Multidimensional Features Helping Predict Failures in Production SSD-Based Consumer Storage Systems

Xinyan Zhang[†], Zhipeng Tan[*†], Dan Feng[*†], Qiang He[*†], Wan Ju[†], Jiang Hao[‡], Ji Zhang[‡], Lihua Yang[†], Wenjie Qi[†]
[†]Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology
[‡]Huawei Technologies, China
Corresponding Email: (tanzhipeng, dfeng, qianghe)@hust.edu.cn*

***Abstract*—As SSD failures seriously lead to data loss and service interruption, proactive failure prediction is often used to improve system availability. However, the unidimensional *SMART*-based prediction models hardly predict all drive failures. Some other features applied in data centers and enterprise storage systems are not readily available in consumer storage systems (*CSS*). To further analyze related failures in production SSD-based *CSS*, we study nearly 2.3 million SSDs from 12 drive models based on a dataset of SMART logs, trouble tickets, and error logs. We discover that *SMART*, *FirmwareVersion*, *WindowsEvent*, and *BlueScreenofDeath* (*SFWB*) are closely related to SSD failures. We further propose a multidimensional-based failure prediction approach (*MFPA*), which is portable in algorithms, SSD vendors, and PC manufacturers. Experiments on the datasets show that *SFWB*-based *MFPA* achieves a high true positive rate (98.18%) and low false positive rate (0.56%), which is 4% higher and 86% lower than the SMART-based model. It is robust and can continuously predict for 2-3 months without iteration, substantially improving the system availability.**

***Index Terms*—SSD, multidimensional features, failure prediction, machine learning, system availability**

I. Introduction

SSDs have been the preferred choice for users as they offer better performance and durability than HDDs and are more resistant to shock and vibration. However, failures of system components may cause data loss, increase data recovery costs, and affect services and system availability. Failures of the storage drive are reported as the major component failures [1]–[3]. The reported downtime cost for 63 data centers increased from $5,617/min in 2010 to $8,851/min in 2016 [4].

Therefore, passive fault tolerance mechanisms including replication, Erasure Codes (EC), and Redundant Arrays of Independent Disks (RAID) have been proposed to reduce the impact. But the system availability is still severely reduced as these measures are taken after failures. By contrast, proactive fault tolerance mechanisms can anticipate failures and migrate data and services out of the unhealthy storage drives, which can reduce downtime costs and significantly improve system availability. There have been many studies [5]–[12] on proactive fault tolerance mechanisms based on HDDs for their comparatively long history.

However, as a novel storage drive, the proactive fault tolerance mechanism of NAND flash-based SSDs has received relatively less attention. Some studies on SSD reliability focus on specific error analysis of simulated workload in a controlled laboratory environment [13], [14]. Some focus on error types and their relationship with workloads, drive age, and wear-out [12], [15]–[18]. Several [19]–[22] develop several failure prediction models in data centers. But these unidimensional features-based models with high false alarm rate are hard to predict all drive failures [11].

Besides, the current research on SSD mainly focuses on the data center or enterprise-grade SSDs, and there is little research on consumer-grade SSDs. The data of consumer storage systems (*CSS*, e.g., Office/Desktop Computers, Laptops) are as important as data of enterprise storage systems. Enterprise-grade and consumer-grade SSDs differ significantly in multiple dimensions, including flash, P/E cycle, performance, and usage scenarios. However, for cost concerns, the current *CSS* does not have passive fault-tolerant mechanisms widely deployed in data centers or enterprise storage systems, let alone proactive fault tolerance mechanisms such as failure prediction. *CSS* only provides failure detection technologies based on SMART (self-monitoring, analysis, and reporting technology) thresholds, which increases the risk of data loss for individual users. A large percentage of users rarely think about backing up their data. The backup data is not real-time, which is hard to ensure that the data is up-to-date. Some data is also not suitable for backup to the cloud. Once an SSD fails, data recovery is difficult and costly (even several times the price of the SSD).

Furthermore, the failure prediction model used for HDD and enterprise-grade SSD in data centers and enterprise storage systems is not fully applicable to consumer-grade SSD. The differences exist not only between HDDs and SSDs but also between enterprise-grade and consumer-grade SSDs. The prediction model of HDD/enterprise-grade SSD cannot be used to predict consumer-grade SSD failures. SMART attributes are widely utilized to predict HDD failures [6], [7], [9], [10]. Some others predicted HDD failures by combining SMART with I/O, performance, and location attributes [8], [11]. Jacob et al. [19] develop models for SSDs in data centers only based on error logs. But these features other than SMART are not readily available to *CSS* as users are relatively scattered and vary in usage time and habits. Other dimensional compelling features should be explored to improve the model prediction effect.

In this paper, we study nearly 2.3 million SSDs of 12 drive models spanning nearly two years. An in-depth study of trouble tickets shows that multidimensional features ***SFWB***, namely ***S****MART,* ***F****irmwareVersion,* ***W****indowsEvent, and* ***B****lueScreenofDeath*, are intrinsically related to SSD failures. We further propose a **M**ultidimensional-based **F**ailure **P**rediction **A**pproach (***MFPA***) to predict SSD failures.

