MSFRD: Mutation Similarity based SSD Failure Rating and Diagnosis for Complex and Volatile Production Environments

Yuqi Zhang, Tianyi Zhang, Wenwen Hao, Shuyang Wang, Na Liu, and Xing He, *Samsung R&D Institute China Xi'an, Samsung Electronics;* Yang Zhang, Weixin Wang, Yongguang Cheng, Huan Wang, Jie Xu, Feng Wang, and Bo Jiang, *ByteDance Inc.;* Yongwong Gwon, Jongsung Na, Zoe Kim, and Geunrok Oh, *Samsung Electronics*

https://www.usenix.org/conference/atc24/presentation/zhang-yuqi

This paper is included in the Proceedings of the 2024 USENIX Annual Technical Conference.

July 10–12, 2024 • Santa Clara, CA, USA

978-1-939133-41-0

Open access to the Proceedings of the 2024 USENIX Annual Technical Conference is sponsored by

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of Science and Technology

MSFRD: Mutation Similarity based SSD Failure Rating and Diagnosis for Complex and Volatile Production Environments

Yuqi Zhang[1], *Tianyi Zhang*[1], *Wenwen Hao*[1], *Shuyang Wang*[1], *Na Liu*[1], *Xing He*[1],
Yang Zhang[2], *Weixin Wang*[2], *Yongguang Cheng*[2], *Huan Wang*[2], *Jie Xu*[2], *Feng Wang*[2], *Bo Jiang*[2],
Yongwong Gwon[3], *Jongsung Na*[3], *Zoe Kim*[3], *Geunrok Oh*[3]
[1]*Samsung R&D Institute China Xi'an, Samsung Electronics*
[2]*ByteDance Inc.* [3]*Samsung Electronics*

Abstract

SSD failures have an increasing impact on storage reliability and performance in data centers. Some manufacturers have customized fine-grained Telemetry attributes to analyze and identify SSD failures. Based on Telemetry data, this paper proposes the mutation similarity based failure rating and diagnosis (MSFRD) scheme to predict failures in dynamic environment of data centers and improve failure handling efficiency. MSFRD dynamically detects the internal mutations of SSDs in real time and measures their similarity to the mutations of historical failed SSDs and healthy SSDs for failure prediction and early rating. Based on the rating, unavailable SSDs with serious failures are handled immediately, while available SSDs with less serious failures will be continuously tracked and diagnosed. The MSFRD is evaluated on real Telemetry datasets collected from large-scale SSDs in data centers. Compared with the existing schemes, MSFRD improves precision by 23.8% and recall by 38.9% on average for failure prediction. The results also show the effectiveness of MSFRD on failure rating and progressive diagnosis.

1 Introduction

Nowadays, with the development of the Internet, the scale of data is growing exponentially and storage plays a more and more crucial role in computer systems. NAND flash-based solid state drives (SSDs) have better performance and lower power consumption than hard disk drives (HDDs), and are increasingly used in data centers. However, as the storage density of SSDs increases, the durability and reliability are decreasing [18, 21], which poses a challenge to the storage reliability of large data centers with even millions of SSDs. SSD failures have received more and more attention due to the following two impacts. First, although passive failure tolerance mechanisms (such as replication [33] and RAID [26]) are used to avoid data loss, SSD failures would cause instability in online services, such as jitter performance and long-tail latency. Second, SSD failures bring additional maintenance costs, and failure handling may be inaccurate and result in more recovery costs. Therefore, SSD failure prediction, as a proactive failure tolerance mechanism, is a powerful supplement to passive failure tolerance mechanisms. The industry expects to reduce the impact and cost of SSD failures through early failure detection and handling (such as service migration and SSD replacement).

Traditional failure prediction is mainly based on Self-Monitoring, Analysis, and Reporting Technology (SMART) originated from HDDs. Based on SMART logs, the existing HDD and SSD failure prediction schemes adopt classification or anomaly detection algorithms to distinguish failed disks from healthy disks. Specifically, classification algorithms such as random forest learn existing patterns of failed disks and healthy disks to perform binary classification [6, 16, 20, 22, 35, 38, 44]. Anomaly detection algorithms such as autoencoder learn the pattern of healthy disks, and identify a failed disk when its SMART data are very different from those of most healthy disks [4, 7, 42]. Based on these algorithms and SMART logs, HDD failure prediction has achieved high prediction accuracy [20], but SSD failure prediction has not [6]. Unlike mechanical-based HDDs, the internal mechanism of flash-based SSDs is more complicated. The SSD has more components and processes that require fine-grained monitoring. Moreover, it has various error tolerance mechanisms [15], which makes it difficult to distinguish between real failures and tolerable errors. SSD monitoring and failure prediction are still facing great challenges in industry.

To enhance SSD monitoring and failure warning, some manufacturers (e.g., Samsung) have customized comprehensive Telemetry logs with more attributes to monitor SSD's internal mechanisms and components in detail. However, we find that the existing schemes face three challenges and are not fully suitable for complicated Telemetry information and dynamic environment in practice. To this end, we propose the mutation similarity based failure rating and diagnosis (**MSFRD**) scheme. The existing challenges and the main ideas of our solution are summarized as follows and introduced in detail in Section 3.

- **Changes in data importance from model training to online prediction.** The existing schemes design feature engineering based on training data to obtain key information from data. For example, feature selection is usually used to select important monitoring attributes for failure prediction [4, 6, 22, 34, 38, 42]. However, in practice, the training data are historical data, but actual failure prediction is performed on future online data [39, 40]. As SSDs wear out, failure-irrelevant attributes in the training data may become failure-relevant in future online data, especially when there are many Telemetry attributes. In this case, traditional static feature engineering based on historical training data would miss some attributes that are important for online prediction. Different from static feature engineering, we propose to dynamically extract failure-related mutations (i.e., rare and sudden changes) from Telemetry data in real time as features, thereby focusing on the real critical information in future online data even if they are not critical in historical training data. (See Section 3.1)

- **Unseen patterns in dynamic data center environments.** For failure prediction in practice, traditional classification models are trained based on historical data and can effectively distinguish health/failure patterns seen in historical data. However, in practice, data patterns continuously change and unseen patterns would appear [4]. Some studies [4, 42] adopt anomaly detection models to predict failures by detecting outlier patterns including unseen patterns, but outlier patterns are not exactly equivalent to failure patterns. It is still a challenge to accurately capture both seen and unseen failure patterns. To take advantage of classification models and anomaly detection models, we adopt the idea of similarity measurement that exist in both types of models. The similarity of mutations in SSDs is measured to capture both seen and unseen patterns. (See Section 3.2)
- **Diverse failure phenomena and degrees.** Most existing schemes have a single definition and handling measure for failures [1, 4, 6, 17, 38, 42]. However, in complex production environments, SSD failures are diverse and vary in degree [19, 39]. In addition to unavailable SSDs with serious failures that should be replaced immediately, some available SSDs with less serious failures (e.g., performance degradation) can be further diagnosed and handled accordingly. The previous schemes cannot capture fine-grained failure status and lack suggestions for failure analysis and handling, which brings troubles to operators. In this paper, we adopt failure rating to distinguish the detailed status and health/failure level of SSDs to reduce operators' burden. In particular, by identifying and diagnosing available SSDs with less serious failures, the cost of unnecessary SSD replacement can be reduced. (See Section 3.3)

We carried out evaluations on large-scale datasets from data centers. The results show that the proposed scheme improves precision by 23.8% and recall by 38.9% on average for failure prediction and improves accuracy rate by at least 38.7% for failure rating, compared with existing schemes. Our contributions are summarized as follows.

1. We propose a dynamic mutation feature extraction method to locate abnormal changes and failure symptoms of SSDs in real time, avoiding the impact of changes in data importance. The deviation of data trends from expectations is adopted to capture mutations, and self-learned weights are designed to reflect the rarity and importance of mutations.
2. We propose a mutation based similarity measure approach to capture failure patterns. When patterns that have been seen in historical data are captured through similarity measurement, unseen patterns are also perceived based on their outlier degree and health/failure tendency.
3. We propose failure rating to predict and distinguish the status and degree of SSD failure, and suggest corresponding measures. In particular, for available SSDs with less serious failures, we continuously track and diagnose the fine-grained failure status and perform progressive processing.

2 A Look at Field Data

We have two large-scale Telemetry datasets collected from the data centers of large Internet companies such as ByteDance. The Telemetry logs are the snapshots of SSD internal attributes on a regular basis (e.g., one Telemetry log per day for each SSD). The first dataset has over 41 million Telemetry logs collected from more than 120,000 SSDs over a year and a half, called the 41-M dataset in this paper. The SSDs carry various businesses and workloads. The second dataset from another company contains over 10 million Telemetry logs collected within seven months from more than 35,000 SSDs, called the 10-M dataset in this paper. These data are collected from Samsung's current data center-level PM9A3 SSD, since Samsung has customized up to 85 Telemetry attributes for PM9A3. These attributes monitor the fine-grained status and mechanisms inside the SSD. In particular, there are various error tolerant mechanisms inside SSDs to enhance their reliability. In addition to common read/write and temperature-related attributes, error tolerance and uncorrectable situations are also recorded. Some key Telemetry attributes are shown in Table 1.

