Multi-view Feature-based SSD Failure Prediction: What, When, and Why

Yuqi Zhang and Wenwen Hao, *Samsung R&D Institute China Xi'an, Samsung Electronics;* Ben Niu and Kangkang Liu, *Tencent;* Shuyang Wang, Na Liu, and Xing He, *Samsung R&D Institute China Xi'an, Samsung Electronics;* Yongwong Gwon and Chankyu Koh, *Samsung Electronics*

https://www.usenix.org/conference/fast23/presentation/zhang

This paper is included in the Proceedings of the 21st USENIX Conference on File and Storage Technologies.

February 21–23, 2023 • Santa Clara, CA, USA

978-1-939133-32-8

Open access to the Proceedings of the 21st USENIX Conference on File and Storage Technologies is sponsored by

Multi-view Feature-based SSD Failure Prediction: What, When, and Why

Yuqi Zhang[†], *Wenwen Hao*[†], *Ben Niu*[‡], *Kangkang Liu*[‡], *Shuyang Wang*[†], *Na Liu*[†], *Xing He*[†], *Yongwong Gwon*[*], *Chankyu Koh*[*]
[†]*Samsung R&D Institute China Xi'an, Samsung Electronics*
[‡]*Tencent* [*]*Samsung Electronics*

Abstract

Solid state drives (SSDs) play an important role in large-scale data centers. SSD failures affect the stability of storage systems and cause additional maintenance overhead. To predict and handle SSD failures in advance, this paper proposes a multi-view and multi-task random forest (MVTRF) scheme. MVTRF predicts SSD failures based on multi-view features extracted from both long-term and short-term monitoring data of SSDs. Particularly, multi-task learning is adopted to simultaneously predict what type of failure it is and when it will occur through the same model. We also extract the key decisions of MVTRF to analyze why the failure will occur. These details of failure would be useful for verifying and handling SSD failures. The proposed MVTRF is evaluated on the large-scale real data from data centers. The experimental results show that MVTRF has higher failure prediction accuracy and improves precision by 46.1% and recall by 57.4% on average compared with the existing schemes. The results also demonstrate the effectiveness of MVTRF on failure type and time prediction and failure cause identification, which helps to improve the efficiency of failure handling.

1 Introduction

Compared with hard disk drives (HDDs), NAND flash-based solid state drives (SSDs) have higher performance and lower power consumption [8, 19] and thus have become popular in enterprise storage systems and large data centers. However, to reduce costs, the storage density of SSDs is increasing, which reduces the endurance and reliability of SSDs [6, 25, 48]. Large-scale data centers usually have hundreds of thousands or even millions of SSDs. Such a large-scale deployment of SSDs poses a challenge to data center reliability. Although redundancy mechanisms (such as replication [38] and RAID [32]) have been used to protect data from loss, SSD failure still causes two major problems. First, even if the data center adopts a redundant protection scheme, SSD failure also affects the performance of the storage system and the stability of online services. Second, SSD failure leads to additional maintenance costs due to failure location, failure recovery, etc. Therefore, SSD failure prediction, as a proactive fault tolerance mechanism, has received increasing attention recently. Compared with the passive redundancy mechanisms, it can identify and proactively handle potential SSD failures in advance, thereby improving the reliability of the storage system and reducing the costs of failure location and recovery. In the large-scale storage system, it is significant to monitor the symptoms of SSD failures and predict failures in advance.

A common monitoring solution for modern storage devices (HDDs and SSDs) is S.M.A.R.T (Self-Monitoring, Analysis, and Reporting Technology), which can monitor and record the internal reliability-related attributes of drives. SMART logs are usually captured regularly, for example, there are one or several SMART logs captured per day for each device (in this paper, a log refers to a snapshot of SSD monitoring attributes). Since SMART logs originate from HDDs, many previous works [4, 9, 14, 21, 23, 24, 29, 42, 46, 49, 51] have studied HDD failure prediction based on SMART logs, and only some works [27, 30, 45, 50] focus on SSDs. In recent years, to better monitor SSDs, some SSD manufacturers have customized more attributes about SSD reliability and failure. Based on these custom attributes, some works [3,7,16] predict SSD failures more effectively.

For SSD failure prediction algorithms, most current schemes are based on supervised learning. They regard failure prediction as a binary classification problem (healthy SSD and failed SSD), and build classification models (such as random forest and neural network) to identify failed SSDs [3, 16, 27, 30, 45]. Some other works [7, 50] adopt anomaly detection approaches (such as isolated forest and autoencoder) to predict SSD failures, based on unsupervised learning. These schemes are primarily designed to learn the pattern of healthy SSDs. When the monitoring log of an SSD is very different from that of most healthy SSDs, it is considered that a failure may occur.

The previous works still face the following challenges. First, most of them [3,7,27,30,45] predict SSD failures based

on one or several short-term monitoring logs, and pay less attention to the long-term logs of SSDs. However, through our analysis, some SSD failures may not be reflected in short-term local information, but hidden in long-term information. A few works [16, 50] use sequence models such as long short-term memory (LSTM) [17] to directly learn from long-term data, but the sequence lengths of SSD monitoring data are too long and the lengths vary greatly, which affect the performance of sequence models. For long-term data, their trends and distributions are actually important for judging SSD failures (see Section 2.2). Second, although the failure prediction has screened the possible failed SSDs, it lacks instructive suggestions for verifying and handling failures. The operator only knows that a failure may occur, but not know what it is, when and why it will occur. Predicting or analyzing more information such as failure type, lifespan (remaining working time before failure) and failure cause is helpful for operators to verify whether it is an internal SSD failure, and judge what measures to take and whether it is urgent. For example, operators would deal with different types of failures with different urgency and measures (see Section 2.1).

In order to solve these challenges, we propose a multi-view and multi-task random forest (MVTRF) scheme. First, in addition to the short-term raw data, we generate histogram statistics and sequence-related features to reflect the long-term pattern of monitoring data. MVTRF adopts multiple input and groups decision trees to learn these multi-view features in parallel. It can predict SSD failures with both short-term and long-term information. Second, MVTRF employs multi-task learning to jointly learn the failure pattern through the associated failure type classification and remaining lifespan prediction. With these two tasks, the operator knows what type of failure it is and when it will occur, and can take corresponding actions. Finally, according to the decision process, we extract key decisions from MVTRF to reveal why the failure occurs and help operators verify and deal with SSD failures quickly. Experiments on real data from data centers show that MVTRF is effective and outperforms existing schemes. Our contributions are summarized as follows:

- We design histogram features and sequence-related features to characterize the distribution and trend of long-term monitoring data. MVTRF is proposed to jointly learn failure patterns from these features and short-term raw data, thereby improving the prediction accuracy.
- We propose SSD failure type prediction and combine remaining lifespan prediction to suggest proactive measures, in addition to failure prediction. Multi-task learning is adopted for these three tasks, since joint learning of related tasks can improve the model performance for each task.
- We propose a similar decision extraction (SDE) approach to extract the key decisions of MVTRF, so as to identify the symptoms and causes of SSD failures, and provide more information for verifying and handling failures.

2 Data Analysis and Motivation

2.1 Dataset

The large-scale SSD monitoring datasets of Samsung PM1733 and PM9A3 SSDs were collected from the data center of Tencent cloud. The datasets include more than 70 million monitoring logs within nine months from more than 300,000 SSDs with different lifespans in Tencent's data center. The log information consists of SSD serial number, server serial number, timestamp and SSD internal attribute values. Besides SMART attributes, Samsung has customized more internal attributes to enhance SSDs' self-monitoring capability, which makes it possible to predict and analyze more failure information. There are a total of 40 internal attributes for PM1733 and 85 for PM9A3. All these attributes, including standard SMART attributes and custom attributes, are called Telemetry attributes in this paper, and some of them are shown below.

- media_errors: the number of unrecovered data integrity errors detected by the controller
- controller_busy_time: the amount of time the controller spends on I/O commands
- temperature: the current temperature of internal composite
- read_recovery_attempts: total count of uncorrectable NAND reads that require retrying
- wear_leveling_max: maximum erase cycle of internal blocks
- nand_bytes_written: the number of NAND sectors written ($1\ count = 32MB$)

The failure lists of both PM1733 and PM9A3 were also provided by Tencent. The lists contain the information of SSD failures collected by Tencent operators, including the serial number of failed SSDs, failure's report date, failure description, and handling time and measures. There are totally 409 failure records in the lists. After checking by operators, most of them were SSD failures, and a few of them were failures of other devices such as the server backplane. Since manually checking and verifying each failure is a burden, operators need additional failure information (such as failure causes) to verify failures more efficiently.