To conclude, we make the following contributions:

978-3-9819263-7-8/DATE23/© 2023 EDAA

- To the best of our knowledge, we are the first to study consumer-grade SSD failures in the actual *CSS* and find that SSD failures can be manifested as drive-level and system-level failures.
- We discover that new multidimensional features *SFWB* are associated with SSD failures. The drive-level features including *SMART* and *Firmware Version* and the system-level features including *WindowsEvent* and *BlueScreenofDeath* can indicate SSD errors.
- Based on *SFWB*, we further develop *MFPA* — an effective SSD failure prediction method. Through the optimization for discontinuous data, identification of failure timestamp, time-series-based model optimization, and validation of multiple algorithms, *MFPA* achieves 98.18% true positive rate (*TPR*) and 0.56% false positive rate (*FPR*).
- Experiment results prove that *SFWB*-based *MFPA* is robust and insensitive to machine learning (ML) algorithms, manufacturers and time periods.

II. Background and Motivation

Features Used for Failure Prediction in data centers: SMART is designed to monitor drives' health without interruption [5]. Current disk failure prediction is mainly based on *SMART* attributes [6], [7], [9], [10], [20], [23], [24]. Beyond the SMART attributes, Researchers utilize other features (error/performance/location logs, etc.) to build the failure prediction model [8], [11], [19].

Methods for Failure Prediction: Almost all disk vendors use the original threshold-based algorithms [5] to trigger a failure alarm when a single *SMART* attribute exceeds the threshold value. However, the *TPR* is only 3%-10%, and *FPR* is 0.1% in such cases. Statistical Methods can improve failure detection accuracy, mainly including parametric and non-parametric models. However, the *TPR* only increases to 56%-70%, and *FPR* decreases to nearly 1% [10]. To improve prediction capability, ML algorithms (such as Bayes, SVM, RF, LSTM, CNN_LSTM, etc.) are widely used for the failure prediction of storage drives [7]–[11], [23], [25], [26].

Research on SSD Availability: Fewer studies have been done on SSD failure prediction, especially for SSD in *CSS*. Most existing studies [12]–[17] have focused on the statistical analysis of the availability of SSDs deployed in enterprise storage systems and large data centers, while related research on SSDs in *CSS* lacks. Some studies [19]–[22] build several failure prediction models for SSD in data centers. While these research mainly focus on enterprise-grade SSD, relating research for consumer-grade SSD lacks.

The data of *CSS* and enterprise storage systems are equally important. The above research of failure prediction based on HDDs and SSDs mainly focuses on data centers or enterprise storage systems. However, they are not directly applicable to *CSS*. The failure prediction of storage drives in *CSS* faces many challenges that data centers do not face.

(1) It is necessary to utilize the proactive failure prediction for SSD-based *CSS*. Original *SMART*-threshold-based failure detection technologies provided by PC manufacturers cannot predict failures and reduce the risk of data loss. Users in *CSS* do not have the passive fault tolerance mechanisms adopted by users in enterprise storage systems and data centers. Even if users adopt other storage media for data backup, it is not easy to ensure the timeliness of backup. Some data is also not suitable for backup to the cloud. Data recovery is complicated and costly when a failure happens on SSD. Hence, introducing proactive failure prediction technologies into *CSS* can significantly alleviate this problem.

(2) Data discontinuity and uncertainty of failure time in *CSS* reduce the quality of data and affect the performance of model prediction. There are differences between data centers and *CSS*. Individual users are scattered and cannot be centrally managed as conveniently as data center users, especially in data collection and failure collection. Data centers provide service on a 24/7 basis, which can easily collect data regularly. While the startup time of *CSS* is irregular, data collection at the hr/min level is unrealistic, resulting in the discontinuity of the dataset. Individual users would not seek repair immediately once the drive fails, so the interval between failure and repair makes it challenging to determine the SSD's actual failure time.

(3) The differences between enterprise-grade and consumer-grade SSD make the model for the former cannot be directly used for that of the latter. There are some differences between client SSDs and enterprise SSDs at both the drive level (e.g. controller, NAND quality, and FTL algorithms) and the user level (e.g., workload and power on/off behavior). It is incorrect to use a model for HDD or SSD with SLC/MLC flash and PCIe/SAS interface to predict the failure of SSD with TLC flash and SATA interface.

(4) Beyond *SMART*, other features for failure prediction in data centers are not entirely applicable to *CSS*. Due to the high *FPR* or low *TPR*, the *SMART*-based models may raise the misclassification overhead, leading to additional data migration, unnecessary service interruption, and latent economic losses. Other features (workloads/location logs) relevant to SSD failures in data centers are not readily available in *CSS*. The application usage habits of individual users vary considerably, and so does the relevant load and performance information. The location information (disk/server/rack/room) is related to drive failures in data centers, while that of servers in *CSS* is so widely distributed that it is of little help for failure prediction.