Besides Telemetry data and corresponding SSD information (serial number, model, firmware, etc.), the failure lists are also collected by the operators, including serial number of failed SSDs, failure's report date, failure description, and handling measures. There are 1,126 and 318 records in the failure list of 41-M dataset and 10-M dataset respectively. Many of the failed SSDs have been confirmed as unavailable and then replaced by operators. Some SSDs that temporarily resulted in online problems (e.g., performance degradation or SSD lost) were reported in failure list but later confirmed to be usable by operators, so they are not replaced for the time being.

3 Background and Motivation

Recently, machine learning-based failure prediction schemes have become mainstream in HDD and SSD failure prediction

Uncorrectable error

- lifetime_uecc_count: the count of NAND's uncorrectable error-correction code (ECC) errors
- dram_uecc_count: the count of DRAM's uncorrectable ECC errors
- ETE_uncorrectable_error: the count of end-to-end uncorrectable errors

Error tolerance (correctable error)

- dram(sram)_cecc_count: the count of DRAM(SRAM)'s correctable ECC errors
- dram(sram)_cecc_address_count: the count of distinct addresses of DRAM(SRAM)'s correctable ECC errors
- read_recovery_attempts: the count of NAND reads that require retrying
- read_reclaim_count: the count of blocks that have been reallocated to maintain data integrity
- bad_user(system)_nand_block_count: the count of user (system) NAND blocks that have been retired

Read/write

- lifetime_user_reads(writes): the count of bytes read(written) by the host
- physical_media_units_read(written): the count of bytes read (written) by the media
- trailing_hour_WAF: the write amplification factor within one hour

Temperature

- highest_temperature: the highest temperature of the device
- lowest_temperature: the lowest temperature of the device
- over_temperature_minutes: the number of minutes the device exceeds the specified maximum operating temperature

Wear and capacitor

- wear_level_avg(max, min): the average (maximum, minimum) erase cycle of internal blocks
- endurance_estimate: a current estimate of the total number of data bytes that can be written to the device over its lifetime
- capacitor_health: an indicator of capacitor health and it represents capacitor energy margin

Table 1: The description of key Telemetry attributes. (Similar attributes are described together, such as dram_cecc_count and sram_cecc_count)

[40] because of their good performance. Failure prediction based on machine learning usually has the following three steps. 1) Feature engineering. Key information for failure prediction is extracted as features from raw monitoring data to reduce noise information. 2) Prediction model. The features are input to the machine learning model to predict failures. 3) Failure alarm and handling. Failures are alarmed and handled in advance based on the prediction results. Next, we will introduce the background and challenges of each of the above steps, as well as our motivations and ideas.

3.1 Feature Engineering

Background. Feature engineering aims to extract the key information from monitoring data to predict failures. The existing failure prediction schemes [4, 6, 22, 34, 38, 42] usually use feature selection to select monitoring attributes related to failures as features, thereby removing other irrelevant attributes to reduce noise. Many schemes use correlation coefficients [2] or J-index [9] to identify the attribute correlation for distinguishing healthy and failed SSDs, and select attributes with high correlation as features. The existing approaches all rely on training data, because they select attributes that are more relevant to failures in the training set. However, in practice, training data are historical data, and failure prediction are actually performed on future data [39, 40]. Furthermore, an evaluation period is required before the model goes online. As workloads change and SSDs wear out, the patterns of monitoring data will change and some unseen data patterns will appear, and the correlation between attributes and failures will also change. Accordingly, the features selected based on training data would not be fully applicable to future online data.

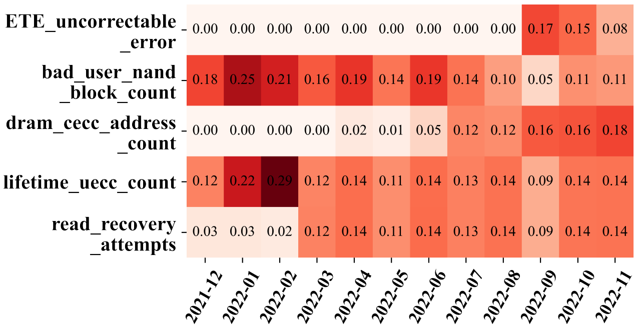
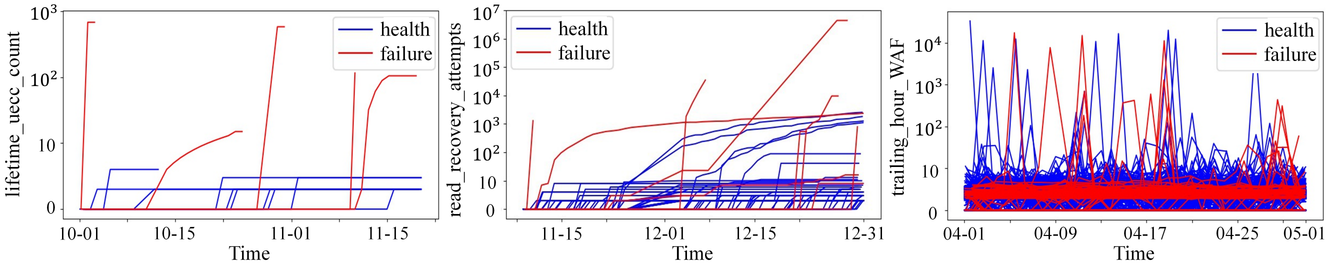


Figure 1: The Pearson correlation coefficients between Telemetry attributes and the failures per month. The top five attributes with the highest correlation coefficients during this period are displayed.

Figure 1 shows that the Pearson correlation coefficients [2] between attributes and failures vary greatly over time. For example, the attribute ETE_uncorrectable_error changes from failure-irrelevant (0) to failure-relevant (0.17), but it will be discarded and cannot contribute to future online prediction if feature selection is based on early data. In fact, after collecting data for over a year and a half, we still observe significant changes in the correlation between attributes and failures on latest data. This indicates that current static feature selection based on training data does not work well in practical time-based training-evaluation-prediction situation. According to our practice, after feature engineering and model training, the model should be evaluated for at least three months to verify its effectiveness before it is actually launched for online prediction, thus widening the time gap between training data and future online data.

In addition to feature selection, some previous schemes [39, 40] also extract time-series features (such as difference and slope) from the data of multiple monitoring logs to reflect



(a) NAND UECC trends.

(b) read_recovery_attempts trends.

(c) WAF trends.

Figure 2: Attribute trends of healthy and failed SSDs. The horizontal coordinate is the date (month-day), and the vertical coordinate is the value of corresponding attribute. Most of failed SSDs have no subsequent data since they have been replaced.

the changes in the SSD, which is helpful for capturing failure symptoms. The time-series features can reflect changes in monitoring values over time, but there are many normal changes inside SSD that are not related to failures, which will also introduce noise information. How to extract key information of failure-related changes remains a challenge.

Motivation and analysis. Based on the existing limitations, we conceive of dynamically capturing abnormal changes related to failures in online data as features, instead of performing static feature engineering based on historical training data. In this way, even if some attributes are not related to failures on historical training data, their information will still be extracted when they change abnormally in future online data. At the same time, by extracting abnormal changes, normal changes and corresponding attributes unrelated to failures in online data can be implicitly eliminated, thereby reducing noise information.

To dynamically capture abnormal changes related to SSD failures, we need to understand what changes are abnormal before failures occur. On the key Telemetry attributes such as error- and read/write-related attributes, the data trends of failed SSDs and healthy SSDs in our dataset are compared. We found that failed SSDs usually have rare, sudden, rapid changes (called mutations in this paper) in Telemetry attributes before the failure, as shown in Figure 2.

Figure 2(a) compares failed and healthy SSDs whose NAND UECC (i.e., lifetime_uecc_count) increases. It shows that these failed SSDs have similar symptoms, i.e., the rare and rapid increase of UECC. Some healthy SSDs' UECC also increases, but the increase tends to be slow and small. UECC means that the SSD has experienced some data read errors, and sporadic UECC may happen by accident, so the SSD still work later. However, the rapid and substantial increase of UECC implies that there are continuous unresolvable problems inside the SSD, and it is on the verge of failure. Figure 2(b) shows that the failed SSDs are also more likely to experience sudden increases of read_recovery_attempts compared with healthy SSDs. Read_recovery_attempts means that the SSD needs to solve some problems by read retry. When it increases rapidly, the read-retry mechanism may not solve the problems, thus resulting in a failure. Although most mutations are rare and more likely to appear on failed SSDs, we also find some common mutations that widely exist on both healthy and failed SSDs, such as WAF (i.e., trailing_hour_WAF) mutations in Figure 2(c). WAF is susceptible to workloads, so its mutation is more common and less meaningful for failure prediction compared with error- and wear-related mutations. In general, Telemetry mutations are abnormal changes related to failures, especially rare mutations which are more likely to appear before failures occur.

Ideas. To focus on actual important attributes and abnormal changes in online data, we suggest dynamically extracting mutation information in Telemetry attributes to reflect the failure symptoms and patterns. Compared with normal slow changes, mutations are rare, sudden, and unexpected changes. If we predict future attribute trend as the expected trend based on historical trend, the prediction error (i.e., the difference between predicted trend and actual trend) for mutations will be much larger than normal changes. Therefore, we adopt the prediction errors of attribute trends as the mutation features to represent the degree of mutations. For normal slow changes, the prediction errors will be very small and the feature values tend to be zero, so they are implicitly eliminated. For mutations, there will be larger prediction errors, and the feature values will also be larger and more significant. Meanwhile, since common mutations on healthy SSDs are less important, we also recommend estimating the rarity of mutations to reflect their importance.