By analyzing the failure description and handling measures in Tencent's failure lists, we found that failures can be divided into eight types, and different measures were taken at different times to deal with different types of failures. These failure types are called Check Failed, Cancelling I/O, Media Error, SSD Drop, Fail Mode, PLP, Read Only, and Reliability Degradation, and the relevant descriptions are shown in Table 1. Based on the measures and time to handle different failures, we also give a corresponding reference in terms of urgency.

Table 1: Eight SSD failure types.

Failure type	Description	Urgency
Check Failed	Health or performance check failed	High
Cancelling I/O	NVMe cancelling I/O	Medium
PLP	Power loss protection test failed	Medium
SSD Drop	SSD cannot be detected by host	Medium
Fail Mode	Device fail mode	Medium
Media Error	Some data cannot be read correctly	Medium
Read Only	Unable to write data to SSD	Medium
Reliability Degradation	NVMe reliability degradation	Low

For example, the SSDs with Check Failed were processed in an average of four days, and almost all of them were directly replaced, so its urgency is high. In contrast, the SSDs with Reliability Degradation were processed in an average of 19 days, and a small number of them were replaced. Reliability Degradation only means that there may be a problem with the SSD, but no real failure has occurred, while Check Failed generally means that the SSD has an unspecified serious failure with the impact on the performance of storage system. Some definite failure types, such as Media Error and Read Only, have definite effect and may be mitigated by redundancy mechanisms, and the processing urgency is medium.

Finding 1: The failure needs to be checked manually to confirm whether it is an internal SSD failure, and the urgency and measures to deal with the failure may vary depending on the confirmed SSD failure phenomenon and type. Detailed failure information is significant for failure handling.

2.2 Failure Analysis

To gain insight into SSD failures for failure prediction, we analyzed the failed SSDs in Tencent datasets based on Telemetry attributes. First, the distribution of Telemetry attributes of failed SSDs and healthy SSDs were analyzed to mine their differences. We evenly divided the value range of each attribute from minimum to maximum into multiple buckets, and used histograms to compare the data distribution of failed SSDs and healthy SSDs in each bucket.

Figure 1 and Figure 2 compare the data distribution of failed SSDs and healthy SSDs with nand_bytes_written and temperature attributes, respectively. The horizontal coordinate is the bucket index, and the vertical coordinate is the proportion of data that falls in the bucket. Figure 1 shows that most nand_bytes_written values of the failed SSDs and healthy SSDs fall in the buckets 1–7. However, the values of failed SSDs have a larger proportion than healthy SSDs in the later buckets. Figure 2 shows that the data distribution of failed SSDs and healthy SSDs differs greatly in the buckets 20–23 of temperature attribute, but the distribution before bucket 17 is more similar. Overall, the Telemetry values of

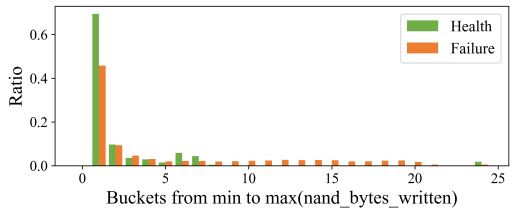


Figure 1: Distribution of nand_bytes_written of failed SSDs and healthy SSDs.

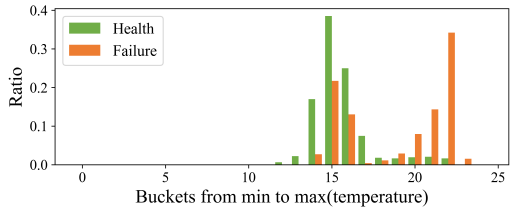


Figure 2: Distribution of temperature of failed SSDs and healthy SSDs.

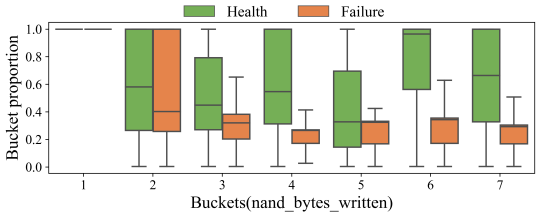


Figure 3: Bucket proportion of long-term data on buckets 1–7 of nand_bytes_written for failed SSDs and healthy SSDs.

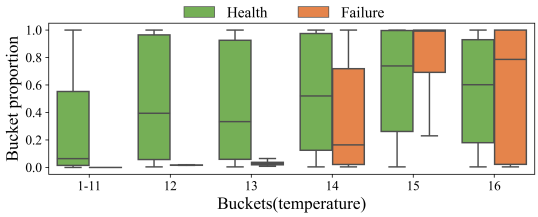


Figure 4: Bucket proportion of long-term data on buckets 1–16 of temperature for failed SSDs and healthy SSDs.

failed SSDs and healthy SSDs are somewhat different, but the distributions in some ranges are similar.

To further distinguish the similar distributions of attributes for failed SSDs and healthy SSDs, we explored the distribution differences of statistics of long-term Telemetry data (each SSD has multiple Telemetry logs over time). Multiple values of each attribute of each SSD over a long time fall into different buckets, and we calculated the proportion of the number of these values in each bucket to the number of values in all buckets, which is called the bucket proportion. Then, we used boxplots (the line in the middle of the box is the median, the lower edge of the box is first quartile, and the upper edge is third quartile) to compare the distribution of bucket proportions for failed SSDs and healthy SSDs.

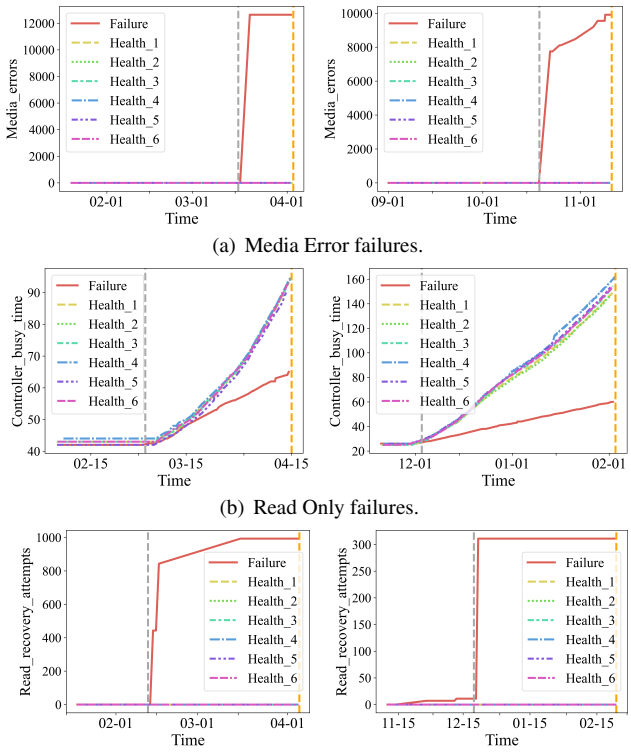
For the nand_bytes_written attribute, Figure 3 shows the bucket proportion of long-term data for buckets 1–7 whose

distributions are similar in Figure 1. The horizontal coordinate is still the bucket index, and the vertical coordinate is the bucket proportion of long-term data. It shows that on these buckets with similar distributions of values, the distribution of bucket proportions for long-term data of failed SSDs and healthy SSDs was different. On buckets 3–7 with small nand_bytes_written, the bucket proportions for long-term data of healthy SSDs were significantly larger than those of failed SSDs. It shows that healthy SSDs suffered from fewer writes over the long term, and thus they were less prone to failure. For the temperature attribute, Figure 4 shows the bucket proportion of long-term data before bucket 17 whose distributions were relatively similar in Figure 2. On buckets 1–13 with low temperature, the bucket proportions for long-term data of healthy SSDs were obviously larger and this indicates that low temperature is good for SSD health. In conclusion, based on the statistics of long-term SSD data, the difference between failed SSDs and healthy SSDs tended to be amplified.

Finding 2: There were some differences in the distribution of Telemetry attributes between failed SSDs and healthy SSDs, and the difference was more significant based on the statistics of long-term Telemetry data of each SSD (i.e., bucket proportion).

