Exploit both SMART Attributes and NAND Flash Wear Characteristics to Effectively Forecast SSD-based Storage Failures in Clusters

Yunfei Gu and Chentao Wu, *Shanghai Jiao Tong University;*
Xubin He, *Temple University*

https://www.usenix.org/conference/atc24/presentation/gu-yunfei

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

Exploit both SMART Attributes and NAND Flash Wear Characteristics to Effectively Forecast SSD-based Storage Failures in Clusters

Yunfei Gu[1,2], Chentao Wu[1,2,*], Xubin He[3]
[1]*Department of Computer Science and Engineering, Shanghai Jiao Tong University*
[2]*Shanghai Key Laboratory of Trusted Data Circulation and Governance in Web3, Shanghai Jiao Tong University*
[3]*Department of Computer & Information Sciences, Temple University*
[*]*Corresponding Author: wuct@cs.sjtu.edu.cn*

Abstract

Solid State Drives (SSDs) based on flash technology are extensively employed as high-performance storage solutions in supercomputing data centers. However, SSD failures are frequent in these environments, resulting in significant performance issues. To ensure the reliability and accessibility of HPC storage systems, it is crucial to predict failures in advance, enabling timely preventive measures. Although many failure prediction methods focus on improving SMART attributes and system telemetry logs, their predictive efficacy is constrained due to the limited capacity of these logs to directly elucidate the root causes of SSD failures at the device level. In this paper, we revisit the underlying causes of SSD failures and first utilize the device-level flash wear characteristics of SSDs as a critical indicator instead of solely relying on SMRAT data. We propose a novel Aging-Aware Pseudo Twin Network (APTN) based SSD failure prediction approach, exploiting both SMART and device-level NAND flash wear characteristics, to effectively forecast SSD failures. In practice, we also adapt APTN to the online learning framework. Our evaluation results demonstrate that APTN improves the F1-score by 51.2% and TPR by 40.1% on average compared to the existing schemes. This highlights the potential of leveraging device-level wear characteristics in conjunction with SMART attributes for more accurate and reliable SSD failure prediction.

1 Introduction

In recent years, the growing complexity of scientific simulations [75] and data-intensive artificial intelligence applications [8] has led to a greater need for massive and rapid data access. This has driven high-performance computing clusters to adopt more advanced storage infrastructures. To meet this demand, flash-based solid-state drives (SSDs) are widely used in High-Performance Computing (HPC) systems as an alternative to hard disk drives (HDDs), resulting in significant performance improvements in data access [10, 48]. Even major service providers like Alibaba, Huawei, and Amazon have developed their own in-house SSDs to achieve top-notch performance. However, the increase in storage density comes with a trade-off of decreased endurance, and the prevalence of SSD failures poses a new challenge to overall HPC reliability [3, 4, 18, 36]. To tackle this challenge, a range of reactive fault-tolerance and storage redundancy schemes has been implemented, including Redundant Arrays of Independent Disks (RAID) [18, 55, 56], High-Availability (HA) pairs [37], and Replication. Nonetheless, storage failures lead to transient recovery and repair overheads, impacting the cost and tail latency of storage systems [11]. Hence, it is crucial to predict disk failures so that appropriate proactive actions can be taken in a timely manner [38, 85]. For example, replacing disks that are likely to fail soon before actual disk failures occur can help prevent data loss and reduce fault-tolerance overheads [3, 4, 18, 20, 26, 36, 56].

State-of-the-art proactive prediction solutions typically use Machine Learning (ML) techniques to train models with historical disk failure data, focusing on Self-Monitoring Analysis and Reporting Technology (SMART) [1]. However, these studies primarily focus on conventional hard disk drives (HDDs) [7, 39, 40, 46, 58, 59, 78, 82, 84, 90, 91, 95]. Nevertheless, these approaches are not entirely applicable to SSDs [67, 81], due to the distinct architecture and intricate device characteristics of SSDs [27, 52, 54, 61, 72, 73, 80]. To understand the root causes, researchers inspect specific error types and device-level errors (such as cell wear-out, program/read disturb errors, power faults, etc.) within SSDs through simulated or controlled laboratory environments [9, 21, 45, 51, 73, 94]. Furthermore, several in-depth studies also analyze the effects of correlated factors on SSD reliability in real-world production environments, including Google [2], Facebook [51], Alibaba [27], and NetApp [49].

In order to enhance failure prediction accuracy and achieve a high True Positive Rate (TPR)[1] while maintaining a low False Positive Rate (FPR)[2], certain ML-based methods [2, 31, 41, 47, 61, 93] strive to enhance SSD monitoring and augment learning features by incorporating additional customized

[1]The proportion of correctly predicted failed SSDs over all failed SSDs.
[2]The proportion of healthy SSDs that are falsely predicted as failed SSDs.

SMART attributes and system telemetry logs. Meanwhile, others [26, 81] attempt to extract the most relevant SMART attributes for training their models. Nevertheless, these solutions face challenges in significantly improving prediction accuracy for three key reasons: ❶ Neither these SMART attributes nor system telemetry logs can directly elucidate the fundamental causes of SSD failures at the device level. ❷ Customized logs vary across different data centers, making it difficult to reproduce consistent results [81]. ❸ These logs are reported by system-level maintenance software, which can experience issues such as data omissions due to software upgrades or system crashes.

In this paper, we revisit the underlying causes of SSD failures and first utilize the device-level flash wear characteristics of SSDs, known as Aging BEC data (explained in Section 3), as a critical indicator instead of solely relying on SMRAT data. To effectively integrate the SSD aging BEC data and SMART attributes, which come from different temporal and spatial dimensions, to improve the SSD failure prediction performance, we propose a novel approach called Aging-Aware Pseudo Twin Network (APTN). This approach is based on the principle of distance-based anomaly detection. It can effectively predict SSD failures not only in a short time but also in a longer time frame.

Our contributions are summarized as follows,

- We collect and analyze a device-level 3D-TLC NAND flash wear dataset in the field with the in-house team from collaborated SSD vendors. This dataset is produced through the SSD aging Bit Error Counts (BEC) benchmarking procedure, which requires customizing firmware functions within the SSD controller.
- We develop a novel SSD failure prediction approach named APTN. This approach adeptly leverages both SMART attributes and NAND flash wear characteristics within a high-dimensional space, facilitated by the pseudo twin network architecture [42]. Additionally, we apply APTN to the practical realm of online learning framework [26].
- We conduct a series of experiments that demonstrate APTN's proficiency in accurately predicting disk failures, even in the long leading time. This robust performance enhancement elevates the reliability of the SSD storage system. Our proposed APTN shows an impressive 51.2% improvement in F1-score and a remarkable 40.1% boost in TPR on average, compared to existing schemes.

2 Background and Motivation

In this section, we explore commonly used machine learning-based methods for predicting failures, the main technical challenges in designing disk failure prediction models, important factors contributing to SSD failures, and the limitations of the selected SSD SMART dataset. Finally, we clarify our underlying motivation.

2.1 Disk Failure Prediction

To enhance the reliability of high-performance distributed storage systems, numerous proactive disk failure prediction methods have been proposed. Existing approaches predominantly employ machine learning algorithms to train prediction models using SMART data, subsequently utilizing these models to anticipate faulty disks. Common machine learning algorithms include Linear Regression (LR) [44, 78, 83], Random Forest (RF) [15, 74, 77, 86], Adaptive Boosting (Support Vector Machine) [90, 91], K-Nearest Neighbor (KNN) [35, 76], Decision Tree (DT) [82, 86], and Neural Networks (NN) [35, 74, 76]. These methods typically necessitate manual feature extraction and selection. Furthermore, the realm of deep learning methodologies has also found its application within this domain. Long Short-Term Memory (LSTM) with under-sampling is introduced to handle time series-driven SMART data and address imbalanced dataset issues [65, 69, 70]. Additionally, Transfer Learning (TL) [7, 66, 79, 90–92], Twin Networks (also referred to as Siamese Networks) [89], and Online Learning (OL) [26, 78] are introduced to overcome challenges like minority drive detection and model aging [7, 26]. The machine learning algorithms used for SSD failure prediction are almost the same as, or the alternatives of what has been introduced above, but most of them [2, 31, 41, 47, 61, 93] utilize the customized SMART attributes and system telemetry logs to obtain more learning features on various scopes, lacking of generality for real-world implementation.

2.2 Challenges

Several primary technical challenges arise when designing disk error prediction models for large-scale HPC storage clusters: ❶ Weak correlations among SMART attributes and SSD failures: SMART attributes solely encompass the healthy state and statistical data of disks. They inadequately indicate SSD failures comprehensively, as demonstrated in Section 2.4 [2, 31, 41, 73, 81]. Notably, some failures lack error information in SMART attributes, particularly device-level flash errors that are pivotal for SSD failure analysis (such as NAND flash wear-and-tear, a dominant factor in SSD reliability [2]). Gunawi *et al*. [23] further advocate for vendors to provide device-level performance statistics to support failure studies. ❷ Dataset imbalance: In operational storage systems, the number of healthy SSDs significantly outweighs that of faulty ones. This dataset imbalance skews most machine learning models toward the majority class (healthy SSDs) and consequently introduces inaccuracies [31, 39, 89]. ❸ Poor failure detection of minority disk drive models: Within extensive storage setups, minority disk models may possess an insufficient sample count for training learning models, leading to subpar prediction accuracy due to underfitting issues [7, 89]. ❹ Failure prediction model aging: In real-world storage systems, SMART logs continually accumulate from disks over time. When expanding the data center or

introducing new drive models, solely relying on past SMART data for training becomes impractical, causing a degradation in predictive performance over long-term use [26, 78].

2.3 Important Factors of SSD Failures

Contemporary SSDs are semiconductor devices that persistently store data in NAND flash arrays, consisting of floating gate transistors [11]. Each NAND flash cell can endure a finite number of Program/Erase (P/E) cycles before wearing out permanently. An SSD comprises NAND flash chips, an SSD controller, interfaces, and essential peripherals. To orchestrate the functionality of these components and manage the SSD, firmware is embedded in the SSD controller. Given the internal organization of SSDs as described above, failures can be classified into two types based on whether the faulty units are directly accessible by the system: system-level failures (related to the controller and interfaces) and device-level failures (related to NAND flash). The SMART technology is integrated into the disk controller to record disk conditions and relay information about gradual degradation and potential defects in disks [1]. However, due to non-transparent firmware [2], it tends to signal symptoms of system-level failures rather than device-level ones.