3.2 Prediction Model

Background. There are mainly two types of machine learning models for failure prediction: classification models and anomaly detection models. Most previous schemes [1, 6, 17, 22, 38, 40] treat failure prediction as a binary classification problem and use classification models such as random forest to classify healthy and failed SSDs. The classification models learn the patterns of healthy and failed SSDs at the same time, and classify them by their differences, so these models are good at distinguishing patterns seen in historical training data.

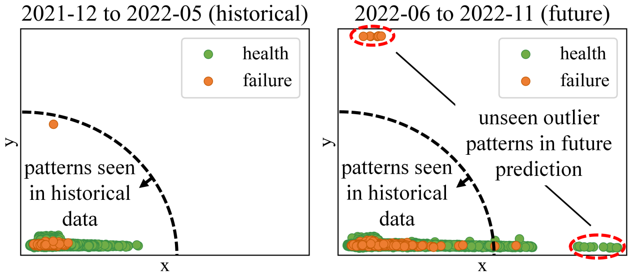


Figure 3: Data patterns over time. The same PCA is used to reduce the dimension of each SSD's monthly data into two dimensions (x and y) for visualization. It is a scenario that the historical six-month data are used for model training and the future six-month data for evaluation and online prediction.

However, compared to healthy SSDs, the number of failed SSDs is very small, and there are limited failure patterns seen in historical training data [4,42]. In dynamic production environment, unseen failure patterns will continuously appear, and the classification models have difficulty handling them [4].

To reflect the changes of data patterns in practical time-based training-evaluation-prediction situation, we adopt the widely used principle component analysis (PCA) method [24] to reduce the dimension of each SSD's monthly data to two dimensions (x and y), and thus they can be visualized in a coordinate system, similar to the previous work [4]. Figure 3 shows that some outlier data patterns not seen in historical data will appear in future prediction. To perceive these unseen patterns, given that there are far more healthy SSDs than failed SSDs, anomaly detection models (also called 1-class models) are applied to failure prediction [4,42]. The anomaly detection models learn the data patterns of healthy SSDs, and identify outlier patterns that are different from the health patterns as failure patterns. However, outlier patterns are not exactly equivalent to failure patterns. Figure 3 shows that both healthy and failed SSDs may have outlier patterns that are far away from the historical data patterns, and there are also some failed SSDs with non-outlier patterns.

Motivation and analysis. Based on the dynamic environment in practice, we need to distinguish patterns that have been seen in historical training data, and identify unseen patterns as well. Drawing on the idea of classification models, for the patterns seen in historical training data, it is important to match them with historical health patterns and failure patterns. Based on the idea of anomaly detection models, the outlier patterns should be detected to perceive unseen patterns. The remaining question is which of the unseen outlier patterns are actually failure patterns. Figure 3 shows that although these outlier patterns in the future are far away from the historical patterns, the outlier patterns of the failed SSDs are still closer to the historical failure patterns, and those of the healthy SSDs are closer to the historical health patterns. This tendency can be exploited to further distinguish outlier patterns.

Ideas. In addition to identifying the patterns that have been seen in historical training data, unseen patterns and their tendency also need to be perceived. We adopt the idea of similarity measurement that is applicable to both classification and anomaly detection. First, the patterns that have been seen in historical training data can be effectively distinguished in prediction by measuring their similarity with historical health and failure patterns. Second, the similarity measure can capture unseen patterns which tend to be outlier, and is helpful for estimating their health/failure tendency.

3.3 Failure Alarm and Handling

Background. The existing schemes predict failures and take measures (e.g., disk replacement) in advance to reduce the impact of failures on online services. When failures are reported, some healthy SSDs may also be wrongly reported as failed ones. To reduce false alarms, some schemes [4,22,40] suggest using scrub technology to perform a full scan on the alarmed SSD to confirm whether it is failed. A scrubber is a background process provided by commercial storage systems, RAID, or file systems, and it detects data integrity errors through full disk scanning [22].

Most previous schemes use a unified mechanism to alert and handle all failures. In practice, however, there are various failures [19,39] and the measures are also different (for example, the failure list includes the cases of replacing the SSD and not replacing the SSD). In particular, there are diverse error-tolerant mechanisms inside the SSD, which increase the reliability of the SSD, but also make it difficult to distinguish whether the SSD is really failed. Some error-tolerant mechanisms such as read retry may cause long tail latency in I/O or timeout error, thereby affecting service performance [25]. These SSDs are available but suffer performance degradation or transient error, and may also be reported in failure list. Some researchers call this event gray failure [14,39] or fail-slow [10,19]. Such failures cannot even be verified with scrub technology as they do not have data integrity issues. In general, coarse-grained failure prediction and handling cannot fully adapt to practical scenarios with various failures.

Motivation and analysis. SSD failures involve various phenomena and degrees, and it is meaningful to identify fine-grained failure status and take corresponding measures. Unavailable SSDs with serious failures should be predicted in advance and handled immediately via data migration, disk replacement, etc. However, for available SSDs without serious failures such as gray failures, we can further check their detailed status. If they work normally later or can be repaired, unnecessary overheads (e.g., data migration and SSD retirement) would be reduced.

The detailed status of SSD is related to the internal error tolerance mechanism and error occurrence [22]. Table 2 shows the rates of SSDs with different types of errors in healthy SSDs and failed SSDs, and the replacement propor-

Status	Rate in health	Rate in failure	Replacement proportion
With uncorrectable errors	0.03%	9.71%	94.12%
With correctable errors	44.74%	47.43%	65.06%
Without errors	55.23%	42.86%	46.67%

Table 2: The rates of SSDs with different error status in healthy SSDs and failed SSDs, and the proportion of failed SSDs that were actually replaced. (For example, 9.71% of failed SSDs have uncorrectable errors, and 94.12% of them have been replaced.)

tions in failed SSDs. SSD failures with uncorrectable errors are usually serious failures, so the replacement proportion is very high. We also found that some SSDs that did not report failures had uncorrectable errors. These errors may occur accidentally at non-critical addresses and there was no or unnoticed impact on the online services, so these SSDs were not reported as failed ones and we call them problematic health. For failed SSDs with correctable errors, the replacement rate is not so high, indicating that there are available SSDs such as those with gray failures. For SSDs without errors, they appear more often in healthy SSDs. In fact, since Telemetry data with error information may not be collected in time when failures occur, the actual rate of SSDs without errors among failed SSDs is lower than 42.86%. Internal errors, especially uncorrectable ones, reflect the health/failure level of the SSD.

Ideas. To distinguish different failures and their degrees, we recommend fine-grained rating for SSDs. Based on the above analysis, we define four levels from failure to health, namely serious failure, gray failure, problematic health, and perfect health. According to the occurrence of errors, failure list and replacement status, historical SSDs fall into these four levels (see Section 4.2 for more details). Combined with the idea of similarity measurement described in Section 3.2, we can identify the health/failure level of an SSD based on its similarity with the four levels of historical SSDs. Then, serious failures can be handled directly, while gray failures and problematic health are further diagnosed with corresponding measures (e.g., latency monitoring or scrub scanning).

4 Methodology

The architecture of mutation similarity based failure rating and diagnosis (MSFRD) scheme is shown in Figure 4. Based on our motivations and ideas, the MSFRD scheme is divided into the following three parts. 1) To characterize key failure symptoms, the mutations in attribute trends are dynamically extracted. As shown in Section 3.1, the mutation means that the data trend deviates significantly from expectations, so it is less predictable than the normal trend. Therefore, we predict the subsequent attribute trend, and calculate the error between predicted trend and actual trend to reflect the degree of mutation. Considering that failures and the corresponding mutations are generally rare, we also introduce self-learned weights in the prediction model to measure the rarity of mutations. 2) Based on the idea in Section 3.2, we adopt similarity measurement to capture both seen and unseen failure patterns. To identify the fine-grained failure status and degree, we divide historical SSDs into four levels from serious failure to perfect health, i.e., level 1 to level 4, based on the analysis in Section 3.3. For a new mutation of an SSD to be identified, the failure level of the SSD is estimated by measuring the similarity of this mutation to the mutations of failed and healthy SSDs of these four levels. 3) Measures are provided based on the failure level to reduce operation and maintenance overhead. Serious failures can be directly handled, and less serious failures would be gradually diagnosed and processed according to their similarity trend with historical SSDs, the development of mutations, and the inspection results of corresponding tools (i.e., scrubber and latency monitor). Next, we will introduce these three parts in detail.

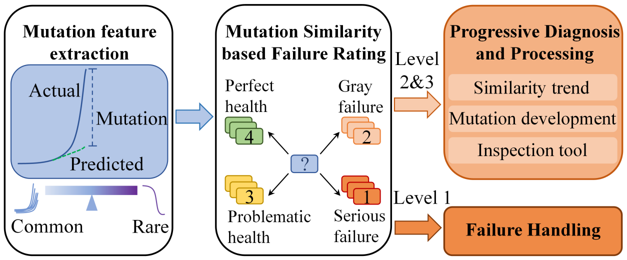


Figure 4: Overall architecture of MSFRD.

4.1 Mutation Feature Extraction

We recommend dynamic mutation feature extraction due to its two benefits. 1) By dynamically extracting important mutations from data in real time instead of static feature engineering, we can adapt to the changes of data importance over time. 2) By capturing the detailed mutation status of each Telemetry attribute of an SSD, we can focus on the abnormal changes that truly reflect the failure status and avoid the interference caused by normal changes.

