General Feature Selection for Failure Prediction in Large-scale SSD Deployment

Fan Xu[1], Shujie Han[2], Patrick P. C. Lee[2], Yi Liu[1], Cheng He[1], and Jiongzhou Liu[1]
[1]Alibaba Group [2]The Chinese University of Hong Kong

***Abstract*—Solid-state drive (SSD) failures are likely to cause system-level failures leading to downtime, enabling SSD failure prediction to be critical to large-scale SSD deployment. Existing SSD failure prediction studies are mostly based on customized SSDs with proprietary monitoring metrics, which are difficult to reproduce. To support general SSD failure prediction of different drive models and vendors, this paper proposes Wear-out-updating Ensemble Feature Ranking (WEFR) to select the SMART attributes as learning features in an automated and robust manner. WEFR combines different feature ranking results and automatically generates the final feature selection based on the complexity measures and the change point detection of wear-out degrees. We evaluate our approach using a dataset of nearly 500 K working SSDs at Alibaba. Our results show that the proposed approach is effective and outperforms related approaches. We have successfully applied the proposed approach to improve the reliability of cloud storage systems in production SSD-based data centers. We release our dataset for public use.**

I. Introduction

Maintaining storage reliability is critical for large-scale data centers. As solid-state drives (SSDs) have become the mainstream building blocks in large-scale data centers, the storage reliability of data centers depends on the reliability of SSDs. However, large-scale SSD deployment is often challenged by prevalent SSD failures. Traditional redundancy protection schemes, such as replication and RAID, are often adopted to tolerate prevalent SSD failures. To complement the traditional redundancy protection schemes, we explore the SSD failure prediction with *machine learning* techniques as a proactive fault tolerance mechanism.

In the literature, many prior studies [3], [6], [10], [15]–[17], [20], [34] have investigated the failure prediction for hard disk drives (HDDs) based on SMART (Self-Monitoring Analysis and Reporting Technology) logs. Nevertheless, due to the complicated characteristics of SSDs [11], [18], [19], [21], [24], [25], [33], only a few studies [1], [17], [23], [24] predict the failure prediction for SSDs based on the proprietary attributes at Google, which may be inapplicable for general production environments. Thus, it is essential to study general SSD failure prediction using SMART logs in large-scale production environments.

The prerequisite of general SSD failure prediction is *feature selection*, which aims to select the effective learning features. In particular, some learning features are weakly correlated to failures and bring noises into the failure prediction. Such uncorrelated learning features may decrease the prediction accuracy. Thus, we need to select the effective learning features and filter out uncorrelated ones.

However, there exist some challenges of feature selection for SSD failure prediction. First, heterogeneity [13] arises in modern data centers due to the deployment of heterogeneous SSDs in terms of drive models, capacity, reliability, etc, meaning that one feature selection approach may not be applicable for all drive models. Our evaluation also shows that no single feature selection approach can always select the best set of learning features for SSD failure prediction across all drive models (Section V-B). Second, various feature selection approaches may select different sets of learning features (see details in Section III-B). How to find an effective set of learning features for prediction is still an open issue. Third, most existing studies [1], [3], [17] select learning features based on statistical analysis, without considering the characteristics of SSDs.

This motivates us to explore a general feature selection approach for SSD failure prediction. In this paper, we propose *Wear-out-updating Ensemble Feature Ranking (WEFR)* as a general approach to select learning features for SSD failure prediction. WEFR combines various feature selection approaches to achieve robust feature selection. It adopts complexity measures to automate feature selection across various drive models. It also updates the selected features by considering the wear-out degree of SSDs. To summarize, this paper makes the following contributions:

- We motivate our work via a measurement study on a dataset of nearly 500 K SSDs from six drive models in SSD-based data centers at Alibaba. We find that the learning features have different correlations with SSD failures across different drive models, and various feature selection approaches select different sets of learning features. Also, the selected learning features vary with the wear-out degree for a drive model.
- We propose WEFR, which selects the SMART attributes as learning features in an automated and robust manner and updates the selected features with the wear-out degree. WEFR can be used for large-scale SSD failure prediction of different drive models and vendors.
- We conduct trace-driven evaluation on WEFR using our dataset of nearly 500 K SSDs at Alibaba. WEFR improves the F0.5-score by 10% and the precision by 22% compared to no feature selection. It also improves the F0.5-score by 4-6% and the precision by 10-14% compared to existing feature selection approaches, with comparable runtime performance.

Our dataset (including SMART logs and trouble tickets) is now made available at **https://github.com/alibaba-edu/dcbrain/tree/master/ssd_smart_logs**.

SMART attribute name	MA1	MA2	MB1	MB2	MC1	MC2
Raw Read Error Rate (RER)	✗	✗	✗	✗	✓	✓
Reallocated Sectors Count (RSC)	✓	✓	✓	✓	✓	✓
Power-On Hours (POH)	✓	✓	✓	✓	✓	✓
Power Cycle Count (PCC)	✓	✓	✓	✓	✓	✓
Program Fail Count (PFC)	✓	✓	✓	✓	✓	✓
Erase Fail Count (EFC)	✓	✓	✓	✓	✓	✓
Media Wearout Indicator (MWI)	✓	✓	✓	✓	✓	✓
Power Loss Protection Failure (PLP)	✓	✓	✗	✗	✗	✗
Unexpected Power Loss Count (UPL)	✓	✓	✗	✗	✓	✓
Available Reserved Space (ARS)	✓	✓	✓	✓	✓	✓
Downshift Error Count (DEC)	✗	✓	✓	✓	✓	✓
End-to-End error (ETE)	✓	✓	✓	✓	✓	✓
Reported Uncorrectable Errors (UCE)	✓	✓	✓	✓	✓	✓
Command Timeout (CMDT)	✓	✗	✗	✗	✓	✓
Enclosure Temperature (ET)	✓	✓	✓	✓	✓	✓
Airflow Temperature (AFT)	✓	✓	✓	✓	✓	✓
Reallocated Event Count (REC)	✓	✗	✗	✗	✓	✓
Current Pending Sector Count (PSC)	✓	✓	✓	✓	✓	✓
Offline Scan Uncorrectable Error (OCE)	✓	✗	✗	✗	✓	✓
UDMA CRC Error Count (CEC)	✓	✓	✓	✓	✓	✓
Total LBAs Written (TLW)	✗	✓	✓	✗	✗	✗
Total LBAs Read (TLR)	✗	✓	✓	✗	✗	✗

TABLE I: Overview of SMART attributes (✓ means an attribute is included in the drive model; ✗ means otherwise).