Through the device-level analysis, the literature [2, 27, 49, 51] reveal that **Age**, **Device wear**, **Flash technology**, **Lithography** and **Capacity** are the most important factors impacting SSDs failures. The wear characteristics of the flash drive have a substantial impact on SSD reliability and are crucial for predictive performance [2, 61, 72, 81]. The study [22] further explains that utilizing burn-in wear data from NAND flash aids in predicting early failures in SSDs. Additionally, Meza *et al.* [51] emphasize that SSDs using the same flash technology (like SLC, MLC, 3D-TLC, etc.) and capacity demonstrate similar wear properties throughout their lifespan, regardless of the specific model.

2.4 SSD SMART Dataset and Limitations

The SSD SMART dataset that we have chosen is collected at Alibaba data center [27, 81]. It covers a population of nearly 500K SSDs of six drive models, spanning two years from 2018-01-01 to 2019-12-31 [81]. We conduct our analyses on 3D-TLC SSDs containing two drive models, MC1 and MC2. We exclude MLC drives since the SSDs from which we collect device-level NAND flash wear characteristics are also based on 3D-TLC SSDs, which are mismatched with MLC-based drives. The overview of the SSD samples we used is shown in Table 1.

Table 1: Overview of the SMART Dataset for Evaluation

Disk Model	Capacity	Flash Tech.	Litho-graphy	Disk Count	# failures
MC1	1920GB	3D-TLC	V1	199655	10508
MC2	960GB	3D-TLC	V1	23803	1131

This SMART dataset spans 43 SMART attributes in total, containing SMART logs and trouble tickets. The SMART logs provide daily records of SSD attribute statistics, while the trouble tickets capture the drive ID and corresponding timestamp upon the occurrence of a failure. However, this dataset encounters issues with missing data, wherein SMART data is not logged on specific days, due to software maintenance, upgrades, or system crashes. These omitted data points may contain crucial anomaly features that significantly impact prediction outcomes, leading to a bias in prediction models [27]. To assess the correlations between SMART attributes and SSD failures, we perform an analysis using the Spearman Correlation Coefficient rank [6]. The outcomes of this analysis are depicted in Figure 1, revealing the correlation coefficients of the top 11 most indicative SMART attributes. In essence, they suggest that the static values of SMART attributes alone are inadequate in predicting SSD failures effectively [31].

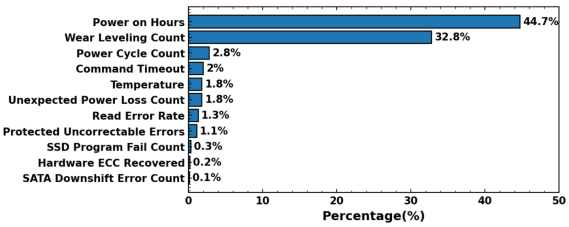


Figure 1: Spearman Correlation Coefficient rank analysis of the SMART attributes.

2.5 Motivation

The above preliminaries show that the device-level flash wear characteristics mainly determine their failures. However, the prevailing approaches in most previous studies are obtaining more learning features by customizing the SMART attributes or system telemetry logs. These solutions have the following limitations. ❶ SMART attributes have weak correlations with SSD failures, SMART and system telemetry logs cannot explain the failures at the device level. ❷ Customized logs are diverse, which makes it difficult to reproduce by different data centers. ❸ In practical implementations, SMART and system telemetry logs are reported by the system-level maintenance system, having data missing issues due to software upgrades or system crash problems.

Motivated by the above studies, we intend to use device-level flash wear characteristics of SSDs, which dominate SSD failures, as extended learning features, instead of obtaining more system-level data (i.e., customized SMART and system logs). Then we attempt to take advantage of SSD system-level and device-level features simultaneously, to improve the SSD failure prediction accuracy.

3 Flash Wear Characteristics Dataset Collection

With the aforementioned motivation, we proceed to gather and analyze the device-level flash wear characteristics of SSDs. Additionally, it's worth noting that the present trend of cloud service providers developing in-house SSDs

Algorithm 1: SSD BEC Aging Algorithm

```
Data: Blocks grouped into chunks, number of cycles to operate
/* Step 1 */
do
    Current_chunk ← first chunk of blocks;
    do
        Erase all the blocks in Current_chunk;
        Program Current_chunk with random pattern;
        Update Max temp;
        Current_chunk ← next chunk of blocks;
    while not all the blocks programmed;
while not all the cycles finished;
Idle Stage;
/* Step 2 */
Erase all the blocks;
Current_chunk ← first chunk of blocks;
do
    Program Current_chunk with random pattern;
    Update Max temp;
    Read Current_chunk;
    Calculate Failing bit count per codeword/AU;
    Current_chunk ← next chunk of blocks;
while not all the blocks finished;
```

facilitates practical real-world implementations. During the data collection process, we select in-house SSDs possessing identical NAND technology, lithography, and capacity parameters as the MC1 and MC2 drive models in the SMART dataset. This allows us to acquire the NAND flash wear characteristics. The architecture of a NAND flash chip is depicted in Figure 2, encompassing components such as dies, planes, chunks, blocks, and pages. We employ **Bit Error Counts** (BEC)[3] as the measure of NAND flash wear characteristics. The dataset for aging SSD BEC collection is generated through the SSD grey-box testing process [30]. The operation details are expounded in Algorithm 1. This collection procedure comprises two steps. ❶ Step 1 involves preparing the drive for subsequent aging by conducting drive preconditioning. ❷ Step 2 entails obtaining aging **BEC Bucket Numbers**[4], and iteratively recording SSD aging BEC data through various loops.

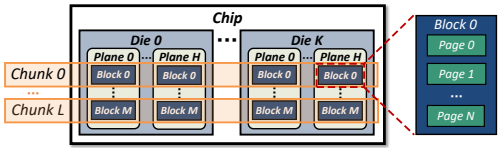


Figure 2: The structural relationships among chips, dies, planes, chunks, blocks, and pages in an SSD.

Step 1 performs *Erase* and *Program* on SSDs in the first several loops, aiming at reducing the false bad block marking to avoid the annealing effects [25] which tend to cause higher BEC or even lead to false Error-Correcting Code (ECC) errors [34] in extreme cases. Then it sets the proper temperature compensation and adjust the voltage threshold of NAND flash before it is scanned for aging cycling in this step. Step 2 performs *Erase*, *Program*, and *Read* operations. This step can be divided into the following procedures. ❶ Parameterize the chunk that has been illustrated in Figure 2, erase all blocks in a chunk, program all blocks in that chunk starting from page 0, and then move to the next page. Repeat the above operations until all pages of all blocks in this chunk are fully programmed. ❷ Progress to the next chunk of blocks until the entire drive has been erased and programmed. ❸ Repeat the outlined actions as necessary for the desired number of cycles. If an erase error or a program error occurs, the corresponding block will be designated as bad. ❹ During the *Read* operations, the BEC of each 4K data frame is tallied as a histogram for each Logical Unit (LUN). For instance, if a data frame contains no errors, the process increments the BEC bucket number of 0-bit. Extending this logical progression, we ultimately obtain the SSD aging BEC histogram.

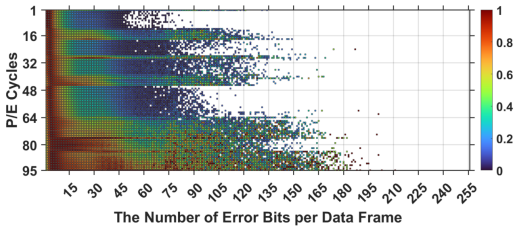


Figure 3: SSD aging BEC data heatmap representation. The X-axis represents the number of bit errors per data frame. The Y-axis represents the number of P/E cycles.

To facilitate interpreting the aging BEC histogram, we transfer the histogram to a heatmap by marking the normalized magnitude of each bucket by color brightness (values are normalized because of the Non-Disclosure Agreement). Figure 3 shows a fragment of a heatmap on aging BEC data upon a chosen die from an SSD. The colorbar's definition associates paler colors, such as blue, with smaller corresponding BEC bucket numbers. while more intense and vibrant colors, like deep red, represent larger values. It is worth noting that the spikes at 15, 30 and 45 P/E cycles, since the early defective blocks that have intrinsic weak LUN are wearing out in these cycles. This is consistent with the conclusion in literature [51] and [2].

SSD aging BEC data serves as an important factor that can efficiently reflect the underlying characteristics of internal NAND flash wear, due to the strong correlation between BEC data and ECC. ECC holds significant importance in SSD data protection, as it manages the correction of raw bit errors occurring throughout the entire lifespan of the drive. If the BEC surpasses the hardware ECC threshold (e.g., 254), the soft-decision [13] will be triggered with a latency penalty. In cases where BEC exceeds the capacity of the soft-decision ECC, it can lead to uncorrectable errors and failures [53]. Figure 4 provides an example of the aging distribution of LUNs within a drive across different P/E cycles (e.g., 0, 800, 1600, 2400, and 3200 cycles). The cumulative percentage of LUNs is represented by the area under the curve, with the red vertical line denoted as $X = n$ indicating the point where the

[3] BEC means bit error counts. In our discussion, it is the number of bit errors per 4K data frames specifically.

[4] BEC bucket number of n bits means the number of data frames that contains n error bits. This bucket number is recorded per die (LUN).

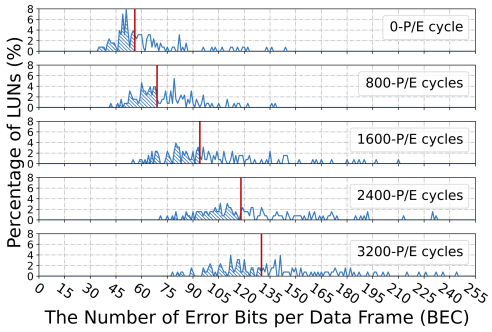


Figure 4: Percentage of LUNs versus the maximum BEC among data frames in the LUN for a given drive. The X-axis: the maximum number of error bits among all data frames in a LUN. The Y-axis: the percentage of LUNs that has at least one data frame containing x error bits and no data frame in this LUN containing more than x error bits.

area is divided in half (i.e., 50% of LUNs contain at least one data frame with n bit errors). In this figure, it is obvious that the distribution of the maximum BEC progressively shifts to the right as P/E cycles increase, indicating that the failure probability of this drive raises as it undergoes more P/E cycles.