Since mutations are rare, sudden, rapid changes that deviate more from expectations than normal slow trends, the errors between predicted trends (i.e., expectations) and actual trends are adopted as mutation features. We adopt Informer model [43] to predict data trends, since it is an accurate time series prediction model derived from the widely used Transformer model [31]. First, after data preprocessing by min-max normalization, we train an Informer model with the time-series data of large-scale historical healthy SSDs (i.e., healthy SSDs in the training set). In this way, the normal trends in healthy data are easily predicted by the model, while the mutations are still less predictable. Then, the trained Informer model is used to predict subsequent trend of each attribute of each SSD, and the prediction error between the predicted trend and the actual trend is used to characterize the mutation on each attribute (see Figure 4).

Specifically, based on the past time-series data $\{dn_{T-H},...,dn_{T-1},dn_T\}$ of the n-th attribute, we use the trained Informer model to predict the subsequent time-series data $\{pn_{T+1},...,pn_{T+2},pn_{T+F}\}$, and later calculate the difference between them and the actual time-series data $\{dn_{T+1},dn_{T+2},...,dn_{T+F}\}$. The prediction errors $\{dn_{T+1}-pn_{T+1},dn_{T+2}-pn_{T+2},...,dn_{T+F}-pn_{T+F}\}$ are used as the mutation feature MUT_n of the n-th attribute. The larger prediction error represents the larger mutation, and the prediction error for normal change tends to be small, because it is expected and thus correctly predicted.

In addition, since failed SSDs are pretty rare relative to healthy SSDs, the major internal mutations that lead to failures also tend to be rare. As shown in Section 3.1, the rarity of a mutation reflects its importance in identifying failures. Therefore, we add a fully connected layer on the last hidden layer of the Informer network, and use the outputs to estimate the rarity weights of possible mutations. The estimated rarity weights W are automatically learned during training by adding them to the loss function of the Informer model. Based on the original mean squared error (MSE) loss function (i.e., $\frac{1}{F\times N}\sum_{t=T+1}^{T+F}\sum_{n=1}^{N}(dn_t-pn_t)^2$), our loss function is designed as follows.

$$loss=\frac{1}{F\times N}\sum_{t=T+1}^{T+F}\sum_{n=1}^{N}\left((dn_t-pn_t)^2\times Wn+e^{-Wn}\right) \quad (1)$$

where Wn is the rarity weight of the mutation for the n-th attribute, and e^{-Wn} is designed as the penalty term to prevent Wn from approaching zero [3]. Since mutations are less predictable, squared error is usually larger when mutations occur. As analyzed in Section 3.1, common mutations of healthy SSDs (e.g., WAF mutations) are of little significance for failure prediction. Compared with rare mutations, common mutations and corresponding large squared errors appear frequently in training samples, so the corresponding Wn would decrease with the convergence of loss in training. At the same time, the penalty term e^{-Wn} is introduced to prevent the rarity weights from being all zero. In this way, self-learned Wn can reflect the rarity of the corresponding mutation, and the mutation with higher Wn is rarer and more important.

Through the above process, we obtained the mutation feature MUT_n of the n-th attribute represented by the prediction error, and the corresponding rarity weight Wn. They reflect the degree and importance of the mutation respectively. In this way, the noise introduced by normal changes is significantly reduced, and the key information of mutations is dynamically extracted to capture fine-grained failure symptoms in real time, without relying on the attribute importance and value range in historical training data.

4.2 Mutation Similarity based Failure Rating

Based on the extracted mutation features, a machine learning based prediction model is needed to identify the detailed status of SSDs. As analyzed in Section 3.2, the unseen failure patterns would appear in future prediction in practice, and existing classification models and anomaly detection models have their own strengths and weaknesses in catching seen and unseen failure patterns. We adopt the idea of similarity measurement (i.e., k-nearest neighbor [30, 41]) that exist in these two types of models to take their advantages. The patterns seen in historical data are distinguished based on the nearest historical neighbors, and the outlier degree is also considered to make the health/failure tendency of unseen patterns more apparent.

It is an option to only distinguish failed SSDs and healthy SSDs by the similarity measure of mutations. However, failed SSDs have different failure status and degrees, and available SSDs with less serious failures may not need to be replaced in practice. Therefore, failure rating is meaningful to reduce unnecessary maintenance overhead and additional impact on online services. Based on the ideas in Section 3.3, historical SSDs are automatically divided into four levels from failure (levels 1 and 2) to health (levels 3 and 4), which is reflected by SSDs' error, failure and replacement status. The four levels are defined as follows.

- Level 1: serious failure, such as unavailability or data integrity failure. The failed SSDs (i.e., those in failure list) that are replaced or have uncorrectable errors are in this level.
- Level 2: gray failure, such as performance degradation. The failed SSDs that are not replaced and have no uncorrectable errors are usually available SSDs with gray failures, so they are placed in this level.
- Level 3: problematic health, with self warning or data issues that have no or unnoticed impact. This level includes healthy SSDs (i.e., those not in failure list) with critical warnings [8] or uncorrectable errors.
- Level 4: perfect health, without any issues. The healthy SSDs without critical warnings and uncorrectable errors are in this level.

Figure 5 shows our overall process of failure rating based on the mutation similarity with historical SSDs. For a new mutation of an SSD (the mutation feature of the n-th attribute is $MUTnew_n$), we measure its similarity to all mutations of historical SSDs at four levels. Rare mutations are more important (see Section 3.1), so they should make a greater contribution in similarity measurement. To this end, we adopt weighted Euclidean distance (a widely used method in similarity measurement) [23], and use normalized rarity weights to amplify

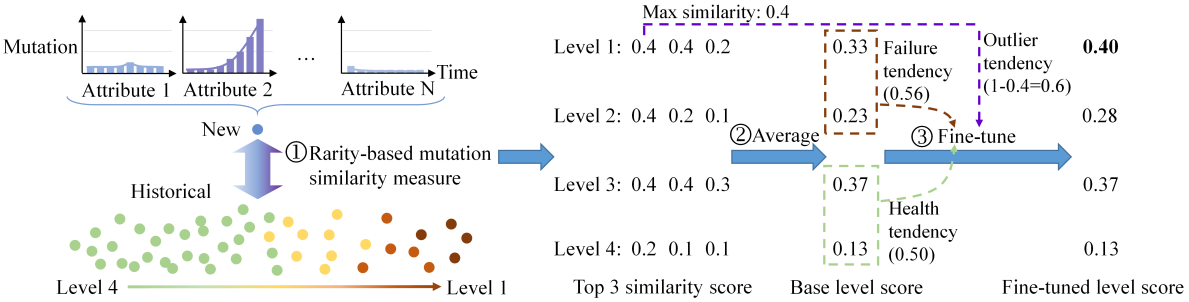


Figure 5: Overall process of mutation similarity based failure rating. ① Similarity measure between a new mutation of an SSD and all historical SSD mutations of 4 levels. ② Averaging top 3 similarity scores of each level as base level scores. ③ Fine-tuning level scores with new mutation's health, failure, and outlier tendencies.

the proportion of rare attribute mutations in distance calculation. In all J mutations of historical SSDs, assuming that the mutation feature of the *n-th* attribute of the *j-th* mutation is $MUTj_n$, we use the following formula to measure the distance between the new mutation and the *j-th* historical mutation.

$$Distance_j = \sum_{n=1}^{N} (\|MUTnew_n - MUTj_n\|_2 \times W_n) \tag{2}$$

where Wn is the rarity weight introduced in Section 4.1. A larger Wn means that the mutation on this attribute is rarer and more important, and it would occupy a larger proportion in the distance calculation. Then, $Distance_j$ is transformed to the similarity score $Similarity_j$ in the range of 0-1 by the following common formula:

$$Similarity_j = \frac{1}{1 + Distance_j} \tag{3}$$

Through the above formula, we can obtain the similarities between the new mutation and all historical mutations of the four levels, and each level takes the top k (3 by default) similarity scores for subsequent process. The new mutation is more likely to be at the level with high similarity scores. Therefore, for *i-th* level, the average of the top k similarity scores is regarded as the base confidence score $Base_i$.

In addition to the base confidence, which level the new mutation falls into also depends on whether it is more like a mutation of failure, a mutation of health, or an unseen mutation. Therefore, we introduce failure tendency (FT), health tendency (HT), and outlier tendency (OT) to fine-tune the base confidence scores. FT is designed as the sum of base confidence scores of levels 1 and 2, reflecting the confidence that the new mutation belongs to a failure. Meanwhile, HT is designed as the sum of base confidence scores of levels 3 and 4, reflecting the confidence that the new mutation belongs to a healthy SSD. In addition, if the new mutation is far away from all historical mutations, the outlier tendency OT should be larger. When the global maximum similarity score is $Similarity_{max}$, OT is defined as $1 - Similarity_{max}$, which reflects how far the new mutation is from historical mutations.