II. Background

We introduce our data collection methodology and review the failure prediction and feature selection approaches.

A. Data Collection

We collect data from five SSD-based data centers at Alibaba. The dataset covers a population of nearly 500 K SSDs of six drive models from three vendors over a two-year span from January 2018 to December 2019. We refer to the three vendors as MA, MB, and MC, and each vendor includes two drive models (denoted by a number after vendors). Our dataset includes two data types: SMART logs and trouble tickets.

SMART logs. SMART is a widely adopted tool for monitoring the statistics of disk drive status (called *attributes*). SMART attributes are vendor-specific. Each of them has both raw and normalized numerical values (denoted by "_R" and "_N" after the SMART attribute names, respectively). We collect SMART attributes for each SSD on a daily basis. Table I shows an overview of the collected SMART attributes for each drive model. The dataset spans 22 SMART attributes in total.

Trouble tickets. Our maintenance system deploys monitoring daemons on each server to perform rule-based detection periodically for checking abnormal behaviors and failures. Once detecting abnormal behaviors and failures, the maintenance system generates failure reports (called *trouble tickets*). Each trouble ticket records the drive ID and the timestamp of the failure occurrence. The dataset covers 7 K trouble tickets of SSD failures in total.

Summary of statistics. Table II shows the basic statistics in our dataset, including the flash technology, the percentage of SSDs in the whole SSD population, the percentage of SSD failures in the whole SSD failures, and the annualized failure rates (AFRs) [21]. We define the AFR as [11], [13], [21]: $AFR(\%) = \frac{f \times 365 \times 100}{n_1+n_2+...+n_{two-year}}$, where f is the total number of SSD failures reported in the trouble tickets and n_i is the number of operational SSDs on day i over the two-year span. The AFRs of TLC SSDs are higher than that of MLC SSDs.

Drive model	Flash technology	Total%	Failures%	AFR (%)
MA1	MLC	10.0%	20.9%	2.36%
MA2	MLC	25.7%	8.5%	0.46%
MB1	MLC	8.9%	15.7%	2.52%
MB2	MLC	10.4%	6.0%	0.71%
MC1	TLC	40.4%	37.8%	3.29%
MC2	TLC	4.6%	11.2%	3.92%

TABLE II: Summary of statistics. "Total%" represents the percentage of SSDs in the whole SSD population; "Failures%" represents the percentage of SSD failures in the whole SSD failures.

B. Failure Prediction

We formulate SSD failure prediction as an offline classification problem to predict whether an SSD will fail within a future period of time (e.g., within the next 30 days). We view raw and normalized values of each SMART attribute as two *learning features* and call a vector of learning features the *input variables*. We view the drive status as an indicator variable (called the *target variable*) (0 means healthy and 1 means failed). We regard the learning features and the drive status for an SSD on each day as a sample. We refer to the samples corresponding to the occurrences of failed SSDs and healthy SSDs as *positive samples* and *negative samples*, respectively. The workflow of offline failure prediction includes data preprocessing, feature selection, feature generation, training the prediction model, validating the prediction model, and prediction.

C. Feature Selection Approaches

We consider five state-of-the-art feature selection approaches for SSD failure prediction.

- *Pearson correlation* [14] measures the linear relationship between learning features and the target variable.
- *Spearman correlation* [29] measures the monotonic relationship between learning features and the target variable (not only linear relationships). It is used by the prior work [1] for SSD failure prediction.
- *J-index* [8] uses classification tasks to measure the ability of a learning feature to classify classes of the target variable correctly. It is used by the prior work [16] for predicting HDD failures.
- *Random forest* [4] provides feature importance evaluation, which measures the degree of reduction of classification accuracy after adding noises to a learning feature. A feature with a higher feature importance score has a greater impact on the HDD failure prediction accuracy [28]. Random forest is also used by the prior work [21] for predicting SSD failures.
- *XGBoost* [5] also provides feature importance evaluation, which measures the number of splits for training all boosted trees with a learning feature and the average gain of using the feature in trees.

III. Feature Importance Characterization

We measure the feature importance for SSD failure prediction (Section III-A), the feature importance using different feature

	MA1		MA2		MB1		MB2		MC1		MC2	
	Feature	Score	Feature	Score	Feature	Score	Feature	Score	Feature	Score	Feature	Score
Top 3	PLP_N	0.218	POH_R	0.104	ARS_N	0.326	REC_N	0.303	OCE_R	0.168	UCE_R	0.663
	PLP_R	0.217	PLP_R	0.092	RSC_N	0.246	POH_R	0.214	UCE_R	0.146	OCE_R	0.203
	MWI_N	0.185	TLR_R	0.078	DEC_R	0.167	UCE_N	0.201	CMDT_R	0.030	CMDT_R	0.021
Last 3	CMDT_N	0.001	PSC_N	0.001	CEC_N	0.001	EFC_R	0.001	ETE_N	0.001	ARS_N	0.001
	PSC_R	0.001	RSC_N	0.001	PFC_R	0.001	TLR_R	0.001	ARS_N	0.001	REC_R	0.001
	PSC_N	0.001	PSC_R	0.001	EFC_R	0.001	POH_N	0.001	ETE_R	0.001	CEC_R	0.001

TABLE III: Top and last three important learning features for predicting failures by ranking feature importance scores of Random Forest.

