction accuracy of 86.2% on previously unseen data. Subsequently, a refinement process focusing on feature importance leads to a reduced feature set of 34, significantly enhancing the accuracy to a noteworthy 87.9%.

When considering the Average HDDs per week in detail, the test cycle for each product model becomes apparent. This enables the prediction of malfunctions up to 3 weeks in advance, facilitating control over the daily maintenance workload. Moreover, it enhances maintenance readiness and efficiency by enabling advanced planning for spare parts orders. This proactive approach also leads to a reduction in the number of HDDs requiring retesting, as servicing testing cells before damage occurs allows tests to be completed seamlessly. Additionally, it minimizes the waiting time for HDDs to join the post-production testing queue. Ultimately, it contributes to a decrease in the number of regraded products resulting from errors in testing cells.

Looking forward, this research trajectory involves the practical application of the developed algorithm to predict device performance in real-time, especially within real-world applications. This aims to push the boundaries of predictive accuracy and enhance practical utility.

Funding

This research received support from Suranaree University of Technology, Thailand. The authors also extend their gratitude to the Program Management Unit for Human Resources & Institutional Development, Research, and Innovation (PMU-B), Thailand, as well as Western Digital Storage Technologies (Thailand) Ltd., for their generous support through a graduate scholarship.

Conflict of Interest

The authors declare no conflict of interest.

Data availability statement

Not applicable.

Author contribution statement

Conceptualization, M.L. and J.S.; methodology, M.R. and J.S.; software, M.R.; validation, M.R. and M.L.; formal analysis, M.R.; investigation, M.R. and J.S.; resources, M.R.; data curation, M.R. and M.L.; writing—original draft presentation, M.R. and J.S.; writing—review and editing, S.K. and J.S.; visualization, M.R.; supervision, S.K. and J.S.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

References

1. P. Chommuangpuck, T. Wanglomklang, S. Tantrairatn, J. Srisertpol, Fault tolerant control based on an observer on PI servo design for a high-speed automation machine, Machines **8** (2020) 22
2. P. Chommuangpuck, T. Wanglomklang, J. Srisertpol. Fault detection and diagnosis of linear bearing in auto core adhesion mounting machines based on condition monitoring, Syst. Sci. Control Eng. **9** (2021) 290–303
3. T. Wanglomklang, P. Chommaungpuck, K. Chomniprasart, J. Srisertpol. Using fault detection and classification techniques for machine breakdown reduction of the HGA process caused by the slider loss defect, Manufactur. Rev. **9** (2022) 21
4. C. Sapapporn, S. Seangsri, J. Srisertpol, Classifying and optimizing spiral seed self-servo writer parameters in manufacturing process using artificial intelligence techniques, Systems **11** (2023) 268
5. S.J. Wagh, M.S. Bhende, A.D. Thakare, Fundamentals of data science, 1st edn. Taylor & Francis Group, London, 2021, 296 pp
6. M.U. Malik, Python scikit-learn for beginners: Scikit-learn specialization for data scientist, 1st edn. AI Publishing, Michigan, 2021, 405 pp
7. A.D. Mauro, Data analytics made easy, 1st edn. Packt Publishing, UK, 2021, 406pp
8. G. Wang, L. Zhang, W. Xu, What can we learn from four years of data center hardware failures? in 2017 47th *Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, IEEE (2017)
9. A. Lawi, S.L. Wungo, S. Manjang, Identifying irregularity electricity usage of customer behaviors using logistic regression and linear discriminant analysis, in 2017 3rd *International Conference on Science in Information Technology (ICSITech)*, IEEE (2017)
10. G.B.G. Pereira, L.P. Fernandes, J.M.R., d.S. Neto, H.D., d.M. Braz, L. da Silva Sauer, A comparative study of linear discriminant analysis and an artificial neural network performances in breast cancer diagnosis, in *2020 IEEE Andean Conference*, IEEE (2020)
11. A. Singh, B.S. Prakash, K. Chandrasekaran, A comparison of linear discriminant analysis and ridge classifier on Twitter data, in *International Conference on Computing, Communication and Automation (ICCCA2016), IEEE* (2016)
12. R.K. Grace, M.I. Priyadharshini, Wind speed prediction using extra tree classifier, in *International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), IEEE* (2023)
13. B. Dhananjay, N.P. Venkatesh, A. Bhardwaj, J. Sivaraman, Cardiac signals classification based on extra trees model, in *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE (2021).
14. J. Shen, J. Wan, S.J. Lim, L. Yu, Random-forest-based failure prediction for hard disk drives, Int. J. Distrib. Sens. Netw. **14** (2018)
15. O. Alshboul, A. Shehadeh, G. Almasabha, A.S. Almuflih, Extreme gradient boosting-based machine learning approach for green building cost prediction, Sustainability **14** (2022) 6651
16. K. M, V.S. Akshaya, A XGBOOST Based algorithm for early prediction of human sepsis, in 2022 4th *International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE (2022)
17. M.E. Irawati, H. Zakaria, Classification model for Covid-19 detection through recording of cough using XGboost classifier algorithm, in *2021 International Symposium on Electronics and Smart Devices (ISESD)*, IEEE (2021)
18. Q. Li, H. Li, K. Zhang, Prediction of HDD failures by ensemble learning, in 2019 IEEE 10th *International Conference on Software Engineering and Service Science (ICSESS)*, IEEE (2020)
19. Z. Miller, O. Medaiyese, M. Ravi, A. Beatty, F. Lin, Hard disk drive failure analysis and prediction: an industry view, in 2023 53rd *Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S)*, IEEE (2023)
20. P. Bruce, A. Bruce, P. Gedeck, Practical statistics for data scientists, Published by O'Reilly Media, Inc. 2020
21. Z.H. Nasiruddin, W.M. Diyana W Zaki, S.A. Hudaibah, A.H. Nur Asyiqin, Automated retinal blood vessel feature extraction in digital fundus images, in *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, IEEE (2022)
22. I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, N. Doulamis, Multiclass confusion matrix reduction method and its application on net promoter score classification problem, Technologies **9** (2021) 81
23. T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. **27** 861–874 (2006) 861–874
24. S. Yang, G. Berdine, The receiver operating characteristic (ROC) curve, Southwest Respirat. Critical Care Chronicles **5** (2017) 34–36
25. L. Zhang and N. Hu, ROC analysis based condition indicator threshold optimization method, in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, IEEE (2017)
26. H. Wan, Q. Liu, Y. Ju, Utilize a few features to classify presynaptic and postsynaptic neurotoxins, Comput. Biol. Med. **152** (2023) 106380
27. P. Kumar, M. Sharma, Feature-importance feature-interactions (FIFI) graph: A graph-based novel visualization for interpretable machine learning, in *2021 International Conference on Intelligent Technologies (CONIT)*, IEEE (2021)
28. J. Yu, C. Xia, H. Zhang, Research on feature importance of gait mechanomyography signal based on random forest, in *2020 International Conference on Computer Vision*, Image and Deep Learning (CVIDL), IEEE (2020)

Cite this article as: Maneerat Rakcheep, Metinan Laosakun , Sorada Khaengkarn, Jiraphon. Srisertpol, Enhancing testing cell set efficiency: A machine learning approach on hard disk drive data, Manufacturing Rev. **11**, 11 (2024)