The Telemetry attributes of each SSD varied over the long term. Next, we analyzed the long-term changing trends of Telemetry attributes to explore the differences between failed SSDs and healthy SSDs. Since the workload was usually similar for most SSDs on the same server, we compared the changing trends of attributes of the failed SSD with other healthy SSDs on the same server before the failure occurred. Figure 5 shows the changing trend of main abnormal attributes of failed SSDs with different failure types. The horizontal coordinate represents the collection time, and the vertical coordinate represents the attribute value. Figure 5 shows that the attribute trends of healthy SSDs on the same server were similar, while the trend of failed SSD was different over the long term. Moreover, the curve of a failed SSD could involve multiple stages such as slow change, rapid change, and stability.

For the Media Error failure type, Figure 5(a) shows the changing trend of the media_errors attribute of two failed SSDs and the healthy SSDs on the same server. Although there are differences in the value range of two failed SSDs, they both showed a rapid growth trend in about 20 days before the failure occurred. Rapid growth of media_errors usually indicates an unrecoverable component problem inside the SSD and is one of the symptoms of SSD failure. Figure 5(b) shows the changing trend of the controller_busy_time attribute for the Read Only failure type. Compared with the healthy SSDs, both failed SSDs show smaller growth rate of the controller_busy_time attribute, and this trend occurred one to two months before the failure. This trend indicates that the



(a) Media Error failures.

(b) Read Only failures.

(c) Check Failed failures.

Figure 5: Attribute trends before the failures. For each failure type, the attribute trends of two failed SSDs and their respective server's healthy SSDs are shown. The gray and orange vertical dashed lines represent the date of symptom onset and the date of failure report respectively.

SSD successfully processed fewer I/Os and experienced performance anomalies, and finally the SSD went into read-only mode. Figure 5(c) shows that the SSDs with Check Failed went through a rapid rise for the read_recovery_attempts attribute in about two months before the failure occurred. Too many read retries generally indicate there exists a problem inside the SSD. In general, the same failure type may have similar changing trends of the same Telemetry attribute, but different failure types usually showed different symptoms which may have appeared at different times.

Finding 3: Due to similar workload and environment, SSDs on the same server usually have similar attribute trends, but failed SSDs may have different trends. The attribute value may change over a long time before SSD fails, and the change may go through multiple stages.

Finding 4: The same types of failures may have similar symptoms in the long-term trends of attributes, and the symptoms may appear at similar times before failures. Different failure types usually exhibit different failure symptoms in attribute trends. This makes it possible to predict the failure type and remaining lifespan of SSDs.

3 Design and Implementation

3.1 Overview

The overall architecture of our multi-view and multi-task random forest (MVTRF) scheme is shown in Figure 6. Based on the analysis in Section 2, our MVTRF design mainly follows three ideas: 1) the distribution and trend related features of long-term data are designed to capture long-term failure patterns; 2) features from different views are combined with group learning and joint decision to predict SSD failures accurately; and 3) detailed failure information is predicted and extracted to improve the efficiency of failure handling.

Specifically, Figure 6 shows that the MVTRF scheme is divided into two parts: offline training and online prediction. Offline training mainly involves two steps. The first is feature extraction. We perform preprocessing and data cleaning on the collected large-scale Telemetry data, and extract raw features, histogram features and sequence-related features. Raw features focus on the values of short-term SSD data, while histogram features and sequence-related features focus on the distribution and trend of long-term SSD data, and they are introduced in detail in Section 3.2. The second step is MVTRF training. The extracted features are trained in groups with MVTRF to obtain information from different views. To predict detailed failure information, we also introduce multi-task learning to simultaneously perform multiple prediction tasks through a single model, including the prediction of health or failure, failure type, and remaining lifespan.

Online prediction involves the following four steps. The first is feature extraction. The three features are extracted from the online data in the same way as offline training. The second is MVTRF prediction. Based on the extracted features, the trained MVTRF model combines decisions from different views to predict whether the SSD will fail, as well as the specific failure type and remaining lifespan. The third is failure cause identification. When an SSD failure is predicted, the key decisions in the judgment process of MVTRF model are extracted to analyze the possible causes of the failure. Through multi-task prediction and failure-cause identification, MVTRF not only identifies the failed SSD, but also answers what the failure is, when and why it will occur. Based on the information above, the fourth step is to verify the failure and take corresponding measures. Furthermore, we regularly train the model offline (e.g., training a new model monthly or quarterly) and update it online to ensure that the model can adapt to data changes. Next, we will introduce multi-view feature extraction, MVTRF, failure-cause identification and failure handling in detail.

3.2 Multi-view Feature Extraction

According to our observations of the symptoms of SSD failures in Section 2.2, we found that SSD failures were not only

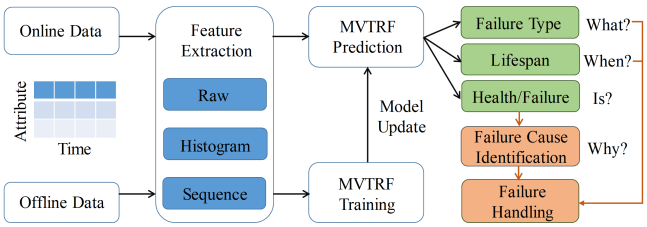


Figure 6: Overall architecture.

reflected in the abnormal value of short-term data, but also hidden in the distribution and trend of long-term data. It is an option to directly feed long-term data into sequence models such as LSTM. However, due to different usage periods and irregular collection, the number of Telemetry logs of different SSDs varies greatly (from a few to several thousands in our datasets). It is difficult for sequence models to process sequence data with such different lengths [20]. Moreover, overly lengthy sequences also affect the performance of sequence models (for example, LSTM has the vanishing gradient problem in the case of long sequences [47]), and lead to excessive computational complexity and overhead.

To avoid using long-term data directly, we extract features from long-term data to represent its distribution and trend. The analysis in Section 2.2 shows that the bucket statistics of long-term data help to distinguish between failed SSDs and healthy SSDs, therefore we first introduce histogram features based on bucket statistics. Then, from Section 2.2, we observed that the fluctuation and trend of long-term data also implies the failure symptoms, and thus we introduce sequence-related features that can characterize the degree of sequence fluctuation and change. Histogram features and sequence-related features extract key information from long-term data and discard redundant information. These features and short-term raw data constitute multi-view information for SSD failure prediction. Specifically, when the T-th Telemetry data of an SSD is collected, we extract raw features, histogram features and sequence-related features as follows.

3.2.1 Raw Features

After preprocessing and data cleaning, the data of a Telemetry log are the raw features. We discard attributes with exactly the same value in offline training, and do the same in online prediction. Assuming that there are N attributes remaining after data cleaning, the raw features of the T-th Telemetry data of SSD are defined as $D_T = \{a_{1T}, a_{2T}, ..., a_{nT}, ..., a_{NT}\}$, where $a_{1T}, a_{2T}, ..., a_{nT}, ..., a_{NT}$ are the values of N attributes. We mainly use raw features to capture short-term abnormal value of attributes, so they come from a single Telemetry log by default.

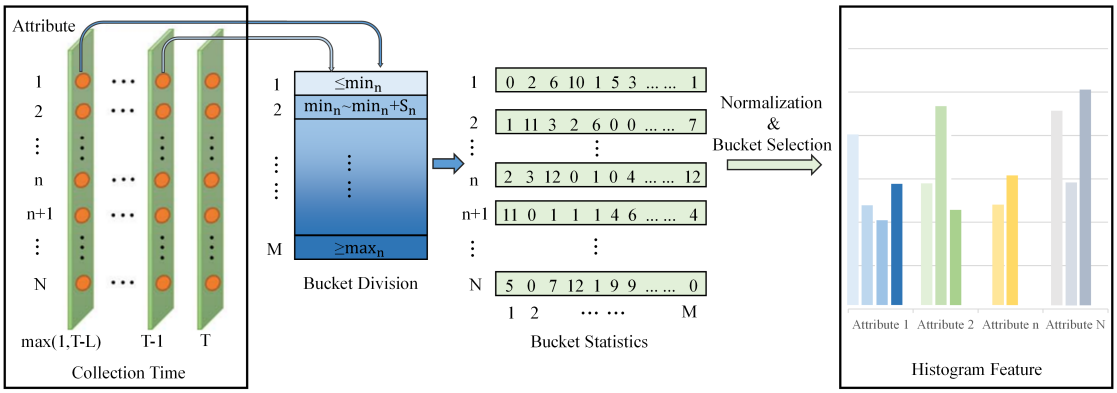


Figure 7: Overall process of generating histogram features.