Rank	Pearson	Spearman	J-index	Random Forest	XGBoost
1	OCE_R	OCE_R	OCE_R	OCE_R	UCE_R
2	POH_R	UCE_R	ARS_R	UCE_R	OCE_R
3	ARS_R	RER_R	RER_R	CMDT_R	RSC_R
4	RSC_N	ARS_N	UCE_R	ARS_R	RER_R
5	ARS_N	RSC_R	OCE_N	OCE_R	OCE_N

TABLE IV: Rankings of the top five important features for MC1 with the five feature selection approaches.

selection approaches (Section III-B), and the feature importance for different wear-out degrees (Section III-C).

A. Feature Importance for SSD Failure Prediction

Recall in Section II-B that we regard the raw or normalized value of each SMART attribute as a learning feature. We refer to the effectiveness of learning features for predicting SSD failures as *feature importance*. We calculate the feature importance scores of *all* learning features using Random Forest feature importance evaluation [4] (Section II).

Table III shows the top and last three important learning features to predict SSD failures for each drive model. In addition to the important learning features, we observe that there also exist trivial ones (e.g., the feature importance scores of PSC_N and PSC_R are only 0.001 for MA1 and MA2). Such trivial learning features may decrease the SSD prediction accuracy as noises. Thus, feature selection is necessary for predicting SSD failures.

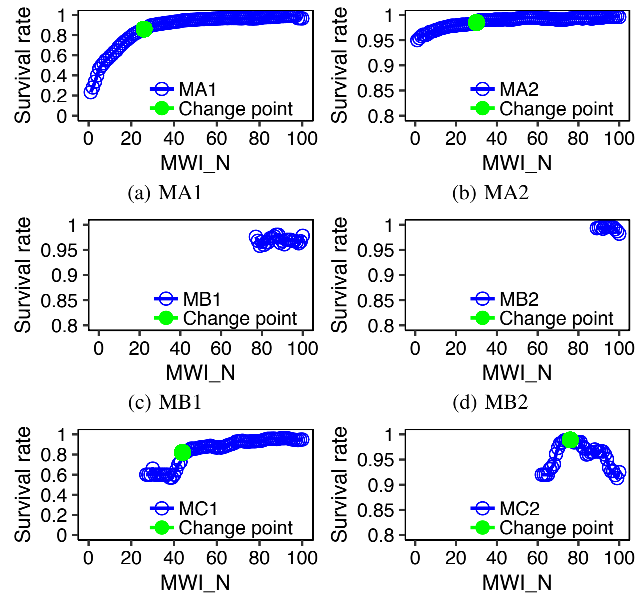
B. Feature Importance using Different Selection Approaches

We measure the feature importance with the five state-of-the-art feature selection approaches (Section II) to examine whether the selected features are different across feature selection approaches. We focus on the drive model MC1, which has the most numerous SSDs and failures.

Table IV shows that the rankings of the top five important learning features for MC1 vary across feature selection approaches. The finding also holds for the other drive models (not shown in the table). Such different rankings bring the following issues: (i) Which feature selection approach is more effective for SSD failure prediction? (ii) How many important features should we select? We will provide an automated and robust feature selection method to address the issues (Section IV).

C. Feature Importance with Different Wear-out Degrees

As read/write workloads are correlated with SSD failures via the media wear-out [21], we study the relationship between the wear-out degree and SSD failures. To quantify the wear-out degree, we use MWI_N, which indicates the percentage of the remaining erase cycles for an SSD; a lower MWI_N means a higher wear-out degree. To reflect the failure probability for different wear-out degrees, we define the *survival rate* on a value of MWI_N as the ratio between the number of survival SSDs (healthy SSDs) at the end of our dataset and the total number of all SSDs (including failed SSDs) on the value of MWI_N. We measure the survival rate on each value in the range of MWI_N.



(a) MA1 (b) MA2 (c) MB1 (d) MB2 (e) MC1 (f) MC2

Fig. 1: Relationships between the survival rate and MWI_N. Green points are change points detected using the Bayesian change point detection.

Figure 1 shows the relationships between the survival rate and MWI_N for the six drive models. We observe that the survival rate changes with MWI_N for MA1, MA2, MC1, and MC2, while it does not show an obvious trend for MB1 and MB2 due to a small range. Specifically, the survival rate decreases as MWI_N decreases for MA1, MA2, and MC1, while for MC2, with the decrease of MWI_N, the survival rate first increases to around 70 of MWI_N and then decreases due to some problems of firmware which are gradually fixed. It shows that SSD characteristics may change with the wear-out degree. Thus, the feature importance may also change with different values of MWI_N.