III. Our Proposed Scheme

A. *Scheme Overview*

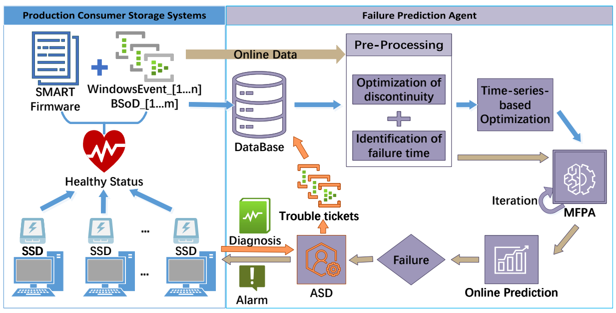


Fig. 1: Scheme overview

Fig.1 shows the overview of our proposed scheme, which consists of two parts: the data mining and application of SSD failures in production *CSS*. The former is used for mining the multidimensional features related to SSD failures in *CSS*. An in-depth study is conducted on nearly 2.3 million SSDs spanning nearly two years, in which multidimensional features (*SMART, Firmware, WindowsEvent, BlueScreenofDeath*, etc.) are proved to be related with SSD failures. The latter is the application of the failure prediction model of SSD. It builds an effective failure prediction model for SSDs by optimizing data preprocessing and model training.

B. *Multidimensional features (SFWB) related to SSD failures*

The upper-layer services could already be affected before a complete SSD failure. Many of the system-level events are early signals of disk errors. The trouble tickets provided by the after-sales department detail various hardware and software problems or prompts (such as machine case/screen/motherboard/SSD damage or software faults).

TABLE I: RaSRF—Replaced as SSD_Related Failures

Failure Level	Category	Causes	Pct.
Drive Level (31.62%)	Components failure	Storage drive failure	31.13%
		Firmware upgrade failure	0.42%
		Overtemperature	0.07%
System Level (68.38%)	Boot/ Shutdown failure	Blue/Black screen after startup	21.44%
		Unable to boot/ shutdown	15.84%
		Bootloop	7.73%
		Stuck startup icon	3.20%
	System running failure	Response delay/ blue screen	8.66%
		Unauthorized system installation	5.43%
		System partition damage	2.58%
		Automatic shutdown/ restart	1.94%
		System upgrade/ recovery failure	0.78%
	Application error	Apps crash/ report errors/ stuck	0.77%

By mining the trouble tickets, we discover the details related to SSD failure (named *RaSRF*-Replaced as SSD_Related Failures) . As presented in Table I, SSD failures can be manifested as drive-level and system-level ones. ***Drive-level** failures* (31.62%) include descriptions of SSDs directly identified as faulty, and the symptoms include slow read/write, data loss, data failure after firmware upgrade, automatic restart, etc. ***System-level** failures* (68.38%) contain system startup or software running errors. 48.21% of the failures occur during system startup or shutdown, and 20.16% occur during system running. The combination of driver-level and system-level errors helps comprehensively identify the eventual SSD failures.

TABLE II: SMART attributes

ID #	Attribute Name	ID #	Attribute Name
1	Critical Warning	9	Host Write Commands
2	Composite Temperature	10	Controller Busy Time
3	Available Spare	11	Power Cycles
4	Available Spare Threshold	12	Power On Hours
5	Percentage Used	13	Unsafe Shutdowns
6	Data Units Read	14	Error Media and Data Integrity Errors
7	Data Units Written	15	Number of Error information Log Entries
8	Host Read Commands	16	Capacity

Observation #1: *SMART* (S) comprehensively reflects the health status of SSD. SMART is designed to detect and report various indicators of drive reliability, including multiple types of errors and operational data. It has been widely used for failure prediction of storage drives. Except for capacity, the vendors only provide 15 SMART features for M.2 SSDs, which is listed in Table II. Based on *S_12* (*power on hours*), we plot the failure time distribution of SSDs in *CSS*. As shown in Fig.2, the failure numbers are higher in infancy, tend to be stable, and then gradually increase during the wear-out period. It fits the bathtub curve of the SSD lifecycle.

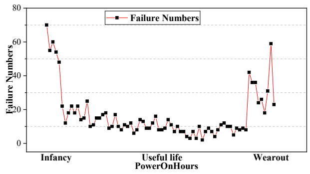


Fig. 2: Failure distribution

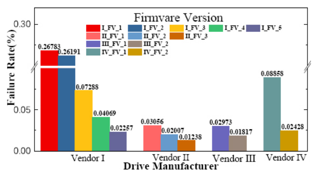


Fig. 3: Failure rate of *FV*s

Observation #2: *Firmware* (F) affects SSD availability. SSD firmware update fixes known bugs that might otherwise trigger driver errors or even SSD failures in older versions. Vendors have different naming conventions for *FirmwareVersion* (*F*), ranging from strings to numeric values. In Fig.3, we calculate the failure rate of different *FirmwareVersion* for faulty SSDs. We name *F_j* according to the time sequence of Vendor *i* (i.e., *i_F_j*). Vendor I has 5 different *FirmwareVersion*s with high failure probability, especially *I_F_1* and *I_F_2*. Vendor II has 3 *FirmwareVersion*s, and Vendor III and IV have 2. For all the 5 SSD vendors, the failure rate of the earlier *FirmwareVersion* is higher than that of the latter. The earlier the firmware version, the higher the failure rate. We observe that most SSDs in the historical dataset remain on the fixed *F* rather than update. The possible cause may be that the SSD management software does not push the update notification, or the user does not perform the update-driven action.