3.2.2 Histogram Features

Histogram features are proposed to represent the distribution of Telemetry attributes over the long term. They are obtained by bucket statistics on the long-term raw features D_{T-L}–D_T of the SSD. L defaults to 256 and the time span of 256 logs is generally more than three months, which can cover the time span of failure symptoms analyzed in Section 2.2. The overall process of generating histogram features is shown in Figure 7. First, the minimum and maximum values of each attribute of all data are calculated during offline training, and the min and max of the *n-th* attribute are defined as min_n, max_n. Then, the min to max range of each attribute is divided into M buckets (100 by default), and the M ranges of the *n-th* attribute are defined as $\{(-\infty, min_n], (min_n, min_n+S_n], ..., (min_n+(M-3)\times S_n, min_n+(M-2)\times S_n), [max_n, +\infty)\}$, where $S_n = (max_n - min_n)/(M-2)$. Since the min and max of many attributes have special meaning, the min and max buckets are independent. Afterwards, we divide the features of each attribute of D_{T-L}–D_T into each bucket and count them. The statistics of the *n-th* attribute on M buckets is defined as $\{C_{n1}, C_{n2}, ..., C_{nm}, ..., C_{nM}\}$, where C_{nm} is the count of this attribute falling in the range of bucket m and $\sum_{m=1}^{M} C_{nm} = L$. In particular, when the number of SSD logs is less than L, all raw features (D_1–D_T) of the SSD are counted in buckets. In order to avoid the influence of this special case, we divide the bucket counts by the number of logs to get the proportions, and the formula for normalizing C_{nm} to the proportion P_{nm} is as follows.

$$P_{nm} = \begin{cases} \frac{C_{nm}}{T}, & T < L \\ \frac{C_{nm}}{L}, & T \geq L \end{cases} \quad (1)$$

Then the normalized *M-dimensional* feature $\{P_{n1}, P_{n2}, ..., P_{nm}, ..., P_{nM}\}$ of the *n-th* attribute is obtained, and $\sum_{m=1}^{M} P_{nm} = 1$. Through normalization, we solve the problem of large differences in the number of SSD logs. Next, we concatenate the *M-dimensional* features of all N attributes to get the histogram features whose dimension is $N \times M$. Since some buckets are less meaningful for failure prediction (e.g., the bucket for $wear\_leveling\_max = 0$), we adopt recursive feature elimination with cross-validation (RFECV) [28] to remove some buckets. During offline training, the RFECV algorithm forms multiple bucket subsets by recursively eliminating the least important buckets, and then selects the bucket subset with highest discrimination between failed SSDs and healthy SSDs through cross-validation. During online prediction, we only need to calculate the values of these selected buckets as the final histogram features. This not only reduces the noise from buckets with low discrimination, but also decreases the feature dimension and computational complexity.

3.2.3 Sequence-related Features

Sequence-related features are proposed to represent the fluctuation and trend of long-term raw features D_{T-L}–D_T of SSD. As stated in Finding 3 (see Section 2.2), the attribute trends of failed SSDs may change over a long time, and there may be multiple change stages. We introduce the coefficient of variation [2] to characterize the fluctuation of the attribute, and introduce kurtosis [10] and slope to characterize the trend of the attribute. To capture the multiple changing stages that may exist in long-term data, we also divide D_{T-L}–D_T into G segments equally in the time dimension (G is 4 by default), and calculate the coefficient of variation, kurtosis and slope separately for each segment. Assuming that the *g-th* segment starts at t_s and ends at t_e ($T-L \leq t_s < t_e \leq T$) and the raw features are D_{t_s}–D_{t_e}, the sequence-related features are calculated as follows.

Coefficient of variation. The coefficient of variation can measure the dispersion degree of the attribute over a long period of time. Relative to variance or standard deviation, the coefficient of variation can eliminate the effect of different scales for different attributes and different SSDs. We calculate the coefficient of variation for each segment window of each

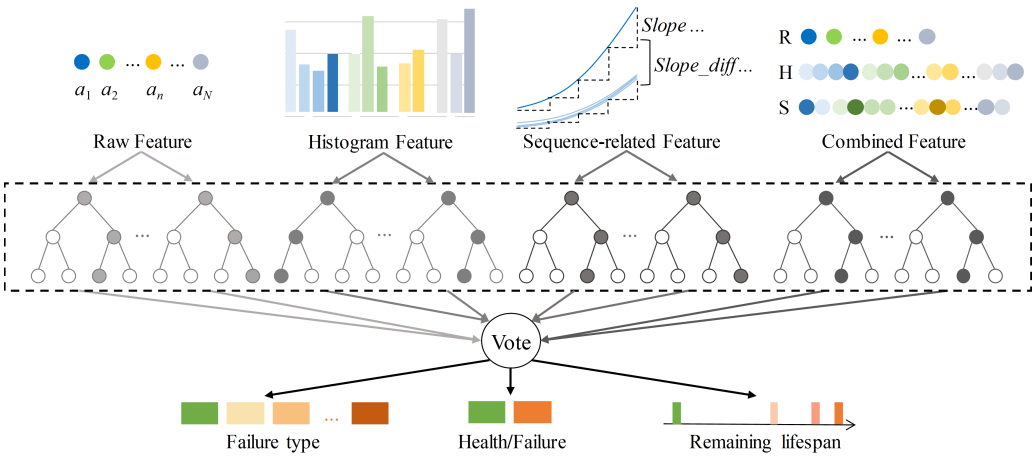


Figure 8: The structure of MVTRF.

attribute of the long-term raw features D_{T-L}–D_T respectively, and the calculation formula of the coefficient of variation $CVAR_{ng}$ for the g-th segment of the n-th attribute is as follows:

$$CVAR_{ng} = \frac{\sqrt{\frac{G}{L}\sum_{t=t_s}^{t_e}(a_{nt}-\mu_{ng})^2}}{\mu_{ng}} \tag{2}$$

where $\mu_{ng} = \frac{G}{L}\sum_{t=t_s}^{t_e} a_{nt}$.

Kurtosis. Kurtosis reflects the steepness of an attribute's distribution over the long term. The calculation formula of the kurtosis $KURT_{ng}$ for the g-th segment of the n-th attribute is shown below:

$$KURT_{ng} = \frac{\frac{G}{L}\sum_{t=t_s}^{t_e}(a_{nt}-\mu_{ng})^4}{(\frac{G}{L}\sum_{t=t_s}^{t_e}(a_{nt}-\mu_{ng})^2)^2} - 3 \tag{3}$$

Slope. Slope can reflect the changing trend of an attribute over time. The slope $SLOPE_{ng}$ for the g-th segment of the n-th attribute is calculated as follows:

$$SLOPE_{ng} = \frac{a_{nt_e}-a_{nt_s}}{t_e-t_s} \tag{4}$$

In addition, when the number of raw features of an SSD is less than L, the above features are calculated based on all the raw features of the SSD (i.e., D_1–D_T, and $L=T$ in the above formula), thereby avoiding the impact of various sequence lengths.

As stated in Finding 3, the trends of some attributes of failed SSDs may be quite different from those of other healthy SSDs on the same server. Therefore, for the above *CVAR*, *KURT*, and *SLOPE*, we calculate the difference between their values of an SSD and the average values of the same feature of other SSDs on the same server, defined as *CVAR_diff*, *KURT_diff*, *SLOPE_diff*. SSDs in the same server usually have similar workloads, so differences of attribute fluctuations and trends between these SSDs can provide more information for failure prediction. Next, we concatenate the *CVAR*, *KURT*, *SLOPE* and *CVAR_diff*, *KURT_diff*, *SLOPE_diff* of G windows of all N attributes to obtain sequence-related features of the SSD, with a dimension of $N\times G\times 6$. Finally, RFECV is also used to select more effective features from these features, similar to the approach in Section 3.2.2.

3.3 MVTRF

To learn the pattern of the extracted features, we chose random forests [5] as our base model for three reasons. First, existing studies have demonstrated good performance of random forests on SSD failure prediction [3, 27, 30, 45]. Second, a random forest is composed of multiple decision trees, and each decision tree divides the samples into different classes through a series of judgments on features. Its interpretability is good, which is helpful in further identifying failure causes through the judgment process (see Section 3.4). Third, the computational complexity of random forests is lower compared with neural network-related models, which is beneficial in reducing overhead during offline training and online prediction.