Based on the obtained FT, HT, and OT, the base confidence scores are fine-tuned. The ratio of the failure tendency to the healthy tendency ($\frac{FT}{HT}$) is adopted to represent the comprehensive health/failure tendency of the new mutation. If the ratio is greater than 1, the new mutation is more similar to the mutations in historical failed SSDs relative to healthy SSDs. Otherwise, it is less similar to the mutations in failed SSDs. Therefore, this ratio is used to adjust the base confidence scores for failure-related levels (i.e., levels 1-2). At the same time, since the unseen pattern with large OT is far away from historical mutations and its health/failure tendency is not obvious, OT is used to increase its health/failure tendency (i.e., $\frac{FT}{HT}$). The base confidence score $Base_i$ of the *i-th* level is adjusted as follows:

$$Level_i = \begin{cases} Base_i \times (\frac{FT}{HT})^{OT+1} & i = 1 \text{ or } 2 \\ Base_i & i = 3 \text{ or } 4 \end{cases} \tag{4}$$

where $Level_i$ is the fine-tuned confidence score of the *i-th* level. It is then normalized through dividing it by the sum of fine-tuned confidence scores of all levels. The new mutation belongs to the level with the highest confidence score. When the new mutation is similar to the mutations of historical failures, its score of levels 1-2 will be larger after fine-tuning with its health/failure tendency (i.e., $\frac{FT}{HT}$). When the new mutation is far away from all historical mutations but slightly inclined to the mutations of failures, it would also belong to level 1 or 2 after fine-tuning with $\frac{FT}{HT}$ amplified by the large OT. In this way, both seen failure patterns with similar mutations in historical data and unseen failure patterns with outlier mutations would be captured.

4.3 Progressive Diagnosis and Processing

After failure rating, the SSDs judged to be level 1 are about to face a serious failure and need to be replaced immediately, while the SSDs of levels 2 and 3 which would face gray failures or problems require tracking and diagnosis to confirm their impact and future status. For these SSDs, automatic diagnosis is first performed based on their subsequent similarity trend and mutation development. Based on the mutation similarity, if the SSD's level-1 score increases further, it is more likely to face serious failure. Moreover, mutations represent abnormal changes that may lead to failure. Therefore, we focus on the attribute with the largest mutation, which reflects the biggest anomaly inside the SSD. If the attribute value with the largest mutation further increases or decreases in the direction of mutation, it means that the mutation is still ongoing and the issue is getting more serious. Based on these two points, for an SSD rated at level 2 or 3, when the confidence score of level 1 further increases and the attribute value with the largest mutation continuously changes, it is further diagnosed as level 1 and needs to be processed immediately. In this way, we can catch the issues that may develop into serious failures and reduce unnecessary handling for SSDs that are no longer bad.

In addition to automatic trend-based diagnosis above, we also recommend using some tools for diagnosis and treatment. Most SSDs rated at level 2 are available but may face performance issues. Therefore, the impact of SSD performance issues on online services needs to be evaluated to decide what measures to take. Since I/O latencies reflect the quality of services, and many mainstream SSDs for data centers (e.g., Samsung PM9A3) support device-side latency tracing, monitoring latency is a good choice. By continuously monitoring the occurrence of SSD's long-tail latency, we can find SSD performance degradation in time and confirm whether it is acceptable or needs to be fixed. For SSDs rated at level 3, the main problem is data integrity, and the scrub technology with full scan is helpful for monitoring data integrity errors. In particular, the addresses of data integrity errors can be found to confirm their impact. If these errors occur accidentally at non-critical addresses and will not occur again, they may not need to be handled. Otherwise, data migration and SSD replacement are necessary. By using relevant tools to diagnose SSDs at different levels, we can clarify the fine-grained SSD status and suggest corresponding measures.

5 Evaluation

The effectiveness of the proposed MSFRD was evaluated based on the real data collected from large-scale SSDs of data centers. We first compared MSFRD with existing schemes on failure prediction in Section 5.1. The prediction is required to be within one month before failure report, and for MSFRD, early ratings of levels 1 and 2 are considered true alarms. The evaluation is performed on three datasets. In addition to our 41-M and 10-M PM9A3 Telemetry datasets introduced in Section 2, the MB2 SMART dataset publicly available from Alibaba [38] is also used in this evaluation. The MB2 SSD model only has 14 SMART attributes and its Multi-Level Cell (MLC) flash technology is different from Triple-Level Cell (TLC) based PM9A3. On the three datasets of SSDs with different monitoring attributes, flash technologies, and users, the failure prediction performance and generalizability of the schemes can be fully evaluated.

Dataset	Exp. round	Train set (month)	Val set (month)	Test set (month)
41-M Telemetry	1	1–10th	11th	12–14th
	2	1–13th	14th	15–17th
10-M Telemetry	1	1–3th	4th	5th
	2	1–4th	5th	6th
	3	1–5th	6th	7th
MB2 SMART	1	1–17th	18th	19–21th
	2	1–20th	21th	22–24th

Table 3: Data partitions on three datasets.

Similar to the previous work [38], we divide each dataset into training set, validation set and test set in chronological order respectively. The training set is used to build the model, the validation set to fine-tune the model's hyper-parameters, and the test set to evaluate the model. We conduct independent experiments on two or three different data partitions for each of the three datasets, as shown in Table 3. The data partitioning is based on the real situations, i.e., the prediction models are trained on the historical data and then used to predict SSD failures with future data. The multiple data partitions also simulate real scenarios. Although the proposed scheme has the ability to cope with data changes and unseen failures, the online model is still updated every few months to further adapt to data changes through iterative training and verification on the latest data. For each dataset, since there are two or three independent experiments, their average results are deemed as the final evaluation results.

Besides failure prediction evaluation, the failure rating accuracy, the effect of each module in MSFRD, the model transferability, and the real failure rating examples are also discussed in Section 5.2, 5.3, 5.4, and 5.5 respectively. Then we shall introduce the evaluation metrics used in this paper. Precision, recall, and F0.5-Score are adopted to evaluate the accuracy of failure prediction.

Precision: the proportion of true alarms (i.e., correctly predicted failed SSDs) to both true alarms and false alarms.

Recall: the proportion of true alarms to all failed SSDs.

$F0.5$-*score*: $\frac{(1+0.5^2)\times Precision\times Recall}{0.5^2\times Precision+Recall}$. It is obtained by combining precision and recall for a more comprehensive evaluation. In the production environment, precision tends to be more important to avoid too many false alarms [38], so it has a larger weight in the calculation of F0.5-Score.

In addition, we use accuracy rate (the proportion of true alarms with correctly identified failure levels to all true alarms) to evaluate failure rating performance in Section 5.2.

Methods	**41-M Telemetry**			**10-M Telemetry**			**MB2 SMART**			**Average**		
	Precision	Recall	F0.5	Precision	Recall	F0.5	Precision	Recall	F0.5	Precision	Recall	F0.5
RF [38]	0.61	0.19	0.43	0.64	0.18	0.38	0.72	0.24	0.52	0.66	0.20	0.44
EC [6]	0.59	0.24	0.44	0.63	0.21	0.44	0.85	0.24	0.57	0.69	0.23	0.48
AE [4]	0.57	0.26	0.46	0.61	0.23	0.46	0.53	0.25	0.43	0.57	0.25	0.45
MVT-RF [40]	0.62	0.28	0.50	0.70	0.27	0.52	**0.87**	0.25	0.58	0.73	0.27	0.53
MSFRD(Ours)	**0.72**	**0.37**	**0.61**	**0.87**	**0.34**	**0.66**	**0.87**	**0.27**	**0.60**	**0.82**	**0.33**	**0.62**

Table 4: The evaluation of failure prediction on three datasets.

5.1 Failure Prediction

For failure prediction, the proposed MSFRD is compared with existing methods: Random Forest [38], Ensemble Classifier [6], Autoencoder [4], and Multi-view and Multi-task Random Forest [40]. Their detailed descriptions are as follows. 1) Random Forest (RF): A combined feature selection method is adopted to select important attributes and the corresponding data are input into the trained random forest model for failure/health classification. 2) Ensemble Classifier (EC): SMART/Telemetry data after feature selection are input into multiple classification models (e.g., random forest and gradient boosted decision tree), and the outputs are combined to get the final failure/health result. 3) Autoencoder (AE): SMART/Telemetry data after feature selection are reconstructed by a neural network-based encoder and decoder, and failures are predicted according to the reconstruction loss. It is an anomaly detection algorithm, and the key idea is that the data of failed SSDs are abnormal and difficult to reconstruct. 4) Multi-view and Multi-task Random Forest (MVT-RF): The multi-view time-series related features are extracted from raw SMART/Telemetry data, and then input into multiple random forests to vote for the failures.

These methods' separate and average results on the three datasets are shown in Table 4. RF and EC, as classification methods, lack the perception of unseen failure patterns, and thus obtain relatively low recall. AE predicts SSD failures by detecting outlier patterns, which helps to discover unseen failure patterns, so it improves recall to 0.25 on average. However, outlier patterns are not exactly equivalent to failure patterns, so the average precision of AE is only 0.57. Besides, these methods use static feature selection to find important attributes and data based on historical training data, which cannot fully adapt to changes of data importance over time in practice, so the overall performance is not good with low F0.5-Score. MVT-RF performs better than RF, AE, and EC, and obtains an average F0.5-Score of 0.53, since it extracts multi-view time-series related features which can capture the failure symptoms in data trends.

Compared with the average result of three datasets of four existing methods, the proposed scheme improves precision, recall, and F0.5-Score by 23.8%, 38.9%, and 30.5%, respectively. MSFRD dynamically extracts mutation features to locate abnormal changing trends in real time and capture failure symptoms more accurately, thereby achieving 0.82 in average precision. Furthermore, MSFRD captures failure patterns seen in training set through similarity measurement, and introduces outlier tendency to perceive unseen failure patterns. Therefore, it predicts SSD failures more comprehensively and improves the recall to 0.33 on average. The average F0.5-Score of 0.62 further demonstrates that our scheme can accurately predict more failed SSDs.