Before we justify whether the feature importance changes with different values of MWI_N, we first need to address how to detect the significant changes of the survival rate (i.e., changes of failure probability) impacted by MWI_N. We identify the changes of the survival rate using the Bayesian change point detection [7]. Specifically, we regard the survival

Drive model	MWI_N	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
MA1	Low	MWI_N	PLP_N	PLP_R	POH_R	REC_R
	High	PLP_N	PLP_R	REC_R	RSC_N	RSC_R
MA2	Low	POH_R	PLP_R	MWI_N	TLR_R	PLP_N
	High	PLP_R	POH_R	TLR_R	PLP_N	POH_R
MC1	Low	OCE_R	POH_R	UCE_R	MWI_N	CMDT_R
	High	OCE_R	UCE_R	CMDT_R	PCC_R	AFT_N
MC2	Low	UCE_R	OCE_R	POH_R	CMDT_R	MWI_N
	High	UCE_R	OCE_R	CMDT_R	RSC_R	AFT_N

TABLE V: Feature ranking results using Random Forest feature importance evaluation under different groups of MWI_N.

rates corresponding to each value in the range of MWI_N as a sequence, and compute the change probability (i.e., the posterior distribution of the sequence up to a survival rate given the sequence before the point) of each survival rate. Given the sequence of change probabilities, we use the z-score (i.e. the number of standard deviations from the mean of change probabilities), which we set as ± 2.5 of the standard deviation with the confidence level of 98.76%, to measure if the change is significant. If we detect multiple change points, we select the point with the most significant change.

We again use Figure 1 to show the change points of survival rates with MWI_N. We find that for MA1, MA2 and MC1, the survival rates of MWI_N have a significant change point between 20 and 45 of MWI_N, while for MC2, the change point is at 72 of MWI_N. For MB1 and MB2, there are no change points due to a small range of MWI_N.

We next divide SSDs into two groups of different values of MWI_N by the threshold of MWI_N corresponding to the most significant change point, and examine the feature importance with different groups of MWI_N. Table V shows that the top five important features among all learning features vary across the different groups of MWI_N for the four drive models (no change points for MB1 and MB2). MWI_N and POH_R are more important for the groups with low MWI_N values than those with high MWI_N values. This observation indicates that MWI_N has greater impact on predicting SSD failures with low MWI_N values. Also, we should update the learning features when predicting SSD failures with different MWI_N for drive models.

IV. Wear-out-updating Ensemble Feature Ranking

We propose Wear-out-updating Ensemble Feature Ranking (WEFR), a practical feature ranking method to select learning features among SMART attributes in an automated and robust manner, so as to generalize SSD failure prediction for different models and vendors. WEFR addresses the following challenges in feature selection for SSD failure prediction:

- **Robust feature selection.** Different feature selection approaches may select different learning features (Section III-B). Also, a feature selection approach is not always the optimal for various drive models. Thus, WEFR should combine feature selection approaches to select features in a robust manner.
- **Automated feature selection.** The effectiveness of SMART attributes as learning features varies, and hence the optimal number of selected features, vary across drive models. Thus, WEFR should determine the number of selected features automatically for different drive models.
- **Updating feature selection.** The selected features vary across SSDs with different wear-out degrees (Section III-C). Also, the wear-out degrees of SSDs increase with time. Thus, WEFR should select learning features for SSDs with different values of MWI_N and update the selected features over time.

Algorithm 1 Wear-out-updating Ensemble Feature Ranking

1: Input learning features of all SMART attributes
2: Initialize preliminary feature selection approaches
3: **for** Each preliminary feature selection approach **do**
4:     Calculate the ranking of learning features
5: **end for**
6: Remove the rankings with large deviations from others
7: Obtain the final rankings by the mean of rankings
8: Determine the feature count and select final learning features
9: Detect change points for survival rate of MWI_N
10: **if** Change point occurs for survival rate of MWI_N **then**
11:     Divide SSDs into high MWI_N and low MWI_N groups
12:     **for** Each MWI_N group **do**
13:         Repeat Lines 3-8 and update learning features
14:     **end for**
15: **end if**
16: Output learning features (for each MWI_N group if any)

A. Workflow Overview

Algorithm 1 shows the workflow of WEFR. Specifically, it takes the learning features of SMART attributes from the same drive model as inputs. It performs preliminary feature selection with common feature selection approaches and ranks learning features with feature importance. To prevent the bias of some methods (i.e., ineffective feature selection), it removes the rankings with large deviations from others and obtains the final rankings by the mean of rankings (Lines 1-7) (Section IV-B). It automatically determines the feature count based on the final rankings and selects the final learning features (Line 8) (Section IV-C). If it detects change points for the survival rate of MWI_N, it updates the selected features for the SSD groups with different groups of MWI_N (Lines 9-15) (Section IV-D). Finally, it outputs the learning features for all SSDs in the same drive model or for each group of MWI_N (Line 16).

B. Preliminary Feature Selection

Given all learning features of SMART attributes for a drive model, WEFR performs preliminary feature selection with common feature selection approaches. In our case, WEFR uses the five state-of-the-art feature selection approaches (Section II) to calculate the rankings of features. As the rankings of features for some feature selection approaches may substantially differ from other feature selection approaches, it is necessary to check and remove such outliers of rankings automatically.

WEFR first examines the similarity between the rankings from two feature selection approaches with the Kendall Tau rank distance [31]. Specifically, it defines the rankings of all learning features for two feature selection approaches A and B as R_A and R_B, respectively. For a pair of two distinct learning features i and j, it uses an indicator variable $\Theta_{i,j}(R_A, R_B)$ to indicate whether the orders of rankings of i and j are the same in R_A and R_B (i.e., the ranking of i is smaller/larger than that of j in both R_A and R_B) (0 means the same or 1 otherwise). It measures the Kendall Tau rank distance (denoted by D) between R_A and R_B by $D(R_A,R_B)=\sum_{i,j}\Theta_{i,j}(R_A,R_B)$ for all pairs of distinct learning features.

WEFR next examines the outliers of rankings for different feature selection approaches. Specifically, it computes D for all pairs of two feature selection approaches and calculates the mean of D (denoted by $\bar{D}$) for one feature selection approach between its rankings and those of the others. If $\bar{D}$ for a feature selection approach has a 1.96× standard deviation (i.e., with the 95% confidence level) from the mean of $\bar{D}$ over all feature selection approaches, it regards the rankings of the feature selection approach as outliers and discards its rankings. It finally takes the mean of the remaining rankings for each learning feature as the final rankings.