Section 3.2 introduced raw features, histogram features and sequence-related features. Each of them actually characterizes the state of SSDs from a different view, and we concatenate these three features together to form combined features with a global view. It is an option to adopt combined features as the input of all decision trees of random forest. However, it would be more reliable to predict SSD failures from these different views independently and then make decisions together. Therefore, we designed MVTRF with different sets of decision trees to learn different types of features in parallel. As shown in Figure 8, all decision trees of a random forest are equally divided into four sets, which learn raw features, histogram features, sequence-related features and combined features respectively. Then, all decision trees of the four sets vote to get the final prediction result. The class with the most votes is the predicted class, and the vote share is the con-

fidence probability. In this way, we combine features from different views to obtain the final judgment.

As stated in Finding 1 (see Section 2.1), more failure information can help operators take actions, and thus we recommend predicting the failure type and remaining lifespan when predicting the SSD failure. We adopt multi-task learning [1] to allow the single model to learn these three prediction tasks simultaneously. Multi-task learning and prediction with a single model has the following two advantages over using three independent models to learn and predict three tasks. First, our three tasks are related to each other. For example, Finding 4 in Section 2.2 shows the correlation between failure type and the time window of failure symptom. Joint learning of related tasks tends to improve the prediction accuracy of the model for each task. Second, learning and predicting three tasks simultaneously via a single model can reduce the time and overhead of training and prediction.

The specific definitions of the three tasks are as follows. 1) Failure prediction. We define it as a binary classification task. The data of healthy SSDs and failed SSDs are labeled 0 and 1 respectively. 2) Failure type prediction. We define it as a multi-classification task. The data of healthy SSDs and failed SSDs are labeled 0 and 1–O respectively. Our datasets have eight failure types, so $O = 8$. 3) Remaining lifespan prediction. Regression is more suitable for this task, but in order to maintain unity with the above two tasks, we also define it as a multi-classification task. The data more than one week from the failure are labeled 0, the data from one day to one week from the failure are labeled 1, the data within one day from the failure are labeled 2, and the data around the time of failure are labeled 3. Through multi-task learning, the prediction accuracy of each task is improved and more information is available for recommending proactive measures.

3.4 Cause Identification and Failure Handling

In a production environment, some SSD anomalies may actually be caused by failures of other devices, such as the server backplane. When a failure is predicted, operators need to understand the symptoms and causes of the failure to confirm exactly what device is failing. In fact, one of the reasons to use the random forest algorithm lies in its interpretability. Random forests are based on decision trees which are essentially a series of threshold decisions. It is in line with human thinking, that is, the final result is obtained through the combination of multiple judgments. By analyzing the decision process, we can reveal why there is a failure, thereby identifying the symptoms and causes of failure. However, a random forest is an ensemble of multiple decision trees, and it is difficult to analyze so many decision processes. Therefore, we propose similar decision extraction (SDE) to obtain key decisions from multiple decision trees in MVTRF to reflect the overall decision process and find the failure causes.

Figure 9 shows an example of how SDE works and there are three steps involved. First, each decision is chosen by the decision tree due to its distinguishing ability, and we extract similar decisions that appear more frequently in multiple decision trees as key decisions. Two decisions are considered to be similar when they meet the following conditions: 1) the features and decision logic (i.e., $\leq$ or $>$) for the two decisions are the same; and 2) the decision thresholds of both decisions are similar, and the difference between the two thresholds is within $\propto$ (10% by default). We look for similar decisions in other decision trees for each decision, and the number of similar decisions is used as the weight of this decision.

After calculating the weights of all decisions, the second step is to remove redundant similar decisions. Drawing on the idea of Non-Maximum Suppression [31], SDE retains decisions with higher weights as key decisions and discards similar decisions with lower weights. The main process is as follows. 1) Sort the weights of all decisions; 2) Select the decision with the highest weight from the unprocessed decisions; 3) Remove other decisions similar to this decision; and 4) Repeat operations 2 and 3 above until the weight of the selected decision is less than half of the global highest weight. In this way, redundant similar decisions are represented by the key decisions with higher weights. Finally, the weights of key decisions with the same features and decision logic can be integrated, and the most strict threshold (i.e., the maximum value for $>$ and the minimum value for $\leq$) is retained to show the outlier.

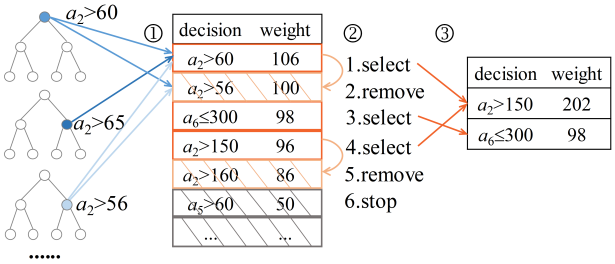


Figure 9: SDE. ①: Statistics of similar decisions within 10% threshold difference; ②: Non-maximum suppression on similar decisions; ③: Integrating key decisions with the same features and decision logic.

The key decisions extracted by SDE can reveal the failure cause and thus help to confirm whether it is an internal failure of the SSD. The key decisions of many failures involve SSD internal errors (e.g., excessive media errors, bad blocks or program failures), indicating that SSDs are failing. When key decisions involve the communication or environment, such as PCI errors or temperature, operators also need to check external devices (such as backplane) or the environment in addition to the SSD. The failure causes revealed by key decisions can significantly improve the efficiency of operators in verifying failures.

When an SSD failure is confirmed, the measures taken are based on the predicted failure type and remaining lifespan. As described in Section 2.1, different failure types may have different processing urgency. For SSDs with a high-urgency failure type, operators can replace them directly. The failures with low or medium urgency and long remaining lifespan can be further analyzed by operators, for example, by regular full-disk scans using scrub technology [27]. Depending on the urgency and remaining lifespan, the scan interval can also be adjusted accordingly. In this way, the impact on healthy SSDs can be significantly reduced while real failures are dealt with in time.

4 Evaluation

We evaluated our MVTRF scheme on real datasets from data centers. The following gives dataset setup and the evaluation metrics.

Dataset setup: For failure prediction, MVTRF was compared with existing schemes on three datasets. Besides the PM1733 and PM9A3 Tencent Telemetry datasets introduced in Section 2.1, the Alibaba public SMART dataset [45] was also used to evaluate the generalizability of MVTRF. This public dataset has multiple SSD models, but the number of failed SSDs for some models is inconsistent with the description of their paper, such as the MA1 and MC1 models. Except these models, we selected the MB1 model with the most failed SSDs for the experiment, as more samples can reduce the test error. There were 42,594 healthy SSDs and 1,807 failed SSDs with two-year SMART data of 16 standard attributes for the MB1 model.

We evaluated the performance of schemes in real scenarios, i.e., the history data were used to train models and new data online were used to predict SSD failures. Similar to the previous work [45], each dataset was divided into a training set, a validation set and a test set in chronological order. The training set was used to train the model, the validation set was used to tune model's hyper-parameters by evaluating the model during training, and the test set was used for the final evaluation of the model. For each dataset, we conducted two or three independent experiments on different data partitions, as detailed in Table 2. The average results of the independent experiments were deemed as the final results.

To further evaluate the generalizability of MVTRF for failure prediction on a new batch of SSDs, we performed a five-fold cross-validation on SSDs of PM1733 Tencent dataset. The further discussion and analysis in Section 4.2 to Section 4.4 were also performed on the PM1733 dataset.

Metrics: We used precision, recall, F0.5-Score and ROC_AUC to evaluate the prediction accuracy.

Precision: The proportion of correctly predicted failed SSDs (true alarms) to all predicted failed SSDs (both true alarms and false alarms).

Table 2: Data partitions for three datasets.

Dataset	Experiment round	Train set (month)	Val set (month)	Test set (month)
Samsung PM1733 (Tencent)	1	1–7th	8th	9th
	2	1–6th	7th	8th
Samsung PM9A3 (Tencent)	1	1–7th	8th	9th
	2	1–6th	7th	8th
MB1 (Alibaba) [45] (detailed model unkown)	1	1–22th	23th	24th
	2	1–21th	22th	23th
	3	1–20th	21th	22th

Recall: The proportion of correctly predicted failed SSDs to all actual failed SSDs, also called the true positive rate (TPR).