Observation #3: Some *WindowsEvents* (W) are early signals of SSD failures. As shown in Table I, when an SSD starts to fail, the SSD/system/application will gradually experience various errors. *WindowsEventViewer* helps troubleshoot various windows problems.

TABLE III: WindowsEvent logs

ID#	Description
W_7	The device has a bad block
W_11	The driver detects a controller error on Disk_i
W_15	The Disk_i is not ready for access yest
W_49	Configuring the page file for crash dump fails.
W_51	An error is detected on device during a paging operation
W_52	The driver detects that device has predicted it will fail.
W_154	The IO operation at logical block address 0x5e50d0 for Disk_i fails due to a hardware error
W_157	Disk has been surprisingly removed
W_161	MSExchageIS(303) File System error during IO on database.

It displays logs of application and system messages, including errors, information messages, and warnings. We investigate *WindowsEvent* descriptions in Table III and find some of them are related to SSD failures. We track and collect the number of *W*s occurring each day. Fig.4 lists the cumulative total of *W_161* metrics distributions for healthy and faulty storage drives before failure. Faulty SSDs (*F1-F4*) are more likely than healthy SSDs (*N1-N4*) to experience various *W* errors before the eventual failure.

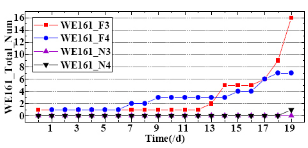


Fig. 4: Sum of W_161

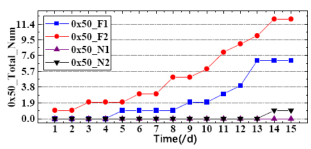


Fig. 5: Sum of B_50

Observation #4: Some *BlueScreenofDeath* (B) logs are early signals of SSD failures. *BlueScreenofDeath* is an error screen displayed on a windows computer after a fatal system error. It is caused by various problems, such as general hardware failures or unexpected terminations of critical processes. Through further investigation into the causes of *B*, we find that many *B* errors are related to SSD failures (see Table IV). Damaged storage drives, bad sectors, and various storage drive issues all result in *B*. We daily collect the number of *B* listed in Table IV. Fig.5 presents the cumulative distribution of *B_50* for healthy and faulty SSDs. Compared to the healthy SSDs (*N1-N4*), the faulty SSDs (*F1-F4*) are more likely to encounter various *B* errors before the eventual failure.

TABLE IV: BlueScreenofDeath logs

ID#	Attributed Name	ID#	Attributed Name
0x23	FAT_FILE_SYSTEM	0xE4	WORKER_INVALD
0x24	NTFS_FILE_SYSTEM	0xFC	ATTEMPTED_EXECUTE_OF_ NOEXECUTE_MEMORY
0x48	CANCEL_STATE_IN _COMPLETED_IRP	0x10C	FSRTL_EXTRA_CREATE_ PARAMETER_VIOLATION
0x50	PAGE_FAULT_IN _NONPAGED_AREA	0x12C	EXFAT_FILE_SYSTEM
0x6B	PROCESSL_INITIALIZATION _FAILED	0x135	REGISTRY_FILTER_ DRIVER_EXCEPTION
0x77	KERNEL_STACK_INPAGE _ERROR	0x13B	PASSIVE_INTERRUPT_ERROR
0x7A	KERNEL_DATA_INPAGE _ERROR	0x157	KERNEL_THREAD_PRIORITY _FLOOR_VILOATION
0x80	NMI_HARDWARE_FAILURE	0x17E	MICROCODE_REVISION_MISMATCH
0x9B	UDFS_FILE_SYSTEM	0x189	BAD_OBJECT_HEADER
0xC7	TIMER_OR_DPC_INVALID	0x1DB	IPI_WATCHDOG_TIMEOUT
0xDA	SYSTEM_PTE_MISUSE	0xC00	STATUS_CANNOT_LOAD

C. *SFWB-based failure prediction model (MFPA)*

Based on the above discovered multidimensional features (*SFWB*), we build the failure prediction model for SSD in *CSS*.

(1) Optimization of the discontinuous data. The dataset consists of *serial number (S/N), model, timestamp, interface, capacity, S*$\{1...m\}$*, F, W*$\{1...i\}$*, B*$\{1...j\}$. *S/N* is the identifier of an SSD. The *model* shows the vendors of SSD. The *interface* is *PCIe 3.0*4*. *Label encoding* technology is adopted to handle the firmware version that is a character variable. We calculate the accumulative values of *W* and *B* as the input features because the daily number of *W* and *B* is hard to detect trends.