C. Automated Feature Selection

Recall that the trivial learning features may decrease the prediction accuracy for SSD failures (Section III-A). WEFR needs to determine the feature count automatically and select learning features based on the final rankings.

Before determining the feature count automatically, WEFR first measures the effectiveness of a learning feature to classify different classes (failed and healthy SSDs) by applying the ensemble of complexity measures [26]. Specifically, WEFR computes the complexity measures for each learning feature using three complexity measure approaches individually, including the maximum Fisher's discriminant ratio [32], volume of overlap region [12], and maximum feature efficiency [2] (denoted by F_1, F_2, and F_3, respectively). It defines the ensemble of the three complexity measure approaches as F, where $F=\frac{1/F_1+F_2+1/F_3}{3}$ [26]. It calculates the final complexity measure for each learning feature (denoted by e) by $e=\alpha\cdot F+(1-\alpha)\cdot\xi$, where α is a parameter in the interval $[0,1]$ (set $\alpha=0.75$ [26]) and ξ is the percentage of scanned learning features from top to bottom in the final rankings over all learning features (see details below).

WEFR then determines the feature count and selects the learning features by applying the approach in [27]. Specifically, it defines a partial cumulative complexity measure of the scanned learning features as $E_p \coloneqq E_p+e$ and a total cumulative complexity measure as $E \coloneqq E+E_p$. It scans each learning feature from top to bottom in the final rankings. It initializes E_p and E by calculating them iteratively over the top $\log_2(\text{\# all learning features})$ learning features (by default, we select these learning features due to their high rankings [27]). For each learning feature after the top $\log_2(\text{\# all learning features})$ ones, it first updates E_p and compares E_p with the current E. If $E_p<E$, it updates E and continues to repeat the process for the next learning feature; otherwise, it breaks the loop and outputs the number of learning features before the last scanned learning feature (i.e., the determined feature count denoted by n). Finally, it selects the top n learning features from the final rankings.

D. Updating Feature Selection

WEFR updates the selected features with changes of MWI_N. It detects the change points of the survival rate using the Bayesian change point detection [7] and divides SSDs into the different groups of MWI_N with the threshold of MWI_N (i.e., the change point of the survival rate corresponding to MWI_N) (Section III-C). It periodically checks the change points of MWI_N (one week in our case) and updates the selected features for each group of MWI_N with the above processes (Sections IV-B and IV-C).

V. Evaluation

We evaluate via trace-driven experiments the effectiveness of WEFR on both the prediction accuracy of SSD failures and system performance. We summarize our findings as follows:

- Overall, WEFR improves the prediction accuracy compared to no feature selection and the five state-of-the-art feature selection approaches across the six drive models. (Exp#1)
- WEFR automatically determines the optimal number of selected features for SSD failure prediction. (Exp#2)
- WEFR improves the prediction accuracy compared to without updating selected features for different groups of MWI_N. (Exp#3)
- The runtime of WEFR is comparable to those of the state-of-the-art feature selection approaches. (Exp#4)

A. Methodology

We divide the 24-month samples, including positive and negative samples, of our dataset (Section II) into the training and testing phases by time. Specifically, we divide the last three months into three non-overlapping testing phases, and use the months before each testing phase as the training phase (i.e., we use the first 21, 22, and 23 months as the training phases for the 22th, 23th, and 24th month as the testing phases, respectively). In the training phase, we set the ratio of training period length to validation period length as 8:2 (by day). In the training period, we select learning features and train the prediction model for each drive model, while in the validation period, we validate the effectiveness of the trained prediction model. We use Random Forest [4] as the prediction model, as also evaluated by prior studies [17], [21]. We set the number of trees in Random Forest as 100 and the maximum depth of trees as 13. In each testing phase, we predict the status of SSDs within the next 30 days on a daily basis.

In addition to the original features selected from the SMART attributes, we generate statistical features for each original feature including the maximum, minimum, mean, standard deviation, difference between the maximum and minimum, and weighted moving average within three-day and seven-day time windows. In total, each sample comprises 64 to 200 learning features, varied across the six drive models.

Metrics. We evaluate the prediction accuracy of WEFR with the following metrics:

- *Precision:* The fraction of correctly predicted failed SSDs over all (correctly and falsely) predicted failed SSDs.

Methods	MA1			MA2			MB1			MB2			MC1			MC2			All drive models		
	P	**R**	**F0.5**	**P**	**R**	**F0.5**	**P**	**R**	**F0.5**	**P**	**R**	**F0.5**	**P**	**R**	**F0.5**	**P**	**R**	**F0.5**	**P**	**R**	**F0.5**
No feature selection	50%	37%	47%	39%	32%	37%	58%	34%	51%	60%	32%	51%	37%	18%	30%	39%	19%	32%	49%	22%	39%
Pearson correlation	**63%**	37%	**55%**	**58%**	32%	**50%**	69%	34%	57%	71%	32%	57%	40%	18%	32%	43%	19%	34%	59%	22%	44%
Spearman correlation	56%	37%	51%	49%	32%	44%	**75%**	34%	**61%**	78%	32%	61%	46%	18%	35%	46%	19%	36%	61%	22%	45%
J-index	59%	37%	53%	52%	32%	46%	74%	34%	60%	77%	32%	60%	47%	18%	35%	42%	19%	34%	62%	22%	45%
Random Forest	58%	37%	52%	53%	32%	47%	65%	34%	55%	75%	32%	59%	**49%**	18%	**36%**	49%	19%	37%	57%	22%	43%
XGBoost	58%	37%	52%	57%	32%	49%	71%	34%	58%	84%	32%	63%	38%	18%	31%	41%	19%	33%	56%	22%	43%
WEFR	**63%**	37%	**55%**	57%	32%	49%	**75%**	34%	**61%**	**85%**	32%	**64%**	**49%**	18%	**36%**	**52%**	19%	**38%**	**71%**	22%	**49%**

TABLE VI: Exp#1 (Effectiveness of robust feature selection). "All drive models" represents the overall prediction accuracy of the six drive models.