$F0.5\text{-}Score$: $\frac{(1+0.5^2)\times Precision\times Recall}{0.5^2\times Precision+Recall}$. It is the harmonic average of precision and recall, where precision is weighted higher. To avoid more false alarms for SSD failure prediction in practice, operators pay more attention to precision [45], and thus we use F0.5-Score to more comprehensively evaluate the effectiveness of schemes in a production environment.

ROC_AUC: For the above three indicators, the discrimination threshold of binary classification was fixed. In practice, different discrimination thresholds may be used. For example, to predict more failed SSDs, the discrimination threshold can be set lower, although there may be more false alarms at the same time. Therefore, we introduce the area under the curve of receiver operating characteristic (ROC) [12] to reflect the diagnostic ability of the binary classification model at different discrimination thresholds. The ROC curve is created by plotting the TPR versus the false positive rate (FPR, the proportion of false alarms to all healthy SSDs) at various thresholds. The area under the ROC curve (ROC_AUC) is a single score that can reflect the ability of the model to distinguish between failed SSDs and healthy SSDs across discrimination thresholds [7].

4.1 Comparison with Existing Schemes

In this section, we compare the proposed MVTRF with Random Forest, Neural Network, Autoencoder, and Ensemble LSTM on failure prediction. The descriptions of these existing methods are as follows. 1) Random Forest (RF): The raw features of a single monitoring log are used as the input of the random forest to predict SSD failures [3], which is the same as the single-task RF with raw features discussed in Section 4.3. 2) Neural Network (NN): SSD failures are predicted based on raw features using a neural network [3]. 3) Autoencoder (AE): The raw features of healthy SSDs are used as the input and they are reconstructed through an encoder and decoder. The reconstruction loss (i.e., the Euclidean distance between the input and reconstructed output) is used to predict SSD failures [7]. 4) Ensemble LSTM (LSTM): LSTM is used to capture failure symptoms from sequence data (the

Table 3: Comparison of MVTRF with existing methods for failure prediction on three datasets.

Methods	PM1733 Tencent				PM9A3 Tencent				MB1 Alibaba				Average			
	P	R	F	AUC	P	R	F	AUC	P	R	F	AUC	P	R	F	AUC
RF [3]	0.58	0.31	0.48	0.69	0.75	0.33	0.60	0.75	0.56	0.37	0.51	0.87	0.63	0.34	0.53	0.77
NN [3]	0.63	0.14	0.36	0.58	**0.85**	0.31	0.58	0.61	0.72	0.46	0.64	**0.89**	0.73	0.30	0.53	0.69
AE [7]	0.54	0.14	0.34	0.77	0.54	0.33	0.40	0.78	0.36	0.46	0.31	0.88	0.48	0.31	0.35	0.81
LSTM [16]	0.36	**0.40**	0.36	0.69	0.52	0.25	0.28	0.62	0.63	0.61	0.62	0.87	0.50	0.42	0.42	0.73
MVTRF(Ours)	**0.90**	**0.40**	**0.72**	**0.81**	0.70	**0.42**	**0.61**	**0.83**	**0.89**	**0.76**	**0.86**	0.86	**0.83**	**0.53**	**0.73**	**0.83**

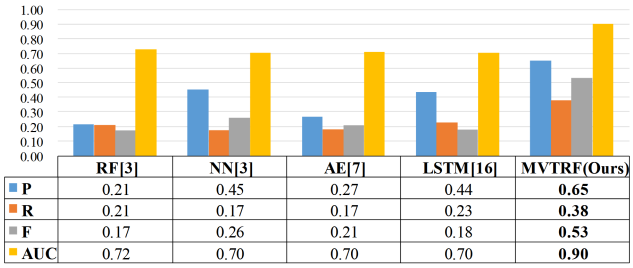
(P: precision; R: recall; F: F0.5-Score; AUC: ROC_AUC)

sequence length is also set to 256 for comparison), and multiple LSTMs are integrated to jointly predict SSD failures [16]. We re-implemented these algorithms, since the source code was not available.

Table 3 shows the results of these methods on the three datasets and the average results. RF, NN and AE are based on the raw features of a single monitoring log and cannot find failure patterns in long-term data, so they produce lower recall. AE predicts SSD failures only by learning the pattern of healthy SSDs and its average precision (0.48) is the lowest. However, its average ROC_AUC reaches 0.81, indicating that AE can better distinguish between failed SSDs and healthy SSDs at lower discrimination thresholds. LSTM achieves the average recall of 0.42 and outperforms the previous methods. This is because LSTM directly takes long-term sequence data as input and can capture more long-term failure symptoms. However, its precision and ROC_AUC is low, since the excessively long sequence length and the difference in lengths bring noise to the LSTM model.

For the average results of three datasets, our MVTRF improves precision by 46.1%, recall by 57.4%, F0.5-Score by 64.5%, and ROC_AUC by 11.1% on average compared with the four existing methods. We extract histogram features and sequence-related features from long-term sequence data to reflect the distribution and trend, thereby reducing noise and redundant information. MVTRF learns these features and raw features separately and predicts SSD failures by combining different views, which is more accurate and comprehensive. In addition, MVTRF performs better on the MB1 Alibaba dataset with a longer time span and more failed SSDs, which is conducive to the learning of long-term failure patterns. Although the three datasets have different SSD models (PM1733, PM9A3, and MB1), monitoring attributes (40 Telemetry attributes, 85 Telemetry attributes, and 16 SMART attributes), and time spans (9 months or two years), our MVTRF shows better performance on all three datasets, which demonstrates its robustness and generalizability.

Furthermore, a five-fold cross-validation on the PM1733 Tencent dataset was performed to further evaluate the effectiveness and generalizability of MVTRF in terms of failure prediction on a new batch of SSDs. Similar to previous work [3], the dataset was divided into five parts according to the serial numbers of the SSDs, and there were five independent experiments accordingly. In each experiment, four parts were selected for training and validation and one for testing. Therefore, SSDs in the test set do not appear in the training set for each experiment, and the test sets of the five experiments contain all SSDs. Fig. 10 shows that the cross-validation results were roughly consistent with the results in Table 3, with some reduction in prediction accuracy. The data patterns of unseen SSDs may be slightly different, which has some impact on the prediction. Compared with the existing methods, our MVTRF showed great improvements in four metrics. It implies that MVTRF is also more effective in failure prediction of unseen SSDs.



	RF[3]	NN[3]	AE[7]	LSTM[16]	MVTRF(Ours)
P	0.21	0.45	0.27	0.44	**0.65**
R	0.21	0.17	0.17	0.23	**0.38**
F	0.17	0.26	0.21	0.18	**0.53**
AUC	0.72	0.70	0.70	0.70	**0.90**

Figure 10: Cross-validation on PM1733 dataset. (P: precision; R: recall; F: F0.5-Score; AUC: ROC_AUC)

4.2 Discussion on Multi-view Features

Besides the raw features, this paper proposes histogram features and sequence-related features based on long-term data. These features reflect the state of SSDs from different views. By concatenating these three features, the combined features have a more comprehensive view. We first trained RF with each feature separately and compared their prediction accuracy to analyze the impact of different features on SSD failure prediction. Then, RF, NN, and AE with combined features and our MVTRF were also compared together to evaluate the effectiveness of MVTRF.

Table 4 shows the results on the PM1733 Tencent dataset. Raw features focus on abnormal attribute values, which are easy to judge, and thus their recall is relatively high. However, the short-term raw features cannot capture some failure symptoms in the long-term information, therefore the ROC_AUC was the lowest (0.69), implying that it is difficult to find more

failed SSDs at lower discrimination thresholds. The histogram features and sequence-related features reflect the distribution and trend of long-term data, and more failure symptoms can be found, so their ROC_AUC is higher. The combined feature contains the above three features. Since it contains multi-view information, RF with combined features performed well in each indicator. However, NN and AE with the same features did not perform so well. The combined features are comprehensive but also contain too much information, and this leads to the overfitting problem of these two models in training, while RF reduces overfitting through the joint decision of various decision trees [41]. Finally, MVTRF reached 0.90, 0.40 and 0.72 in precision, recall and F0.5-Score respectively. It enables different sets of decision trees to capture failure symptoms from different views, thereby further reducing overfitting caused by mixed excess information during training.

Table 4: Comparison of MVTRF and existing methods with different features.