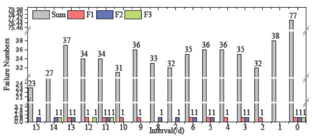


Fig. 6: Data Discontinuity

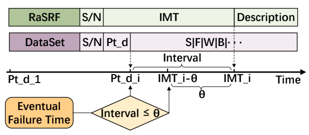


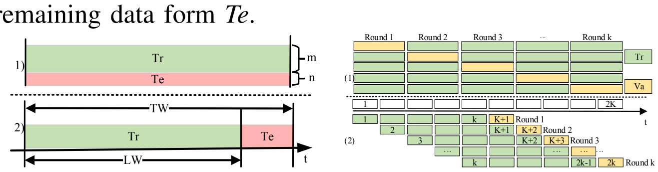
Fig. 7: Eventual failure time

Unlike data centers that continuously collect data, the *discontinuity* of data is the particularity of *CSS*. The healthy data cannot be recorded daily as there is no guarantee that users will turn on their computers daily. The timestamp of the dataset in *CSS* is relatively scattered. As shown in Fig.6, Vendor I's total number of faulty SSDs ranges from 23 to 77 within a single interval over two months. The faulty SSD *F1* has the logs at timestamps *(0, 2-6, 9-13)*. That of *F2* and *F3* are *(0, 3, 5-8, 11, 13-15)* and *(0, 11-14)* respectively. If the data of some faulty SSDs is discontinuous and the time window between adjacent time points is too long (e.g., *F3*), the data cannot be used for subsequent model training; otherwise, the model performance will be affected. We remove the data with a long interval ($\geq$ 10) and fill the mean value of adjacent time windows(= 3) for partial discontinuous data.

(2) Identification of the eventual failure time. Unlike the datasets in the data center that can be conveniently labeled as faulty or healthy, the data of *CSS* require manual intervention and labeling. The SSDs of *CSS* are labeled through the corresponding *S/N* in *RaSRF* from trouble tickets. As presented in Fig.7, *RaSRF* records SSD failures in trouble tickets, including *S/N*, initial maintenance time (*IMT*), and corresponding failure descriptions. *Dataset* stands for the history SSD logs, including *S/N*, tracking point date (*Pt_d*), and value of *SFWB*. It is inaccurate to directly label *IMT_i* in the *RaSRF* closest to *Pt_d_i* in the dataset as the failure time of SSD *i*. There may be a time interval (*ti*) between *Pt_d_i* and *IMT_i* because a faulty SSD may not be immediately sent to the after-sales department. A threshold θ is taken to address the problem. If $ti \leq \theta$, the *Pt_d_i* closest to *IMT_i* is used as the SSD failure time point. Otherwise, the corresponding $\{$*IMT_i* - $\theta\}$ is taken. The attribute value of the faulty disk generally changes during a period before the failure occurs. The value of θ is set to 7 through the sensitive test. If the threshold is too high, the feature value of the faulty disk around $\{$*IMT_i* - $\theta\}$ is similar to that of the healthy disk, increasing the model's FPR; if it is too low, many faulty disks have no data around $\{$*IMT_i* - $\theta\}$, reducing the model's TPR.

(3) Time-series-based optimization. The datasets of SSD health status are characterized by imbalance and time series. Faulty SSDs data collected within 7, 14, or 21 days before failures are generally selected as positive samples. The negative samples are selected from the healthy SSDs in proportion to the positive ones (e.g., *3:1 or 5:1*). We adopt the *RandomUnderSampler* algorithm to balance the minority and majority classes. We further optimize *sample segmentation* and *cross-validation*.

(3.1)Timepoint-based sample segmentation. As shown in Fig.8(a) (1), a dataset is usually divided into a training set (*Tr*) and a test set (*Te*) in a particular proportion (*m:n*) (e.g., 10:1). However, this method does not consider the time series of data. The training set may contain future data, while the test set may include historical data, which increases the possibility of inaccurate prediction. Therefore, as shown in Fig.8(a) (2), a timepoint-based sample segmentation method is adopted. Initially, the data of *Tr* and *Te* are chosen from the given time window (*TW for short*) in the historical dataset. Then, *Tr* and *Te* are segmented by the time point in a learning time window (*LW for short*). The data within *LW* are used as *Tr*, and the remaining data form *Te*.