Table 4 shows that MSFRD outperforms the existing methods on almost all metrics of three datasets. These datasets were collected from different companies with different monitoring attributes, workloads, and collection periods, and thus the results on them show the effectiveness and generalizability of MSFRD. In addition, compared with existing methods, MSFRD shows a larger performance improvement on Telemetry datasets relative to SMART dataset. The MB2 SMART dataset only has 14 attributes, while Telemetry datasets have 85 attributes. So many Telemetry attributes enhance the monitoring of various components and mechanisms in SSD, but are not friendly for capturing key failure-related information. The attribute importance and value ranges change over time in practice, which brings more troubles to static feature engineering. The proposed MSFRD dynamically captures mutations and estimates their rarity, which accurately locates key attributes and failure-related changes in real time and reduces noise caused by meaningless changes, thus working well on the Telemetry datasets.

5.2 Failure Rating

To reduce the impact on available SSDs without serious failures, the proposed MSFRD grades the degree of failures and tracks the status of available SSDs through progressive diagnosis. In this section, the effectiveness of our failure rating and progressive diagnosis is evaluated. RF, EC, and MVT-RF can also perform the classification of four levels of SSDs, and are compared with our scheme together based on the same level definitions. Figure 6 shows the accuracy of our scheme and existing schemes on the ratings of their true alarms on the 41-M dataset. For failure rating, EC outperforms RF and MVT-RF. EC combines the results of various machine learning methods, which would be more accurate and robust than the single method.

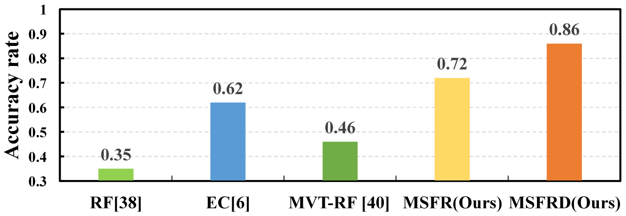


Figure 6: The evaluation on failure rating.

Figure 6 shows that our scheme has higher accuracy compared to the existing methods. The mutation similarity based failure rating (MSFR) rates an SSD by its mutation similarity to historical mutations of four levels, and introduces health, failure, and outlier tendencies in similarity measure to further fine-tune the ratings. It obtains an accuracy rate of 0.72 in failure rating. Coupled with automatic trend-based diagnosis (i.e., MSFRD), the accuracy rate finally reaches 0.86 which is 38.7% higher than EC. Progressive diagnosis is able to track available SSDs at levels 2 and 3, and adjust the rating to level 1 when they go bad. In conclusion, through failure rating and progressive diagnosis, MSFRD effectively distinguishes the detailed health/failure status of SSDs to provide accurate recommendations.

5.3 Discussion on MSFRD Modules

In this section, we shall discuss the effectiveness of each module or process in the proposed scheme. The following seven methods are compared on the 41-M dataset, from the baseline random forest method, to using partial MSFRD and finally the whole MSFRD. 1) Raw data + RF: The raw data are input into the trained random forest model for failure/health classification. 2) Feature selection + RF: The data after static feature selection are input into the trained random forest model for failure/health classification. It is the same as RF in Section 5.1. 3) Mutation feature + RF: The dynamic mutation features are input into the trained random forest model for failure/health classification. 4) Mutation feature + similarity based failure rating (SFR): Based on the mutation feature, failed SSDs and their levels are identified based on their similarities to historical failed and healthy SSDs of four levels. 5) Mutation feature with rarity (Mutation(rarity)) + SFR: the difference from mutation feature + SFR is that the self-learned rarity weights (Equation 1 in Section 4.1) are used in similarity measure (Equation 2 in Section 4.2). 6) Mutation(rarity) + SFR with fine-tuned score (SFR(tuned)): the difference from Mutation(rarity) + SFR is that the fine-tuned confidence score (Equation 4 of Section 4.2) is used instead of base confidence score for failure prediction and rating. 7) MSFRD: coupled with automatic diagnosis (see Section 4.3) on Mutation(rarity) + SFR(tuned), the whole MSFRD is used to predict failures.

Methods	Precision	Recall	F0.5
Raw data+RF	0.55	0.21	0.41
Feature selection+RF	0.61	0.19	0.43
Mutation feature+RF	0.70	0.24	0.51
Mutation feature+SFR	0.66	0.27	0.52
Mutation(rarity)+SFR	0.67	0.30	0.54
Mutation(rarity)+SFR(tuned)	0.69	0.36	0.58
MSFRD	**0.72**	**0.37**	**0.61**

Table 5: Comparison of MSFRD modules.

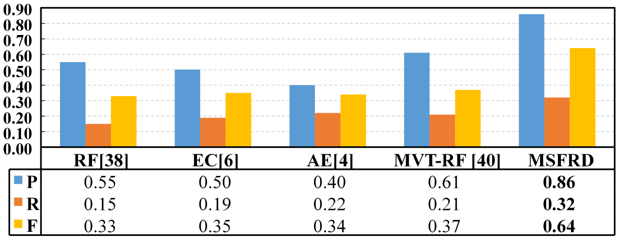
Table 5 shows the results of these seven methods. Static feature selection selects failure-relevant attributes based on historical training data, which reduces the noise introduced by failure-irrelevant attributes, and thus Feature selection + RF has higher precision than Raw data + RF. However, some removed failure-irrelevant attributes may become failure-relevant in the future, making it difficult to identify the corresponding failures and resulting in a decrease in recall. Instead of static feature selection, we extracted dynamic mutation feature to capture the failure symptoms in real time, therefore mutation feature + RF performs better than Feature selection + RF and improves precision to 0.70 and recall to 0.24.

Mutation feature + SFR adopts similarity measure instead of the RF method. It is based on the traditional k-nearest neighbor classification idea, so it obtains similar overall performance (F0.5-Score) with the RF classification method. Mutation(rarity) + SFR introduces the rarity weights of mutations in similarity measurement and focuses on more rare and important mutations, thus achieving 0.67 in precision and 0.30 in recall. Mutation(rarity) + SFR can easily find failure patterns that have been seen in historical data, but it is difficult to identify ambiguous mutations, such as mutations of unseen failures. Mutation(rarity) + SFR(tuned) adopts health, failure and outlier tendencies to fine-tune the failure confidence scores of new mutations. For ambiguous mutations that are slightly inclined to the mutations of failures, especially the mutations of unseen failures with higher outlier tendency, the failure confidences are significantly improved after fine-tuning, and thus Mutation(rarity) + SFR(tuned) greatly improves recall to 0.36. Finally, through further diagnosis, some SSDs that go bad are further identified, and the whole MSFRD scheme achieves 0.61 in F0.5-Score.

5.4 Model Transferability

Due to the differences in workload, server, and SSD firmware, there are some differences in Telemetry data from different companies or data centers, and this poses a challenge to the transferability of the prediction model. In this section, we use the models trained on the 41-M dataset to perform failure prediction directly on the 10-M dataset (different company and different SSD firmware version) to evaluate their transferability and generalizability.

Figure 7 shows the performance of MSFRD and existing schemes in this case. RF, EC and AE use static feature selec-



	RF[38]	EC[6]	AE[4]	MVT-RF [40]	MSFRD
P	0.55	0.50	0.40	0.61	**0.86**
R	0.15	0.19	0.22	0.21	**0.32**
F	0.33	0.35	0.34	0.37	**0.64**

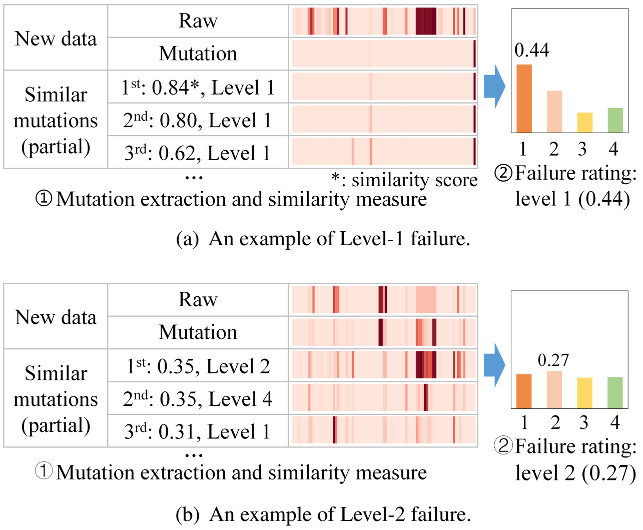
Figure 7: Model transferability from the 41-M dataset to the 10-M dataset. (P: precision; R: recall; F: F0.5-Score)

tion to select important attributes. However, some attributes that are important on the 41-M dataset may not be important on the 10-M dataset, which introduces noisy data and results in lower precision. In particular, AE treats outlier patterns as failure patterns. However, with the data differences caused by dataset migration, there would be more outlier patterns on the 10-M dataset based on the patterns in the 41-M dataset. Therefore, the precision of AE is the lowest (only 0.40). MVT-RF pays more attention to data trends which are also susceptible to the data differences between two datasets, so it does not work well too and the F0.5-Score is 0.37.