Method	Precision	Recall	F0.5-Score	ROC_AUC
RF + Raw	0.61	0.34	0.52	0.69
RF + Histogram	**1.00**	0.17	0.48	0.72
RF + Sequence	0.50	0.25	0.38	**0.81**
RF + Combined	0.83	0.37	0.66	0.78
NN + Combined	0.79	0.17	0.39	0.74
AE + Combined	0.88	0.14	0.40	0.77
MVTRF	0.90	**0.40**	**0.72**	**0.81**

4.3 Multi-task Learning and Prediction

In addition to failure prediction, this paper introduces the tasks of failure type prediction and remaining lifespan prediction (see Section 3.3). Since joint learning of related tasks is often beneficial for each task, we perform multi-task learning and prediction through a single model. On the baseline RF with raw features and our MVTRF, the impact of multi-task learning on each task was evaluated. Table 5 compares the performance of two models under single-task learning and multi-task learning for three tasks. For failure prediction, the performance was better and the F0.5-Score of two models improved by 0.05 on average with multi-task learning and prediction. For failure type prediction and remaining lifespan prediction, we used the accuracy rate to evaluate the performance, since both tasks are multi-classification tasks and they only make sense when SSD failures are correctly predicted. The accuracy rate is defined as the proportion of SSDs with correctly predicted failure type (or remaining lifespan) to all correctly predicted failed SSDs. After using multi-task learning, Table 5 shows that the accuracy rate of two models for failure type prediction and remaining lifespan prediction increased by 0.04 and 0.09 on average, respectively. In conclusion, multi-task learning and prediction boosted the model's performance on three tasks.

Table 5 shows that our MVTRF with multi-task learning achieved an accuracy rate of 0.95 in failure type prediction and 0.55 in remaining lifespan prediction. It demonstrates that both predictions are effective. According to the urgency of different failure types and the remaining lifespan, operators can decide whether to directly replace the SSD or further analyze it, so that the failures can be handled in a timely and accurate manner.

Table 5: Comparison of single-task learning and multi-task learning.

Method		P	R	F	AUC	Type Acc	Lifespan Acc
RF + Raw	**Single-task**	0.58	0.31	0.48	0.69	0.88	0.44
	Multi-task	0.61	0.34	0.52	0.69	0.93	0.53
MVTRF	**Single-task**	0.83	0.37	0.66	0.79	0.93	0.47
	Multi-task	**0.90**	**0.40**	**0.72**	**0.81**	**0.95**	**0.55**

(P: precision; R: recall; F: F0.5-Score; AUC: ROC_AUC; Acc: accuracy rate)

Another benefit of using a single model for multi-task learning is that it can reduce model training and prediction time compared with using three models to predict three tasks. Table 6 shows the dimensions of different features, and compares the total time required for separate training/prediction and joint training/prediction on the three tasks based on these features. Table 6 reveals that adopting multi-task learning can reduce training/prediction time in most cases. It also shows that the training/prediction time of MVTRF mainly depends on the training/prediction time of the combined features with the highest dimension. In addition, MVTRF with multi-task learning completes the prediction of one million Telemetry data within three minutes and thus can fully support the online real-time prediction of large-scale SSDs.

Table 6: Total training/prediction time of single-task model and multi-task model.

Method	Feature NO.	Training time(s)		Prediction time(s)	
		Single	Multi	Single	Multi
RF + Raw	26	1230.9	599.8	36.5	62.8
RF + Histogram	102	1852.1	978.5	171.6	111.2
RF + Sequence	104	2867.9	1378.1	65.2	69.7
RF + Combined	232	3171.9	1707.2	232.2	130.9
MVTRF	464	3262.7	1775.2	245.2	143.0

(Train and predict on one million data.)

4.4 Similar Decision Extraction

According to the decision process of MVTRF model, we propose SDE to obtain key decisions and find the failure causes (see Section 3.4). Table 7 shows the key decisions extracted from the decision process of a failed SSD. SDE extracts five key decisions from a total of original 3,825 decisions and gives them weights (as described in Section 3.4, the weight is the number of similar decisions). The extracted

key decisions can certainly identify this failed SSD, but may lead to false alarms due to the large reduction in joint decisions. We reapplied these key decisions to all data to evaluate their effectiveness based on the false alarms introduced by them. Table 7 shows that the decision with the highest weight only had three false alarms, which indicates that the extracted key decisions have a strong distinguishing ability. Then, all false alarms were eliminated by combining subsequent key decisions. It can be concluded that the decisions extracted by the SDE approach are critical and they can represent the major decision process. According to the key decisions, we figured out that the direct cause of this failure was the rapid increase of media errors ($media\_errors\_slop > 126.44$ and $media\_errors > 6015.5$), and thus it was verified to be an internal failure of the SSD. In addition, the changes of temperature and wear leveling may be potential factors ($temperature\_kurt <= -1.11$ and $wear\_leveling\_max\_kurt > -0.047$).

Table 7: Key decisions of an SSD failure. False alarms were reduced with the combination of key decisions.

	Key decision	Feature type	Weight	False alarms
①	$media\_errors\_slope > 126.44$	Sequence	122	3 (①)
②	$media\_errors\_bkt0 <= 0.99$	Histogram	120	3 (① - ②)
③	$temperature\_kurt <= -1.11$	Sequence	117	1 (① - ③)
④	$media\_errors > 6015.5$	Raw	113	1 (① - ④)
⑤	$wear\_leveling\_max\_kurt > -0.047$	Sequence	96	0 (① - ⑤)

We also extracted several sets of key decisions from the judgment process of all failed SSDs to evaluate the overall discriminative ability of key decisions, as shown in Table 8. It shows there were 53,663 decisions in total for failed SSDs, and our SDE approach extracts 49 key decisions. Reapplying these key decisions to all data achieved the same precision and recall as all original decisions. The 49 key decisions performed almost the same as the original 53,663 decisions in distinguishing failed SSDs and healthy SSDs, which illustrates the effectiveness of the proposed SDE approach. Then, the failure causes can be identified and analyzed based on these decisions, which lays the foundation for verifying and handling SSD failures.

Table 8: Comparison of key decisions with all decisions.

	Decision NO.	Precision	Recall
All decisions	53663	0.90	0.40
Key decisions	49	0.90	0.40

5 Related Work

Many previous studies have investigated and analyzed the impact of drive errors and failures on large data centers [13, 15, 34, 35, 37, 43, 44]. In order to take proactive measures (such as replacing drives) before failures occur, drive failure prediction has received extensive attention and research. Since HDDs have been widely used for a long time, there are many works on HDD failure prediction [9, 11, 18, 21, 26, 36, 39, 42, 46, 49, 51, 52]. Most of these works [9, 18, 21, 26, 39, 42, 51, 52] are based on short-term monitoring data, as the symptoms of HDD failure generally appear days or hours leading up to the failure [24]. Unlike SSDs that are based on electrical signals, HDDs are mechanically based, and their problems would quickly develop into serious failures.

In recent years, with the popularization of SSDs, more and more research studies have been done on SSD failure prediction [3, 7, 16, 22, 27, 30, 33, 40, 45, 50]. Alter [3] et al. adopted classification algorithms to predict SSD failures based on machine learning algorithms, including logistic regression, support vector machine, random forest, and neural network. They also analyzed the failure characteristics of SSDs in different periods. Chandranil et al. [7] introduced the unsupervised anomaly detection algorithms, isolation forest and autoencoder, to predict SSD failures. These algorithms only learn the patterns of healthy SSDs and consider the ones with large pattern differences to be failed SSDs. Hao et al. [16] introduced LSTM, a recurrent neural network, to capture failure symptoms from the sequences of monitoring data. In addition, they proposed Ensemble LSTM to enhance the prediction accuracy through ensemble learning. Xu [45] et al. studied the impact of feature selection algorithms on SSD failure prediction. They proposed a feature selection approach, Wear-out-updating Ensemble Feature Ranking (WEFR), to improve the performance of random forest algorithm by selecting SMART attributes with strong representational ability.