Eventually, we use the SSD aging BEC data as the main clue to reflect the inherent wear characteristics of SSD NAND flash memory. Together with the in-house teams of our partner SSD vendors, we collected BEC data at the smallest granularity of die from 20,000 3D-TLC flash-based SSDs.

4 Design and Implementation

4.1 Overview

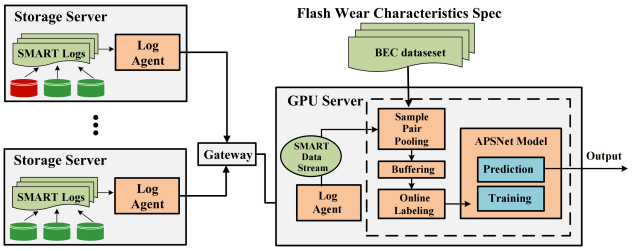


Figure 5: Overall architecture

SSD BEC aging data, obtained through the firmware's API interface, serves as a representation of SSD's internal wear patterns under different P/E cycles. Unlike SMART data, SSD BEC aging data is not collected in real time. It is gathered in an offline environment through wear tests conducted by the SSD manufacturer's in-house team at the firmware level. SMART data, on the other hand, represents real-time operational statistical information obtained during SSD runtime. These two datasets exist in disparate dimensions and cannot be directly combined. However, BEC and SMART data share a common clue, which is the current P/E cycle. Therefore, we design a novel failure prediction approach based on a pseudo-twin network, called Aging-aware Pseudo-Twin Network (APTN). It maps SSD BEC aging data and SMART data in a sparse high-dimensional space and calculates the Euclidean distance, we perform similarity comparisons to predict whether an SSD is likely to fail. The overall architecture of this approach is illustrated in Figure 5. It contains four stages: ❶ SMART logs are collected from storage servers via log agents and then transmitted to the inference node (GPU server). Then the GPU server transforms the SMART logs into a SMART data stream and executes the SMART/Aging sampling pooling procedure. ❷ Pre-train the APTN model in the offline model. ❸ The online prediction of SSD failure is executed, yielding 'failed' or 'healthy' outcomes. ❹ APTN is iteratively trained using the online learning framework, effectively addressing the challenge of model aging.

4.2 SMART/Aging Sample Pooling

The SMART/Aging module has three parts as follows:

4.2.1 SMART Data Transformation

Our SSD SMART data includes attribute values and ground truth labels (healthy/failed drive). The data is recorded daily, and we take consecutive days of SMART data (T_s) as input to the SMART/Aging sample pooling module, forming a "SMART time series." Each SMART instance at day t is represented as $I^t = \{I^t_0, I^t_1, \ldots, I^t_{(n-1)}\}$, and a SMART time series starting from day t is given by $D^t_s = \{I^t, I^{t+1}, \ldots, I^{t+T_s-1}\}$. For a healthy SSD, the SMART time series moves through a sliding window, theoretically moving forward each day when data is available consecutively for more than T_s days. The number of SMART time series that can be generated for a given disk with T recorded consecutive days ($T \geq T_s$) and a chosen stride s between two adjacent series is $\lfloor \frac{T-T_s}{s} \rfloor + 1$. To reduce correlation, the stride s should be no less than the length of the time series ($s \geq T_s$). For a failed SSD, we only consider the last T_s consecutive days of data as SMART time series samples, based on the gradual deterioration of a failed disk from health to failure.

4.2.2 SSD Aging BEC Data Transformation

The SSD aging BEC data reveals the wear characteristics of NAND flash memory cells within the SSD. This data, as elaborated in Section 3, is collected over multiple P/E cycles. An aging instance with N attributes at P/E cycle P can be represented as $I^P = I^P_0, I^P_1, \ldots, I^P_{N-1}$. We denote a *sequence of aging P/E cycles* as the collection of BEC data over P_s consecutive P/E cycles. An *aging P/E cycle series* starting from P/E Cycle P is denoted as $D^P_a = I^P, I^{P+1}, \ldots, I^{P+P_s-1}$. Notably, all SSD aging BEC data are generated from healthy SSDs with varying degrees of wear.

4.2.3 SMART/Aging Sample Pair Construction

Ideally, as input pairs for APTN, the *SMART time series* and the *aging P/E cycle series* should originate from the same SSD and be recorded at approximately the same time to ensure

perfect temporal correspondence between SMART and aging data. In practice, however, the *SMART time series* and the *aging P/E cycle series* are derived from distinct sets of SSDs. The SMART data pool is dynamic, while the aging data pool remains static. As mentioned in Section 2.3, evolving aging data are not accessible during SSD usage and are instead provided by the manufacturer. On the contrary, SMART data is captured during the operational life of SSDs in data centers. To promise a fair comparison between different feature values in machine learning algorithms, we implement the widely-used min-max feature normalization technique [47, 89] to normalize (SMART, Aging) sample pairs. We normalize both the average power-on-hours information in SMART data and the average P/E Cycles in SSD aging BEC data. The normalized values are denoted as H_n^t and P_n^c, respectively. The normalization process is outlined in Equations 1, wherein H_{min}, H_{max}, H_{avg}^t, P_{min}, P_{max}, and P_{avg}^c represent the minimum, maximum, and average power-on hours and P/E cycles in the SMART time series and aging BEC series, respectively. This sampling module constructs the valid SMART/Aging pairs, whose *aging series* is sorted as the one that has the closest P_n^c to H_n^t of the given *SMART aging series*. In this way, the wear-out status of the two data sources roughly matches in high dimensions, though the status is not necessarily equal.

$$H_n^t = \frac{H_{avg}^t - H_{min}}{H_{max} - H_{min}} \qquad P_n^c = \frac{P_{avg}^c - P_{min}}{P_{max} - P_{min}} \tag{1}$$

When given a *SMART time series* as input to the NN, the *aging P/E cycle series* input to the LSTM is determined by selecting the one with the closest P_n^c value to H_n^t from the *SMART aging series*. This approach aligns the wear status of the two data sources in higher dimensions, although their states may not be identical. With the recorded SSD aging BEC data and SMART data, we follow the mechanisms described above to transform the raw SMART attributes and aging BEC data into suitable SMART/Aging sample pairs $< S_L, A_L >$. These pairs are then used as inputs for the subsequent APTN-based learning model. The process of data transformation is illustrated in Figure 6, where the *Aging Samples* are highlighted in red and the *SMART Samples* are shown in green. Each element A_p in the *Aging Samples* corresponds to the BEC data of a healthy SSD over a fixed number of P/E cycles P_s (typically set to 30 cycles). Each element S in the *SMART Samples* represents <SMART attributes S_p, label l> for a specific SSD over a fixed duration of days T_s (usually set to 30 days). Since the focus of SSD failure detection lies in those with recorded SMART data, we divide the *SMART Samples* into separate training and test sets, and employ the *Aging Samples* as guidance for predictions during both the training and testing phases.

4.3 Design of APTN Model

Twin Networks (also called Siamese Network) typically consist of two identical sub-networks with shared weights, making them suitable for tasks involving pair matching

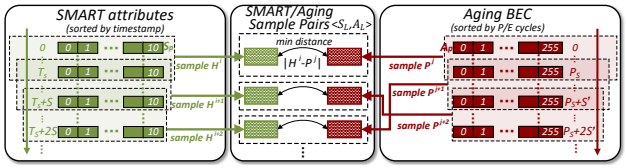


Figure 6: The instance of SMART/Aging sample pair construction mechanism implementation.

or identifying similarities between input pairs [50]. This concept aligns well with our objective of assessing the resemblance between SSD SMART and aging BEC data within a high-dimensional space. However, the lengths of the input vectors for the NN sub-network and the LSTM sub-network are different, resulting in non-identical sub-networks. To elaborate, as indicated in Table 2 and Table 3, we have 11 chosen SMART attributes and 5 fields derived from 256 BEC for input. Moreover, the physical significance of the *SMART time series* and the *aging P/E cycle series* are distinct. Therefore, we present our APTN-based failure prediction approach within a pseudo twin network framework. In this framework, the two sub-networks non-identical and the weights are not shared. This methodology directs the failure prediction by finding if there is a similarity between a SMART data sample and aging BEC data in a given sample pair within a high-dimensional space. A specific SMART sample is deemed healthy if its high-dimensional distance from the corresponding aging sample falls below a threshold associated with the SMART sample.

Table 2: Selected SMART attributes

ID	SMART Attribute
1	Read Error Rate
9	Power-on Hours
171	Number of Program Errors
172	Number of Erase Errors
173	Wear Leveling Status
174	Unexpected Power Loss
180	Unused Reserved Blocks
183	Number of SATA Errors
187	Number of Uncorrectable Errors
188	Command Timeout
194	Temperature

Table 3: Aging attributes generated from BEC

Symbol	Formula	Description
$N_{(1)}^p$	$\sum_{i=48}^{63} n_i^p$	Sum of BEC from 48 to 63 bits
$N_{(2)}^p$	$\sum_{i=64}^{79} n_i^p$	Sum of BEC from 64 to 79 bits
$N_{(3)}^p$	$\sum_{i=80}^{95} n_i^p$	Sum of BEC from 80 to 95 bits
$N_{(4)}^p$	$\sum_{i=96}^{127} n_i^p$	Sum of BEC from 96 to 127 bits
$N_{(5)}^p$	$\sum_{i=128}^{255} n_i^p$	Sum of BEC from 128 to 255 bits

The network structure of the proposed Aging-Aware Pseudo Twin Network (APTN) is shown in Figure 7. It consists of ❶ a sample pairs input layer, ❷ a pseudo twin network that contains two sub-networks, NN for SMART data and LSTM for aging BEC data, and ❸ an RF-based adaptive discriminator.