(a) Sample segmentation

(b) Cross-Validation

Fig. 8: Timeseries-based Optimization

(3.2)Time-series-based Cross-Validation. The *k-fold Cross Validation* is widely applied for model training. The training datasets are divided into *k* subsets in Fig. 8(b) (1). Each iteration will choose one subset as a validation sample (*Va*), while the remaining *k-1* subsets are integrated into the training sample (*Tr*). However, the timing features of the data are still underutilized. The training *k-1* subsets may contain future data, while the validation subsets may contain historical ones, reducing

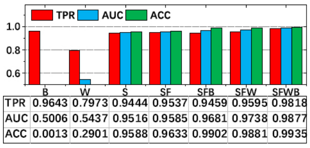


Fig. 9: Across Features(a)

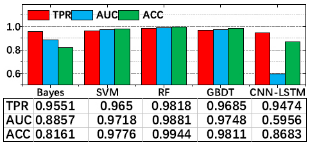


Fig. 10: Across MLs(a)

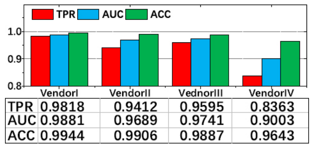


Fig. 11: Across Vendors(a)

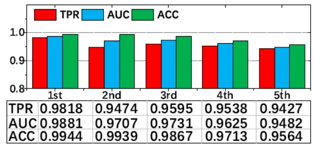


Fig. 12: Across Months(a)

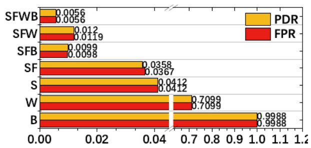


Fig. 13: Across Features(b)

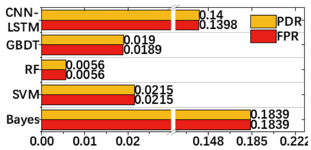


Fig. 14: Across MLs(b)

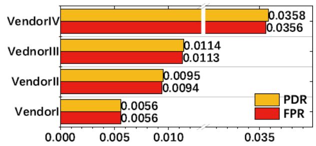


Fig. 15: Across Vendors(b)

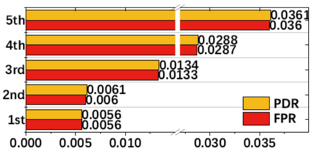


Fig. 16: Across Months(b)

prediction accuracy. The *time-series-based cross-validation* is proposed to address the problem in Fig.8(b) (2). The data are divided into *2*k* subsets (labeled *1, ..., 2k*) in chronological order. The corresponding *Tr* consists of records occurring only before the records that form *Va*. Consequently, the model will not be trained by future samples. The consecutive k subsets are for building the model in each iteration, and the following k+1 subset is for validation.

(4) Multiple ML Algorithms and Hyperparameter Optimization. With the proposed *SFWB*, the optimization of data preprocessing and model training, we realize the failure prediction model based on multiple *ML* algorithms, including *Bayes, SVM, RF, GBDT*, and *CNN_LSTM*. The hyperparameter controls the learning process, and other parameters (node weights) are obtained by training. Different model training algorithms need different hyperparameters, such as maximum tree depth and max features for *RF* and the neural network's learning rate and mini-batch size. We utilize *Grid Search*, combined with time-series-based cross-validation, to optimize the value of hyperparameters.

(5) Feature Group Sets. We adopt various input datasets to verify the validity of different features. As presented in Table V, we divide them into seven different feature groups: *SFWB, SFW, SFB, SF, S, W,* and *B*. Group *S* is the baseline. Although each group set contains multiple attributes, not all of them are associated with SSD failures. We implement a *sequential forward selection* algorithm [27] to select the optimal subset.

TABLE V: Feature Groups

	SMART	Firmware	WindowsEvent	BlueScreenofDeath
SFWB	16	1	5	23
SFW	16	1	5	NaN
SFB	16	1	NaN	23
SF	16	1	NaN	NaN
S	16	NaN	NaN	NaN
W	NaN	NaN	5	NaN
B	NaN	NaN	NaN	23

IV. Evaluation and Analysis

(1) Dataset. The dataset consists of nearly 2.3 million M.2 SSDs based on 3D TLC NAND flash that supports the NVM Express protocol. As listed in Table.VI, they are mainly from four manufacturers (I to IV) with 12 models of different capacities (from 128GB to 1TB) and layers (from 32-layer to 96-layer). The four manufacturers' total Replacement Rate (*RR*) is 0.0068, 0.0007, 0.0005, and 0.0011 respectively. We train the prediction model based on vendors rather than the traditional model based on disk series.

TABLE VI: Dataset

Manu. /Model	F/F	Protocol	FlashTech	Total	Sum_failure	Sum_RR
I	M.2 (2280)	NVMe1.*	3D TLC	270,325	1850	0.0068
II				1,001,278	669	0.0007
III				908,037	463	0.0005
IV				152,405	172	0.0011

The *confusion_matrix* (displays the number of true positives (*TP*), false positives (*FP*), false negatives (*FN*), and true negatives (*TN*)) is widely used to evaluate the effectiveness of the classification model. Accuracy (*ACC=(TP+TN)/(TP+TN+FP+FN)*) is the ratio of all correctly predicted cases and all cases of the data. True positive rate (*TPR=TP/(TP+FN)*) is the proportion of correctly predicted cases. False positive rate *(FPR=FP/(FP+TN))* refers to the expectancy of the false positive ratio. *AUC* (the area under the *ROC* curve) represents the trade-off between *TPR* and *FPR*. Beyond the above metrics, the newly introduced *Positive Detection Rate* (*PDR=(TP+FP)/(TP+TN+FP+FN)*) is proposed to reflect the ratio of all predicted positive cases and all cases.