The proposed MSFRD outperforms existing schemes, achieving 0.86 in precision, 0.32 in recall and 0.64 in F0.5-Score, which is very similar to the results shown in Section 5.1 for both training and testing on the 10-M dataset. MSFRD dynamically extracts key failure symptoms by capturing mutations, reducing the impact of data differences between two datasets. Besides, for patterns with large differences after dataset migration, MSFRD can estimate their health/failure tendency through similarity measurement. Based on these mechanisms, MSFRD maintains high prediction accuracy after dataset migration.

5.5 Practical Examples

The MSFRD has been applied for online prediction, and Figure 8 shows the visual results of two true alarms. Figure 8(a) shows an example where a level-1 failure is correctly predicted and rated. With dynamic mutation extraction, only a few attributes of this SSD have mutations (with deep color), while other attributes that are changing normally in the raw data are implicitly eliminated. In this way, the key information is captured. After the similarity measure in mutations, this example is mainly similar with level-1 failures and the similarity scores are high, so it is a pattern seen in historical data and is definitely rated level 1. Figure 8(b) shows an example that has not been seen in historical data and the similarity scores are only about 0.3. Its health/failure tendency is not obvious, but its outlier tendency is large. MSFRD uses the outlier tendency to amplify its slight failure tendency, thereby increasing the confidence scores of levels 1 and 2, and this SSD is finally correctly rated at level 2. Through the visualization of MSFRD output, operators can clearly understand the mutation status of SSD, the information of historical neighbors, and the failure level and confidence, which are useful for failure handling.


New data
Raw
Mutation
Similar mutations (partial)
1st: 0.84*, Level 1
2nd: 0.80, Level 1
3rd: 0.62, Level 1
…
*: similarity score
① Mutation extraction and similarity measure
0.44
1 2 3 4
② Failure rating: level 1 (0.44)

(a) An example of Level-1 failure.

New data
Raw
Mutation
Similar mutations (partial)
1st: 0.35, Level 2
2nd: 0.35, Level 4
3rd: 0.31, Level 1
…
① Mutation extraction and similarity measure
0.27
1 2 3 4
② Failure rating: level 2 (0.27)

(b) An example of Level-2 failure.

Figure 8: The correct failure rating examples of MSFRD. The heat map shows the values of N attributes of corresponding data at a moment.

6 Related Work

Numerous studies have investigated and analyzed the impact of disk errors and failures on large data centers [11, 12, 27, 28, 32, 36, 37]. To take proactive measures before failures occur, disk failure prediction has received extensive attention. There are many studies on HDD failure prediction, since HDDs have been widely used for a long time. Most researchers use machine learning to predict HDD failures [5, 7, 16, 20, 29, 35, 44], as machine learning is more accurate and flexible. Some of them have achieved high accuracy for HDD failure prediction. However, since flash-based SSDs have different storage technologies and more error-tolerant mechanisms, these previous results do not apply to SSDs [1, 4].

In recent years, with the large-scale use of SSDs in data centers, SSD failure prediction has received increasing attention. Most of existing studies [1, 13, 17, 22, 38, 40] focus on predicting failures with classification algorithms. Based on random forest classifier, Xu et al. [38] explored the impact of different feature selection methods on prediction results. They also combined the rankings of multiple feature selection methods to obtain the mean ranking for better performance. Chen et al. [6] trained multiple decision tree-based classification models with different hyper-parameters and boosting methods, and combined the outputs of different models to obtain the final prediction results. Zhang et al. [40] extracted time-series related features to capture the failure symptoms in long-term data trends, and achieved better prediction accuracy. Besides failure prediction, they also introduced failure type classification and lifespan estimation. It should be pointed out that the failure type definitions are not completely objective, and there are some differences in the definitions from different companies. The failure ratings in this paper are based on more objective SSD error, failure, and replacement status, so they are more universal.

In addition to classification algorithms, Chandranil et al. [4] adopted anomaly detection algorithms to capture failed SSDs which are considered far away from most healthy SSDs, and thus they can perceive unseen outlier patterns. They compared the isolation forest and autoencoder algorithms, and autoencoder has higher accuracy for failure prediction.

7 Conclusion

In this paper, we propose MSFRD for failure prediction, early rating and progressive diagnosis. MSFRD first dynamically detects data mutations through the prediction error of time-series Telemetry data and estimates the importance of mutations. Then, failed SSDs are predicted and rated based on mutation similarity with historical failed SSDs and healthy SSDs. Finally, failures are handled incrementally to minimize the impact on available SSDs without serious failures. The evaluations on multiple real datasets show that MSFRD significantly improves the accuracy of failure prediction, rating and handling.

References

[1] Jacob Alter, Ji Xue, Alma Dimnaku, and Evgenia Smirni. SSD Failures in the Field: Symptoms, Causes, and Prediction Models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '19, New York, NY, USA, 2019. Association for Computing Machinery. https://doi.org/10.1145/3295500.3356172.

[2] Agustin Garcia Asuero, Ana Sayago, and AG González. The correlation coefficient: An overview. *Critical reviews in analytical chemistry*, 36(1):41–59, 2006. https://doi.org/10.1080/10408340500526766.

[3] Kurt Bryan and Yosi Shibberu. Penalty functions and constrained optimization. *Dept. of Mathematics, Rose-Hulman Institute of Technology.*, 2005. https://www.rose-hulman.edu/~bryan/lottamath/penalty.pdf.

[4] Chandranil Chakraborttii and Heiner Litz. Improving the Accuracy, Adaptability, and Interpretability of SSD Failure Prediction Models. In *Proceedings of the 11th ACM Symposium on Cloud Computing*, SoCC '20, page 120–133, New York, NY, USA, 2020. Association for Computing Machinery. https://doi.org/10.1145/3419111.3421300.

[5] Iago C. Chaves, Manoel Rui P. de Paula, Lucas G.M. Leite, Lucas P. Queiroz, Joao Paulo P. Gomes, and Javam C. Machado. BaNHFaP: A Bayesian Network Based Failure Prediction Approach for Hard Disk Drives. In *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 427–432, Oct 2016. https://doi.org/10.1109/BRACIS.2016.083.

[6] Lei Chen, Zongpeng Zhu, Anyu Li, Najmeh Mashhadi, Robert Frickey, Jinhe Ye, and Xin Guo. SSD Drive Failure Prediction on Alibaba Data Center Using Machine Learning. In *2022 IEEE International Memory Workshop (IMW)*, pages 1–4, 2022. https://doi.org/10.1109/IMW52921.2022.9779284.

[7] Yan Ding, Yunan Zhai, Yujuan Zhai, and Jia Zhao. Explore deep auto-coder and big data learning to hard drive failure prediction: a two-level semi-supervised model. *Connect. Sci.*, 34(1):449–471, 2022. https://doi.org/10.1080/09540091.2021.2008320.

[8] NVM Express. SMART/Health Information. In *NVMe Base Specification, Revision 1.4c*, page 120–121, 2021. https://nvmexpress.org/wp-content/uploads/NVM-Express-1_4c-2021.06.28-Ratified.pdf.

[9] Ronen Fluss, David Faraggi, and Benjamin Reiser. Estimation of the Youden Index and its associated cutoff point. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 47(4):458–472, 2005. https://doi.org/10.1002/bimj.200410135.

[10] Haryadi S. Gunawi, Riza O. Suminto, Russell Sears, Casey Golliher, Swaminathan Sundararaman, Xing Lin, Tim Emami, Weiguang Sheng, Nematollah Bidokhti, Caitie McCaffrey, Gary Grider, Parks M. Fields, Kevin Harms, Robert B. Ross, Andree Jacobson, Robert Ricci, Kirk Webb, Peter Alvaro, H. Birali Runesha, Mingzhe Hao, and Huaicheng Li. Fail-Slow at Scale: Evidence of Hardware Performance Faults in Large Production Systems. In *16th USENIX Conference on File and Storage Technologies (FAST 18)*, pages 1–14, Oakland, CA, February 2018. USENIX Association. https://www.usenix.org/conference/fast18/presentation/gunawi.

[11] Chuanxiong Guo, Lihua Yuan, Dong Xiang, Yingnong Dang, Ray Huang, Dave Maltz, Zhaoyi Liu, Vin Wang, Bin Pang, Hua Chen, Zhi-Wei Lin, and Varugis Kurien. Pingmesh: A Large-Scale System for Data Center Network Latency Measurement and Analysis. In *Proceedings of the 2015 ACM Conference on Special Interest*

Group on Data Communication, SIGCOMM '15, page 139–152, New York, NY, USA, 2015. Association for Computing Machinery. https://doi.org/10.1145/2785956.2787496.

[12] Shujie Han, Patrick P. C. Lee, Fan Xu, Yi Liu, Cheng He, and Jiongzhou Liu. An In-Depth Study of Correlated Failures in Production SSD-Based Data Centers. In *19th USENIX Conference on File and Storage Technologies (FAST 21)*, pages 417–429. USENIX Association, February 2021. https://www.usenix.org/conference/fast21/presentation/han.

[13] Wenwen Hao, Ben Niu, Yin Luo, Kangkang Liu, and Na Liu. Improving accuracy and adaptability of SSD failure prediction in hyper-scale data centers. *SIGMETRICS Perform. Eval. Rev.*, 49(4):99–104, jun 2022. https://doi.org/10.1145/3543146.3543169.