4.3.1 Sample Pairs Input Layer

This layer extracts the sample pairs from the SMART/Aging sample pair construction module in the form of $< S_L, A_L >$, where S_L represents a *SMART time series*, and A_L corresponds

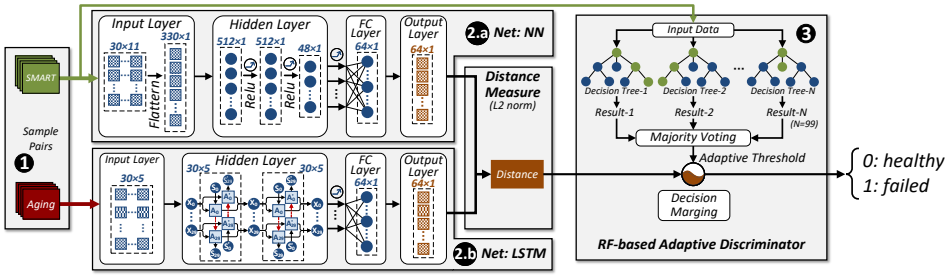


Figure 7: Network Structure of APTN-based SSD failure prediction approach

to an *aging P/E cycle series* (as explained in Section 4.2).

4.3.2 Pseudo Twin Network

Our pseudo twin network consists of two sub-networks: NN for SMART data, and LSTM for aging data. ❶ **NN for SMART Data:** In our investigation of SSD failure prediction solely relying on SMART data, we examined seven models discussed in Section 2.1, including NN, RF, LSTM, LR, KNN, DT, and SVM. After comprehensive analysis, we identified that NN achieves a well-balanced performance, with metrics such as precision and AUC score (validated in Section 5.2). Hence, we select NN as the appropriate sub-network for SMART data within the APTN architecture. In this design, a *SMART time series* is fed into the input layer of the neural network. The input time series has dimensions of (T_s, n), which is then flattened by the input layer. Subsequently, the flattened input of dimensions $(n \cdot T_s, 1)$ is propagated through 3 hidden layers. The output from these layers is then passed to a fully connected layer, followed by an output layer of matching dimensions. The neural network's specifications are illustrated in Figure 7. ❷ **LSTM for Aging Data:** LSTM, a type of Recurrent Neural Network (RNN), excels in retaining both short-term and long-term states of inputs, making it particularly suitable for prediction tasks based on sequential data.Given that SSD aging data is collected continuously in relation to P/E Cycles, we employ LSTM for failure prediction utilizing aging data. The LSTM network takes an aging sample of size (T_s, n') and passes it along the input layer. Then this input is directed through 2 hidden layers and further into a fully connected layers. The output dimensions of the LSTM network match those of the neural network. The distance measurement unit within APTN computes the L2 norm distance between these two outputs. Elaborate specifications can be observed in Figure 7. Additionally, in Section 5.2.2, we conduct **Exp#2** by substituting the sub-networks of the pseudo twin network to affirm that the current combination of sub-networks 1 and 2 within APTN already yields optimal results.

4.3.3 RF-based Adaptive Discriminator

The distance d between the outputs of the NN and LSTM is positively correlated with the probability that S_L and A_L belong to different health states. The larger d is, the less similar the two outputs are. Since the *aging P/E cycle series* are generated from healthy drives, a substantial distance d suggests a high likelihood that the target *SMART time series* originates from an unhealthy drive. Specifically, when the distance d exceeds the threshold D_{th} (i.e., $d > D_{th}$), the prediction result y_t will be 1, implying that we predict the target SMART SSD to be on the verge of failure. To avoid the limitations of a static threshold, we introduce an RF-based adaptive discriminator that dynamically adjusts the threshold associated with each given SMART sample. This discriminator acts as a safeguard for the predicted outcome. Without this adaptation, prediction accuracy could fluctuate when the output distance is close to the preset threshold. To mitigate this concern, the RF-based adaptive discriminator tailors the threshold according to the characteristics of the specific SMART sample. The effectiveness of this approach is demonstrated in **Exp#3** in Section 5.2.3.

4.3.4 Learning Model Hyperparameter Tuning

We used grid searching [88] to tune the hyper-parameters. Eventually, the network is tuned as below. The NN sub-network contains 3 hidden layers with sizes of 512×1, 512×1 and 48×1, respectively. Following the ReLu activation function, the output is forwarded to a 64×1 FC layer and mapped to a high-dimensional space of 64 dimensions. In LSTM sub-network, it contains 2 hidden layers, with each hidden layer having a size of $64 \times 30 \times 11$. In the RF-based discriminator, we utilize entropy as a measure of purity. We set the maximum tree depth to 100, with each leaf node containing a minimum of 1 sample. Furthermore, each node can split at most 3 times, and we employ a total of 150 decision trees in the ensemble. We choose Euclidean distance metric as the distance measurement. Furthermore, we choose the contrastive loss function [43] for the learning model training of this pseudo twin network. For a given sample pair $< S_L, A_L >$, we set the label $Y = 1$ if the SMART sample S_L comes from a healthy SSD, and $Y = 0$ if it comes from an unhealthy one. Let $G_{W_S}(S_L)$ denote the output of the NN which represents the SMART data embedding in the high dimension, where W_S is the weight of NN. Similarly, we define $G'_{W_A}(A_L)$ as the output of the LSTM network, where W_L is the weight of the LSTM network. Denote $W = \{W_S, W_L\}$ for simplicity. The distance between the two outputs can be measured as Equation 2,

$$D_W(S_L, A_L) = \left\| G_{W_S}(S_L) - G'_{W_L}(A_L) \right\|^2 \quad (2)$$

The total contrastive loss [24] is given by Equation 3 and 4,

$$\mathcal{L}(W) = \sum_{i=1}^{m} L(W,(Y,S_L,A_L)^i) \tag{3}$$

$$L(W,(Y,S_L,A_L)^i) = \frac{1}{2}(1-Y)\cdot(D_W(S_L,A_L))^2 + \frac{1}{2}Y\cdot\{max(0,D_{th}-D_W(S_L,A_L))\}^2 \tag{4}$$

where m is the number of sample pairs.

4.4 Learning Process

Algorithm 2: Model Training Algorithm

Input: SMART/Aging samples
Output: $\{W_R, <W_S, W_L>\}$ updating
`/* training RF-based adaptive discriminator */`
for *each epoch for APTN* **do**
  for *batch data in the training set* **do**
    $set_{SL} \leftarrow$ a batch of SMART time series;
    $labels \leftarrow$ a batch of labels;
    $W_R \leftarrow$ weights in random forests;
    RF_train ($set_{SL}, labels$);
    update W_R;

`/* training pseudo twin network */`
for *each epoch for APTN* **do**
  for *batch data in the training set* **do**
    $set_{SL} \leftarrow$ a batch of SMART time series;
    $labels \leftarrow$ a batch of labels;
    $set_{AL} \leftarrow \emptyset$;
    for *each* S_L *in* set_{SL} **do**
      construct $< S_L, A_L >$ as a sample pair, where A_L is an aging P/E cycle series;
      add A_L to set_{AL};
    $W_S \leftarrow$ weights in NN;
    $W_L \leftarrow$ weights in LSTM;
    $distances \leftarrow$ train_calc_dist (set_{SL}, set_{AL});
    $thresholds \leftarrow$ RF_vote (set_{SL}, W_R);
    $y_t \leftarrow$ determine_status ($distances, thresholds$);
    $W_S, W_L \leftarrow$ backward_prop ($labels, predictions$);

The procedures of overall APTN-based learning model training are illustrated in Algorithm 2. The RF-based adaptive discriminator and the pseudo twin network are trained separately due to the distinction that RF does not necessitate backpropagation, whereas the pseudo twin network relies on it in the training process. During each epoch, the RF-based adaptive discriminator takes a batch of SMART time series from the training set and obtains the corresponding labels that indicate whether the sources are healthy or unhealthy SSDs. After the RF-based adaptive discriminator has finished training, its parameters are fixed. Then, the pseudo twin network is trained. In each epoch, the pseudo twin network takes a batch of SMART time series from the training set and the corresponding labels. For each of them in the batch, the pseudo twin network finds the corresponding aging P/E cycle series and constructs the sample pairs. The NN and LSTM then process the corresponding samples. The distance of their outputs is calculated. For each sample pair, the random forests vote for whether they consider the SMART data as coming from a healthy disk ($y = 0$) or an unhealthy disk ($y = 1$), and their results are used to determine the threshold for that SMART data. After performing predictions and comparing them with the correct labels, APTN triggers backpropagation and updates the weights of NN and LSTM.

4.5 Online Learning Adaption

For practical implementation, we adapt APTN to support online learning, aiming to enhance the applicability of our failure prediction model in real-world product environments.

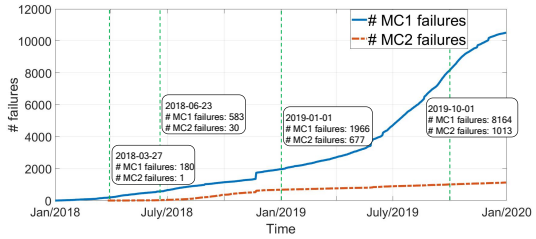


Figure 8: The accumulated failure occurrence distribution of MC1 and MC2 drive models over time

The traditional offline machine learning methods used for creating SSD failure prediction models assume that the entire training dataset is accessible and sufficient for building precise models. However, in real-world scenarios, new SMART data is generated daily, leading to changes in the underlying distribution of SMART attributes over time. Moreover, using offline-trained models without parameter updates for extended periods can result in significant performance degradation, known as the "model aging problem" [78]. To demonstrate this, we examine *trouble tickets* from the Alibaba SMART dataset and display the accumulated distribution of failure occurrences for MC1 and MC2 drive models over time in Figure 8. It's important to note that the initial MC2 drive model didn't fail until 2018-03-27, while the first MC1 drive model failed on 2018-03-02, with a total of 180 failures recorded on that day. This difference suggests that if a predictor is trained before 2018-03-27, its ability to predict MC2 failures might be limited due to the lack of learning features specific to MC2 failures. Additionally, the near future of 2018-03-27 poses a challenge related to the detection of minority samples. By this date, the number of MC2 failures has only reached 30, which might be insufficient for effectively training learning models and could become a performance bottleneck (**Exp#4** in Section 5.2.4 confirms this observation). As time progresses, the number of both MC1 and MC2 failures increases, indicating that the learning models can extract more valuable features to enhance predictive performance [17, 26, 62, 78].