- *Recall:* The fraction of correctly predicted failed SSDs over all actual failed SSDs.
- *F0.5-score:* $\frac{(1+0.5^2)\times\text{Precision}\times\text{Recall}}{0.5^2\times\text{Precision}+\text{Recall}}$.

From our practical experience, once an SSD is predicted as failed (regardless of correctly or falsely), administrators will decommission the SSD for further inspection. Thus, we use the F0.5-score, instead of the F1-score, to weigh the precision twice as important as the recall, since the cost of replacing a healthy SSD that is falsely predicted as a failure (i.e., low precision) is higher than that of missing a failed SSD that is falsely predicted as a healthy SSD (i.e., low recall). Also, we evaluate the prediction accuracy based on the first time when an SSD is predicted as failed.

B. Results

Exp#1 (Effectiveness of robust feature selection). We compare the prediction accuracy of WEFR with no feature selection (i.e., using all learning features) and the five state-of-the-art feature selection approaches (Section II-C). For the five state-of-the-art feature selection approaches, we tune the percentage of selected features linearly from 10% to 100% to find the highest prediction accuracy.

Table VI shows that WEFR improves the precision and F0.5-score, subject to a fixed recall, by 13% (8%), 18% (12%), 17% (10%), 25% (13%), 12% (6%), and 13% (6%) for MA1, MA2, MB1, MB2, MC1, and MC2, respectively, compared to no feature selection. Overall, WEFR improves the precision (F0.5-score) by 22% (10%) compared to no feature selection for all drive models, confirming the importance of feature selection for SSD failure prediction (Section III-A).

Table VI also shows that WEFR generally improves the precision and F0.5-score for the six drive models compared to the five state-of-the-art feature selection approaches. Specifically, with a fixed recall, WEFR improves the F0.5-score by 4-6% and the precision by 10-14% compared to the five feature selection approaches. WEFR outperforms the other feature selection approaches for MA1, MB1, MB2, MC1, and MC2 from three different vendors. For MA2, the difference of F0.5-score between WEFR and the best result (50% of F0.5-score from the Pearson correlation) is only 1%. The reason of the robust performance of WEFR is that WEFR avoids the bias of a single feature selection approach by combining different feature selection approaches.

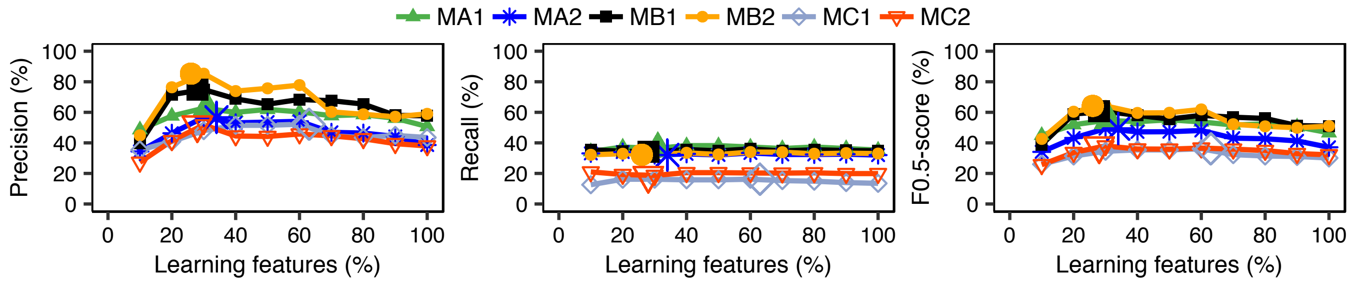
We observe that the five feature selection approaches cannot always perform the best for SSD failure prediction of different drive models and vendors. For example, the Spearman correlation achieves the highest F0.5-score (61%) for MB1. However, for MA1, the F0.5-score of the Spearman correlation is lower than that of the other approaches. Also, we observe that Random Forest has a close prediction accuracy to WEFR for MC1 and MC2, but a lower F0.5-score than WEFR for MB1 or MB2 by at least 5%. Thus, it is critical to combine various feature selection approaches to achieve robust feature selection for heterogeneous drive models.

Exp#2 (Effectiveness of automated feature selection). We evaluate the effectiveness of automated feature selection in WEFR and compare it with using a fixed percentage of selected features (varied linearly increasing from 10% to 100%).

Figure 2 shows that the F0.5-score of WEFR is always higher than or equal to the highest F0.5-score when fixing the percentage of selected features for the six drive models. Specifically, the percentages of selected features automatically determined by WEFR are 31%, 34%, 28%, 26%, 63%, and 28% for MA1, MA2, MB1, MB2, MC1, and MC2, respectively, which are close to the percentages of selected features corresponding to the highest F0.5-score when fixing the percentage of selected features. Note that using automated feature selection is also more flexible than tuning for the appropriate percentage of selected features in production.

Exp#3 (Effectiveness of updating feature selection). To evaluate the effectiveness of the wear-out-updating component in WEFR, we compare WEFR with and without updating feature selection for SSD failure prediction on MA1, MA2, MC1, and MC2 (recall that there is no change point of the survival rate for MWI_N for MB1 and MB2 (Section III-C)). Note that WEFR without updating feature selection (i.e., WEFR (No update)) means skipping Lines 10-15 in Algorithm 1.