[14] Peng Huang, Chuanxiong Guo, Lidong Zhou, Jacob R. Lorch, Yingnong Dang, Murali Chintalapati, and Randolph Yao. Gray Failure: The Achilles' Heel of Cloud-Scale Systems. In *HotOS '17*, page 150–155, New York, NY, USA, 2017. Association for Computing Machinery. https://doi.org/10.1145/3102980.3103005.

[15] Bryan S Kim, Jongmoo Choi, and Sang Lyul Min. Design Tradeoffs for SSD Reliability. In *FAST*, pages 281–294, 2019. https://www.usenix.org/conference/fast19/presentation/kim-bryan.

[16] Jing Li, Xinpu Ji, Yuhan Jia, Bingpeng Zhu, Gang Wang, Zhongwei Li, and Xiaoguang Liu. Hard Drive Failure Prediction Using Classification and Regression Trees. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 383–394, June 2014. https://doi.org/10.1109/DSN.2014.44.

[17] Peng Li, Wei Dang, Congmin Lyu, Min Xie, Quanyang Bao, Xiaofeng Ji, and Jianhua Zhou. Reliability Characterization and Failure Prediction of 3D TLC SSDs in Large-Scale Storage Systems. *IEEE Transactions on Device and Materials Reliability*, 21(2):224–235, June 2021. https://doi.org/10.1109/TDMR.2021.3063164.

[18] Chun-Yi Liu, Yunju Lee, Myoungsoo Jung, Mahmut Taylan Kandemir, and Wonil Choi. Prolonging 3D NAND SSD lifetime via read latency relaxation. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 730–742, 2021. https://dl.acm.org/doi/10.1145/3445814.3446733.

[19] Ruiming Lu, Erci Xu, Yiming Zhang, Zhaosheng Zhu, Mengtian Wang, Zongpeng Zhu, Guangtao Xue, Minglu Li, and Jiesheng Wu. NVMe SSD Failures in the Field: the Fail-Stop and the Fail-Slow. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, pages 1005–1020, Carlsbad, CA, July 2022. USENIX Association. https://www.usenix.org/conference/atc22/presentation/lu.

[20] Sidi Lu, Bing Luo, Tirthak Patel, Yongtao Yao, Devesh Tiwari, and Weisong Shi. Making Disk Failure Predictions SMARTer! In *18th USENIX Conference on File and Storage Technologies (FAST 20)*, pages 151–167, Santa Clara, CA, February 2020. USENIX Association. https://www.usenix.org/conference/fast20/presentation/lu.

[21] Yixin Luo, Saugata Ghose, Yu Cai, Erich F. Haratsch, and Onur Mutlu. Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(3), dec 2018. https://doi.org/10.1145/3224432.

[22] Farzaneh Mahdisoltani, Ioan Stefanovici, and Bianca Schroeder. Proactive error prediction to improve storage system reliability. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, pages 391–402, Santa Clara, CA, July 2017. USENIX Association. https://www.usenix.org/conference/atc17/technical-sessions/presentation/mahdisoltani.

[23] José M Merigó and Anna M Gil-Lafuente. On the use of the OWA operator in the Euclidean distance. *International Journal of Computer Science and Engineering*, 2(4):170–176, 2008. https://www.researchgate.net/publication/242681011_On_the_Use_of_the_OWA_Operator_in_the_Euclidean_Distance.

[24] Wolfgang Müller, Nocke Thomas, and Schumann Heidrun. Enhancing the visualization process with principal component analysis to support the exploration of trends. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation*, pages 121–130, 2006. https://dl.acm.org/doi/pdf/10.5555/1151903.1151922.

[25] Jisung Park, Myungsuk Kim, Myoungjun Chun, Lois Orosa, Jihong Kim, and Onur Mutlu. Reducing Solid-State Drive Read Latency by Optimizing Read-Retry. In *ASPLOS '21*, page 702–716, New York, NY, USA, 2021. Association for Computing Machinery. https://doi.org/10.1145/3445814.3446719.

[26] Stefan Savage and John Wilkes. AFRAID–A Frequently Redundant Array of Independent Disks. In *USENIX 1996 Annual Technical Conference (USENIX ATC 96)*, San Diego, CA, January 1996. USENIX Association. https://www.usenix.org/conference/

usenix-1996-annual-technical-conference/afraid-frequently-redundant-array-independent.

[27] Bianca Schroeder, Raghav Lagisetty, and Arif Merchant. Flash Reliability in Production: The Expected and the Unexpected. In *14th USENIX Conference on File and Storage Technologies (FAST 16)*, pages 67–80, Santa Clara, CA, February 2016. USENIX Association. https://www.usenix.org/conference/fast16/technical-sessions/presentation/schroeder.

[28] Bianca Schroeder, Arif Merchant, and Raghav Lagisetty. Reliability of nand-Based SSDs: What Field Studies Tell Us. *Proceedings of the IEEE*, 105(9):1751–1769, Sep. 2017. https://doi.org/10.1109/JPROC.2017.2735969.

[29] Jing Shen, Yongjian Ren, Jian Wan, and Yunlong Lan. Hard Disk Drive Failure Prediction for Mobile Edge Computing Based on an LSTM Recurrent Neural Network. *Mobile Information Systems*, 2021:1–12, feb 2021. https://doi.org/10.1155/2021/8878364.

[30] Kashvi Taunk, Sanjukta De, Srishti Verma, and Aleena Swetapadma. A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260, 2019. https://doi.org/10.1109/ICCS45141.2019.9065747.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[32] Guosai Wang, Lifei Zhang, and Wei 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 (DSN)*, pages 25–36, June 2017. https://doi.org/10.1109/DSN.2017.26.

[33] Yang Wang, Lorenzo Alvisi, and Mike Dahlin. Gnothi: Separating Data and Metadata for Efficient and Available Storage Replication. In *2012 USENIX Annual Technical Conference (USENIX ATC 12)*, pages 413–424, Boston, MA, June 2012. USENIX Association. https://www.usenix.org/conference/atc12/technical-sessions/presentation/wang.

[34] Ziyao Wang and Jie Xu. SSD Failure Prediction Based on Classification Models and Data Engineering. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress*, pages 1–8, 2022. https://ieeexplore.ieee.org/document/9927939.

[35] Jiang Xiao, Zhuang Xiong, Song Wu, Yusheng Yi, Hai Jin, and Kan Hu. Disk Failure Prediction in Data Centers via Online Learning. In *Proceedings of the 47th International Conference on Parallel Processing*, ICPP 2018, New York, NY, USA, 2018. Association for Computing Machinery. https://doi.org/10.1145/3225058.3225106.

[36] Erci Xu, Mai Zheng, Feng Qin, Jiesheng Wu, and Yikang Xu. Understanding SSD Reliability in Large-Scale Cloud Systems. In *2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS)*, pages 45–53, Nov 2018. https://doi.org/10.1109/PDSW-DISCS.2018.00010.

[37] Erci Xu, Mai Zheng, Feng Qin, Yikang Xu, and Jiesheng Wu. Lessons and Actions: What We Learned from 10K SSD-Related Storage System Failures. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pages 961–976, Renton, WA, July 2019. USENIX Association. https://www.usenix.org/conference/atc19/presentation/xu.

[38] Fan Xu, Shujie Han, Patrick P. C. Lee, Yi Liu, Cheng He, and Jiongzhou Liu. General Feature Selection for Failure Prediction in Large-scale SSD Deployment. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 263–270, June 2021. https://doi.org/10.1109/DSN48987.2021.00039.

[39] Yong Xu, Kaixin Sui, Randolph Yao, Hongyu Zhang, Qingwei Lin, Yingnong Dang, Peng Li, Keceng Jiang, Wenchi Zhang, Jian-Guang Lou, Murali Chintalapati, and Dongmei Zhang. Improving Service Availability of Cloud Systems by Predicting Disk Error. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 481–494, Boston, MA, July 2018. USENIX Association. https://www.usenix.org/conference/atc18/presentation/xu-yong.

[40] Yuqi Zhang, Wenwen Hao, Ben Niu, Kangkang Liu, Shuyang Wang, Na Liu, Xing He, Yongwong Gwon, and Chankyu Koh. Multi-view Feature-based SSD Failure Prediction: What, When, and Why. In *21st USENIX Conference on File and Storage Technologies (FAST 23)*, pages 409–424, 2023. https://www.usenix.org/conference/fast23/presentation/zhang.

[41] Ming Zhao, Jingchao Chen, and Yang Li. A Review of Anomaly Detection Techniques Based on Nearest Neighbor. In *2018 International Conference on Computer Modeling, Simulation and Algorithm (CMSA 2018)*, pages 290–292. Atlantis Press, 2018. https://www.atlantis-press.com/proceedings/cmsa-18/25897526.

[42] Hao Zhou, Zhiheng Niu, Gang Wang, XiaoGuang Liu, Dongshi Liu, Bingnan Kang, Hu Zheng, and Yong Zhang. A Proactive Failure Tolerant Mechanism for SSDs Storage Systems based on Unsupervised Learning. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pages 1–10, June 2021. https://doi.org/10.1109/IWQOS52092.2021.9521302.

[43] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11106–11115, 2021. https://doi.org/10.1609/aaai.v35i12.17325.

[44] Bingpeng Zhu, Gang Wang, Xiaoguang Liu, Dianming Hu, Sheng Lin, and Jingwei Ma. Proactive drive failure prediction for large scale storage systems. In *2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–5, May 2013. https://doi.org/10.1109/MSST.2013.6558427.