Existing solutions tackle the above problem by re-training and updating the failure prediction models. However, they do not have the ability to remember what they have learned from the previous data used to train the predictor, which lowers the learning efficiency for real-time streaming SMART data. Therefore, we adapt our proposed APTN into an online learning framework by realizing the buffering and automatic online labeling [26, 57] mechanisms and applying incremental

Algorithm 3: APTN with Online Learning

```
Initialize APTN SSD failure prediction model
buffer buf ← [];
for t = 1, ··· do
    if buf is full then
        Slide one day for buf
    Extract a batch of learning features {< S_L, A_L >, label}_t;
      buffer{< S_L, A_L >, label}_t into buf;
    Call Algorithm 1 (APTN model training) to train APTN using
      buf;
    if buf is full then
        ỹ_t ← determine_status (distances, thresholds);
    Update weights {W_R, <W_S, W_L>};
```

online learning [57] with APTN. The algorithm of APTN with online learning is shown in Algorithm 3. Firstly, initialize the APTN model and the buffer buf used for receiving samples for online labeling. For each day t, if buf is full, we extract the learning features $\{< S_L, A_L >\}_t$ from all SSDs and buffer $\{< S_L, A_L >, label\}_t$ into buf. Finally, if buf is full, it indicates that the buffer has enough labeled samples for training, and thus we use APTN with $\{< S_L, A_L >\}_t$ to output the prediction results $\tilde{y}_t$ and update the weights $\{W_R, <W_S, W_L>\}$.

5 Evaluation

In this section, we present a series of experiments aimed at evaluating the performance of our APTN model compared to state-of-the-art approaches.

5.1 Methodology and Testbed

5.1.1 Dataset and Attribute Selection

(i) SSD SMART Attributes: In our experiments, we select the raw data of the following correlated attributes which are listed in Table 2, according to the Spearman correlation coefficient rank of the attributes shown in Figure 1 in Section 2.4. **(ii) SSD aging BEC Dataset**: In order to make this dataset fit the design of APTN. we reshape the SSD aging BEC data as Table 3 demonstrated. n_i^p represent the BEC bucket of i bits at the p^{th} P/E cycle. To evaluate the performance of our model, we employ 5-fold cross-validation on both the SMART and Aging datasets, a widely-used technique [19]. The entire dataset is split into training and testing sets using an 8:2 ratio [81, 87] for each validation fold.

5.1.2 Experiment Setup

❶ **Baselines:** As outlined in Section 2.1, we reproduce the state-of-the-art ML-based SSD failure prediction models, including RF [93], DT [86], LR [83], SVM [91], KNN [35], LSTM [65], and NN [74], as the baseline evaluation methods that exclusively utilize SMART attributes for learning. ❷ **Learning Hyperparameters Tuning:** To optimize hyper-parameters, we employ grid searching for hyperparameters selection [88]. Specifically, we set the number of trees to 100 in RF, while for KNN, we use 5 neighbors with the Euclidean Distance [5]. L2 regularization [12] is employed in LR with a regularization parameter set to 1. DT employs Gini coefficients [14] to gauge split quality. For SVM, we adopt the Radial Basis Function (RBF) kernel [28] with a regularization parameter of 1. In the case of LSTM, NN, and APTN, we set the epoch count to 1000, the initial learning rate to 0.001, and utilize the Adam optimizer. The pseudo twin network within APTN employs the contrastive loss and employs the Euclidean distance metric, whereas LSTM and NN employ BCE [63] as their loss function.

5.1.3 Performance Metrics

We use the following metrics to report the results in our experiments which are commonly used for evaluating the capability of binary classification (i.e., a disk is healthy or failed) model in machine learning [68].

a) TPR: True Positive Rate, also called recall. It captures the proportion of correctly predicted failed SSDs (denoted as *Positive* instances) over all the failed SSDs. A higher TPR means a better model.

b) FPR: False Positive Rate. It represents the proportion of healthy SSDs that are falsely predicted as failed. The lower the FPR is, the better the model is.

c) F1-score: F-score is a measurement to take both precision and TPR into consideration to comprehensively assess the classification model, given by $\frac{2\times\text{Precision}\times\text{TPR}}{\text{Precision}+\text{TPR}}$. It is designed to work well on imbalanced data. There F1-score higher, the better the prediction performance the model has.

d) AUC: AUC is the area under this ROC Curve. It represents the measure of separability of different classes [29]. The higher the AUC, the better performance of the model at distinguishing failed and healthy SSDs [71].

e) C-MTTDL: C-MTTDL is the economic analysis metric and evaluate the reliability and availability of the system based on different methods. $C-MTTDL = \frac{MTTDL}{Cost} \approx \frac{MTTF}{(1-\frac{k\mu}{\mu+\gamma})(C_a\times FP+C_b\times FN)}$. Where MTTDL is the Mean Time To Data Loss. FP (FN) is the number of true healthy (failed) SSDs that are falsely predicted as failed (healthy). C_a and C_b represent the associated costs for misclassifications. $\gamma = 1/(lookaheaddays\times 24hours)$. μ is the inverse value of Mean Time To Repair (MTTR) [89].

5.1.4 Simulating Practical Long-Term Availability

For practical purposes, it is preferable to anticipate drive failures further in advance to enhance drive maintenance and procure new spare drives efficiently. Therefore, we adopt a failure prediction sliding window upon the SMART series to simulate the experiments of practical long-term availability via predicting SSD failures ahead of time. The size of the sliding window is L. The prediction is 1 to N days lookahead. Let S denote the stride. The organization of sliding is shown in Figure 9. Here, we set L to 30, set N to 5, 7, 15, 30, 45, 60, 90, and 120, respectively, and assess the performance of predicting N days ahead using the mentioned metrics.

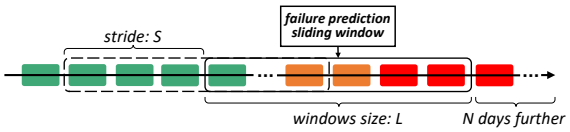


Figure 9: The sliding window for predicting SSD failures ahead in time for practical long-term usage.

5.2 Experimental Results

We show the experimental results to comprehensively evaluate our proposed APTN's prediction performance of *Short-Term Lookahead Prediction*[5] and *Long-Term Lookahead Prediction*[6]. Exp#1, Exp#2, and Exp#3 are conducted to evaluate all learning models in offline mode. All model training and failure detection tests are based on the complete dataset that contains all samples of MC1 and MC2 drive models. Exp#4 is to evaluate the learning models in online mode, the dataset is in stream data mode. Exp#5 demonstrates the improvement of storage system reliability via cost-MTTDL. Exp#6 shows the overhead of these ML-based methods in practical real-time simulation.

5.2.1 Exp#1 (Effectiveness of proposed APTN)

we compare APTN with the aforementioned state-of-the-art models respectively. Due to the structural difference, RF, DT, LR, SVM, KNN, LSTM, NN and SN models can only feed in SMART attributes as the training/testing data, but APTN is able to utilize both SMART attributes and NAND flash wear characteristics (i.e., BEC aging data) as the learning features to improve the performance.

Figure 10 demonstrates the TPR, FPR, F1-score, and AUC score results of all models in short-term and long-term lookahead failure prediction. Figure 10(a) shows that the TPR of APTN performs best in both short-term and long-term lookahead prediction. In short-term lookahead prediction, the average TPR of APTN is higher RF, DT, LR, SVM, KNN, LSTM, and NN than 40.4%, 27.4%, 78.4%, 78.0%, 54.3%, 30.2% and 20.3% respectively. Moreover, in long-term lookahead prediction, the average TPR of APTN exceeds those of RF, DT, LR, SVM, KNN, LSTM and NN than 45.2%, 33.5%, 81.1%, 80.5%, 59.1%, 26.6% and 31.6% respectively. Figure 10(b) demonstrates that as the days lookahead increases, the FPR of all methods increases. APTN can consistently keep the lowest FPR among state-of-the-art methods. Specifically, the FPR of APTN remains at 0.094 at minimum and stays relatively low at 0.156 when the days lookahead reaches 120.

Overall, The TPR and FPR of APTN achieve 90.1% and 9.4% at best, respectively. APTN improves the F1-score by 51.2% and TPR by 40.1% on average compared with the existing schemes. This experiment exhibits that the APTN achieves the best performance of SSD failure prediction, solving the SMART limitations and imbalanced dataset problems for practical drive maintenance in long-term availability. Moreover, it indicates that APTN has an excellent model generalization behavior via good AUC scores.

[5]Short-Term Lookahead Prediction means operating the failure prediction in 5, 7, 15, and 30 days ahead.

[6]Long-Term Lookahead Prediction means operating the failure prediction in 45, 60, 90, and 120 days ahead.

5.2.2 Exp#2 (Discussion of the alternative designs of sub-networks in the pseudo twin network)

We analyze the prediction behavior of APTN with alternatives of sub-networks 2.a and 2.b in the structure of APTN (See in Figure 7 in Section 4.3). We replace these two sub-networks with different RNN which are proficient to deal with the prediction task in time series. We name these APTNs with alternative sub-networks in Table 4 and show the results in Figure 11. Figure 11(a) and Figure 11(b) show that the TPR of APTN performs surprisingly best in both short-term and long-term lookahead prediction, with 13.4% performance improvement on average. Moreover, the FPR of APTN is the lowest. Figure 11(c) and Figure 11(d) show the F1-score and AUC score variations in failure prediction of APTN, APTN_LL, APTN_GG and APTN_GL models with different days lookahead. APTN outperforms the other models in both F1-score and AUC score, achieving performance gains of at least 6% and 8%, respectively. These results prove that the design of sub-networks in the pseudo twin network in the proposed APTN is reasonable and convincing.