(2) How does *MFPA* perform towards feature groups?

(2.1) As present in Fig.9 and Fig.13, the ***SFWB*** group performs the best (98.18% TPR and 0.56% FPR) across all feature groups, confirming our hypothesis that *SFWB* features are helpful to the SSD failure prediction beyond the traditional SMART-based prediction model. ***W*** and ***B*** errors are good indicators of SSD failures. The *TPR*, *FPR*, and *PDR* of the *SFWB*-based model are 98.18%, 0.56%, and 0.56%, compared to 95.37%, 3.58%, and 3.67% of the *SF*-based model respectively. Adding the ***F*** feature improves the prediction capability, but the improvement is limited(less than 10% benefit in *TPR* and *FPR*). Minor firmware version updates do not mean that previous *F* will cause SSD failures. *F* is not frequently updated unless a severe bug is found.

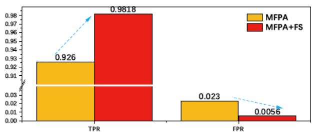


Fig. 17: Feature selection

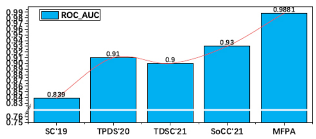


Fig. 18: Comparison

(2.2) Through feature selection shown in Fig.17, the TPR of the model increases from 0.926 to 0.9818, and the FPR decreases from 0.023 to 0.0056. The optimal feature subset obtained from feature selection varies from vendors and data sets. Features such as *Available Spare Threshold, Error Media and Data Integrity Errors, power cycles, W_11, W_49, W_51, W_161, B_50, B_7A* require special attention. *Available Spare Threshold* is less associated with SSD failures.

Meanwhile, we compare *MFPA* and state-of-art studies [19]–[22]. They build the SSD-based failure prediction model with some *ML* algorithms. Fig.18 proves that *MFPA* achieves the best performance, reflecting the effectiveness of *SFWB* groups for SSD failure prediction.

(3) **Is *MFPA* portable across ML algorithms?** As listed in Fig.10 and Fig.14, we evaluate *MFPA* across many widely used algorithms (*Bayes, SVM, RF, GBDT, CNN_LSTM*). The *TPR* of *MFPA* models based on traditional *ML* algorithms can reach more than 95%. *RF* performs best with 98.18% *TPR* and 0.56% *FPR*. *CNN_LSTM* achieves 94.74% *TPR* and 12.98% *FPR*. Data discontinuity is inevitable in *CSS*, which leads to poor data quality, and further affects the performance of time-series-based *CNN_LSTM*. The tree-based model is superior to other models for discontinuous data.

(4) **Is *MFPA* portable across vendors?** These SSDs from four vendors are widely used in *CSS* by various PC manufacturers (*Dell, Lenovo, HP, SAMSUNG* laptop/office computers, etc.) As listed in Fig.11 and Fig.15, *SFWB*-based *MFPA* performs effectively across different vendors from I to III (with 98.81%, 96.89% and 97.41% *AUC*, respectively). The model for vendor IV works not well as it has the fewest faulty SSDs.

(5) Is *MFPA* portable across time periods? We make *MFPA* predict continuously for five months without iteration. As shown in Fig.12 and Fig.16, the model's *TPR* of vendor I keeps stable for five months, while its *FPR* in third month increases to 1.34%. Other manufacturers fared similarly. The model needs iteration every 2-3 months. The historical changes of some feature values that *MFPA* has learned in the past cannot adapt to the new data, causing the increasing *FPR* in the subsequent period. The model needs to be iterated periodically.

(6) Is *MFPA* portable across the lookahead window? Fig.19 presents the *TPR* of *MFPA* in different lookahead windows (up to 21 days). Failure prediction several days in advance is sufficient for subsequent processing (such as data backup and replacement). *MFPA* performs well within 5 days (89% *TPR*). A long lookahead window *N* reduces the difference in feature values between healthy and faulty SSDs, resulting in a misjudgment of the model (55.66% *TPR* at N = 20).

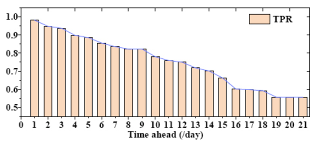


Fig. 19: Across lookahead N

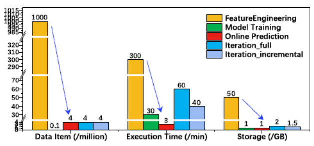


Fig. 20: Overhead

Fig.20 lists the overhead of *MFPA* in various stages. Feature engineering occupies the most overhead regarding the data item, execution time, and storage space. For 4 million real-time data, the model takes only about 3 minutes to complete the prediction. Microsecond prediction can be achieved for the model deployed on the client side. The model is iterated every two months and pushed to the user for updates.