Table VII shows that WEFR improves the precision and F0.5-score with updating selected features for MA1, MA2, MC1, and MC2 compared to no updating feature selection. Specifically, WEFR improves the precision (F0.5-score) by 6% (4%), 4% (2%), 5% (2%), and 6% (2%) in MA1, MA2, MC1 and MC2, respectively, compared to WEFR (No update). For SSDs with low MWI_N, WEFR improves the precision (F0.5-score) by 13% (9%), 12% (8%), 13% (6%), and 13% (6%) for MA1, MA2, MC1 and MC2, respectively, compared to WEFR (No update). It confirms that feature importances vary with MWI_N and it is necessary to update the selected features with MWI_N. The differences of F0.5-score between WEFR and WEFR (No update) for high MWI_N are insignificant and within 1% (not shown in Table VII).



(a) Precision (b) Recall (c) F0.5-score

Fig. 2: Exp#2 (Effectiveness of automated feature selection). Larger isolated points represent WEFR; smaller points on lines represent WEFR using a fixed percentage of selected features.

Model	Metrics	WEFR (No update)		WEFR	
		All	Low	All	Low
MA1	Precision (%)	57%	48%	63%	61%
	Recall (%)	37%	37%	37%	37%
	F0.5-score (%)	51%	45%	55%	54%
MA2	Precision (%)	53%	43%	57%	55%
	Recall (%)	32%	32%	32%	32%
	F0.5-score (%)	47%	40%	49%	48%
MC1	Precision (%)	44%	36%	49%	49%
	Recall (%)	18%	18%	18%	18%
	F0.5-score (%)	34%	30%	36%	36%
MC2	Precision (%)	46%	37%	52%	50%
	Recall (%)	19%	19%	19%	19%
	F0.5-score (%)	36%	31%	38%	37%

TABLE VII: Exp#3 (Effectiveness of updating feature selection). "WEFR (No update)" represents WEFR without updating selected features with MWI_N; "All" represents all SSDs and "Low" represents SSDs with low MWI_N only.

Exp#4 (Runtime of the state-of-the-art feature selection approaches and WEFR). We evaluate the runtime performance of the state-of-the-art feature selection approaches and WEFR using MC1 (i.e., the drive model with the largest population) on a local server. The local server has two 2.6 GHz eight-core Intel(R) Xeon(R) E5-2650 CPUs, 256 GB RAM, and LSI RAID of HDDs with 8 TB in total. For Random Forest and XGBoost, we use 100 trees and enable multi-threading with 32 threads; for WEFR, we execute the state-of-the-art feature selection approaches in parallel.

Table VIII shows the average runtimes of the state-of-the-art feature selection approaches and WEFR with MC1 over 20 rounds. We observe that the Spearman correlation is the slowest approach among the state-of-the-art approaches and takes 20.4 s to select features. As WEFR executes the state-of-the-art feature selection approaches in parallel, the runtime of WEFR is close to that of the slowest one (i.e., the Spearman correlation). Thus, the runtime of WEFR is comparable with those of the state-of-the-art feature selection approaches.

Methods	Pearson	Spearman	J-index	Random Forest	XGBoost	WEFR
Runtime	4.2 s	20.4 s	8.4 s	3.1 s	5.2 s	22.9 s

TABLE VIII: Exp#4 (Runtime of the state-of-the-art feature selection approaches and WEFR).

VI. Related Work

Our work is mainly related to two lines of studies, feature selection and failure prediction. For feature selection, Botezatu *et al.* [3] select features based on statistical measures for HDD failure prediction. Gaber *et al.* [9] adopt machine learning methods to extract SMART attributes to predict HDD failures. Narayanan *et al.* [21] apply permutation feature ranking to identify important features based on the prediction accuracy. Xu *et al.* [34] prune non-predictive features based on SMART attributes and system-level features for HDDs. Lu *et al.* [16] apply J-index [8] to select learning features among SMART attributes, locations, and workloads for HDD failure prediction.

Many studies focus on HDD failure prediction [3], [6], [10], [15], [22], [30], while limited studies focus on SSD failure prediction [1], [17], [21], [23]. For example, Mahdisoltani *et al.* [17] and Alter *et al.* [1] predict sector errors and SSD failures, respectively, based on customized SSDs at Google. Sarkar *et al.* [23] predict SSD failures based on the learning features from firmware functions. Narayanan *et al.* [21] study the important features in SMART logs for predicting SSD failures using Random Forest. The differences between our work and the prior studies are as follows. First, we focus on robust feature selection for generalizing SSD failure prediction on various drive models. Second, we determine the number of selected features automatically across the drive models. Third, we consider SSD characteristics (i.e., wear-out degree) to update the selected features for the groups with different wear-out degrees.

VII. Conclusion

In this paper, we propose Wear-out-updating Ensemble Feature Ranking (WEFR), a general practical feature selection method to select learning features from SMART attributes in an automated and robust manner, for predicting SSD failures across different drive models and vendors. WEFR combines different feature selection approaches for robust feature selection, determines the number of selected features automatically, and updates the selected features with the changes of wear-out degrees of SSDs. Our evaluation with the real-world dataset of nearly 500 K SSDs in production shows that WEFR generally improves the prediction accuracy across six drive models. Our dataset is released for public validation.

Acknowledgments. We thank our shepherd, Devesh Tiwari, and the anonymous reviewers for their comments. This work was supported by Alibaba Group via the Alibaba Innovation Research (AIR) program. The corresponding author is Shujie Han (sjhan@cse.cuhk.edu.hk).

References

[1] J. Alter, J. Xue, A. Dimnaku, and E. Smirni. SSD failures in the field: symptoms, causes, and prediction models. In *Proc. of IEEE SC*, 2019.