Table 4: Sub-network alternatives in APTN structure

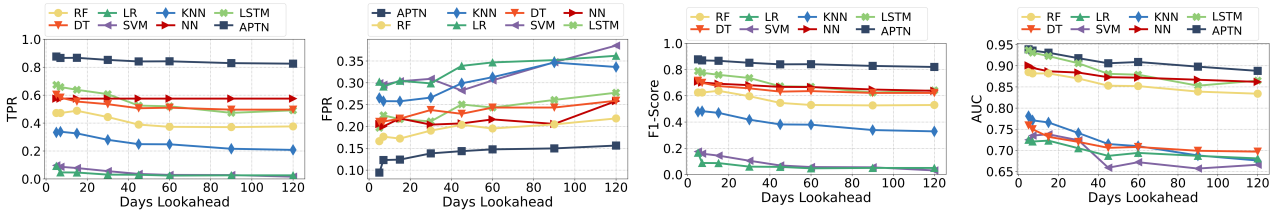
Model Name	Sub-network 2.a Model	Sub-network 2.b Model
APTN	NN	LSTM
APTN_LL	LSTM	LSTM
APTN_GG	GRU	GRU
APTN_GL	GRU	LSTM

5.2.3 Exp#3 (Effectiveness of adaptive RF-based discriminator)

We demonstrate the importance of the design of an RF-based adaptive discriminator in APTN (See in Section 4.3.4) by showing performance degradation without it and showing the results in Figure 12. We name the APTN without RF-based adaptive discriminator after APTN_NRF. Figure 12(a) and Figure 12(b) show that The TPR of APTN performs better in both short-term and long-term lookahead prediction, with 14% performance improvement on average, and APTN achieves lower FPR than APTN_NRF, with 23% deduction on average. Figure 12(c) and Figure 12(d) show that both the F1-score and AUC score of APTN are higher than that of APTN_NRF by 4% at least. These results indicate that the RF-based adaptive discriminator helps APTN performing better.

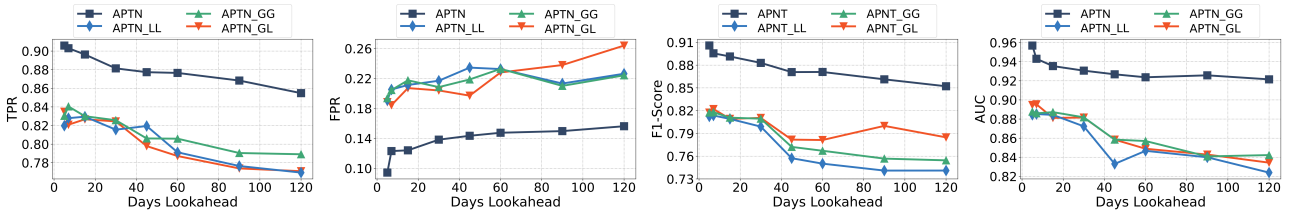
5.2.4 Exp#4 (Adaptability of APTN for online learning)

We simulate the sequential arrival of training data according to the timestamps in the SMART dataset. We observe that all models almost fail to predict the MC2 drive model's failure at that time regarding the excessively low F1-score or high FPR results shown in Figure 13. These above results prove the hypothesis in Section 4.5. Afterward, we evaluate the performance of 30 days lookahead failure prediction (a moderate selection for early warning of disk replacement)



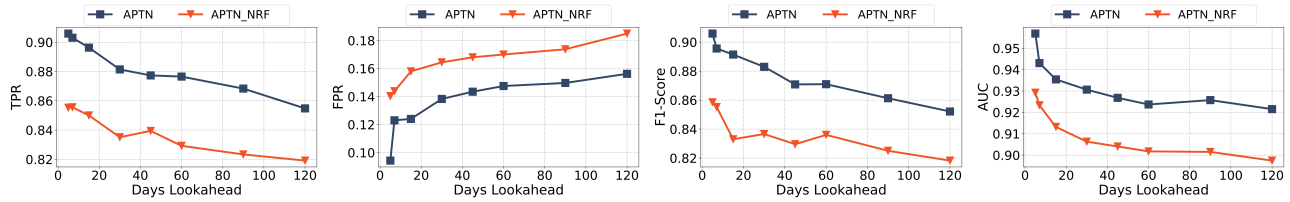
(a) TPR (b) FPR (c) F1-score (d) AUC Score

Figure 10: Prediction performance among APTN, and baseline SSD failure prediction models based on the complete dataset.



(a) TPR (b) FPR (c) F1-score (d) AUC Score

Figure 11: Prediction performance among APTN, APTN_LL, APTN_GG and APTN_GL based on the complete dataset.



(a) TPR (b) FPR (c) F1-score (d) AUC Score

Figure 12: Prediction performance among APTN, and APTN_NRF based on the complete dataset.

of all learning models with online learning on the test set approximately once every 3 months from 2018-04-30, particularly focusing on MC2 drive models. Notably, from our statistics, the accumulated number of failure samples in the MC2 drive models increases from 5 to 88 from 5^{th} to 8^{th} month. It indicates that ML models have only trained on a minority MC2 SSD dataset. Figure 14 demonstrates that, from 5^{th} to 8^{th} month, the TPR and F1-score of APTN are higher than other methods by 40% and 30% at least, respectively. Moreover, the performance of APTN is keeping leading steadily with each model update. Finally, the TPR and F1-Score of APTN stably and gradually increase from 0.788 and 0.781 to 0.882 and 0.893, respectively. Overall, we conclude that APTN with online learning solves the model aging problem and APTN has the outstanding ability to solve the minority disk failure detection challenge as well.

5.2.5 Exp#5 (Improvement of Reliability)

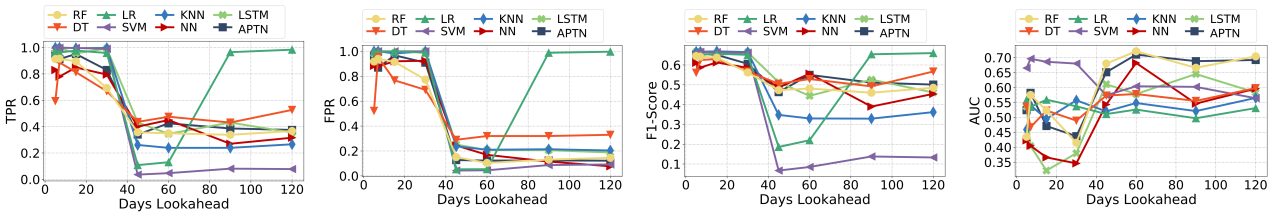
We utilize the economic analysis metric C-MTTDL to quantitatively evaluate the reliability and availability of the system based on different SSD failure prediction approaches. Given that $MTTR = 10$ hours, so that $\mu = 1/10$ hours. The literature [27] reveals that SSD failures follow an exponential distribution with the Mean Time Between Failures (MTBF) (i.e., the number of hours in a year over the overall Annual Failure Rate (AFR) in Alibaba SSDs failure dataset, i.e., $\frac{8760}{1.16\%}$, equals to 757,759 hours. C_a and C_b are set to 400 and 200 dollars respectively in our evaluation. Table 5 shows that our proposed APTN improves C-MTTDL approximately from 2.5× to 15×, which significantly improves the reliability of the storage system at a lower cost.

5.2.6 Exp#6 (Overhead of APTN)

To investigate the overhead of APTN in practical implementation, we evaluate the time cost of APTN in online learning mode, which is usually concerned with practical proactive fault tolerance mechanisms in modern storage systems [89]. The time cost contains training and prediction time. We evaluate that the training and prediction time cost of APTN is 86.7 seconds and 12.6 seconds, respectively. Therefore, we conclude that APTN only costs less than 2 minutes on the daily SMART data of 45K disks, which is acceptable and supposed to satisfy the performance need in large-scale data center deployment [26, 89].

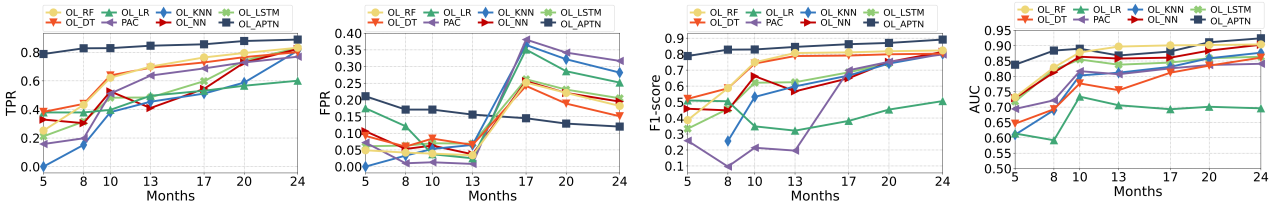
Table 5: Average Improvement of C-MTTDL

Method	TPR	FP	FN	MTTDL (years)	C-MTTDL(hours/dollars)
RF	47.2%	433	6148	154.7	0.96
DT	60.3%	979	4625	198.1	1.31
LR	9.3%	272	10559	94.8	0.37
PAC	9.7%	119	10514	95.1	0.38
KNN	33.3%	703	7759	125.7	0.60
NN	57%	529	4949	186.9	1.36
LSTM	67.3%	428	3801	233.5	2.19
APTN	**88%**	1133	1438	**478.7**	**5.66**



(a) TPR (b) FPR (c) F1-score (d) AUC Score

Figure 13: Prediction performance of predicting SSD failure of MC2 drive model among offline APTN, RF, DT, LR, SVM, KNN, LSTM and NN models based on the SMART dataset in the near future of 2018-03-27.



(a) TPR (b) FPR (c) F1-score (d) AUC Score

Figure 14: Performance of 30 days lookahead SSD failure prediction of MC2 drive model among online APTN, RF, DT, LR, SVM, KNN, LSTM and NN models based on the SMART dataset roughly every 3 months. The x-axis represents the number of months from 2018-01-01. Each ML model in online mode is denoted by "OL_" added in front of its name. In the legends of this figure, PCA represents the Principal Component Analysis, which is the online learning alternative of SVM.

6 Related Work

SSD Failure Prediction based on SMART data. Litz [11] observes that the number of healthy and erroneous disks in disk sample data is severely unbalanced. Hence, they try to drop small-class data during training to improve prediction accuracy. Alter *et al.* [2] prove that the conventional SMART attributes, such as write behavior or error incidence, are not as meaningful, but the age of the drive is important for error prediction. Further, Xu *et al.* [81] propose a wear-updating Ensemble Feature Ranking mechanism to select the SMART attributes learning features with the wear degree for SSD failure prediction of different drive models and vendors. In addition, Hao *et al.* [31] discover that static values of SMART attributes hardly indicate SSD failures, so they customize the SMART attributes and propose LSTM-based ensemble learning scheme to overcome limitations and improve the prediction accuracy. Zhang [93] *et al.* extract static and sequential features from customized SMART and system logs and use multi-task Random Forest to get predictive performance promotion. Different from them, we use the general SMART attributes but combine them with NAND flash wear characteristics to improve the robustness and performance of SSD failure prediction.