V. Conclusion

Through an in-depth study on 2.3 million SSDs from production *CSS*, we find that multidimensional features (*SFWB, SMART, Firmware, WindowsEvent, BlueScreenofDeath*) are correlated with SSD failures. We further put forward *SFWB*-based *MFPA* to predict SSD failures by optimizing discontinuous data and eventual failure time identification, as well as time-series-based sample segmentation and cross-validation. Experiment results show that *SFWB*-based *MFPA* can achieve a high *TPR* (98.18%) and low *FPR* (0.56%), which is 4% higher and 86% lower than the *SMART*-based model. It is robust and portable on ML algorithms, manufacturers and time periods.

Acknowledgment

This work was supported by Key Laboratory of Information Storage System and Engineering Research Center for Data Storage Systems and Technology, Ministry of Education, China.

References

[1] S. Bianca. *et al.*, "Understanding disk failure rates: What does an mttf of 1,000,000 hours mean to you?" *TOS*, vol. 3, no. 3, pp. 8–es, 2007.
[2] V. K. Venkatesh. *et al.*, "Characterizing cloud computing hardware reliability," in *SoCC*, 2010, pp. 193–204.
[3] W. Guosai. *et al.*, "What can we learn from four years of data center hardware failures?" in *DSN*, 2017, pp. 25–36.
[4] L. Ponemon, "Cost of data center outages," *Data Center Performance Benchmark Serie*, 2016.
[5] B. Allen, "Monitoring hard disks with smart," *Linux Journal*, no. 117, pp. 74–77, 2004.
[6] B. M. Madalina. *et al.*, "Predicting disk replacement towards reliable data centers," in *SIGKDD*, 2016, pp. 39–48.
[7] X. Yanwen. *et al.*, "Ome: An optimized modeling engine for disk failure prediction in heterogeneous datacenter," in *ICCD*, 2018, pp. 561–564.
[8] X. Yong. *et al.*, "Improving service availability of cloud systems by predicting disk error," in *ATC*, 2018, pp. 481–494.
[9] X. Yanwen. *et al.*, "Dfpe: Explaining predictive models for disk failure prediction," in *MSST*, 2019, pp. 193–204.
[10] Z. Ji. *et al.*, "Hddse: Enabling high-dimensional disk state embedding for generic failure detection system of heterogeneous disks in large data centers," in *ATC*, 2020, pp. 111–126.
[11] L. Sidi. *et al.*, "Making disk failure predictions smarter!" in *FAST*, 2020, pp. 151–167.
[12] M. Stathis. *et al.*, "A study of *ssd* reliability in large scale enterprise storage deployments," in *FAST*, 2020, pp. 137–149.
[13] C. Yu. *et al.*, "Data retention in mlc nand flash memory: Characterization, optimization, and recovery," in *HPCA*, 2015, pp. 551–563.
[14] M. K. Qureshi *et al.*, "Avatar: A variable-retention-time (vrt) aware refresh for dram systems," in *DSN*, 2015, pp. 427–437.
[15] F. Mahdisoltani *et al.*, "Proactive error prediction to improve storage system reliability," in *ATC*, 2017, pp. 391–402.
[16] S. Bianca. *et al.*, "Flash reliability in production: The expected and the unexpected," in *FAST*, 2016, pp. 67–80.
[17] X. Erci. *et al.*, "Lessons and actions: What we learned from 10k ssd-related storage system failures," in *ATC*, 2019, pp. 961–976.
[18] H. Shujie. *et al.*, "An in-depth study of correlated failures in production ssd-based data centers," in *FAST*, 2021, pp. 417–429.
[19] A. Jacob. *et al.*, "Ssd failures in the field: symptoms, causes, and prediction models," in *SC*, 2019, pp. 1–14.
[20] Z. Ji. *et al.*, "Minority disk failure prediction based on transfer learning in large data centers of heterogeneous disk systems," *TPDS*, vol. 31, no. 9, pp. 2155–2169, 2020.
[21] C. Chakraborttii *et al.*, "Improving the accuracy, adaptability, and interpretability of ssd failure prediction models," in *SoCC*, 2020, pp. 120–133.
[22] R. Pinciroli *et al.*, "Lifespan and failures of ssds and hdds: Similarities, differences, and prediction models," *TDSC*, 2021.
[23] Z. Ji. *et al.*, "Tier-scrubbing: An adaptive and tiered disk scrubbing scheme with improved mttd and reduced cost," in *DAC*, 2020, pp. 1–6.
[24] Z. Xinyan. *et al.*, "Csle: a cost-sensitive learning engine for disk failure prediction in large data centers," in *DATE*. IEEE, 2022, pp. 478–483.
[25] S. Xiaoyi. *et al.*, "System-level hardware failure prediction using deep learning," in *DAC*, 2019, pp. 1–6.
[26] L. Chuan. *et al.*, "Ntam: neighborhood-temporal attention model for disk failure prediction in cloud platforms," in *WWW*, 2021, pp. 1181–1191.
[27] A. W. Whitney, "A direct method of nonparametric measurement selection," *TC*, vol. 100, no. 9, pp. 1100–1103, 1971.