[2] G. H. Ball and D. J. Hall. Some implications of interactive graphic computer systems for data analysis and statistics. *Technometrics*, 12(1):17–31, 1970.

[3] M. M. Botezatu, I. Giurgiu, J. Bogojeska, and D. Wiesmann. Predicting disk replacement towards reliable data centers. In *Proc. of ACM SIGKDD*, 2016.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proc. of ACM SIGKDD*, 2016.

[6] F. D. dos Santos Lima, G. M. R. Amaral, L. G. de Moura Leite, J. P. P. Gomes, and J. de Castro Machado. Predicting failures in hard drives with LSTM networks. In *Proc. of IEEE BRACIS*, 2017.

[7] P. Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Stats and Computing*, 16(2):203–213, 2006.

[8] R. Fluss, D. Faraggi, and B. Reiser. Estimation of the youden index and its associated cutoff point. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 47(4), 2005.

[9] S. Gaber, O. Ben-Harush, and A. Savir. Predicting HDD failures from compound SMART attributes. In *Proc. of ACM SYSTOR*, 2017.

[10] S. Han, P. P. Lee, Z. Shen, C. He, Y. Liu, and T. Huang. Toward adaptive disk failure prediction via stream mining. In *Proc. of IEEE ICDCS*, 2020.

[11] S. Han, P. P. C. Lee, F. Xu, Y. Liu, C. He, and J. Liu. An in-depth study of correlated failures in production SSD-based data centers. In *Proc. of USENIX FAST*, 2021.

[12] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(3):289–300, 2002.

[13] S. Kadekodi, K. Rashmi, and G. R. Ganger. Cluster storage systems gotta have heart: improving storage efficiency by exploiting disk-reliability heterogeneity. In *Proc. of USENIX FAST*, 2019.

[14] P. Karl. Notes on the history of correlation. *Biometrika*, 13(1):25–45, 1920.

[15] J. Li, X. Ji, Y. Jia, B. Zhu, G. Wang, Z. Li, and X. Liu. Hard drive failure prediction using classification and regression trees. In *Proc. of IEEE/IFIP DSN*, 2014.

[16] S. Lu, B. Luo, T. Patel, Y. Yao, D. Tiwari, and W. Shi. Making disk failure predictions SMARTer! In *Proc. of USENIX FAST*, 2020.

[17] F. Mahdisoltani, I. Stefanovici, and B. Schroeder. Proactive error prediction to improve storage system reliability. In *Proc. of USENIX ATC*, 2017.

[18] J. Meza. Large scale studies of memory, storage, and network failures in a modern data center. *arXiv preprint arXiv:1901.03401*, 2019.

[19] N. R. Mielke, R. E. Frickey, I. Kalastirsky, M. Quan, D. Ustinov, and V. J. Vasudevan. Reliability of solid-state drives based on NAND flash memory. *Proceedings of the IEEE*, 105(9):1725–1750, 2017.

[20] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado. Machine learning methods for predicting failures in hard drives: A multiple-instance application. *Journal of Machine Learning Research*, 6(May):783–816, 2005.

[21] I. Narayanan, D. Wang, M. Jeon, B. Sharma, L. Caulfield, A. Sivasubramaniam, B. Cutler, J. Liu, B. Khessib, and K. Vaid. SSD failures in datacenters: What? when? and why? In *Proc. of ACM SYSTOR*, 2016.

[22] E. Pinheiro, W.-D. Weber, and L. A. Barroso. Failure trends in a large disk drive population. In *Proc. of USENIX FAST*, 2007.

[23] J. Sarkar, C. Peterson, and A. Sanayei. Machine-learned assessment and prediction of robust solid state storage system reliability physics. In *Proc. of IEEE IRPS*, 2018.

[24] B. Schroeder, R. Lagisetty, and A. Merchant. Flash reliability in production: The expected and the unexpected. In *Proc. of USENIX FAST*, 2016.

[25] B. Schroeder, A. Merchant, and R. Lagisetty. Reliability of NAND-based SSDs: What field studies tell us. *Proceedings of the IEEE*, 105(9):1751–1769, 2017.

[26] B. Seijo-Pardo, V. Boln-Canedo, and A. Alonso-Betanzos. On developing an automatic threshold applied to feature selection ensembles. *Information Fusion*, 45:227 – 245, 2019.

[27] B. Seijo-Pardo, V. Bolón-Canedo, and A. Alonso-Betanzos. Using data complexity measures for thresholding in feature selection rankers. In *Proc. of CAEPIA*, 2016.

[28] J. Shen, J. Wan, S. J. Lim, and L. Yu. Random-forest-based failure prediction for hard disk drives. *International Journal of Distributed Sensor Networks*, 14(11), 2018.

[29] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 100(3/4):441–471, 1987.

[30] X. Sun, K. Chakrabarty, R. Huang, Y. Chen, B. Zhao, H. Cao, Y. Han, X. Liang, and L. Jiang. System-level hardware failure prediction using deep learning. In *Proc. of ACM DAC*, 2019.

[31] E. M. Voorhees. Evaluation by highly relevant documents. In *Proc. of ACM SIGIR*, 2001.

[32] A. R. Webb. *Statistical Pattern Recognition, 2nd Edition*. John Wiley & Sons, 2003.

[33] E. Xu, M. Zheng, F. Qin, Y. Xu, and J. Wu. Lessons and actions: What we learned from 10K SSD-related storage system failures. In *Proc. of USENIX ATC*, 2019.

[34] Y. Xu, K. Sui, R. Yao, H. Zhang, Q. Lin, Y. Dang, P. Li, K. Jiang, W. Zhang, and J.-G. Lou. Improving service availability of cloud systems by predicting disk error. In *Proc. of USENIX ATC*, 2018.