NAND Flash Endurance Prediction. Hogan *et al.* [32,33] perform a symbolic regression learning method to predict the maximum number of erases in a flash memory block. Fitzgerald *et al.* [16] analyzes the effects such as programming time and erase time in 2D flash memory, and proposes a lifetime prediction model to predict the maximum number of erasures. Moreover, Peleato *et al.* [64] analyze the relationships between the Bit Error Rates (BER) and the number of erasures in 2D flash memory to build a BER prediction model for flash memory blocks. Furthermore, Nakamura *et al.* [60] find that 25% of the programming interference errors were concentrated in 3.5% of the flash memory cells, and they achieve the prediction of the weaker data retention errors based on different strategies. These studies are focusing on SSD internal errors and attempt to improve single-SSD performance. However, we pioneer the use of device-level flash wear characteristics of SSDs as a set of learning features, together with SMART attributes to improve the SSD failure prediction performance in the HPC storage system.

7 Conclusions

In this paper, we collect and analyze the device-level NAND flash wear characteristics from more than 20k in-house SSDs. Then we propose a novel APTN-based SSD failure prediction approach, which significantly improves the predictive performance of short-term and long-term SSD failure prediction, giving a long lead time to take actions before failure occurrence in HPC clusters. Our results show 90.1% TPR and 9.4% FPR. APTN also boosts the F1-score by 51.2% and TPR by 40.1% on average over other methods. Besides, Our proposed APTN has the adaptability to online learning, which is practical for real-world deployment.

8 Acknowledgement

We thank our anonymous reviewers, as well as our colleagues for their valuable feedback. We thank funding support from the National Key R&D Program of China (No.2023YFB4502900).

References

[1] Bruce Allen. Monitoring hard disks with smart. *Linux Journal*, (117):74–77, 2004.

[2] 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*, pages 1–14, 2019.

[3] Muhammad Alfian Amrizal, Pei Li, Mulya Agung, Ryusuke Egawa, and Hiroyuki Takizawa. A failure prediction-based adaptive checkpointing method with less reliance on temperature monitoring for hpc applications. In *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 515–523. IEEE, 2018.

[4] Anupong Banjongkan, Watthana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop. A study of job failure prediction at job submit-state and job start-state in high-performance computing system: Using decision tree algorithms [j]. *Journal of Advances in Information Technology*, 12(2), 2021.

[5] Thorsten Behrens, Karsten Schmidt, Raphael A Viscarra Rossel, Philipp Gries, Thomas Scholten, and Robert A MacMillan. Spatial modelling with euclidean distance fields and machine learning. *European journal of soil science*, 69(5):757–770, 2018.

[6] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[7] 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*, pages 39–48, 2016.

[8] Shawn T Brown, Paola Buitrago, Edward Hanna, Sergiu Sanielevici, Robin Scibek, and Nicholas A Nystrom. Bridges-2: a platform for rapidly-evolving and data intensive research. In *Practice and Experience in Advanced Research Computing*, pages 1–4. 2021.

[9] Yu Cai, Yixin Luo, Erich F Haratsch, Ken Mai, and Onur Mutlu. Data retention in mlc nand flash memory: Characterization, optimization, and recovery. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pages 551–563. IEEE, 2015.

[10] Adrian M Caulfield, Laura M Grupp, and Steven Swanson. Gordon: using flash memory to build fast, power-efficient clusters for data-intensive applications. *ACM Sigplan Notices*, 44(3):217–228, 2009.

[11] 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*, pages 120–133, 2020.

[12] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L2 regularization for learning kernels. *arXiv preprint arXiv:1205.2653*, 2012.

[13] Guiqiang Dong, Ningde Xie, and Tong Zhang. Enabling nand flash memory use soft-decision error correction codes at minimal read latency overhead. *IEEE Transactions on Circuits and Systems I: regular papers*, 60(9):2412–2421, 2013.

[14] Robert Dorfman. A formula for the gini coefficient. *The review of economics and statistics*, pages 146–149, 1979.

[15] Fernando Dione dos Santos Lima, Gabriel Maia Rocha Amaral, Lucas Goncalves 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. IEEE, 2017.

[16] Barry Fitzgerald, Damien Hogan, Conor Ryan, and Joe Sullivan. Endurance prediction and error reduction in nand flash using machine learning. In *2017 17th Non-Volatile Memory Technology Symposium (NVMTS)*, pages 1–8. IEEE, 2017.

[17] Óscar Fontenla-Romero, Bertha Guijarro-Berdiñas, David Martinez-Rego, Beatriz Pérez-Sánchez, and Diego Peteiro-Barral. Online machine learning. In *Efficiency and Scalability Methods for Computational Intellect*, pages 27–54. IGI Global, 2013.

[18] Alvaro Frank, Dai Yang, Andre Brinkmann, Martin Schulz, and Tim Süss. Reducing false node failure predictions in hpc. In *2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, pages 323–332. IEEE, 2019.

[19] Tadayoshi Fushiki. Estimation of prediction error by using k-fold cross-validation. *Statistics and Computing*, 21(2):137–146, 2011.

[20] Ana Gainaru, Franck Cappello, Marc Snir, and William Kramer. Failure prediction for hpc systems and applications: Current situation and open issues. *The International journal of high performance computing applications*, 27(3):273–282, 2013.

[21] Laura M Grupp, Adrian M Caulfield, Joel Coburn, Steven Swanson, Eitan Yaakobi, Paul H Siegel, and Jack K Wolf. Characterizing flash memory: Anomalies, observations, and applications. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 24–33, 2009.

[22] Yunfei Gu, Xingyu Wang, Zixiao Chen, Chentao Wu, Xinfei Guo, Jie Li, Minyi Guo, Song Wu, Rong Yuan, Taile Zhang, Yawen Zhang, and Haoran Cai. Improving productivity and efficiency of ssd manufacturing self-test process by learning-based proactive defect prediction. In *2023 IEEE International Test Conference (ITC)*, pages 226–235, 2023.

[23] Haryadi S Gunawi, Riza O Suminto, Russell Sears, Casey Golliher, Swaminathan Sundararaman, Xing Lin, Tim Emami, Weiguang Sheng, Nematollah Bidokhti, Caitie McCaffrey, et al. Fail-slow at scale: Evidence of hardware performance faults in large production systems. *ACM Transactions on Storage (TOS)*, 14(3):1–26, 2018.

[24] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[25] Jin-Woo Han, Mo Kebaili, and M Meyyappan. System on microheater for on-chip annealing of defects generated by hot-carrier injection, bias temperature instability, and ionizing radiation. *IEEE Electron Device Letters*, 37(12):1543–1546, 2016.

[26] Shujie Han, Patrick PC Lee, Zhirong Shen, Cheng He, Yi Liu, and Tao Huang. Streamdfp: A general stream mining framework for adaptive disk failure prediction. *IEEE Transactions on Computers*, 2022.

[27] Shujie Han, Patrick PC 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, 2021.

[28] Shunjie Han, Cao Qubo, and Han Meng. Parameter selection in svm with rbf kernel function. In *World Automation Congress 2012*, pages 1–4. IEEE, 2012.

[29] David J Hand and Robert J Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2):171–186, 2001.

[30] Mingzhe Hao, Levent Toksoz, Nanqinqin Li, Edward Edberg Halim, Henry Hoffmann, and Haryadi S Gunawi. Linnos: Predictability on unpredictable flash storage with a light neural network. In *OSDI*, pages 173–190, 2020.

[31] Wenwen Hao, Ben Niu, Yin Luo, Kangkang Liu, and Na Liu. Improving accuracy and adaptability of ssd failure prediction in hyper-scale data centers.

[32] Damien Hogan, Tom Arbuckle, and Conor Ryan. Evolving a storage block endurance classifier for flash memory: A trial implementation. In *2012 IEEE 11th International Conference on Cybernetic Intelligent Systems (CIS)*, pages 12–17. IEEE, 2012.

[33] Damien Hogan, Tom Arbuckle, and Conor Ryan. Estimating mlc nand flash endurance: a genetic programming based symbolic regression application. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 1285–1292, 2013.

[34] Ping Huang, Pradeep Subedi, Xubin He, Shuang He, and Ke Zhou. FlexECC: Partially relaxing ECC of MLCSSD for better cache performance. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pages 489–500, 2014.

[35] Song Huang, Song Fu, Quan Zhang, and Weisong Shi. Characterizing disk failures with quantified disk degradation signatures: An early experience. In *2015 IEEE International Symposium on Workload Characterization*, pages 150–159. IEEE, 2015.

[36] Mohammad S Jassas and Qusay H Mahmoud. A failure prediction model for large scale cloud applications using deep learning. In *2021 IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2021.

[37] Saurabh Jha, Shengkun Cui, Subho S. Banerjee, Tianyin Xu, Jeremy Enos, Mike Showerman, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. Live forensics for hpc systems: A case study on distributed storage systems. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2020.

[38] Saurabh Jha, Shengkun Cui, Subho S Banerjee, Tianyin Xu, Jeremy Enos, Mike Showerman, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. Live forensics for hpc systems: A case study on distributed storage systems. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

[39] Tianming Jiang, Jiangfeng Zeng, Ke Zhou, Ping Huang, and Tianming Yang. Lifelong disk failure prediction via gan-based anomaly detection. In *2019 IEEE 37th International Conference on Computer Design (ICCD)*, pages 199–207. IEEE, 2019.

[40] 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. IEEE, 2014.

[41] 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, 2021.

[42] Zhengguang Li, Heng Chen, Xiaochuang Ma, Huayue Chen, and Zhi Ma. Triple pseudo-siamese network with hybrid attention mechanism for welding defect detection. *Materials & Design*, 217:110645, 2022.

[43] Zheng Lian, Ya Li, Jianhua Tao, and Jian Huang. Speech emotion recognition via contrastive loss under siamese networks. In *Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and First Multi-Modal Affective Computing of Large-Scale Multimedia Data*, pages 21–26, 2018.

[44] 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, 2020.