6 Conclusions

In this paper, we propose multi-view and multi-task random forest (MVTRF) to predict SSD failures and other failure information based on short-term and long-term monitoring data. We observed that some failure symptoms are hidden in the distribution and trend of long-term data, and thus histogram features and sequence-related features were introduced. MVTRF learns these features and short-term data in parallel through multiple sets of decision trees, thereby integrating multi-view information to find more failures and reduce false alarms. In addition, we adopted multi-task learning to allow a single model to learn and predict detailed failure information, including failure type and remaining lifespan. We also propose similar decision extraction (SDE) to obtain the key decisions from MVTRF to identify and analyze the failure causes. These details help operators to quickly verify the failure and recommend appropriate actions to handle it more efficiently. Our evaluation on real data from data centers showed that MVTRF significantly improves the accuracy of failure prediction and can predict the failure type and remaining lifespan of SSDs simultaneously and effectively.

References

[1] Multiclass and multioutput algorithms. https://scikit-learn.org/stable/modules/multiclass.html.

[2] Hervé Abdi. Coefficient of variation. *Encyclopedia of research design*, 1:169–171, 2010. https://www.utdallas.edu/~herve/abdi-cv2010-pretty.pdf.

[3] 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.

[4] Mirela Madalina Botezatu, Ioana Giurgiu, Jasmina Bogojeska, and Dorothea Wiesmann. Predicting Disk Replacement towards Reliable Data Centers. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 39–48, New York, NY, USA, 2016. Association for Computing Machinery. https://doi.org/10.1145/2939672.2939699.

[5] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. https://doi.org/10.1023/A:1010933404324.

[6] Yu Cai, Yixin Luo, Saugata Ghose, and Onur Mutlu. Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery. In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 438–449, June 2015. https://doi.org/10.1109/DSN.2015.49.

[7] 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.

[8] Saeideh Alinezhad Chamazcoti, Bardia Safaei, and Seyed Ghassem Miremadi. Can Erasure Codes Damage Reliability in SSD-Based Storage Systems? *IEEE Transactions on Emerging Topics in Computing*, 7(3):435–446, July 2019. https://doi.org/10.1109/TETC.2017.2693424.

[9] 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, October 2016. https://doi.org/10.1109/BRACIS.2016.083.

[10] L. T DECARLO. On the meaning and use of kurtosis. *Psychological methods*, 2(3):292–307, 1997. https://doi.org/10.1037/1082-989X.2.3.292.

[11] 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.

[12] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. https://doi.org/10.1016/j.patrec.2005.10.010.

[13] 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.

[14] Shujie Han, Patrick P. C. Lee, Zhirong Shen, Cheng He, Yi Liu, and Tao Huang. Toward Adaptive Disk Failure Prediction via Stream Mining. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 628–638, November 2020. https://doi.org/10.1109/ICDCS47774.2020.00044.

[15] 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.

[16] 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, June 2022. https://doi.org/10.1145/3543146.3543169.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735.

[18] G.F. Hughes, J.F. Murray, K. Kreutz-Delgado, and C. Elkan. Improved disk-drive failure warnings. *IEEE*

Transactions on Reliability, 51(3):350–357, September 2002. https://doi.org/10.1109/TR.2002.802886.

[19] Massimo Iaculo, Francesco Falanga, and Ornella Vitale. Introduction to SSD. *Memory Mass Storage*, pages 213–236, 2011. https://doi.org/10.1007/978-3-642-14752-4_5.

[20] Pedro Lara-Benítez, Manuel Carranza-García, and José C.Riquelme. An Experimental Review on Deep Learning Architectures for Time Series Forecasting. *CoRR*, abs/2103.12057, 2021. https://arxiv.org/abs/2103.12057.

[21] 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.

[22] 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.

[23] Fernando Dione dos Santos Lima, Gabriel Maia Rocha Amaral, Lucas Gonçalves de Moura Leite, João Paulo Pordeus Gomes, and Javam de Castro Machado. Predicting Failures in Hard Drives with LSTM Networks. In *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 222–227, October 2017. https://doi.org/10.1109/BRACIS.2017.72.

[24] 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.

[25] 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), December 2018. https://doi.org/10.1145/3224432.

[26] Ao Ma, Rachel Traylor, Fred Douglis, Mark Chamness, Guanlin Lu, Darren Sawyer, Surendar Chandra, and Windsor Hsu. RAIDShield: Characterizing, Monitoring, and Proactively Protecting Against Disk Failures. *ACM Trans. Storage*, 11(4), November 2015. https://doi.org/10.1145/2820615.

[27] 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.

[28] Puneet Misra and Arun Singh Yadav. Improving the Classification Accuracy using Recursive Feature Elimination with Cross-Validation. *Int. J. Emerg. Technol*, 11(3):659–665, 2020. http://www.puneetmisra.com/admin/uploads/journals/5f136d202b8ba1.18644117.pdf.

[29] Joseph F. Murray, Gordon F. Hughes, and Kenneth Kreutz-Delgado. Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. *Journal of Machine Learning Research*, 6(27):783–816, 2005. http://jmlr.org/papers/v6/murray05a.html.

[30] Iyswarya Narayanan, Di Wang, Myeongjae Jeon, Bikash Sharma, Laura Caulfield, Anand Sivasubramaniam, Ben Cutler, Jie Liu, Badriddine Khessib, and Kushagra Vaid. SSD Failures in Datacenters: What? When? And Why? In *Proceedings of the 9th ACM International on Systems and Storage Conference*, SYSTOR '16, New York, NY, USA, 2016. Association for Computing Machinery. https://doi.org/10.1145/2928275.2928278.

[31] A. Neubeck and L. Van Gool. Efficient Non-Maximum Suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855, August 2006. https://doi.org/10.1109/ICPR.2006.479.

[32] David A. Patterson, Garth Gibson, and Randy H. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*, SIGMOD '88, page 109–116, New York, NY, USA, 1988. Association for Computing Machinery. https://doi.org/10.1145/50202.50214.

[33] Jay Sarkar, Cory Peterson, and Amir Sanayei. Machine-learned assessment and prediction of robust solid state storage system reliability physics. In *2018 IEEE International Reliability Physics Symposium (IRPS)*, pages 3C.6–1–3C.6–8, March 2018. https://doi.org/10.1109/IRPS.2018.8353565.

[34] 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.

[35] 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, September 2017. https://doi.org/10.1109/JPROC.2017.2735969.

[36] 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, February 2021. https://doi.org/10.1155/2021/8878364.

[37] 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.

[38] 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.

[39] Yu Wang, Eden W. M. Ma, Tommy W. S. Chow, and Kwok-Leung Tsui. A Two-Step Parametric Method for Failure Prediction in Hard Disk Drives. *IEEE Transactions on Industrial Informatics*, 10(1):419–430, February 2014. https://doi.org/10.1109/TII.2013.2264060.

[40] Debao Wei, Liyan Qiao, Mengqi Hao, Hua Feng, and Xiyuan Peng. Reliability prediction model of NAND flash memory based on random forest algorithm. *Microelectronics Reliability*, 100-101:113371, 2019. https://www.sciencedirect.com/science/article/pii/S002627141930472X.

[41] Graham Williams. Random forests. In *Data Mining with Rattle and R*, pages 245–268. Springer, 2011. https://doi.org/10.1007/978-1-4419-9890-3_12.

[42] 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.

[43] 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, November 2018. https://doi.org/10.1109/PDSW-DISCS.2018.00010.

[44] 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.

[45] 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.

[46] 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.

[47] Jinpei Yan, Yong Qi, Qifan Rao, and Tom Chen. LSTM-Based Hierarchical Denoising Network for Android Malware Detection. *Sec. and Commun. Netw.*, 2018, January 2018. https://doi.org/10.1155/2018/5249190.

[48] Ji Hyuck Yun, Jin Hyuk Yoon, Eyee Hyun Nam, and Sang Lyul Min. An Abstract Fault Model for NAND Flash Memory. *IEEE Embedded Systems Letters*, 4(4):86–89, December 2012. https://doi.org/10.1109/LES.2012.2213235.

[49] Ying Zhao, Xiang Liu, Siqing Gan, and Weimin Zheng. Predicting Disk Failures with HMM- and HSMM-Based Approaches. In *Proceedings of the 10th Industrial Conference on Advances in Data Mining: Applications and Theoretical Aspects*, ICDM'10, page 390–404, Berlin, Heidelberg, 2010. Springer-Verlag. https://doi.org/10.1007/978-3-642-14400-4_30.

[50] 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.

[51] 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.

[52] Marwin Züfle, Florian Erhard, and Samuel Kounev. Machine Learning Model Update Strategies for Hard Disk Drive Failure Prediction. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1379–1386, December 2021. https://doi.org/10.1109/ICMLA52953.2021.00223.