[45] 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. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(3):1–48, 2018.

[46] Ao Ma, Fred Douglis, Guanlin Lu, Darren Sawyer, Surendar Chandra, and Windsor Hsu. Raidshield: Characterizing, monitoring, and proactively protecting against disk failures. In *13th USENIX Conference on File and Storage Technologies (FAST 15)*, pages 241–256, 2015.

[47] 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, 2017.

[48] Hosein Mohamamdi Makrani, Hossein Sayadi, Najmeh Nazari, Sai Mnoj Pudukotai Dinakarrao, Avesta Sasan, Tinoosh Mohsenin, Setareh Rafatirad, and Houman Homayoun. Adaptive performance modeling of data-intensive workloads for resource provisioning in virtualized environment. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 5(4):1–24, 2021.

[49] Stathis Maneas, Kaveh Mahdaviani, Tim Emami, and Bianca Schroeder. A study of SSD reliability in large scale enterprise storage deployments. In *18th USENIX Conference on File and Storage Technologies (FAST 20)*, pages 137–149, 2020.

[50] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. Siamese network features for image matching. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 378–383. IEEE, 2016.

[51] Justin Meza, Qiang Wu, Sanjev Kumar, and Onur Mutlu. A large-scale study of flash memory failures in the field. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):177–190, 2015.

[52] Justin J Meza. *Large scale studies of memory, storage, and network failures in a modern data center*. PhD thesis, Carnegie Mellon University, 2018.

[53] Rino Micheloni, Alessia Marelli, and Kam Eshghi. *Inside solid state drives (SSDs)*. Springer, 2013.

[54] Neal R Mielke, Robert E Frickey, Ivan Kalastirsky, Minyan Quan, Dmitry Ustinov, and Venkatesh J Vasudevan. Reliability of solid-state drives based on nand flash memory. *Proceedings of the IEEE*, 105(9):1725–1750, 2017.

[55] Subrata Mitra, Rajesh Panta, Moo-Ryong Ra, and Saurabh Bagchi. Partial-parallel-repair (ppr) a distributed technique for repairing erasure coded storage. In *Proceedings of the eleventh European conference on computer systems*, pages 1–16, 2016.

[56] Bashir Mohammed, Irfan Awan, Hassan Ugail, and Muhammad Younas. Failure prediction using machine learning in a virtualised hpc system and application. *Cluster Computing*, 22(2):471–485, 2019.

[57] Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphael Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, et al. River: machine learning for streaming data in python. 2021.

[58] Joseph F Murray, Gordon F Hughes, and Kenneth Kreutz-Delgado. Hard drive failure prediction using non-parametric statistical methods. In *Proceedings of Icann/Iconip*, 2003.

[59] Joseph F Murray, Gordon F Hughes, Kenneth Kreutz-Delgado, and Dale Schuurmans. Machine learning methods for predicting failures in hard drives: A multiple-instance application. *Journal of Machine Learning Research*, 6(5), 2005.

[60] Yoshio Nakamura, Tomoko Iwasaki, and Ken Takeuchi. Machine learning-based proactive data retention error screening in 1xnm tlc nand flash. In *2016 IEEE International Reliability Physics Symposium (IRPS)*, pages PR–3. IEEE, 2016.

[61] 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*, pages 1–11, 2016.

[62] Nikunj C Oza and Stuart J Russell. Online bagging and boosting. In *International Workshop on Artificial Intelligence and Statistics*, pages 229–236. PMLR, 2001.

[63] Junting Pan, Cristian Canton Ferrer, Kevin McGuinness, Noel E O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.

[64] Borja Peleato, Haleh Tabrizi, Rajiv Agarwal, and Jeffrey Ferreira. Ber-based wear leveling and bad block management for nand flash. In *2015 IEEE International Conference on Communications (ICC)*, pages 295–300. IEEE, 2015.

[65] Francisco Lucas F Pereira, Daniel N Teixeira, Joao Paulo P Gomes, and Javam C Machado. Evaluating one-class classifiers for fault detection in hard disk drives. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 586–591. IEEE, 2019.

[66] Francisco Lucas Falcão Pereira, Fernando Dione dos Santos Lima, Lucas Goncalves de Moura Leite, João Paulo Pordeus Gomes, and Javam de Castro Machado. Transfer learning for bayesian networks with application on hard disk drives failure prediction. In *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 228–233. IEEE, 2017.

[67] Riccardo Pinciroli, Lishan Yang, Jacob Alter, and Evgenia Smirni. Lifespan and failures of ssds and hdds: Similarities, differences, and prediction models. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[68] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[69] Lucas P Queiroz, Joao Paulo P Gomes, Francisco Caio M Rodrigues, Felipe T Brito, Iago C Chaves, Lucas GM Leite, and Javam C Machado. Fault detection in hard disk drives based on a semi parametric model and statistical estimators. *New Generation Computing*, 36(1):5–19, 2018.

[70] Lucas P Queiroz, Francisco Caio M Rodrigues, Joao Paulo P Gomes, Felipe T Brito, Iago C Chaves, Manoel Rui P Paula, Marcos R Salvador, and Javam C Machado. A fault detection method for hard disk drives based on mixture of gaussians and nonparametric statistics. *IEEE Transactions on industrial informatics*, 13(2):542–550, 2016.

[71] Saharon Rosset. Model selection via the auc. In *Proceedings of the twenty-first international conference on Machine learning*, page 89, 2004.

[72] 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, 2016.

[73] 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, 2017.

[74] Jing Shen, Jian Wan, Se-Jung Lim, and Lifeng Yu. Random-forest-based failure prediction for hard disk drives. *International Journal of Distributed Sensor Networks*, 14(11):1550147718806480, 2018.

[75] Mahdi Torabzadehkashi, Siavash Rezaei, Ali HeydariG-orji, Hosein Bobarshad, Vladimir Alves, and Nader Bagherzadeh. Computational storage: an efficient and scalable platform for big data and hpc applications. *Journal of Big Data*, 6(1):1–29, 2019.

[76] Yu Wang, Qiang Miao, Eden WM Ma, Kwok-Leung Tsui, and Michael G Pecht. Online anomaly detection for hard disk drives based on mahalanobis distance. *IEEE Transactions on Reliability*, 62(1):136–145, 2013.

[77] Yu Wang, Qiang Miao, and Michael Pecht. Health monitoring of hard disk drive based on mahalanobis distance. In *2011 Prognostics and System Health Managment Confernece*, pages 1–8. IEEE, 2011.

[78] 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*, pages 1–10, 2018.

[79] Yanwen Xie, Dan Feng, Fang Wang, Xinyan Zhang, Jizhong Han, and Xuehai Tang. Ome: An optimized modeling engine for disk failure prediction in heterogeneous datacenter. In *2018 IEEE 36th International Conference on Computer Design (ICCD)*, pages 561–564. IEEE, 2018.

[80] 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, 2019.

[81] Fan Xu, Shujie Han, Patrick PC 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. IEEE, 2021.

[82] Yong Xu, Kaixin Sui, Randolph Yao, Hongyu Zhang, Qingwei Lin, Yingnong Dang, Peng Li, Keceng Jiang, Wenchi Zhang, Jian-Guang Lou, et al. Improving service availability of cloud systems by predicting disk error. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 481–494, 2018.

[83] Qibo Yang, Xiaodong Jia, Xiang Li, Jianshe Feng, Wenzhe Li, and Jay Lee. Evaluating feature selection and anomaly detection methods of hard drive failure prediction. *IEEE Transactions on Reliability*, 70(2):749–760, 2020.

[84] Wenjun Yang, Dianming Hu, Yuliang Liu, Shuhao Wang, and Tianming Jiang. Hard drive failure prediction using big data. In *2015 IEEE 34th Symposium on Reliable Distributed Systems Workshop (SRDSW)*, pages 13–18. IEEE, 2015.

[85] Amirhessam Yazdi, Xing Lin, Lei Yang, and Feng Yan. Sefee: Lightweight storage error forecasting in large-scale enterprise storage systems. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14. IEEE, 2020.

[86] Yusheng Yi, Jiang Xiao, Song Wu, Huichuwu Li, and Hai Jin. Failure order: A missing piece in disk failure processing of data centers. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 223–230. IEEE, 2019.

[87] Shuangshuang Yuan, Peng Wu, Yuehui Chen, Liqiang Zhang, and Jian Wang. An integrated gan-based approach to imbalanced disk failure data. In *International Conference on Intelligent Computing*, pages 615–627. Springer, 2022.

[88] Leila Zahedi, Farid Ghareh Mohammadi, Shabnam Rezapour, Matthew W Ohland, and M Hadi Amini. Search algorithms for automated hyper-parameter tuning. *arXiv preprint arXiv:2104.14677*, 2021.

[89] Ji Zhang, Ping Huang, Ke Zhou, Ming Xie, and Sebastian Schelter. Hddse: Enabling high-dimensional disk state embedding for generic failure detection system of heterogeneous disks in large data centers. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pages 111–126, 2020.

[90] Ji Zhang, Ke Zhou, Ping Huang, Xubin He, Zhili Xiao, Bin Cheng, Yongguang Ji, and Yinhu Wang. Transfer learning based failure prediction for minority disks in large data centers of heterogeneous disk systems. In *Proceedings of the 48th International Conference on Parallel Processing*, pages 1–10, 2019.

[91] Ji Zhang, Ke Zhou, Ping Huang, Xubin He, Ming Xie, Bin Cheng, Yongguang Ji, and Yinhu Wang. Minority disk failure prediction based on transfer learning in large data centers of heterogeneous disk systems. *IEEE Transactions on Parallel and Distributed Systems*, 31(9):2155–2169, 2020.

[92] Xu Zhang, Junghyun Kim, Qingwei Lin, Keunhak Lim, Shobhit O Kanaujia, Yong Xu, Kyle Jamieson, Aws Albarghouthi, Si Qin, Michael J Freedman, et al. Cross-dataset time series anomaly detection for cloud systems. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pages 1063–1076, 2019.

[93] 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.

[94] Mai Zheng, Joseph Tucek, Feng Qin, and Mark Lillibridge. Understanding the robustness of {SSDs} under power fault. In *11th USENIX Conference on File and Storage Technologies (FAST 13)*, pages 271–284, 2013.

[95] 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. IEEE, 2013.